WIRTSCHAFTSUNIVERSITÄT WIEN
Vienna University of Economics and Business

Bachelor Thesis

Titel of Bachelor Thesis (english)	Developing Austrian POS Software
Titel of Bachelor Thesis (german)	Entwicklung einer POS Software
Author (last name, first name):	Aydin Hüseyin Kaya
Student ID number:	00553559
Degree program:	Bachelor of Business Administration
Examiner (degree, first name, last name):	Prof. Rony G. Flatscher

I hereby declare that:

1. I have written this Bachelor thesis myself, independently and without the aid of unfair or unauthor-ized resources. Whenever content has been taken directly or indirectly from other sources, this has been indicated and the source referenced.
2. This Bachelor Thesis has not been previously presented as an examination paper in this or any other form in Austria or abroad.
3. This Bachelor Thesis is identical with the thesis assessed by the examiner.
4. (only applicable if the thesis was written by more than one author): this Bachelor thesis was written together with

   The individual contributions of each writer as well as the co-written passages have been indicated.

30.01.2018
Date


Unterschrift

Vienna University of Economics and Business / Wirtschaftsuniversität Wien

Bachelor Thesis

Developing Austrian POS Software

in German: Entwicklung einer POS Software

by Hüseyin Kaya Aydin, 00553559

Supervisor: Prof. Rony G. Flatscher

30 January 2018

Abstract:

The Purpose of this Research is developing an Austrian Point of Sale (POS) Software using BSF4ooRexx and JavaFX. It shows how easily a desktop Program can be created by using Open Object Rexx programming language.

Recently the Parliament in Austria introduced a new law which obligated companies to replace existing POS Software in order to comply with the new law. This paper conducts an analysis of POS Softwares and deploys a basic POS Software programmed in BSF4ooRexx which can be run on a Tablet or any PC.

Keywords: pos, point of sale, pos system, bsf4ooRexx, rexx, oorexx, java, till system

Table of Contents:

List of Figures

List of Flow Charts

List of Tables

List of Snippets

1. Introduction

A POS System, which is the abbreviation of "point of sale”, stands for the point where a sale is completed and usually payment is done. A POS system helps the seller not only make sales easier and faster but also record invoices for tax authorities. It is usual that modern POS systems also offer integration with accounting or marketing software, what is more, some POS software do have also own modules like accounting and marketing modules. In the UK the term Till System is common instead of POS System.

Some modern POS Systems even reduce the buying process to 5 seconds. At the Figure below, there is a self-service-checkout POS or ExpressKassa which was introduced by the retailer company "Spar”,at the Vienna Economy University (WU) as a pilot Project. This POS system allows customers to scan products themselves and make payment via Banking card without need of waiting for an employee and long waiting lines are not necessary. In the future we may encounter more self-checkout systems.

Figure 1.1: NCR SelfServ, a Self-service-checkout POS at WU Campus [Source: NCR]

Today the POS Systems are very necessary tools providing graphical user interface for different operations. Not only basic operations like selling process, adding product prices together, but also print invoices with a company logo, complete accounting tasks, store

customer data or inventory stock data on a database are provided functions. This database can later be used for marketing such as customer relationship management.

Modern programming languages like Java make it possible to deploy POS Systems which can be adapted and customised for different kind of stores, restaurant, ice-cream shop, small stores, one-person establishments.

The purpose of this study is deploying a POS Software exactly for such companies which is programmed in the BSF4ooRexx programming language that can easily take advantage of the Java and JavaFX. BSF4ooRexx makes it easy for a newbie like the author of this paper, to deploy a desktop software even without having experience with desktop programming. Creating a Graphical User interface (GUI) is, thanks to JavaFX and Scene Builder, easy. With dragging and dropping one can easily create a GUI and then integrate it to the Rexx programming language.

This paper is divided into eight parts:

- Part 1 contains the introduction. It describes what simply a POS System is, presents abstract, background and objectives of this paper.
- Part 2 presents the requirements of a POS system, such as hardware and software requirements.
- Part 3 and 4 describes the involved components in this POS project and their installation.
- Part 5, 6 and 7 describes how this POS Software has been created, such as GUI, database and used environment.
- The Part 8 concludes the Bachelor thesis.

This paper also includes these supplementary parts at the end: bibliography and the code of the program.

1.1. Background

The Parliament in Austria has introduced a new law and according to it, the pos systems must fulfill new requirements by law in order to prevent manipulation. According to these new requirements, it is necessary to have a certificate issued by a certificate authority and have an AES 256 encrypted invoice by each sale. Since April 2017 not implementing technical necessities in POS system could cause punishment.

1.1. Objectives

The objective of this research is, creating an open source point of sale program which can be run on any Linux, Windows or Mac computer or on a Windows Tablet PC. It doesn't currently support Android Tablets, however working well on a Windows Tablet. The program will create invoices that are fulfilling requirements by law such as a QR code on invoices with AES 256 encryption so that small shops could use this open source software in order to comply with the new law.

2. Analysis of a Point of Sale System (POS)

The term POS system is sometimes used for POS software but also for POS hardware. Nevertheless, a POS system comes into being with a Software and Hardware. There are basicly 2 kind of POS systems, first one is on-premise, which is a software on a local computer, the second type of POS system is cloud software, which gets accessed from any browser of a computer, tablet or mobile.

2.1. Software Analysis

The cloud type of POS software needs an active Internet connection to be accessed. This kind of software allows user to pay for access only as long as one needs it. As the hardware and infrastructure for keeping this software is at the provider side, the provider asks for monthly subscribtion fee for providing this service. The on-enterprise type of POS software is usually

bought once and can be installed at a local computer or tablet. There is mostly no monthly subscription as the hardware and infrastructure is local.

For this project, the on-enterprise model has been selected. This means the user of the software needs to have own hardware and infrastructure. As operating system any of the following could be selected in order to use this POS Software: Windows, Linux, MacOS, Windows Tablets. Android tablets are not yet compatible with the current project.

For developing the current POS software in this research, the programming language ooRexx and for the GUI JavaFX have been used. Chapter 3.3 describes this object oriented programming language.

2.2. Hardware Analysis:

For using the POS software deployed in this paper any PC or windows tablet could be selected as most common operating systems are, thanks to Java and Rexx, supported; Windows, Linux, MacOS. A simple POS hardware includes a PC, (touchscreen) display and a thermal printer to print invoices. A touchscreen display would be a plus, however is not obligatory. A thermal printer can be connected via Ethernet, there are also ones which support Bluetooth or WLAN connection. Epson is a widely used thermal printer producer. Optional hardware could be a barcode reader and a card payment system to accept bank cards.

Figure 2.1: a PC POS [Source: Telemat] and a Tablet POS [Source: Kassensysteme]

3. Involved Components

The Program has been developed on a Linux Computer. It is working also on Windows computer as well as on Windows tablets. It runs on any computer which has the obligatory programs installed i.e. Java (JRE), ooRexx, BSF4ooRexx, MySQL JDBC driver and zxing library.

3.1. Java

Java will be camouflaged as ooRexx using the BSF.CLS Package and Bean Scripting Framework for ooRexx named „BSF4ooRexx“. The lecture Foils of Prof. Flatscher describes how this works [FlatAJ]. Java is highly portable as it can be run on any hardware or operating system through platform compatible Java Virtual Machine. For instance all Android apps are Java-based and majority of big companies use Java as backend development at server-side. It is named after Java coffee and first appeared in 1995. The main advantage of Java is that there are a lot of libraries and code snippets available, so there is no need to write new code all over again. This makes programming with Java very efficient.

Table 3.1. depicts a ranking List of most Popular Programming Languages:

Pos.	Programming Language	Ratings
1	Java	14.215%
2	C	11.037%
3	C++	5.603%
4	Python	4.678%
5	C#	3.754%
6	JavaScript	3.465%
7	Visual Basic .NET	3.261%
8	R	2.549%
9	PHP	2.532%
10	Perl	2.419%
11	Ruby	2.406%
12	Swift	2.377%
13	Delphi/Object Pascal	2.377%
14	Visual Basic	2.314%
15	Assembly language	2.056%
16	Objective-C	1.860%
17	Scratch	1.740%
18	MATLAB	1.653%
19	Go	1.569%
20	PL/SQL	1.429%

21	Dart	1.426%
22	SAS	1.419%
23	Erlang	1.250%
24	COBOL	0.798%
25	ABAP	0.697%
26	Alice	0.652%
27	D	0.628%
28	Lua	0.620%
29	Ada	0.569%
30	REXX	0.562%

Table 3.1 : Programming Language Ratings [Source: Tiobe]

The Table 3.1. shows how popular Java is and that, Rexx is also listed in Top 30 among many programming language. Therefore using Java and Rexx which is implemented thanks to BSF4ooRexx sounds to be very efficient way of programming

3.2. ooRexx

The Open Object Rexx (ooRexx) programming language is a cross-platform open source programming language which was initially produced by IBM.

Rexx (Restructured Extended Executor) is a programming language which was developed at IBM, and several implementations are available under open source licenses. It is a structured high-level programming language which was designed to be both easy to learn and easy to read. Both commercial and open source Interpreters for Rexx are available on a wide range of computing platforms. [RexxIn]

Object Rexx is the object-oriented version of Rexx developed in the 90s. Object Rexx is fully compatible to Rexx, but has a complete internal object oriented structure and a powerful object model (meta-classes, multiple inheritance). [Flatscher]

IBM's programming guide and reference to Object Rexx can be found on [Ibm01] and on [rexxpg].

"Open Object Rexx (ooRexx) which is an Open Source project managed by Rexx Language Association (RexxLA) providing a free implementation of Object Rexx, is an enhancement of classic Rexx; a powerful, full-featured programming language which has a human-oriented syntax." [ooRexx]

Lecture slides by Prof. Dr. Rony G. Flatscher provide a great introduction to Object Rexx. In the lecture "Business Programming 1" tutorials and sample scripts for automating Windows and in the lecture "Business Programming 2" tutorials and sample scripts for camouflaging Java in Rexx can be found. (download available on [Flat02]) . The book of Prof. Flatscher [Flat13]) is also recommended.

Snipper 3.1. depicts a sample of programming in ooRexx

```
/* town.rex */
/* use PULL for user to enter a Data
and use SAY to show it on screen */
SAY 'Which town are you living?'
PULL town
SAY 'Wow, i was in' town
EXIT 0
```

Snippet 3.1 - an ooRexx sample code using SAY and PULL commands

Using rexx town.rex command or simply by double clicking it the program can be runned.

Output would look like following:

```
rexx town.rex

Which town are you living in?

vienna

Wow, i was in vienna!
```

Snippet 3.2 - output of the ooRexx sample code above

Snippet 3.3 is an another sample for arithmetic operations or merging strings

```
/* samplecode.rex */
  /* arithmetic operations or merging strings */
  x = 10; y = 13; z = x + y
  res = y - x
  v = 'golf'; w = '6gti'; vw = v || w
  SAY x y y; SAY v w vw
  say res
```

Snippet 3.3 - an ooRexx sample code for arithmetic operations with variables

Snippet 3.4 shows the output of running Snippet 3.3

```
rexx samplecode.rex

10 13 23

golf 6gti golf6geti

3
```

Snippet 3.4 - output of the ooRexx sample code above

Snipper 3.5 depicts how to include another rexx file into current file

```
  /* mainpage.rex */
  result=addthem(3412,4442)
  SAY result
  ::REQUIRES subpage.rex
```

Snippet 3.5 - an ooRexx sample code for including a file

Snippet 3.6 shows a sample subpage having a routine which is Snippet 3.5

```
/* subpage.rex */
  ::ROUTINE addthem PUBLIC
  result = 0
  USE ARG var1, var2
  result = var1 + var2;
 RETURN result
```

Snippet 3.6 - the file that was included in above file

The mainpage would add the routine in subpage and execute it.

Here is another example which shows how one can create an object in open object Rexx, with class, method and method Attribute the object details can be given

```
  /* student.rex */
  student1 = .Student~NEW
  student1~name = "Kaya"
  student1~matrikelnummer = 1205539
  student1~studienrichtung = "BWL"
  student1~note = "4"

  say "student 1:"
  student1~getAttributes
  say ; say "-"~COPIES(20)
  student1~unbekanntemethode
  say ;  say "-"~COPIES(20)

  Student2= .Student~NEW~~"name="("Daniela")~~"matrikelnummer= -
"(1209443)~~"studienrichtung="(BWL)~~"note="(3)
```

```
say "student 2:"
Student2~getAttributes   ; say

::CLASS Student
::METHOD INIT
    EXPOSE name matrikelnummer studienrichtung note
    USE ARG name, matrikelnummer, studienrichtung, note

::METHOD name ATTRIBUTE
::METHOD matrikelnummer ATTRIBUTE
::METHOD studienrichtung ATTRIBUTE
::METHOD note ATTRIBUTE

::METHOD getAttributes
    SAY self
    SAY self~name
    SAY self~matrikelnummer
    SAY self~studienrichtung
    SAY self~note

::method unknown
say "Fehler: Unbekannte Methode! "
```

Snippet 3.7 - Using Class, Methods and Attributes

More documentation can be found at official website of Open Object Rexx [ooReDo]

3.3. BSF4ooRexx

BSF4ooRexx is abbreviation of Bean Scripting Framework for ooRexx. This Framework allows Java libraries to be used as if they were Object Rexx libraries. [Flat01]

BSF4ooRexx camouflages Java as Object REXX. It was developed by Prof. Dr. Rony G. Flatscher and a former student of his, Peter Kalender, in 2000/01. According to Prof. Flatscher "ooRexx module called "BSF.CLS" makes it possible to import Java classes into ooRexx and interact with them as if they were ooRexx classes. Java objects can be treated as if they were ooRexx objects, e.g. by sending ooRexx messages to them using the ooRexx message operator (the tilde, ~, in ooRexx also called "twiddle"). In essence this support allows to use all of Java as if it was a huge ooRexx class library, that is already ported to all platforms ooRexx runs on." [Flat02]

The Bean Scripting Framework (BSF) was an open source project developed at IBM. In 2003 the code was handed ove to Apache's Jakarta project. BSF is a framework that allows Java to execute scripting languages very easily. Defined interfaces give scripting languages the possibility to interact with Java objects.

As BSF4ooRexx camouflages Java as ooRexx, the advantages of both programming languages can be combined. The huge amount of already available Java code libraries with their classes and methods become accessible in an easily understable programming language like Rexx.

Here is an examples of using BSF.CLS in a Rexx script. The "::REQUIRES BSF.CLS" statement must be at the end of the script to load the Java support.

The Snippet 3.8 gets weather information in Vienna using the "java.net.URL" class, then displays it in a "jawa.art.Frame" by adding buttons, labels into the Frame.

```
/* weather.rex */

URL="http://api.openweathermap.org/data/2.5/weather?id=2761369" -
"&lang=de&units=metric&mode=json&APPID=0c6b6e1ef1faa6ab9007dbe04dbd5c4b"

URL2="http://api.openweathermap.org/data/2.5/forecast/daily?id=2761369" -
"&lang=de&units=metric&mode=json&APPID=0c6b6e1ef1faa6ab9007dbe04dbd5c4b"

dim3 = .bsf~new("java.net.URL",URL)
dimprog = .bsf~new("java.net.URL",URL2)
cont = dim3~getContent
---ch=0
do until ch=-1
   if ch>0 then do
   ch=cont~read
   if (ch>0) then chta=chat || d2c(ch)
end
end
ch=1

contprog = dimprog~getContent

do until ch=-1
   if ch>0 then do
   ch=contprog~read
   if (ch>0) then chtaprog=chtaprog || d2c(ch)
   end
end

datum=DATE("E")                 /* Wochentag        */
t.0="Montag" ; t.1="Dienstag" ;t.2="Mittwoch";t.3="Donnerstag"
t.4="Freitag";t.5="Samstag";t.6="Sonntag"
  /* */
temp=date("B",datum,"E")//7 ; temp2=date("B",datum,"E")//7+1
temp3=date("B",datum,"E")//7+2 ; temp4=date("B",datum,"E")//7+3
```

```
temp5=date("B",datum,"E")//7+4
dimchta=.bsf~new("java.lang.String",chta)
dimchtaprog=.bsf~new("java.lang.String",chtaprog)

say "Bitte warten.."
say "Wetterdaten werden aus OpenWeatherMap geholt..";

start= dimchta~indexOf("temp_min")
end= dimchta~indexOf("temp_max")
vis= dimchta~indexOf('},"visibility":')

min= dimchta~substring(start+10,end-2)
max= dimchta~substring(end+10,vis)

result1=" Heute wird minimum " min "°C Grad kalt"
result1=result1 ||  "   maximum " max "°C Grad warm"

 start1= dimchta~indexOf("description")
end1= dimchta~indexOf("icon")

result2= "" dimchta~substring(start1+14,end1-3)

result2=result2 || " "
start2= dimchta~indexOf('"main":{"temp":')
end2= dimchta~indexOf('pressure')

result2=result2 ||  ", aktuell " -
 dimchta~substring(start2+15,end2-2) "°C Grad"
result2=result2 || " "

---Morgen
start= dimchtaprog~indexOf('"min"')
end= dimchtaprog~indexOf('"max"')
vis= dimchtaprog~indexOf('"night"')

min= dimchtaprog~substring(start+6,end-1)
max= dimchtaprog~substring(end+6,vis-1)

result1di= " "t.temp2" wird minimum " min "°C Grad kalt"
result1di=result1di ||  "   maximum " max "°C Grad warm"

start1= dimchtaprog~indexOf("description",0)
end1= dimchtaprog~indexOf("icon",0)

result2di= "" dimchtaprog~substring(start1+14,end1-3)

start= dimchtaprog~indexOf('"min"',dimchtaprog~indexOf('"min"')+1)
end= dimchtaprog~indexOf('"max"',dimchtaprog~indexOf('"max"')+1)
vis= dimchtaprog~indexOf('"night"',dimchtaprog~indexOf('"night"')+1)

min= dimchtaprog~substring(start+6,end-1)
max= dimchtaprog~substring(end+6,vis-1)

result1mi= " "t.temp3" wird minimum " min "°C Grad kalt"
result1mi=result1mi ||  "   maximum " max "°C Grad warm"

start1= dimchtaprog~indexOf("description",dimchtaprog~indexOf("description")+1)
end1= dimchtaprog~indexOf("icon",dimchtaprog~indexOf("icon")+1)

result2mi= "" dimchtaprog~substring(start1+14,end1-3)

start= dimchtaprog~indexOf('"min"',dimchtaprog~indexOf('"min"')
+100+dimchtaprog~indexOf('"min"'))
end= dimchtaprog~indexOf('"max"',dimchtaprog~indexOf('"max"')
+100+dimchtaprog~indexOf('"max"'))
vis= dimchtaprog~indexOf('"night"',dimchtaprog~indexOf('"night"')
```

```
+100+dimchtaprog~indexOf('"night"'))

min= dimchtaprog~substring(start+6,end-1)
max= dimchtaprog~substring(end+6,vis-1)

result1fre= " "t.temp4" wird minimum " min "°C Grad kalt"
result1fre=result1fre ||  "     maximum " max "°C Grad warm"

start1= dimchtaprog~indexOf("description",dimchtaprog    -
      ~indexOf("description")+100+dimchtaprog~indexOf("description"))

end1= dimchtaprog~indexOf("icon",dimchtaprog~indexOf("icon")
+100+dimchtaprog~indexOf("icon"))

result2fre= "" dimchtaprog~substring(start1+14,end1-3)

rexxCloseEH = .RexxCloseAppEventHandler~new  -- Rexx event handler
rpCloseEH = BsfCreateRexxProxy(rexxCloseEH, , "java.awt.event.WindowListener")

frame  = .BSF~new( 'java.awt.Frame', 'Aktuelles Wetter und Prognose" -
  "für die kommenden Tage')

frame~addWindowListener(rpCloseEH)
-- add RexxProxy event handler
--frame~setLayout( .bsf~new("java.awt.FlowLayout") )
userData = .directory~new
-- a directory which will be passed to Rexx with the event
userData~rexxCloseEH=rexxCloseEH

rp=BsfCreateRexxProxy(.RexxProcessEventHandler~new, userData,
"java.awt.event.ActionListener")

        /* remove layout manager or *strange* things happen ... */
 frame ~~setLayout(.nil)
  frame ~~show ~~hide      -- show and hide frame, now insets are known
  insets=  frame~getInsets      -- get and parse insets
  parse value insets~toString with . "top=" top "," "left=" left  -
          "," "bottom=" bottom "," "right=" right "]" .
  w_width =660                  -- desired inner width
  w_col1  = 10                  -- start of first column
  w_col2  = w_width/2+w_col1    -- start of second column
   w_height=300                 -- desired inner height

  tot_width =w_width +left+right        -- total_width (insets included)
  tot_height=w_height+top +bottom       -- total_height (insets included)

  frame ~~setSize(tot_width, tot_height) ~~setResizable(.true)

dim     = frame~getToolkit~getScreenSize
  scr_height = dim~bsf.getFieldValue('height')
  scr_width  = dim~bsf.getFieldValue('width')

leave =.bsf~new('java.awt.Label', 'Wetter am [ ' t.temp ' ]  -
         ' datum  ' Stadt: Vienna, AT' )
  leave~~setSize(350, 15)
  leave~~setLocation(10,10)
  frame~add(leave)

  input=.bsf~new('java.awt.TextField',result1,400)
  inp_width=200
  input~~setSize(538, 50)
  input~~setLocation(10, 30)
  frame~add(input)
userData~input=input

leaven =.bsf~new('java.awt.Label', ' Datum:'   )
```

```
  leaven~~setSize(70, 50)
  leaven~~setLocation(10,130)
  frame~add(leaven)

leave2 =.bsf~new('java.awt.Button', '   ' t.temp2   )
  leave2~~setSize(120, 50)
  leave2~~setLocation(190,130)
  frame~add(leave2)

userData~leave2=leave2

leave2~addActionListener(rp)

leave3 =.bsf~new('java.awt.Button', '  Heute'   )
  leave3~~setSize(120, 50)
  leave3~~setLocation(100,130)
  frame~add(leave3)

userData~leave3=leave3
rp=BsfCreateRexxProxy(.RexxMittwochEventHandler~new, userData, -
                "java.awt.event.ActionListener")

leave3~addActionListener(rp)

leave4 =.bsf~new('java.awt.Button', '   ' t.temp3   )
  leave4~~setSize(120, 50)
  leave4~~setLocation(280,130)
  frame~add(leave4)

userData~leave4=leave4
rp=BsfCreateRexxProxy(.RexxDonEventHandler~new, userData, "java.awt.event.ActionListener")

leave4~addActionListener(rp)

leave5 =.bsf~new('java.awt.Button', '   ' t.temp4   )
  leave5~~setSize(120, 50)
  leave5~~setLocation(370,130)
  frame~add(leave5)

userData~leave5=leave5
rp=BsfCreateRexxProxy(.RexxFreEventHandler~new, userData, "java.awt.event.ActionListener")

leave5~addActionListener(rp)
input2=.bsf~new('java.awt.TextField',result2,400)
  inp_width=200
  input2~~setSize(538, 50)
  input2~~setLocation(10, 80)
  frame~add(input2)
userData~input2=input2
userData~result1=result1 ; userData~result2=result2
userData~result1di=result1di ; userData~result2di=result2di
userData~result1mi=result1mi ;userData~result2mi=result2mi
userData~result1fre=result1fre ;userData~result2fre=result2fre

frame  ~~setVisible(.true) ~~toFront ~~setLayout( .bsf~new("java.awt.FlowLayout") )
-- layout the Frame object, show it, put to front

rexxCloseEH~waitForExit

-- wait until we are allowed to end the program
call BSF.terminateRexxEngine
-- inhibit callbacks from Java (necessary, if Rexx started Java)

::requires BSF.CLS

::class RexxCloseAppEventHandler
```

```
::method init
/* constructor */
  expose closeApp
-- used as control variable
  closeApp  = .false
::method windowClosing      -- event method (from WindowListener)
  expose closeApp
  closeApp=.true            -- change control variable to unblock
::method unknown            -- intercept unhandled events, do nothing
::attribute closeApp        -- allow to get and set the control variable's value
::method waitForExit        -- blocking (waiting) method
  expose closeApp
  guard on when closeApp=.true
-- blocks (waits) until control variable is set to .true

/* Rexx event handler : "java.awt.event.ActionListener" */
::class RexxProcessEventHandler
::method actionPerformed
  use arg eventObject, slotDir
slotDir~userData~input~setText(slotDir~userData~result1di)

slotDir~userData~input2~setText(slotDir~userData~result2di)
  userData=slotDir~userData         -- get 'userData' directory

::class RexxMittwochEventHandler
::method actionPerformed
  use arg eventObject, slotDir
slotDir~userData~input~setText(slotDir~userData~result1)

slotDir~userData~input2~setText(slotDir~userData~result2)
  userData=slotDir~userData         -- get 'userData' directory

::class RexxFreEventHandler
::method actionPerformed
  use arg eventObject, slotDir
slotDir~userData~input~setText(slotDir~userData~result1fre)

slotDir~userData~input2~setText(slotDir~userData~result2fre)
  userData=slotDir~userData         -- get 'userData' directory

::class RexxDonEventHandler
::method actionPerformed
  use arg eventObject, slotDir
slotDir~userData~input~setText(slotDir~userData~result1mi)

slotDir~userData~input2~setText(slotDir~userData~result2mi)
  userData=slotDir~userData         -- get 'userData' directory
```

Snippet 3.8 - BSF4ooRexx Example code Weather.rex

3.4. SceneBuilder:

The Gluon SceneBuilder is not necessary to use the RexxPOS orogram. However it is a very useful tool for developers to create graphical user interfaces (GUIs) easily. More information can be found at official website: http://gluonhq.com/products/scene-builder/

The main screen of the SceneBuilder is shown in Figure 3.1.

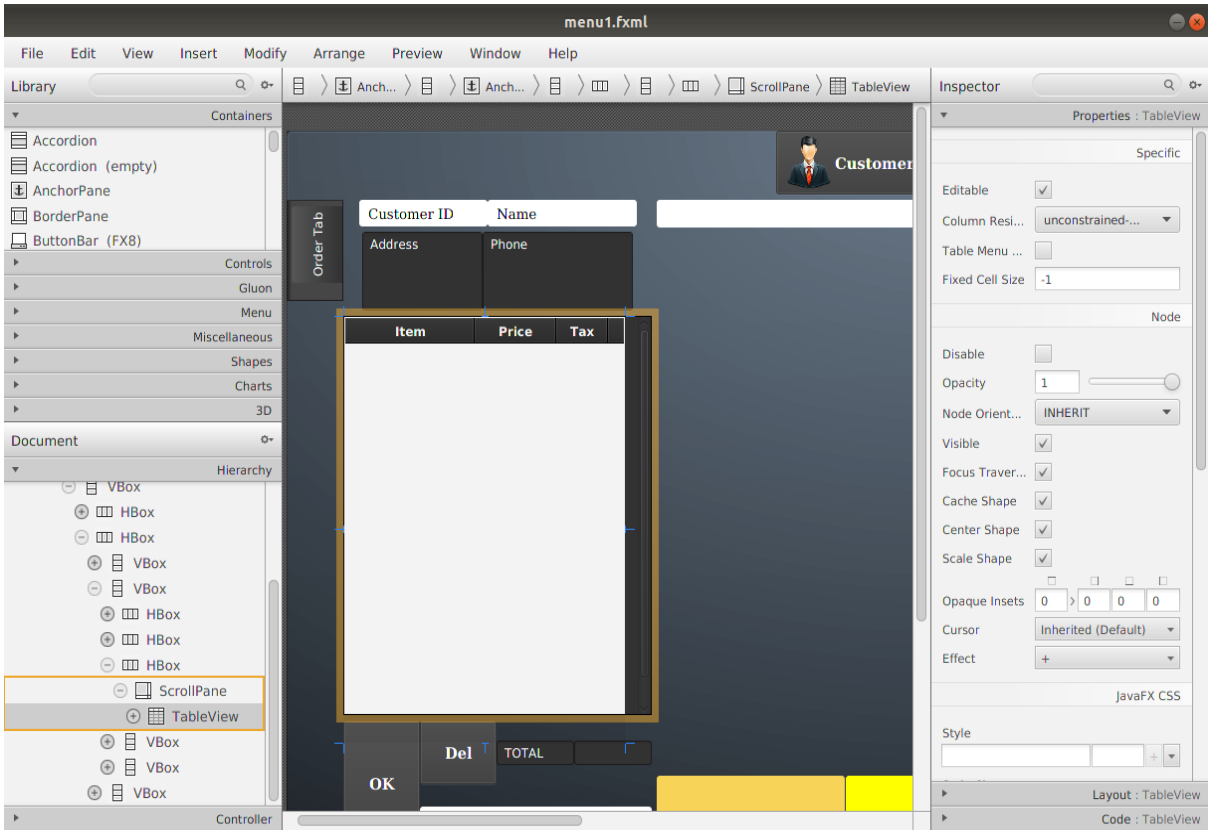


Figure 3.1: Overview of the Scene Builder Program

At the middle of the screen, there is the current view of the GUI design. This file will be saved as an FXML document and can be shown in the main program.

At the left of the screen there are items to add into GUI. For instance, containers which are container that stores all other GUI elements. For example an Anchorpane, HBOX, Gridpane, Tabpane, VBOX are different kind of Containers to be selected. After adding a container to a new document, the other library items could be added. These are for instance, control items; a button, label or textbox. Other elements are menu, shapes, charts etc.

At the right of the screen, there are 3 tabs; Properties, Layout and Code. If you want to give a name to a controller, just name the "fx:id" attribute in the Code section. If you want to add a .CSS stylesheet and want to style an item, then add the Stylesheet under Properties section and add the id of the style into that field.

3.5. MySQL JDBC Connector

In order to establish a database connection with MySql Database, a JDBC driver is necessary.

The SQL Cookbook Paper of Kirisits (2016) is a very useful reference for learning required SQL commands or how to establish a JDBC connection in BSF4ooRexx. {Source: Kiri06]

4. Installation Guide

The following chapter contains an installation guide for the software described in chapter 2. It documents where the needed software can be obtained. The instructions are intended for a Windows or Linux system.

4.1. Java

Before starting any installation of other Programs, check if Java is already installed on the PC, if, not install the Java Runtime Environment (JRE) . The developer version, JDK (Java Development Kit) is not necessary to run program. In comparison to JRE, a JDK also includes a Java compiler (javac) for developers. The JRE fulfills the task of Java bytecode execution. The latest version of Java is available at http://www.java.com/de/download . This page also checks whether Java is already installed or not.

Before starting the download, check if Java is already installed on your computer if yes, which version.

Executing command in Snipper 4.1 at command line (Windows) or terminal (Linux), one can get the installed Java version displayed :

```
java -version
```

Snippet 4.1: Get Java Version

the output of Snippet 4.1 would be similar to Snippet 4.2. displaying the Java Runtime installed

```
java version "1.8.0_151"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_151-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.151-b12, mixed mode)
```

Snippet 4.2: Java version shown [1]

Please check the bitness of the Java (64-Bit or 32-Bit), as the ooRexx and Java bitness must be same. The ouput in Snippet 4.2 shows that a 64-Bit version of Java is installed at this PC. This means, one must install the ooRexx with the same bitness, which is 64-Bit. The bitness is 32 Bit, if it shows i386 or x86. The Bitness is 64 bit, if it shows x64, x86-64 or amd64.

4.2. ooRexx

The programming language ooRexx needs to be installed on your machine. It is free to download from http://www.oorexx.org/download.html . The latest release ath the time of this writing is version 5.0.0. Do not download an earlier version as it is not compatible with current Project. The latest version is available at Sourceforge: https://sourceforge.net/projects/oorexx/files/oorexx/

At download page, click on 5.0.0beta. Then for instance if you are using a Windows PC click on the corresponding .exe file with your Java bitness. The file name ending with x86_32.exe is a 32-Bit version of ooRexx. The other file name ending with x86_64.exe is a 64-Bit version of ooRexx. It is necessary to install the version 5.0 or newer.

If you are using Linux, you must download the installation file ending with ".deb" for instance if you are using Ubuntu or Debian Linux and the file ending with ".rpm" for instance for RedHat, Fedora Linux.

To check if you have installed ooRexx successfully, execute the command in Snippet 4.3. at command line (Windows) or terminal (Linux):

```
rexx -v
```

Snippet 4.3: Get ooRexx version

1 The Java version shown at Snippet 4.2 is 1.8.0_151 and the Bitnes is 64-Bit. These may be different at the computer where the command is executed.

If you have a similar output like in Snippet 4.4. showing the installed ooRexx version, then you have successfully installed ooRexx

```
Open Object Rexx Version 5.0.0
Build date: Oct 19 2017
Addressing mode: 64
Copyright (c) 1995, 2004 IBM Corporation. All rights reserved.
Copyright (c) 2005-2017 Rexx Language Association. All rights reserved.
This program and the accompanying materials are made available under the terms
of the Common Public License v1.0 which accompanies this distribution or at
http://www.oorexx.org/license.html
```

Snippet 4.4: ooRexx version shown [2]

4.3. BSF4ooRexx

Before installing BSF4ooRexx, ensure that you have already installed Java and ooRexx beforehand. In order to install BSF4ooRexx, open following SourceForge page and click the link after text "Download the latest version": https://sourceforge.net/projects/bsf4oorexx/files/

Unpack the .zip archive. Depending on your operating system, there are installation batch files in the folder "bsf4oorexx\install\windows" or "bsf4oorexx/install/linux"

For instance if you are installing under Linux. open the directory "bsf4oorexx/install/linux", then open right click and open a Terminal window in the same directory. If this is first install run "./install.sh" at command line. If you are upgrading to a newer Version or want to reinstall it, run "./reinstall.sh"

On Windows, click on install.cmd if it is a first installation.

If you have troubles after installation, there is a log file created at user's main folder. After installation, you can try some sample scripts to check if BSF4ooRexx is working fine. These samples are found in the bsf4oorexx/samples directory.

If you get any error message regarding BSF.CLS not being found, there must be a problem with the JAVA_HOME directory or the CLASSPATH variable. For instance if a newer Java

2 The ooRexx „version" and „Build Date„ shown at Snippet 4.4 may be different at the computer where the command is executed.

version is installed but the JAVA_HOME variable has a directory inside and it is not the actual Java installed then error messages may be shown. To check the JAVA_PATH of your Java, execute the command in a terminal or command line as shown in Snippet 4.5.

```
on Windows: echo %JAVA_HOME%
on Linux:   echo $JAVA_HOME
on MAC:     java_home
```

Snippet 4.5: Get JAVA_HOME path

The output may be similar to the one in Snippet 4.6.

```
/usr/lib/jvm/jre1.8.0_131
```

Snippet 4.6: JAVA_HOME path shown [3]

If this folder does not corresponds to the actual Java you have installed, then you must edit your JAVA_HOME directory.

Depending on the operating system and version, these are different methods to edit you JAVA_HOME variable:

- under Windows:

If you want to temporarily change JAVA_HOME, run the command in the same command line windows before you run the program as depicted in Snippet 4.7.

```
set JAVA_HOME=/usr/lib/jvm/jre1.8.0_131
```

Snippet 4.7: Setting a JAVA_HOME in Windows [4]

If you want to permanently edit the JAVA_HOME variable then use Search function of Windows to find and open the variable settings page. Search for "Environment" to find tthe environment variables settings page. If your Windows is In German search for: "Umgebungsvariable". After opening the environment variables page, select JAVA_HOME under system variables. Click edit, if it does not exist, click new. In the system variable window and add the new folder of Java into JAVA_HOME

3 The Java version shown at Snippet 4.6 is 1.8.0_131. This output may be different at the computer where the command is executed.

4 The Java path shown at Snippet 4.7 depends on the Java version installed. Here the Path ends with 1.8.0_151 which is the Java version currently installed.

- under Linux:

To change JAVA_HOME enter the command as depicetd in Snippet 4.8.

```
export JAVA_HOME=/usr/lib/jvm/jre1.8.0_131
```

Snippet 4.8: Setting a JAVA_HOME path in Linux [5]

However this will not change JAVA_HOME permanently. When you close the terminal window, the variable will be lost. To permanently change the JAVA_HOME Variable, depending on the Linux version used, the following file must be edited: /etc/profile

editing ~/.profile or *~/.bashrc* *or* */etc/environment* *can be necessary i.e. for setting a system-wide environment variable instead of a specific user.*

4.4. SceneBuilder

The "Gluon Scene Builder" is not necessary to run the software. It is just a tool for developers, to create JavaFX GUIs. It can be downloaded from the official website: http://gluonhq.com/products/scene-builder/

It is free and licensed under the BSD license. The latest version in time of this writing is from October 2017.

Depending on operating system bitness select the appropriate version to download.

4.5. MySQL JDBC Connector

The .JAR package MySQL JDBC connector is necessary to run the program. This file is already included in /bin/lib Folder of RexxPOS Archive. The RexxPOS will automatically add this file to the CLASSPATH. However if you want to install the latest release of this file here is the Instructions: The library can be downloaded from the official MySQL website: https://dev.mysql.com/downloads/connector/j/

When you click download, a sign in or register page will be shown. Just click on "No thanks, just start my download." at the bottom of the page and the download will start.

5 The Java path shown at Snippet 4.8 depends on the Java version installed. Here the Path ends with 1.8.0_151 which is the Java version currently installed.

After downloading the driver, unpack the archive into a folder. Notice the full path and the name of the JAR file. Then you must include this file's full path into CLASSPATH.

Depending Operating System and version, these are the methods to add a JAR file into the CLASSPATH environment variable:

- under Windows:

If you want to temporarily add a .jar file to the CLASSPATH, run the command in same command line windows before you run the program as shown in Snippet 4.9.

```
set CLASSPATH=%CLASSPATH%;/home/pos/PathToJarfile.jar
```

Snippet 4.9: Adding a .jar into CLASSPATH in Windows [6]

Using Windows search function and searching for "Environment" open the environment variables settings page. If your Windows is German search for: "Umgebungsvariable". After opening the environment variables page, select CLASSPATH under system variables, click "edit", if it does not exist, click "new". In the system variable window add the file name including the path name to that directory, to the CLASSPATH

- under Linux:

Open a terminal window. Enter the command as shown in Snippet 4.10.

```
export CLASSPATH=$CLASSPATH:/home/pos/PathToJarFile.jar
```

Snippet 4.10: Adding a .jar into CLASSPATH in Linux

Here you must edit the path to MySql Driver with your directory and filenames.

However this will not change the CLASSPATH permanently. When you close the terminal window, the variable will be lost. To permanently change CLASSPATH variable, depending on the Linux version used, following file must be edited: /etc/profile

Editing ~/.profile or *~/.bashrc* *or* */etc/environment* *can be necessary i.e. for setting a system-wide CLASSPATH variable instead of a specific user*

6 The path shown at Snippet 4.9 or 4.10 is the Path where the java library is saved. This variable may not be same at another computer.

In order to check if this JAR file is added to the CLASSPATH, one can execute the following command in Windows: "set CLASSPATH" or in Linux: "echo $CLASSPATH"

The /etc/profile file may look similar to Snippet 4.11.

```
# /etc/profile: system-wide .profile file for the Bourne shell...

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/jvm/jre1.8.0_131/bin"
export PATH

JAVA_HOME="/usr/lib/jvm/jre1.8.0_131"
export JAVA_HOME

CLASSPATH="home/kaya/Downloads/mysql-connector-java-5.1.40/mysql-connector-java-5.1.40-bin.jar:/opt/openoffice4/program/classes/ridl.jar:/opt/openoffice4/program/classes/jurt.jar:/opt/openoffice4/program/classes/juh.jar:/opt/openoffice4/program/classes/unoil.jar"
export CLASSPATH
```

Snippet 4.11: Setting a permanent CLASSPATH in Linux [7]

5. Program Overview and Code explanation

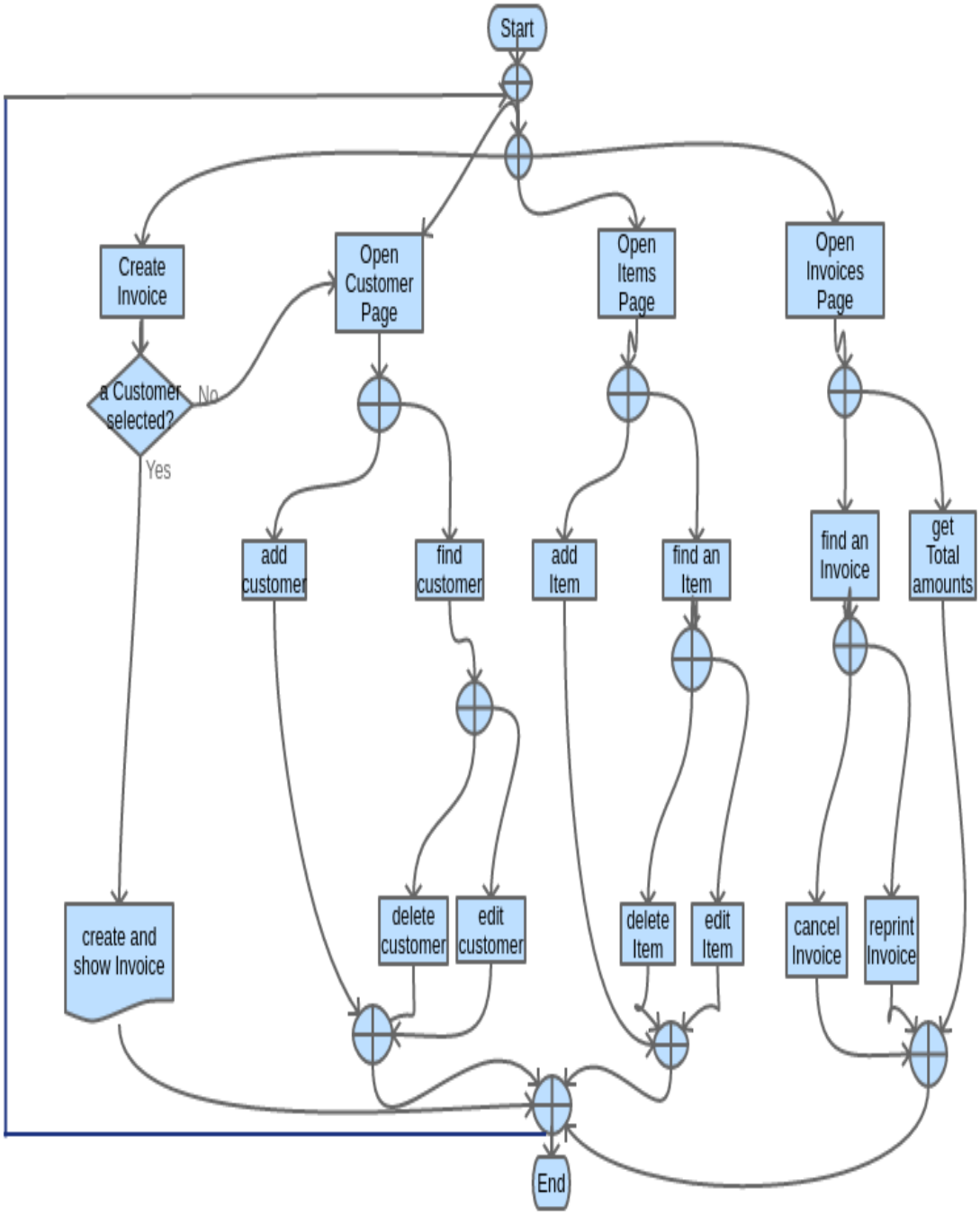
The program can be downloaded from it's official website: http://www.lieferex.at/rexxpos/

Unpack the downloaded .zip file, then read "readme.txt" for addtional instructions.

To start the program the user double clicks "start.rex" file. After this a dashboard will be shown. In the Flowchart 5.1, the sequences of the tasks are shown. User has possibilities to create an invoice (make an order), to open the customer page, to open the invoices page and to open the items Page. There is possibility to edit a dataset by double-clicking. Also adding new data or deleting an existing one is possible. System then will check if a customer is selected, if not one must select a customer to be able to create an invoice. At the invoices page one can search for an invoice or see the total amount of orders.

7 The java path or CLASSPATH shown at Snippet 4.11 may be differen at another computer

The **Flowchart 5.1.** represents which tasks one may execute.



Flowchart 5.1: Flowchart of the Program

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. [WikiFl] For instance if the process is "create invoice", the program will check if a customer is selected. If the answer is "yes" the process will be directed to next step, otherwise "open customer page" will be opened to select a customer.

When the program is started, it first checks if the required libraries are added to the CLASSPATH. If not, it automatically adds these libraries (.jar files) which are located in bin/lib directory to the CLASSPATH. For this, it must get the operating system name and execute the required commands depending on the operating system. Snippet 5.1 shows the relevant code from the file "start.rex".

```
/* ============================================================================
   FIND CURRENT FOLDER NAME AND CHANGE TO SUB DIRECTORY /BIN
============================================================================
*/
PARSE SOURCE . . name
dir=FILESPEC('D', name) || FILESPEC('P', name)
"cd "dir
"cd bin"
/* ============================================================================
   CHECK CLASSPATH and ADD .JAR FILES AUTOMATICALLY INTO IT IF NOT EXISTS
============================================================================
*/
if POS("mysql-connector",value("CLASSPATH",,"ENVIRONMENT"))=.false then
do
ver=SysVersion() -- get operating system version
   if ver~left(1)~upper="W" then -- a Windows OS
      do
         "set CLASSPATH=%CLASSPATH%;"directory()"\lib\mysql-connector-java-5.1.45-bin.jar;"
      end
   else if ver~left(1)~upper="L" | ver~left(1)~upper="U" then -- a Linux OS
      do
         "export CLASSPATH=$CLASSPATH:"directory()"/lib/mysql-connector-java-5.1.45-bin.jar"
      end
   else if ver~left(1)~upper="M" | ver~left(1)~upper="D" then -- a MacOS
      do
         say "*********************************************************************"
         say "Classpath Error:"
         say "The .jar archieve mysql-connector-java-5.1.45-bin.jar or newer must be "
         say "added in  Classpath"
         say "*********************************************************************"
         say
         say "Please install official mysql driver at
https://dev.mysql.com/downloads/connector/j/ "
         say
         say "Please check instruction on www how to edit Classpath on Mac"
         pull .
      end
end
--if classp~contains("zxing")=.false then           -- check zxing .jar file
if POS("core-3.3.0.jar",value("CLASSPATH",,"ENVIRONMENT"))=.false then
do
ver=SysVersion() -- get operating system version
   if ver~left(1)~upper="W" then -- a Windows OS
      do
         "set CLASSPATH=%CLASSPATH%;"directory()"\lib\core-3.3.0.jar;"
      end
   else if ver~left(1)~upper="L" | ver~left(1)~upper="U" then -- a Linux OS
      do
         "export CLASSPATH=$CLASSPATH:"directory()"/lib/core-3.3.0.jar"
      end
```

```
   else if ver~left(1)~upper="M" | ver~left(1)~upper="D" then -- a MacOS
      do
         say "*************************************************************************"
         say "Classpath Error:"
         say "The .jar archieve zxing.jar must be added in  Classpath"
         say "This is necessary for QR Code Barcode creation"
         say "*************************************************************************"
         say
         say "The file allready exists in lib directory of RexxPOS"
         say "Please check instruction on www how to edit Classpath on Mac"
         pull .
      end
end
if POS("javase-3.3.0.jar",value("CLASSPATH",,"ENVIRONMENT"))=.false then
do
ver=SysVersion() -- get operating system version
   if ver~left(1)~upper="W" then -- a Windows OS
      do
         "set CLASSPATH=%CLASSPATH%;"directory()"\lib\javase-3.3.0.jar;"
      end
   else if ver~left(1)~upper="L" | ver~left(1)~upper="U" then -- a Linux OS
      do
         "export CLASSPATH=$CLASSPATH:"directory()"/lib/javase-3.3.0.jar"
      end
   else if ver~left(1)~upper="M" | ver~left(1)~upper="D" then -- a MacOS
      do
         say "*************************************************************************"
         say "Classpath Error:"
         say "The .jar archieve zxing.jar must be added in  Classpath"
         say "This is necessary for QR Code Barcode creation"
         say "*************************************************************************"
         say
         say "The file allready exists in lib directory of RexxPOS"
         say "Please check instruction on www how to edit Classpath on Mac"
         pull .
      end
end
-- ======== CHECK CLASSPATH End ============================================
/* ========================================================================
   RUN MAIN APP   POS.RXJ
========================================================================
*/
call "pos.rxj"
```

Snippet 5.1: Check CLASSPATH and Add Libraries Automatically

After the required Libraries have been added the main program "bin/pos.rxj" starts.

Figure 5.1. depicts the application's main user interface.

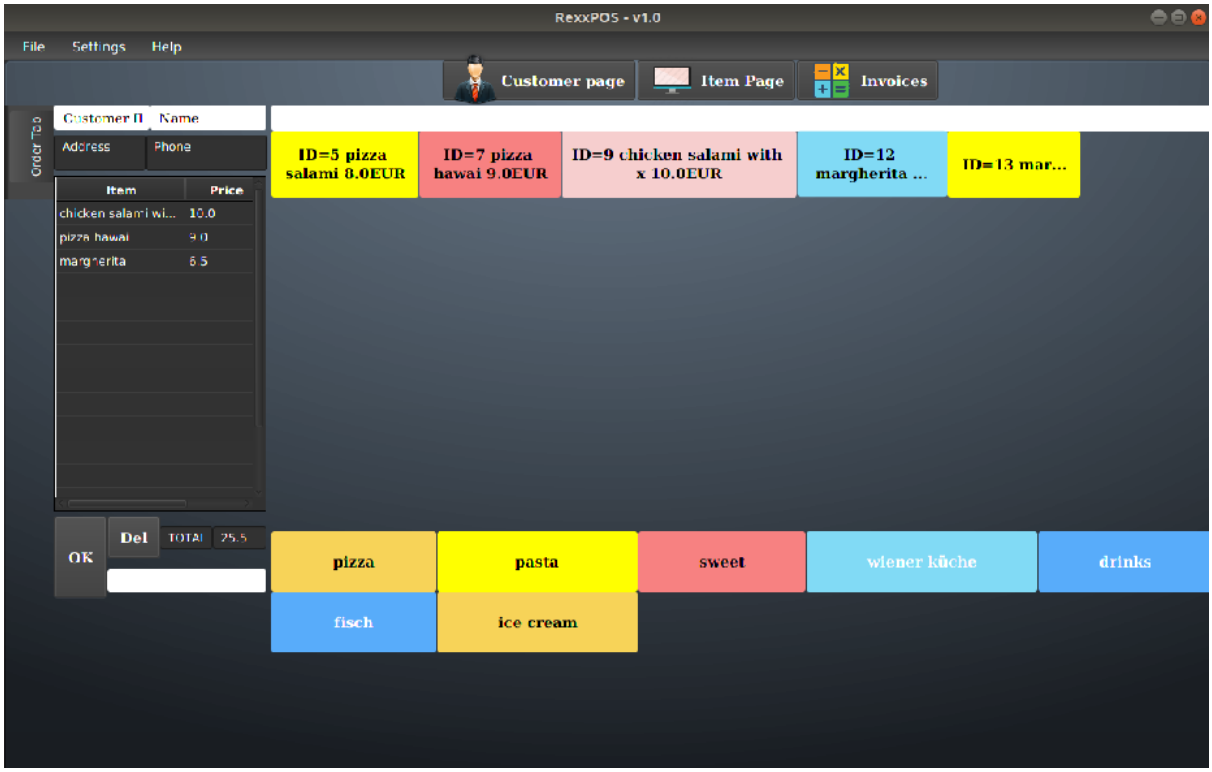


Figure 5.1: Mainpage (Dashboard) of the Application

The navigation between pages is done by using the buttons at the top center of the screen entitled "Customer page", "Item Page" and "Invoices". There are also the "File", "Settings" and "Help" menu items. Clicking "Settings" will open the "Settings" page where the company details can be edited.

Clicking "Customer page" will open "Customer" page in which customers can be added, searched, deleted etc. By selecting a customer and clicking "New order button" at that screen one can create a new order for that selected customer.

Figure 5.2 depicts the "Customer" page where customer searches or data maintenance can be done

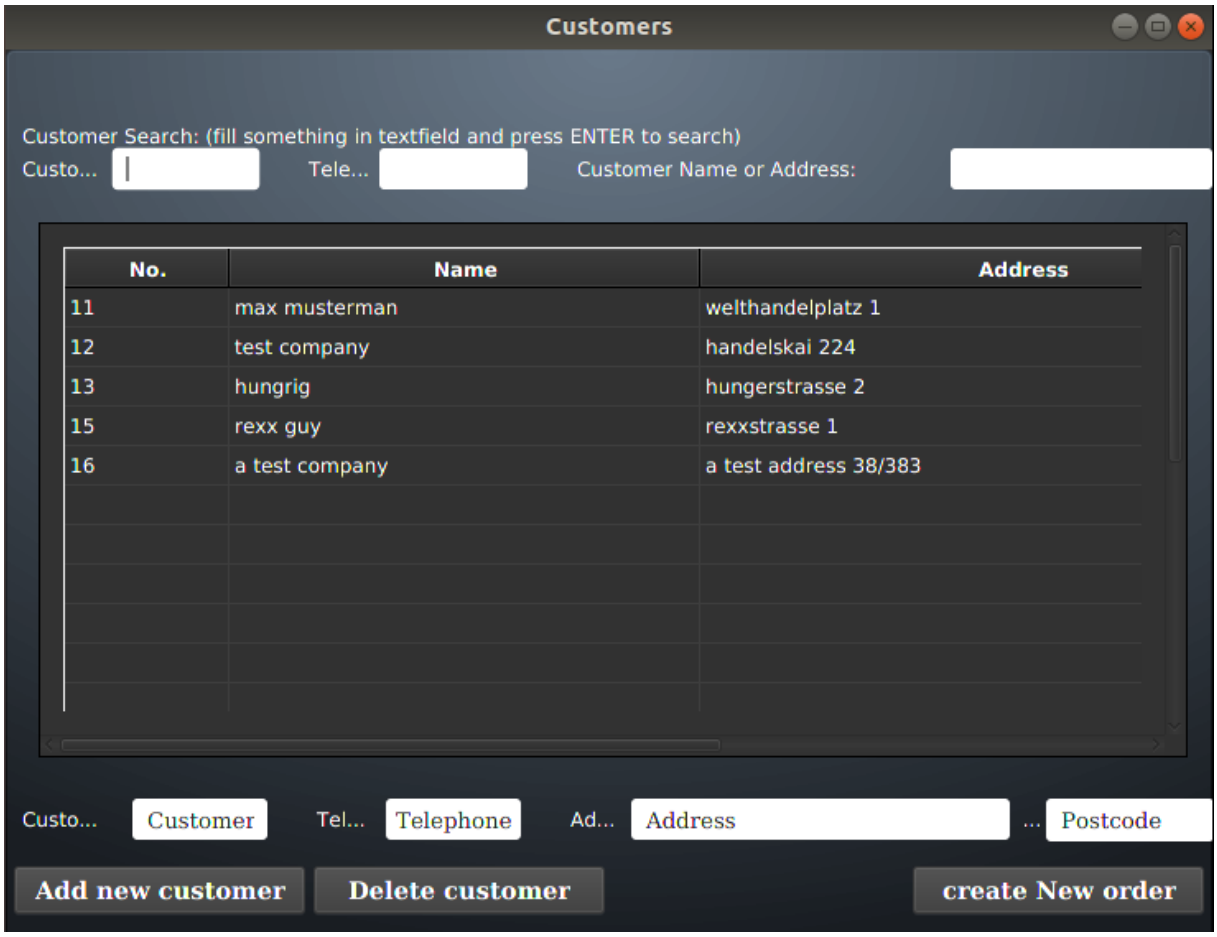


Figure 5.2: Customer Page for Editing, Deleting or Adding a Customer

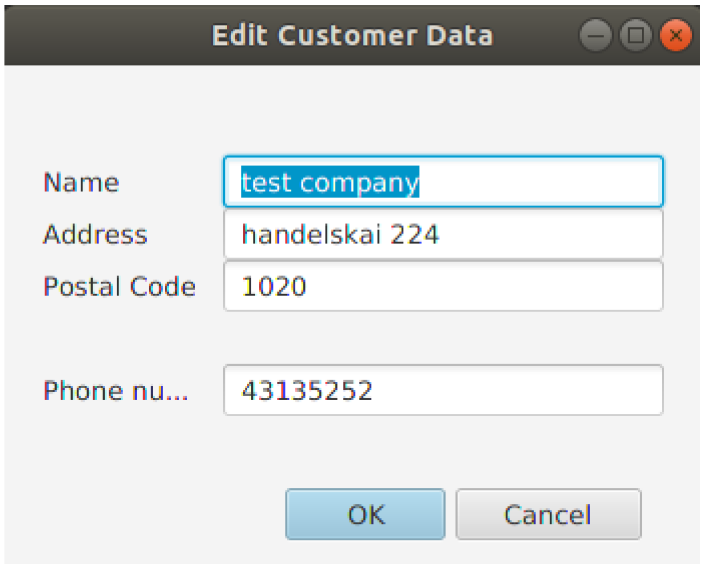


Figure 5.3: Edit Customer Data

By clicking the "Items" page, the user can see the items and the categories that are available.

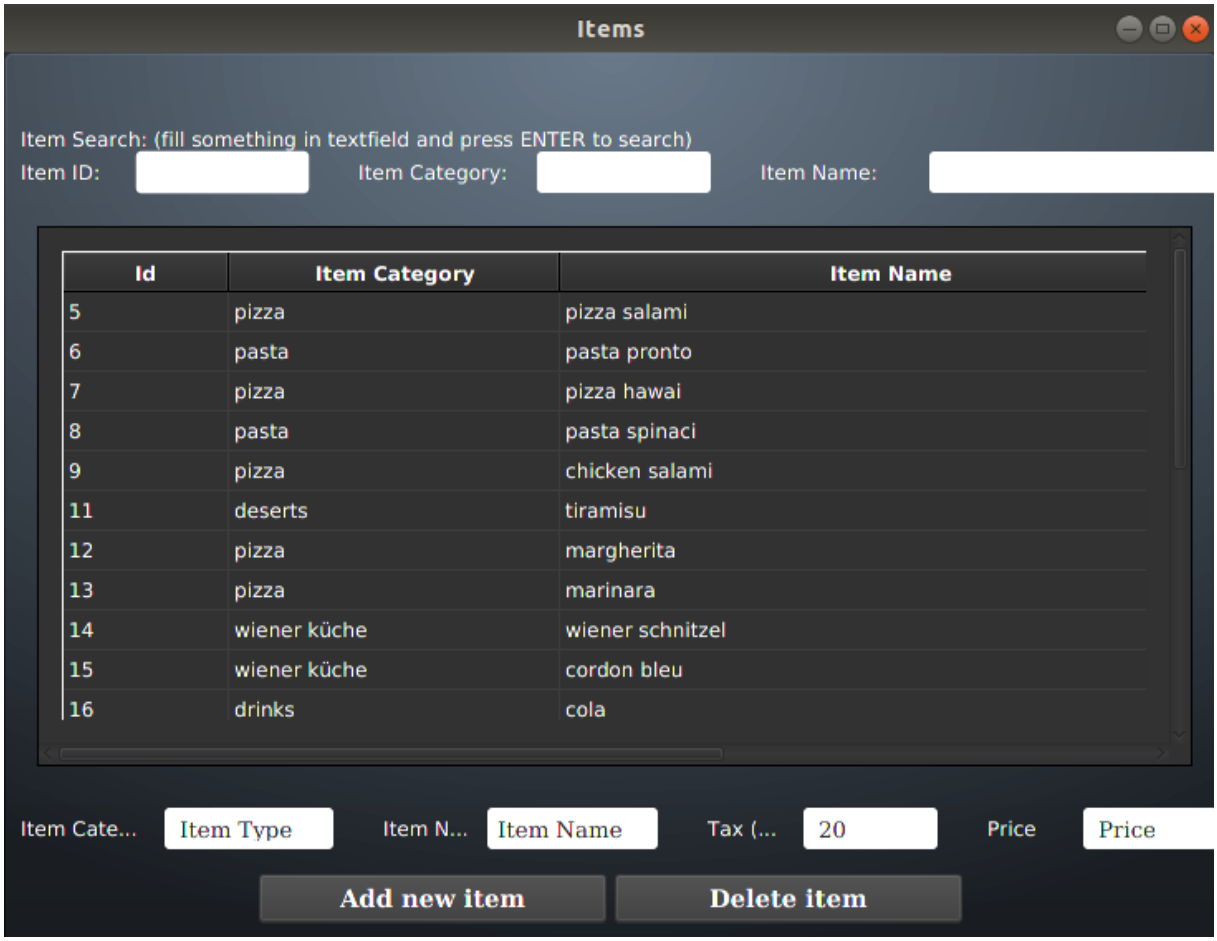


Figure 5.4: Item Page for Editing, Deleting or Adding an Item

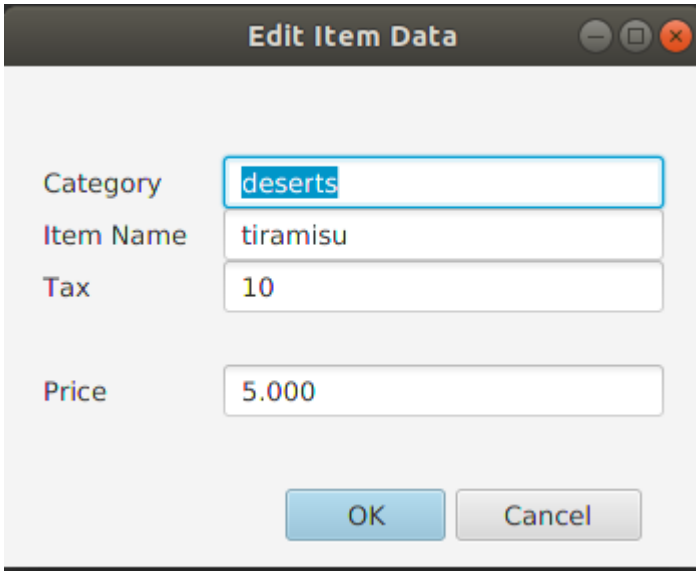


Figure 5.5: Edit Item Data

At "**Invoices Page**" shown at Figure 5.6., the user can show all invoices on a selected Date. The invoices can be shown, re-printed or cancelled if necessary.

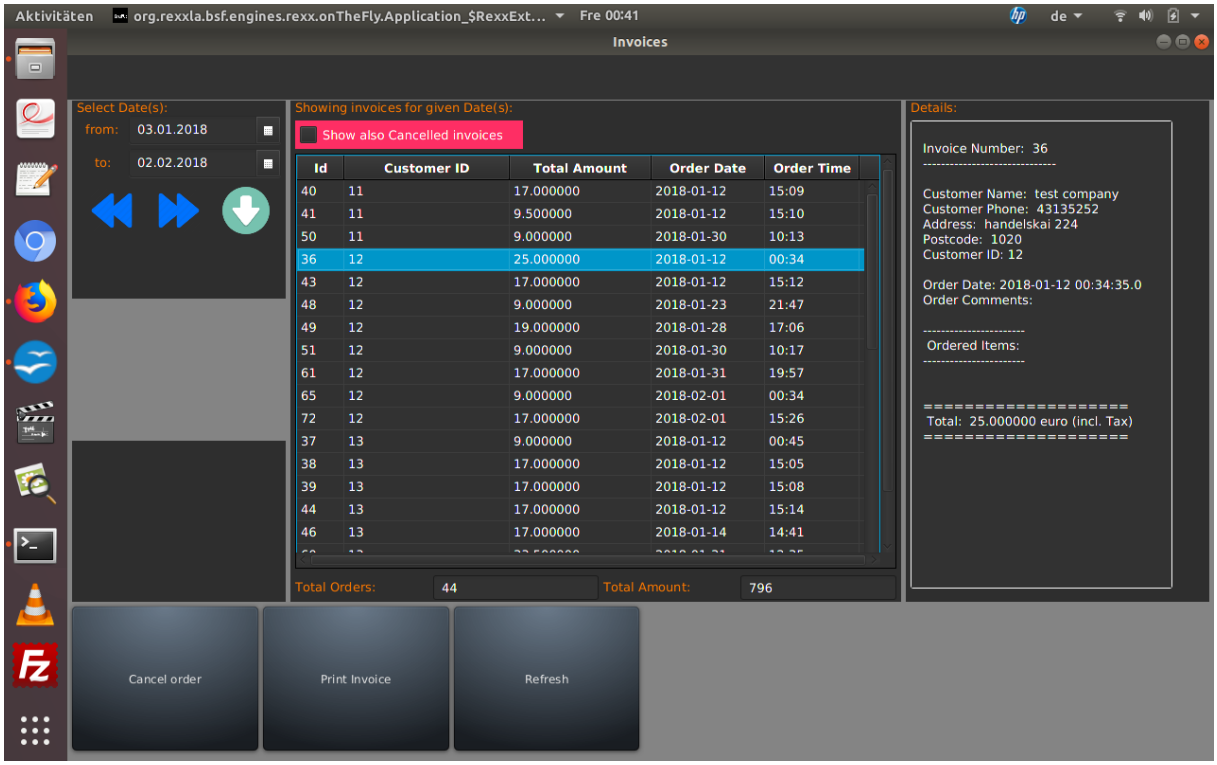


Figure 5.6: Invoices Page

At Main Page (Dashboard), there are buttons in different colors. These are Item Categories shown at bottom and Items in that category once it is clicked are shown at top. So user simply clicks a category to show the items and price of item in it. Once an item is clicked it will be added to the Table at middle left of screen. This is the list of items ordered by a customer. Each item added in to the Table shown with Description, Piece of items, Price of item and Tax amount. If one item is clicked more than once, than the Pieces of item will be increased.

There is also a comment box at the bottom of Table, in which a seller can add comments or special wishes of customers in to the receipt. At top of the Table there is shown the Customer details, such as Customer Number, Name, Address and Phone. These information is also printed on invoice as shipping details.

If seller wants to create order for another customer, he/she can simply open „Customer page“, select the customer and click „create new order“ button.

Invoice Printing Page shown at Figure 5.7. shows the step for printing invoice

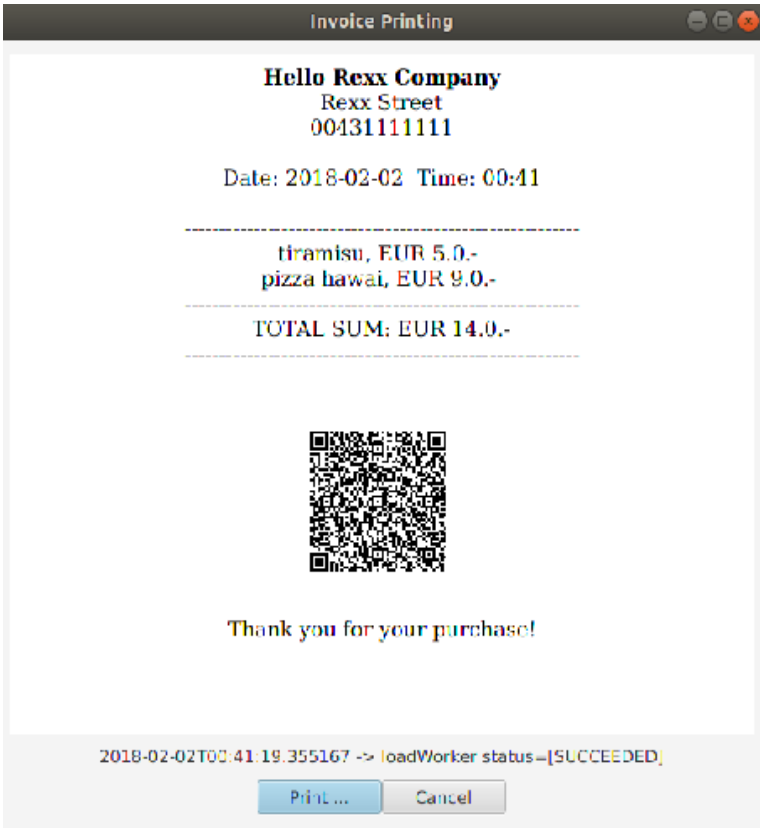


Figure 5.7: Invoice Page showing Invoice to be prin

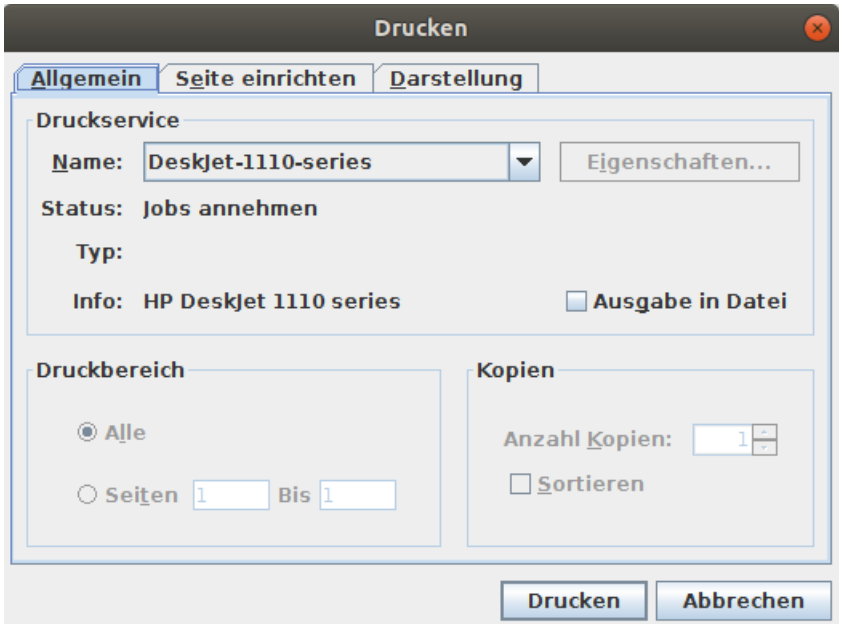


Figure 5.8: Print options window when Print button is clicked

The QR code shown on invoice in Figure 5.7. has been created thanks to following simple code. Snippet 5.2. depicts how it is done. As the invoice must have a QR code which is obligatory due to the new tax law, here is the sample code for creating a QR code and saving it in an image file.

```
call bsf.import "java.awt.Color","Color"
call bsf.import "java.awt.Graphics2D","Graphics2D"
call bsf.import "java.awt.image.BufferedImage","BufferedImage"
call bsf.import "java.io.File"
call bsf.import "java.io.IOException"
call bsf.import "java.util.Hashtable","Hashtable"
call bsf.import "javax.imageio.ImageIO","ImageIO"

call bsf.import "java.nio.file.FileSystems","FileSystems"
call bsf.import "java.nio.file.Path","Path"

call bsf.import "com.google.zxing.BarcodeFormat","BarcodeFormat"
call bsf.import "com.google.zxing.WriterException"
call bsf.import "com.google.zxing.client.j2se.MatrixToImageWriter","MatrixToImageWriter"
call bsf.import "com.google.zxing.qrcode.QRCodeWriter","QRCodeWriter"
call bsf.import "com.google.zxing.common.BitMatrix","BitMatrix"

call bsf.import "com.google.zxing.EncodeHintType","EncodeHintType"
call bsf.import "com.google.zxing.qrcode.decoder.ErrorCorrectionLevel", -
"ErrorCorrectionLevel"

qrCodeText ="_R1-AT100_CASHBOX-POS_Receipt-ID-21" -
"_2018-01-08T23:46:36_9,60_0,00_0,00_0,00_0,00_KCnacMHXtR4=" -
"_21650c9bb7c09a3a__" -
"YktlT0OSej8f2jOcQB67dMMC1DByqSIxCq2RF4Xo9Sa/5ugn1S0mwpKJXt" -
"6PntONG36FJ49rFALdn8bI6YNWCA=="

QR_CODE_IMAGE_PATH = "qrtest.png"
size=128
qrCodeWriter = .QRCodeWriter~new
bitMatrix = qrCodeWriter~encode(qrCodeText,.BarcodeFormat~QR_CODE, size, size)

--path=.Path
--fs=.FileSystems~getDefault()
--path=fs~getPath(QR_CODE_IMAGE_PATH)
path="qrtest.png"
qrFile = .bsf~new("java.io.File",path)
MIW=.MatrixToImageWriter
MIW~writeToFile(bitMatrix, "PNG", qrFile)

/*
filePath = "qrtest.png"

size = 128
fileType = "png"
qrFile = .bsf~new("java.io.File",filePath)
--createQRImage(qrFile, qrCodeText, size, fileType)

ench=.EncodeHintType
errc=.ErrorCorrectionLevel
hintMap = .bsf~new("java.util.Hashtable")
hintMap~put(ench~ERROR_CORRECTION, errc~L);
qrCodeWriter = .QRCodeWriter~new
byteMatrix = qrCodeWriter~encode(qrCodeText,.BarcodeFormat~QR_CODE, size, size, hintMap)

matrixWidth = byteMatrix~getWidth();
```

```
image = .bsf~new("java.awt.image.BufferedImage",matrixWidth,
matrixWidth,.BufferedImage~TYPE_INT_RGB);
image~createGraphics();
graphics = .Graphics2D
graphics = image~getGraphics();
graphics~setColor(.Color~WHITE);
graphics~fillRect(0, 0, matrixWidth, matrixWidth);
graphics~setColor(.Color~BLACK);

i=0
do while i < matrixWidth
   j=0
   do while j < matrixWidth
       if (byteMatrix~get(i, j)<> 0) then graphics~fillRect(i, j, 1, 1)
       j=j+1
   end
i=i+1
end

.ImageIO~write(image, fileType, qrFile)
*/
::REQUIRES BSF.CLS
```

Snippet 5.2: Sample Code to create a QR Code

Figure 5.9. shows the company informations "**Settings Page**":

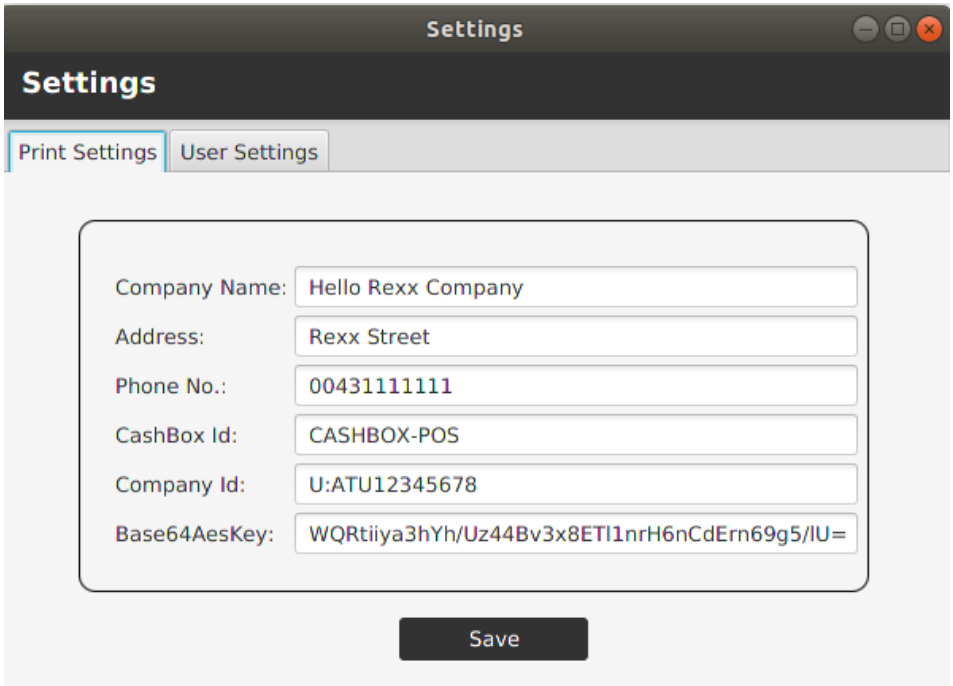


Figure 5.9: Edit Company details in Settings page

6. GUI Design and implementation

For Java™ there are different possibilities for designing such a GUI, AWT, Swing or JavaFX. For this GUI, JavaFX has been selected. The reason was easiness of designing the GUI with SceneBuilder Tool. As having no experience in Desktop programming, this tool makes Design process very easy. More Information on how to create a GUI with SceneBuilder can be found at Chapter 3.4.

In this Chapter, it will be explained how an FXML that has been created via Scenebuilder be edited, so that it can be simply controlled at a Rexx program. Please try not to make an error at FXML file, otherwise the program may give just a generic error code without giving what exact error has been made in FXML file. For instance if you add double fx:id with the same name into different controls, then the Rexx program will give a generic error message and you must yourself search for the error everywhere.

First step is opening an FXML file in a Text Editor. THen add the following lines shown below with a yellow background into the file. First "<?language rexx?>" line must be added into the FXML file, for instance here it has been added just before where Container starts.

Then, you will see some more texts with yellow background. These are fx:id of the input textfields or buttons. In the Rexx program these IDs will be called and Controllers, Handlers will be created. For instance, if one wants to read the Username entered then in the Rexxprogram the textfield with fx:id iUsername will be read. Or if we want to execute a method when Login button is clicked then, we must add an Eventlistener for the button with fx:id loginbutton. After that we can do actions when an Event like clicking on a Button happens.

Then you will see another Line with yellow Background: <fx:script source="contro.rex" /> This line means that the controller file for this FXML file is contro.rex. If you see login button, you will see an onAction Handler: onAction="slotDir=arg(arg()); call Loginaction

slotDir;" This means whenever this button is clicked, the method Loginaction will be called which is found in contro.rex.

One more thing to mention it, that Stylesheet which is added at the bottom of the FXML file or the Image Files at this page are having a "@" symbol at the start of the file name.

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.scene.paint.*?>
<?import javafx.scene.image.*?>
<?import javafx.geometry.*?>
<?import java.lang.*?>
<?import java.net.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>

<?language rexx?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="400.0" prefWidth="600.0" style="-fx-background-color: #DDDDDD;"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">

<fx:script source="controller.rex" />

<children>
      <AnchorPane prefHeight="106.0" prefWidth="600.0" style="-fx-background-color:
#4B4B4B;" />
      <AnchorPane layoutX="145.0" layoutY="49.0" prefHeight="288.0" prefWidth="312.0"
style="-fx-background-color: white; -fx-background-radius: 12px;">
         <children>
            <ImageView fitHeight="72.0" fitWidth="76.0" layoutX="120.0" layoutY="14.0"
pickOnBounds="true" preserveRatio="true">
               <image>
                  <Image url="@User.png" />
               </image>
      </ImageView>
        <TextField fx:id="iUsername" layoutX="61.0" layoutY="116.0" prefHeight="25.0" prefWidth="191.0" />
        <PasswordField fx:id="iPassword" layoutX="61.0" layoutY="172.0" prefHeight="25.0" prefWidth="191.0" />
      <Label layoutX="61.0" layoutY="99.0" text="Username:" />
      <Label layoutX="61.0" layoutY="155.0" text="Password:" />
        <Button layoutX="112.0" layoutY="219.0" mnemonicParsing="false" onAction="slotDir=arg(arg()); call
Loginaction slotDir;" prefHeight="32.0" prefWidth="89.0" style="-fx-background-color: #3196DE;" fx:id="loginbutton"
text="Login" textAlignment="CENTER" textFill="WHITE">
        <font>
                  <Font name="System Bold" size="12.0" />
               </font>
            </Button>
         </children>
      </AnchorPane>
   </children>
  <stylesheets>
    <URL value="@button.css" />
 </stylesheets>
         <fx:script source="put_FXID_objects_into.my.app.rex" />
</AnchorPane>
```

Snippet 6.1: Description of FXML Code of the Login.fxml

At the end of the FXML file there is another line with Yellow background. This is a very useful Rexx Script written by Prof. Rony G. Flatscher. This file is included also in the current Project. It simply adds all fx:ids into the Rexx program, so that it is not necessary to add each fx:id one by one.

The Controller file controller.rex which controls the FXML code above would look like this:

```
  /*  controller.rex sample code    */
  ::requires BSF.CLS
  ::routine LoginAction public
   use arg slotDir
   -- here Code  i.e. what to do when Login Button is clicked
```

Snippet 6.2: Code of a sample Controller.rex

This was a sample Code how to determine a Controller File directly in FXML File and create Handle actions which address the Routines in the Controller File.

In the current Project, however, there is no Controller File determined in FXML Files. All Handles and controllers are directly created as Classes and Methods in Rexx file.

As seen at below sample Code, the RxMainApplication Class has methods, attributes and without any controller file determined, the Rexx will control the FXML file. Please note that, if you want this kind of Control from Rexxfile itself, then you must delete the Line in FXML file which determines the controller.rex as a controller. Otherwise you will get error Message.

```
  .environment~my.app=.directory~new
-- directory to contain objects relevant to this application
  .my.app~bDebug=.false            /* if set to .true, "put_FXID_objects_into.my.app.rex"
will show
        all entries in ScriptContext Bindings on the console        */

  call bsf.import "javafx.fxml.FXMLLoader",                    "fx.FXMLLoader"
  call bsf.import "javafx.scene.Scene",                        "fx.Scene"
   -- create Rexx object that will control the application
  rexxApp=.RxMainApplication~new
  .my.app~mainApp=rexxApp   -- store the Rexx MainApp object in .my.app

     -- instantiate the abstract JavaFX class, the abstract "start"
  jRexxApp=BsfCreateRexxProxy(rexxApp, ,"javafx.application.Application")

     -- launch the application, invoke "start" and then stay up
  jRexxApp~launch(jRexxApp~getClass, .nil)  -- need to use this version

  ::requires "BSF.CLS"          -- get Java support
```

```
  /* ============================================
   implements the abstract method "start" for the Java class
"javafx.application.Application"
     (BSF4ooRexx also supplies another (trailing) slotDir (a Rexx Directory) argument, as
"start" is
     invoked from Java).
     It also controls
  */
  ::class RxMainApplication

     /* for tutorial Part 2: create an ObservableList
           and create and store Person data in it */
  ::attribute personData
  ::attribute kundenData
  ::attribute itemData
  ::attribute invoiceData
  ::attribute cartData
  ::attribute primaryStage
 -- stores the primaryStage supplied via the start method by JavaFX

  ::method init  -- constructor
    expose personData kundenData itemData invoiceData prefs personFilePath cartData   --
observableArrayList (maintained by TableView)

     -- create and save an ObservableList
    personData=.fx.FXCollections~observableArrayList
  kundenData=.fx.FXCollections~observableArrayList
  itemData=.fx.FXCollections~observableArrayList
  invoiceData=.fx.FXCollections~observableArrayList
  cartData=.fx.FXCollections~observableArrayList

  .my.app~cartData=cartData

     /* loads the fxml document defining the GUI elements,
         sets up a scene for it and shows it */
  ::method start    -- will be invoked by the "launch" method
    expose locale primaryStage personFilePath rootLayout
    use arg primaryStage  -- we get the stage to use for our UI

    bAlertsAvailable=(.java.lang.system~getProperty("javafx.runtime.version")>8)
    .my.app~bDialogsAlertsAvailable=bAlertsAvailable
                -- determine availability
    if bAlertsAvailable then    -- available since JavaFX 8u40
    do
    call bsf.import "javafx.scene.control.Alert",   "fx.Alert"
       /* note "AlertType" is public "inner" enum class,
    i.e. it is defined within the class "Alert", hence
     the JavaDocs refer to "AlertType"
      as "Alert.AlertType": however the Java compiler produces the
      class file as "Alert$AlertType"  */

  call bsf.import "javafx.scene.control.Alert$AlertType", "fx.Alert.Type"
    end

    if personFilePath=.nil then primaryStage~setTitle("RexxPOS - v1.1")
                        else primaryStage~title="RexxPOS - v1.1"

     -- create an URL for the FMXLDocument.fxml file
           -- (hence the protocol "file:")
    rootLayoutUrl=.bsf~new("java.net.URL", "file:RootLayout.fxml")
    rootLayout   =.fx.FXMLLoader~load(rootLayoutUrl) -- load the fxml
    .rootLayoutController~new(self)
                -- create an instance of the controller

    scene=.fx.scene~new(rootLayout)   -- create a scene for our document
  scene~getstylesheets()~add("file:button.css")
```

```
    primaryStage~setScene(scene)    -- set the stage to our scene

     -- add application icon
    img=.bsf~new("javafx.scene.image.Image", "file:logo.png")
   primaryStage~getIcons~add(img)

     -- load PersonOverview.fxml, place it into the rootLayout,

    overviewUrl=.bsf~new("java.net.URL", "file:menu1.fxml")
    overviewPage=.fx.FXMLLoader~load(overviewUrl)   -- load the fxml

    .my.app~overviewPage=overviewPage
    rootLayout~setCenter(overviewPage)
    .my.app~personOverviewController=.PersonOverviewController~new
  primaryStage~setMaximized(.true)

    primaryStage~show     -- now show the stage (and thereby our scene)

  ::method showSettings
    expose primaryStage personData btnPrintSettings iSaveUserSettings

    -- Load the fxml file and create a new stage for the popup
    SettingsUrl=.bsf~new("java.net.URL", "file:Settings.fxml")
    page  = .fx.FXMLLoader~load(SettingsUrl)
    scene = .fx.Scene~new(page)

    -- AnchorPane page = (AnchorPane) loader.load();
    dialogStage = .bsf~new("javafx.stage.Stage")
    dialogStage~setTitle("Settings")
    dialogStage~initModality(.fx.Modality~WINDOW_MODAL)

  --dialogStage~initOwner(primaryStage)
    dialogStage~setScene(scene)

    -- Set the persons into the controller
    controller = .PersonOverviewController~new
   -- controller~setPersonData(personData)
    dialogStage~show

sel1~getSettings()
```

Snippet 6.3: A Sample code to create a Controller directly in Rexx code

7. Database

As data saved in the POS System is preferred to be used also for accounting purposes, it is necessary to save the Data in a secure way, so that even after years access will be possible.

There are different possible Database solutions, NoSql, MySQL, SQLLite, MsSql etc.

The MySQL has been selected in this Project, as it is free to use under GPL licensing

It is very suitable for smaller databases. It has very low resource consumption therefore is perfect choice for this JavaFX based ooRexx project.

7.1. Database Structure:

Table 7.1. shows all available tables in the database:

```
mysql> show tables from pos;
+-----------------+
| Tables_in_pos   |
+-----------------+
| customer        |
| invoice         |
| invoice_details |
| item            |
| print_settings  |
| users           |
+-----------------+
```

Table 7.1 : All Tables in Database

The command shown at Table 7.1. is executed in "mysql command line tool" [MySQL1]

The Table 7.1 shows that there are six tables in database; „customer“, „invoice“, „invoice_details“, „item“, „print_settings“ and „users“ table.

In „customer“ table the details of a customer are saved as shown at table 7.2. The primary key is Customer_ID.
In „invoice“ table the details of an invoice are saved as shown at table 7.3. The primary key is ID. This ID is also a secondary key named as „InvoiceID“ column in „invoice_details“ table.
In „invoice_details“ table more details of an invoice are saved for instance, which items have been ordered etc. as shown at table 7.4.
In „item“ table the details of an item are saved as shown at table 7.5.
In „users“ table the login details of an user are saved as shown at table 7.6.
In „print_settings“ table the company details and printing details are saved as shown at table 7.7.

Schema of the Database is shown at Figure 7.1:

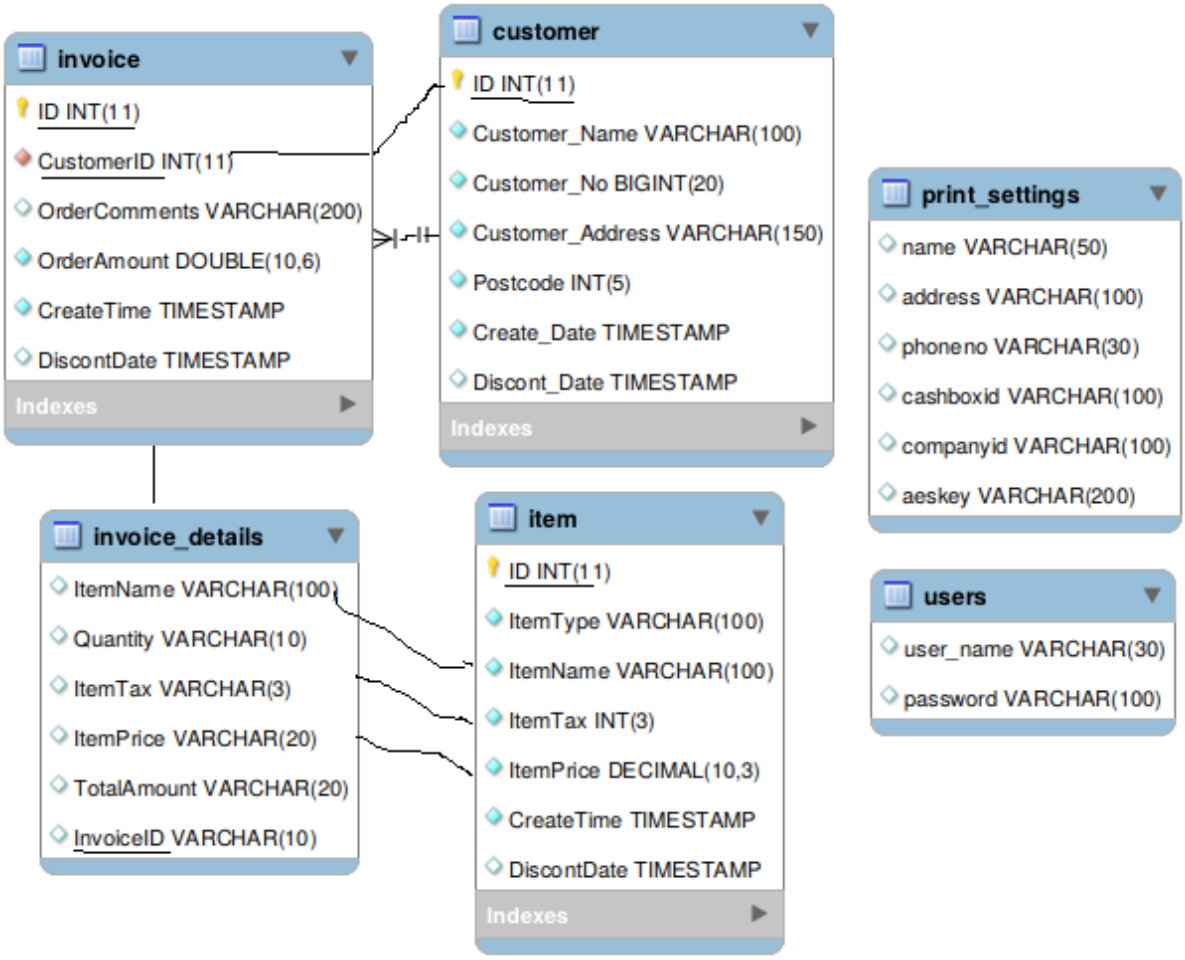


Figure 7.1: Schema of the Database

The Schema shown at Figure 7.1. has been created via "MySQL Workbench" program. MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X. [MySqlW]

It allows creating for instance Schema of the Database. In the Figure 7.1. relationships between different Tables are shown. For instance, the primary key "InvoiceID" of Table "Invoice" is used as a foreign key in the Table. "invoice_details" . The primary key "CustomerID" of table "Customers" is used as foreign key in Table "Invoice"

Figure 7.2. shows an **Entity Relationship Model** (ERM) based on relationships between tables.

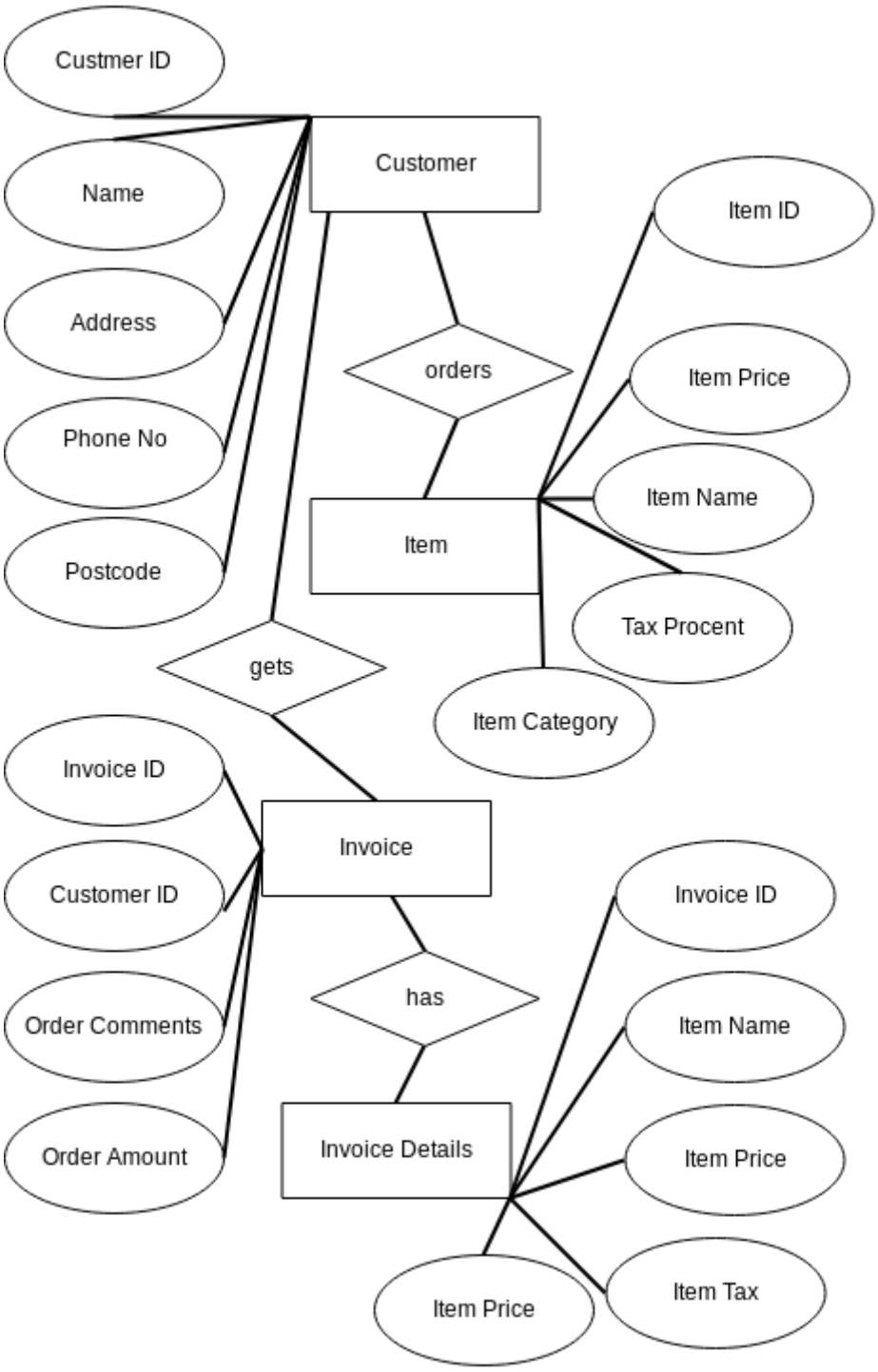


Figure 7.2: Entity Relationship Model (ERM)

According to Wikipedia, Entities can be thought of as nouns. Examples: a computer, an employee, a song, a mathematical theorem. A relationship captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns. Examples: an *owns* relationship between a company and a computer. [WikiER]

The **Customer Table** has the columns in detail shown at Table 7.2.

```
mysql> SHOW COLUMNS FROM pos.customer;
+------------------+-------------+------+-----+-------------------+----------------+
| Field            | Type        | Null | Key | Default           | Extra          |
+------------------+-------------+------+-----+-------------------+----------------+
| ID               | int(11)     | NO   | PRI | NULL              | auto_increment |
| Customer_Name    | varchar(100)| NO   |     | NULL              |                |
| Customer_No      | bigint(20)  | NO   |     | NULL              |                |
| Customer_Address | varchar(150)| NO   |     | NULL              |                |
| Postcode         | int(5)      | NO   |     | NULL              |                |
| Create_Date      | timestamp   | NO   |     | CURRENT_TIMESTAMP |                |
| Discont_Date     | timestamp   | YES  |     | NULL              |                |
+------------------+-------------+------+-----+-------------------+----------------+
```

Table 7.2 : Columns in Customer Table in Database

Invoice Table has the columns in detail shown at Table 7.3.

```
mysql> SHOW COLUMNS FROM pos.invoice;
+---------------+-------------+------+-----+-------------------+----------------+
| Field         | Type        | Null | Key | Default           | Extra          |
+---------------+-------------+------+-----+-------------------+----------------+
| ID            | int(11)     | NO   | PRI | NULL              | auto_increment |
| CustomerID    | int(11)     | NO   | MUL | NULL              |                |
| OrderComments | varchar(200)| YES  |     | NULL              |                |
| OrderAmount   | double(10,6)| NO   |     | NULL              |                |
| CreateTime    | timestamp   | NO   |     | CURRENT_TIMESTAMP |                |
| DiscontDate   | timestamp   | YES  |     | NULL              |                |
+---------------+-------------+------+-----+-------------------+----------------+
```

Table 7.3 : Columns in Invoice Table in Database

Invoice Details Table has the columns in detail shown at Table 7.4.

```
mysql> SHOW COLUMNS FROM pos.invoice_details;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| ItemName    | varchar(100) | YES  |     | NULL    |       |
| Quantity    | varchar(10)  | YES  |     | NULL    |       |
| ItemTax     | varchar(3)   | YES  |     | NULL    |       |
| ItemPrice   | varchar(20)  | YES  |     | NULL    |       |
| TotalAmount | varchar(20)  | YES  |     | NULL    |       |
| InvoiceID   | varchar(10)  | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
```

Table 7.4 : Columns in Invoce_details Table in Database

Item Table has the following columns in detail shown at Table 7.5.

```
mysql> SHOW COLUMNS FROM pos.item;
+-------------+--------------+------+-----+-------------------+----------------+
| Field       | Type         | Null | Key | Default           | Extra          |
+-------------+--------------+------+-----+-------------------+----------------+
| ID          | int(11)      | NO   | PRI | NULL              | auto_increment |
| ItemType    | varchar(100) | NO   |     | NULL              |                |
| ItemName    | varchar(100) | NO   |     | NULL              |                |
| ItemTax     | int(3)       | NO   |     | NULL              |                |
| ItemPrice   | decimal(10,3)| NO   |     | NULL              |                |
| CreateTime  | timestamp    | NO   |     | CURRENT_TIMESTAMP |                |
| DiscontDate | timestamp    | YES  |     | NULL              |                |
+-------------+--------------+------+-----+-------------------+----------------+
```

Table 7.5 : Columns in Item Table in Database

Users Table has following columns in detail shown at Table 7.6.

```
mysql> SHOW COLUMNS FROM pos.users;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| user_name | varchar(30)  | YES  |     | NULL    |       |
| password  | varchar(100) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
```

Table 7.6 : Columns in users Table in Database

Print Settings Table has following columns in detail shown at Table 7.7.

```
mysql> SHOW COLUMNS FROM pos.print_settings;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| name      | varchar(50)  | YES  |     | NULL    |       |
| address   | varchar(100) | YES  |     | NULL    |       |
| phoneno   | varchar(30)  | YES  |     | NULL    |       |
| cashboxid | varchar(100) | YES  |     | NULL    |       |
| companyid | varchar(100) | YES  |     | NULL    |       |
| aeskey    | varchar(200) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
```

Table 7.7 : Columns in Print_settings Table in Database

7.2. SQL Query Examples

To update (add, delete, edit) a Column in database „executeUpdate“ method is used. To select a column and show it executeQuery method has to be used. A simple Rexx code for making a database connection and doing a select or update command is shown at Snippet 7.1.

```
*----------------          LOAD DATABASE DRIVER  ---------------------------------------- */

  call bsf.import "java.sql.DriverManager", "driverMgr"
  call bsf.import "java.sql.Connection", "baglan"
  call bsf.import "java.sql.SQLException", "exc"
  call bsf.import "java.sql.Statement", "Sta"
  call bsf.import "java.sql.ResultSet", "ress"
  call bsf.import "com.mysql.jdbc.Field", "fi"
  call bsf.import "com.mysql.jdbc.SQLError", "sq"
  call bsf.import "com.mysql.jdbc.Driver", "jdbcDriver"
  .jdbcDriver~newinstance

  /*----------- DATABASE CONFIGURATION ----------- */

  .my.app~dbhost="localhost"   -- Set this for localhost
  --.my.app~dbhost="217.160.4.246" -- or this for Remote Host
  .my.app~dbname="databasename"
  .my.app~dbusr="databaseusername"
  .my.app~dbpass="databasepassword"

  /*  ------------------------------------------------------------ */

conn=.driverMgr~getConnection("jdbc:mysql://".my.app~dbhost":3306/" -
                              .my.app~dbname -
"? user=".my.app~dbusr"&password=".my.app~dbpass"&autoReconnect=true" -
               "&useSSL=false&verifyServerCertificate=false")

statement = conn~createStatement
sqlkoma ="select * from pos.print_settings ;"
kunde=statement~executeQuery(sqlkoma)

DO WHILE kunde~next
     iResturantName~setText(kunde~getString("name"))
     iAddress~setText(kunde~getString("address"))
     iPhoneNo~setText(kunde~getString("phoneno"))
END

say "Name of the company: " iResturantName
say "Address of the company: " iAddress
say "Phone number of the company: " iPhoneNo
```

Snippet 7.1: Code to Generate a Database Connection and Execute SQL Commands

Here are some other SQL commands used in this program:

1) Snippet 7.2. represents how to show Invoices between a Start Date and now:

```
"SELECT invoice.ID, customer.Customer_Name, invoice.OrderComments,

invoice.OrderAmount, invoice.CreateTime ,invoice.DiscontDate

FROM pos.invoice INNER JOIN pos.customer on invoice.CustomerID =customer.ID

WHERE invoice.DiscontDate is null and DATE(invoice.CreateTime) =

'"iInvoiceStart~getValue()"'
```

Snippet 7.2: SQL Command to Show Invoices from Date X till Now

2) Snippet 7.3. represents how to show Invoices between two Dates:

```
"SELECT invoice.ID, customer.Customer_Name, invoice.OrderComments,
invoice.OrderAmount, invoice.CreateTime ,invoice.DiscontDate
FROM pos.invoice INNER JOIN pos.customer on invoice.CustomerID =customer.ID
WHERE invoice.DiscontDate is null and DATE(invoice.CreateTime) >=
'"iInvoiceStart~getValue()"' and DATE(invoice.CreateTime)
<='"iInvoiceEnd~getValue()"'
```

Snippet 7.3: SQL Command to Show Invoices Between Two Dates

3) Snippet 7.4. represents how to show Invoice details:

```
"SELECT * FROM pos.invoice_details where InvoiceID='"  lOrderId
```

Snippet 7.4: SQL Command to Show Invoice Details

4) Snippet 7.5. represents how to get Customer Details with a given CustomerId:

```
"SELECT * FROM pos.customer where Discont_Date is null and id like
'%"lCustomerId"%'
```

Snippet 7.5: SQL Command to Show a Customer Information in Details

5) Snippet 7.6. represents how to get Item Categories:

```
"SELECT * FROM pos.item where DiscontDate is null and ItemType LIKE
'%"lItemType"%'
```

Snippet 7.6: SQL Command to get Available Item Categories

6) Snippet 7.7. represents how to get Details of a given Item:

```
"SELECT * FROM pos.item where DiscontDate is null and ID like '%"llItemId"%'
```

Snippet 7.7: SQL Command to Show an Item Information in Details

7) Snippet 7.8. represents how to search for Items:

```
"SELECT * FROM pos.item where DiscontDate is null and ItemName  LIKE
'%"llItemName"%'
```

Snippet 7.8: SQL Command to Search For an Item

8) Snippet 7.9. represents how to add new Item:

```
INSERT INTO pos.item(ItemType, ItemName, ItemTax, ItemPrice) VALUES
('"llItemType"','"llItemName"',"llItemTax", "llItemPrice")
```

Snippet 7.9: SQL Command to Add a New Item

9) Snippet 7.10. represents how to update an Item Price:

```
UPDATE pos.item SET ItemPrice ='"llItemPrice"'
WHERE ID ="llItemId
```

Snippet 7.10: SQL Command to Update an Item Price

10) Snippet 7.11. represents how to get all Items:

```
SELECT * FROM pos.item where DiscontDate is null
```

Snippet 7.11: SQL Command to Get All Items

11) Snippet 7.12. represents how to get all items under a category:

```
SELECT ItemName FROM pos.item Where ItemType ='"lItemTyp
```

Snippet 7.12: SQL Command to Get All Items of a Category

12) Snippet 7.13. represents how to add new Order:

```
INSERT INTO pos.invoice(CustomerID, OrderComments,OrderAmount)
VALUES ("lCustomerId", '" lOrderComments"',"lOrderAmount")
```

Snippet 7.13: SQL Command to Add a New Order

13) Snippet 7.14. represents how to show all Item Categories:

```
SELECT DISTINCT ItemType FROM pos.item
```

Snippet 7.14: SQL Command to Show All Item Categories

14) Snippet 7.15. represents how to search for a Customer using Phone number:

```
SELECT * FROM pos.customer
WHERE Customer_No=" + dCustomerNumberText~getText()
```

Snippet 7.15: SQL Command to Search for a Customer using Phone Number

8. Conclusion

This paper has attempted to illustrate how Java and Rexx can be used successfully through BSF: The author had no desktop programming experience before. It shows how easily one can create a desktop program using advantages of JavaFX and the human oriented programming language ooRexx.

The knowledge for writing the program presented in this paper has been acquired during visiting the lectures of Prof. Rony G. Flatscher at the Vienna University of Economics and Business in just one Semester. By visiting Business Programming 1 & Business Programming 2, the author has rapidly acquired Information how to easily create a desktop program.

The author appreciates the sample JavaFX Codes written by Prof. Rony G. Flatscher and Adrian Baginski which were very helpful during writing of the program in this project. Especially the fxml_99 was of great help, from which a lot of techniques had been learnt.

This paper is on one hand, intended for Austrian companies which want to have a free and open source POS Software, on the other hand is intended to help other readers who want to create great desktop applications through BSF4ooRexx.

9. Bibliography

[BSF4oo] SourceForge web site of BSF4ooRexx. Retrieved January 13, 2018, from https://sourceforge.net/projects/bsf4oorexx/

[Flat13] Flatscher R.G.: Introduction to Rexx and ooRexx. Wien: facultas wuv. (2013)

[Flat04] Flatscher R.G.: "Camouflaging Java as Object REXX", in: Proceedings of the "2004 International Rexx Symposium", Böblingen, Germany, May 3rd - May 6th, 2004

[Flat05] Flatscher R.G.: "Automating OpenOffice.org with ooRexx, Architecture, Gluing to Rexx using BSF4Rexx,", in: Proceedings of the "2005 International Rexx Symposium", Los Angeles, California, U.S.A., April 18th - April 21st, 2005

[FlatAJ] Lecture foils of Prof. Flatscher. Retrieved January 13, 2018, from http://wi.wu-wien.ac.at:8002/rgf/wu/lehre/autojava/material/foils/

[Kiri06] Kirisits, J.: "A SQL Cookbook". Retrieved January 13, 2018, from http://wi.wu-wien.ac.at:8002/rgf/diplomarbeiten/Seminararbeiten/2016/20160204_Kirisits_BSF4ooRexx-SQLCookbook.pdf

[MySQL1] The MySQL Command-Line Tool. Retrieved January 30, 2018, from https://dev.mysql.com/doc/refman/5.7/en/mysql.html

[MySQLW] The MySQL Workbench program. Retrieved January 30, 2018, from https://www.mysql.com/de/products/workbench/

[NCR] NCR Self-checkout. Retrieved January 30, 2018, from http://www.ncr-news.de/2016/04/06/self-checkout-in-5-sekunden-spar-baut-innovationsfuehrerschaft-mit-ncr-selbstbedienungskassen-aus/

[ooRexx] Open Object Rexx website. Retrieved January 13, 2018, from

http://www.oorexx.org/

[ooReDo] Open Object Rexx Documentation. Retrieved January 13, 2018, from http://www.oorexx.org/docs/

[Rexx] Mark Hessling's Rexx page. Retrieved January 13, 2018, from http://www.Rexx.org

[RexxIn] Resource for Rexx programmers. Retrieved January 13, 2018, from http://www.RexxInfo.org/

[RexxLA] Rexx Language Association web site. Retrieved January 13, 2018, from http://www.RexxLA.org

[RexxPG] ooRexx Programming Guide. Retrieved January 13, 2018, from http://wi.wu.ac.at:8002/rgf/tmp/201202/rexxpg.pdf

[Tiobe] TIOBE Index. Retrieved January 30, 2018, from https://www.tiobe.com/tiobe-index/

[WikiFl] Flowchart. Retrieved January 30, 2018, from https://en.wikipedia.org/w/index.php?title=Flowchart&oldid=821509874

[WikiJa] Java programming language. Retrieved January 30, 2018, from https://en.wikipedia.org/w/index.php?title=Java_(programming_language)&oldid=822953044

[WikiER] Entity relationship model. Retrieved January 30, 2018, from https://en.wikipedia.org/w/index.php?title=Entity%E2%80%93relationship_model&oldid=822911366

Appendix

Name of file:	Description:
readme.txt	: Contains information about the software and it's usage
start.rex	: The start file which runs the software
bin/pos.rxj	: The main file in which all controllers are stored.
bin/put_FXID_objects_into.my.app.rex	: A code written by Prof. Flatscher which automates FX:ID add process
/bin/json-rgf.cls	: A code written by Prof. Flatscher which reads/writes a JSON String
bin/item.fxml	: FXML design of the "Item" page
bin/menu1.fxml	: FXML design of the Main page
bin/menu2.fxml	: FXML design of the "Item" page
bin/kunden.fxml	: FXML design of the "Customer" page
bin/Settings.fxml	: FXML design of the "Settings" page
bin/Login.fxml	: FXML design of the "Login" page
bin/RootLayout.fxml	: FXML design of the "Root" page
bin/PersonEditDialog.fxml	: FXML design of the Item/customer editing page
bin/Invoice_printout.html	: HTML version of the printed invoice
bin/PersonPrinterDialog.fxml	: FXML design of the "Invoice" printing page
bin/button.css	: Cascading style sheets file
bin/customer.png	: Icon of the "Customer" page
bin/dashboard.png	: Icon of the main page
bin/img2.png	: Icon of the calculater
bin/img3.png	: Icon "Items"
bin/img4.png	: Icon "edit"
bin/img-search.png	: Icon "search"
bin/qrtest.png	: The QR code saved as image
bin/User.png	: Icon "user"
bin/on.png	: Icon "export"
bin/logo.png	: Logo of the software
bin/arrow.png	: Icon "next"
bin/arrow_back.png	: Icon "previous"
bin/lib/core-3.3.0.jar	: The Java archive of zxing core
bin/lib/javase-3.3.0.jar	: The Java archive of zxing
bin/lib/mysql-connector-java-5.1.45-bin.jar	: The JDBC Driver Java archive

bin/sql/pos.sql : The SQL dump file

Source code of the file "**start.rex**"

```
/*
   RexxPOS v1.1
   New Releases will be published at website: www.lieferex.at/rexxpos
   e-mail: h0553559@gmail.com

   author:  H. Kaya Aydin  (implemented from fxml99 sample of Prof. Rony G. Flatscher)
   date:    2018-01-08
   changes: 2018-01-13 Bugfixes

   purpose: this is RexxPOS, a Point of Sale System (German: Kassensystem) for Austria
            It allows a shop, restaurant etc. create a POS and have Invoices printed
            with QR Code.

        It has been coded during a Bachelor thesis.


            The structure of this program (classes, controller) had been implementet from
            the fxml_99 Sample found under bsf4oorexx/samples Folder
            The controllers are coded in this Rexx file (a "Rexx package") as Rexx classes.

        Prof. Rony G. Flatscher is greatly appreciated for this Sample (fxml_99)
            thanks this code, it is easy to do many things with BSF4ooRexx and JavaFX

   license: Apache License 2.0 (see at bottom)

   invoke:
        double click on start.rxj

   needs: ooRexx 5.0 or higher, and BSF4ooRexx 6.00 or higher

*/

/* ============================================================================
   FIND CURRENT FOLDER NAME AND CHANGE TO SUB DIRECTORY /BIN
============================================================================
*/

PARSE SOURCE . . name
dir=FILESPEC('D', name) || FILESPEC('P', name)
"cd "dir
"cd bin"

/* ============================================================================
   CHECK CLASSPATH and ADD .JAR FILES AUTOMATICALLY INTO IT IF NOT EXISTS
============================================================================
*/

if POS("mysql-connector",value("CLASSPATH",,"ENVIRONMENT"))=.false then
do
ver=SysVersion() -- get operating system version
    if ver~left(1)~upper="W" then -- a Windows OS
        do
            "set CLASSPATH=%CLASSPATH%;"directory() -
          "\lib\mysql-connector-java-5.1.45-bin.jar;"
        end
    else if ver~left(1)~upper="L" | ver~left(1)~upper="U" then -- a Linux OS
        do
            "export CLASSPATH=$CLASSPATH:"directory()         -
              "/lib/mysql-connector-java-5.1.45-bin.jar"
        end
    else if ver~left(1)~upper="M" | ver~left(1)~upper="D" then -- a MacOS
        do
            say "****************************************************************************"
            say "CLASSPATH Error:"
            say "The .jar archive mysql-connector-java-5.1.45-bin.jar or" -
   "newer must be "
```

```
            say "added in  CLASSPATH"
            say "*************************************************************************"
            say
            say "Please install official mysql driver at "  -
 "https://dev.mysql.com/downloads/connector/j/ "
            say
            say "Please check instruction on www how to edit CLASSPATH on Mac"
            pull .
        end
end


--if classp~contains("zxing")=.false then           -- check zxing .jar file
if POS("zxing",value("CLASSPATH",,"ENVIRONMENT"))=.false then

do
ver=SysVersion() -- get operating system version
    if ver~left(1)~upper="W" then -- a Windows OS
        do
            "set CLASSPATH=%CLASSPATH%;"directory()"\lib\javase-3.3.0.jar;"
        end
    else if ver~left(1)~upper="L" | ver~left(1)~upper="U" then -- a Linux OS
        do
            "export CLASSPATH=$CLASSPATH:"directory()"/lib/javase-3.3.0.jar"
        end
    else if ver~left(1)~upper="M" | ver~left(1)~upper="D" then -- a MacOS
        do
            say "*************************************************************************"
            say "CLASSPATH Error:"
            say "The .jar archive zxing.jar must be added in  CLASSPATH"
            say "This is necessary for QR Code Barcode creation"
            say "*************************************************************************"
            say
            say "The file allready exists in lib directory of RexxPOS"
            say "Please check instruction on www how to edit CLASSPATH on Mac"
            pull .
        end
end


--if classp~contains("zxing")=.false then           -- check zxing .jar file
if POS("core-3.3.0",value("CLASSPATH",,"ENVIRONMENT"))=.false then

do
ver=SysVersion() -- get operating system version
    if ver~left(1)~upper="W" then -- a Windows OS
        do
            "set CLASSPATH=%CLASSPATH%;"directory()"\lib\core-3.3.0.jar;"
        end
    else if ver~left(1)~upper="L" | ver~left(1)~upper="U" then -- a Linux OS
        do
            "export CLASSPATH=$CLASSPATH:"directory()"/lib/core-3.3.0.jar"
        end
    else if ver~left(1)~upper="M" | ver~left(1)~upper="D" then -- a MacOS
        do
            say "*************************************************************************"
            say "CLASSPATH Error:"
            say "The .jar archive zxing.jar must be added in  CLASSPATH"
            say "This is necessary for QR Code Barcode creation"
            say "*************************************************************************"
            say
            say "The file allready exists in lib directory of RexxPOS"
            say "Please check instruction on www how to edit CLASSPATH on Mac"
            pull .
        end
end

-- ======== CHECK CLASSPATH End ============================================
```

```
/* ============================================================================
   RUN MAIN APP   POS.RXJ
============================================================================
*/

call "pos.rxj"


/*
      ------------------------ Apache Version 2.0 license -------------------------
         Copyright 2018 H. Kaya Aydin

         Licensed under the Apache License, Version 2.0 (the "License");
         you may not use this file except in compliance with the License.
         You may obtain a copy of the License at

             http://www.apache.org/licenses/LICENSE-2.0

         Unless required by applicable law or agreed to in writing, software
         distributed under the License is distributed on an "AS IS" BASIS,
         WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
         See the License for the specific language governing permissions and
         limitations under the License.
      ----------------------------------------------------------------------------
*/
```

pos.rxj

```
/*
   RexxPOS v1.1
   New Releases will be published at website: www.lieferex.at/rexxpos
   e-mail: h0553559@gmail.com

   author:  H. Kaya Aydin  (implemented from fxml99 sample of Prof. Rony G. Flatscher)
   date:    2018-01-08
   changes: 2018-01-12 Bugfixes

   purpose: this is RexxPOS, a Point of Sale System (German: Kassensystem) for Austria
            It allows a shop, restaurant etc. create a POS and have Invoices printed
            with QR Code.

             It has been coded during a Bachelor thesis.


            The structure of this program (classes, controller) had been implementet from
            the fxml_99 Sample found under bsf4oorexx/samples Folder
            The controllers are coded in this Rexx file (a "Rexx package") as Rexx classes.

             Prof. Rony G. Flatscher is greatly appreciated for this Sample (fxml_99)
            thanks this code, it is easy to do many things with BSF4ooRexx and JavaFX

   license: Apache License 2.0 (see at bottom)

   invoke:

            double click on start.rex

   needs: ooRexx 5.0 or higher, and BSF4ooRexx 6.00 or higher

*/

say "Loading RexxPOS..."

.environment~my.app=.directory~new  -- directory to contain objects relevant to this application
.my.app~bDebug=.false              /* if set to .true, "put_FXID_objects_into.my.app.rex" will show
                                      all entries in ScriptContext Bindings on the console
*/

/* ============================================================
          DATABASE CONFIGURATION
============================================================== */

--.my.app~dbhost="localhost"   -- Set this for localhost
.my.app~dbhost="217.160.4.246" -- or this for demo Remote Host
.my.app~serverport="3306"      -- mysql server port
.my.app~dbname="pos"           -- database name
.my.app~dbusr="posrexx"        -- database username
.my.app~dbpass="Rexx1234"      -- database password

/*
============================================================
*/

   -- import JavaFX classes that we may use more often
call bsf.import "javafx.fxml.FXMLLoader",                      "fx.FXMLLoader"
call bsf.import "javafx.scene.Scene",                          "fx.Scene"
call bsf.import "javafx.beans.property.SimpleStringProperty",  "fx.SimpleStringProperty"
call bsf.import "javafx.beans.property.SimpleIntegerProperty", "fx.SimpleIntegerProperty"

call bsf.import "javafx.collections.FXCollections",            "fx.FXCollections"
call bsf.import "javafx.stage.Modality",                       "fx.Modality"

call bsf.import "java.util.ArrayList",                         "fx.ArrayList"
```

```
call bsf.import "java.util.Arrays",                                  "fx.Arrays"

call bsf.import "java.text.SimpleDateFormat",                       "SimpleDateFormat"
call bsf.import "java.time.LocalDate",                              "LocalDa"
call bsf.import "java.util.Date",                                    "jDate"


call bsf.import "java.awt.Color",                                        "Color"
call bsf.import "java.awt.Graphics2D",                                   "Graphics2D"
call bsf.import "java.awt.image.BufferedImage",                        "BufferedImage"
call bsf.import "java.io.File"
call bsf.import "java.io.IOException"
call bsf.import "java.util.Hashtable",                                   "Hashtable"

call bsf.import "javax.imageio.ImageIO",                                 "ImageIO"

call bsf.import "java.nio.file.FileSystems","FileSystems"
call bsf.import "java.nio.file.Path","Path"

call bsf.import "com.google.zxing.BarcodeFormat","BarcodeFormat"
call bsf.import "com.google.zxing.WriterException"
call bsf.import "com.google.zxing.client.j2se.MatrixToImageWriter","MatrixToImageWriter"
call bsf.import "com.google.zxing.qrcode.QRCodeWriter","QRCodeWriter"
call bsf.import "com.google.zxing.common.BitMatrix","BitMatrix"

call bsf.import "com.google.zxing.EncodeHintType",                     "EncodeHintType"

call bsf.import "com.google.zxing.qrcode.decoder.ErrorCorrectionLevel","ErrorCorrectionLevel"

call bsf.import "java.sql.DriverManager",                              "driverMgr"
call bsf.import "java.sql.Connection"
call bsf.import "java.sql.SQLException"
call bsf.import "java.sql.Statement"
call bsf.import "java.sql.ResultSet"
call bsf.import "com.mysql.jdbc.Field"
call bsf.import "com.mysql.jdbc.SQLError"
call bsf.import "com.mysql.jdbc.Driver",                                 "jdbcDriver"

.jdbcDriver~newinstance   -- create new instance of mysql jdbc driver

   -- create Rexx object that will control the application
rexxApp=.RxMainApplication~new
if arg()>0 then      -- if a command line JSON filename was given, use it to set the attribute
do
   parse arg jsonFile
   rexxApp~personFilePath=strip(jsonFile) -- save filePath, if any, in the attribute "personFilePath"
end
.my.app~mainApp=rexxApp        -- store the Rexx MainApp object in .my.app


   -- instantiate the abstract JavaFX class, the abstract "start" method will be served by rexxApp
jRexxApp=BsfCreateRexxProxy(rexxApp, ,"javafx.application.Application")

   -- launch the application, invoke "start" and then stay up until the application closes
jRexxApp~launch(jRexxApp~getClass, .nil)  -- need to use this version of launch in order to work

::routine dbconstr public
conn=.driverMgr~getConnection("jdbc:mysql://".my.app~dbhost":".my.app~serverport"/".my.app~dbname"?
user=".my.app~dbusr"&password=".my.app~dbpass"&autoReconnect=true&u??
seSSL=false&verifyServerCertificate=false")
return conn

::requires "BSF.CLS"          -- get Java support
::requires "json-rgf.cls"     -- get JSON support for ooRexx (to load and store person data)


/* ============================================================================== */
/* implements the abstract method "start" for the Java class "javafx.application.Application"
```

```
   (BSF4ooRexx also supplies another (trailing) slotDir (a Rexx Directory) argument, as "start" is
   invoked from Java).
 */
::class RxMainApplication

   /* for tutorial Part 2: create an ObservableList and create and store Person data in it */
::attribute personData
::attribute kundenData
::attribute itemData
::attribute invoiceData
::attribute cartData
::attribute primaryStage   -- stores the primaryStage supplied via the start method by JavaFX

   -- tutorial, part 5
::attribute personFilePath -- .nil or the file path to the file containing the attribute records (in
JSON format)

::method init  -- constructor
  expose personData kundenData itemData invoiceData prefs personFilePath cartData   --
observableArrayList (maintained by TableView)

   -- create and save an ObservableList
personData=.fx.FXCollections~observableArrayList
kundenData=.fx.FXCollections~observableArrayList
itemData=.fx.FXCollections~observableArrayList
invoiceData=.fx.FXCollections~observableArrayList
cartData=.fx.FXCollections~observableArrayList

.my.app~cartData=cartData

   /* loads the fxml document defining the GUI elements, sets up a scene for it and shows it */
::method start    -- will be invoked by the "launch" method
  expose locale primaryStage personFilePath rootLayout
  use arg primaryStage  -- we get the stage to use for our UI

   -- starting with JavaFX 8u40 true dialogs got introduced; taking advantage of them if running on
Java 1.8 or higher
   -- get the JavaFX runtime version (only available when running it), e.g. "8.0.111-b14"

  bAlertsAvailable=(.java.lang.system~getProperty("javafx.runtime.version")>8)
  .my.app~bDialogsAlertsAvailable=bAlertsAvailable  -- determine availability
  if bAlertsAvailable then    -- available since JavaFX 8u40
  do
      call bsf.import "javafx.scene.control.Alert",                "fx.Alert"
         -- note "AlertType" is public "inner" enum class, i.e. it is defined within the class
"Alert", hence
         -- the JavaDocs refer to "AlertType" as "Alert.AlertType": however the Java compiler
produces the
         -- class file as "Alert$AlertType"
      call bsf.import "javafx.scene.control.Alert$AlertType",       "fx.Alert.Type"
  end

  if personFilePath=.nil then primaryStage~setTitle("RexxPOS - v1.1")
                         else primaryStage~title="RexxPOS - v1.1"

   -- create an URL for the FMXLDocument.fxml file (hence the protocol "file:")
  rootLayoutUrl=.bsf~new("java.net.URL", "file:RootLayout.fxml")
  rootLayout   =.fx.FXMLLoader~load(rootLayoutUrl) -- load the fxml document
  .rootLayoutController~new(self)   -- create an instance of the controller

  scene=.fx.scene~new(rootLayout)   -- create a scene for our document
scene~getstylesheets()~add("file:button.css")
  primaryStage~setScene(scene)      -- set the stage to our scene

   -- add application icon
  img=.bsf~new("javafx.scene.image.Image", "file:logo.png")
 primaryStage~getIcons~add(img)
```

```
   -- load PersonOverview.fxml, place it into the rootLayout, create the Rexx object controlling the
PersonOverview form
  overviewUrl=.bsf~new("java.net.URL", "file:menu1.fxml")
  overviewPage=.fx.FXMLLoader~load(overviewUrl)   -- load the fxml document, AnchorPage (root)
returned
  .my.app~overviewPage=overviewPage
  rootLayout~setCenter(overviewPage)
  .my.app~personOverviewController=.PersonOverviewController~new
primaryStage~setMaximized(.true)

  primaryStage~show        -- now show the stage (and thereby our scene)
/*
call SysSleep 5
overviewUrl2=.bsf~new("java.net.URL", "file:menu2.fxml")
  overviewPage2=.fx.FXMLLoader~load(overviewUrl2)   -- load the fxml document, AnchorPage (root)
returned
  .my.app~overviewPage2=overviewPage2
  rootLayout~setCenter(overviewPage2)
  .my.app~personOverviewController=.PersonOverviewController~new

  primaryStage~show        -- now show the stage (and thereby our scene)
primaryStage~setMaximized(.true)
*/

--say primaryStage~getOwner()

/* tutorial, step 5: get filePath via Java's Preferences mechanism   */
::method getPersonFilePath
  expose prefs
  filePath=prefs~get("filePath", .nil)
  return filePath

/* tutorial, step 5: set filePath via Java's Preferences mechanism (supplying .nil will remove the
preference)   */
::method setPersonFilePath
  expose prefs primaryStage
  use arg filePath

  if file<>.nil then
  do
     prefs~put("filePath", filePath)
     primaryStage~title="RexxPOS -" --filePath
  end
  else
  do
     prefs~remove("filePath")
     primaryStage~title="RexxPOS"
  end


/* tutorial, step 5: using JSON instead of XML as file format */
::method loadPersonDataFromFile
  expose personData     -- get access to the ObservableList
  use arg filePath

  s=.stream~new(filepath)~~open("read")   -- open stream
  jsonStr=s~charIn(1, s~query("size"))    -- read entire content
  s~close                                 -- close stream
  persons=.Json~new~fromJSON(jsonStr)
  personData~clear      -- clear the ObservableList
  do p over persons     -- add persons to ObservableList
     personData~add(.person~new( p["firstName"], p["lastName"], p["street"], p["postalCode"],
p["city"], p["birthday"]))
  end

/* tutorial, step 5: using JSON instead of XML as file format */
::method savePersonDataToFile
  expose personData     -- get access to the ObservableList
```

```
  use arg filePath

  arr=.array~new
  do p over personData  -- iterate over ObservableList
     dir=.directory~new
     dir["firstName"] =p~firstName
     dir["lastName"]  =p~lastName
     dir["street"]    =p~street
     dir["postalCode"]=p~postalCode
     dir["city"]      =p~city
     dir["birthday"]  =p~birthday

    arr~append(dir)
  end

  s=.stream~new(filepath)~~open("write replace")   -- open stream for replacement
  jsonStr=s~charout(.json~new~toJson(arr)) -- create and write JSON string to stream
  s~close                                  -- close stream
  self~setPersonFilePath(filePath)     -- save this one as preference for next load



/* Load the fxml-form, define a stage for it, create an instance of the Rexx class controlling
   the fxml-form.
   Unlike the original tutorial we do not create the dialog and controller over and over again, we
   just cache them using two attributes and reuse them.
*/
::method showPersonEditDialog
  expose pedDialog pedController primaryStage
  use arg person

  if \var("pedDIALOG") then   -- not yet created, setup PersonEdit data
  do
     -- Load the fxml file and create a new stage for the popup
     personEditDialogUrl=.bsf~new("java.net.URL", "file:PersonEditDialog.fxml")
     page  = .fx.FXMLLoader~load(personEditDialogUrl)
     scene = .fx.Scene~new(page)

     -- AnchorPane page = (AnchorPane) loader.load();
     dialogStage = .bsf~new("javafx.stage.Stage")
     dialogStage~setTitle("Edit Customer Data")
     dialogStage~initModality(.fx.Modality~WINDOW_MODAL)



     dialogStage~setScene(scene)

     -- Set the person into the controller
     pedController = .PersonEditDialogController~new
     pedController~dialogStage=dialogStage   -- this way the pedController is able to close the stage
  end

  pedController~setPerson(person)
  pedController~okClicked=.false          -- make sure we start out with .false

  -- Show the dialog and wait until the user closes it
  dialogStage~showAndWait
  return pedController~OkClicked



::method showItemEditDialog
  expose pedDialog pedController primaryStage
  use arg person

  if \var("pedDIALOG") then   -- not yet created, setup PersonEdit data
  do
     -- Load the fxml file and create a new stage for the popup
```

```
    personEditDialogUrl=.bsf~new("java.net.URL", "file:PersonEditDialog.fxml")
    page  = .fx.FXMLLoader~load(personEditDialogUrl)
    scene = .fx.Scene~new(page)

    -- AnchorPane page = (AnchorPane) loader.load();
    dialogStage = .bsf~new("javafx.stage.Stage")
    dialogStage~setTitle("Edit Item Data")
    dialogStage~initModality(.fx.Modality~WINDOW_MODAL)



    dialogStage~setScene(scene)

    -- Set the person into the controller
    pedController = .ItemEditDialogController~new
    pedController~dialogStage=dialogStage   -- this way the pedController is able to close the stage
  end

  pedController~setPerson(person)
  pedController~okClicked=.false          -- make sure we start out with .false

  -- Show the dialog and wait until the user closes it
  dialogStage~showAndWait
  return pedController~OkClicked


::method showSettings
  expose primaryStage personData btnPrintSettings iSaveUserSettings

  -- Load the fxml file and create a new stage for the popup
  SettingsUrl=.bsf~new("java.net.URL", "file:Settings.fxml")
  page  = .fx.FXMLLoader~load(SettingsUrl)
  scene = .fx.Scene~new(page)

  -- AnchorPane page = (AnchorPane) loader.load();
  dialogStage = .bsf~new("javafx.stage.Stage")
  dialogStage~setTitle("Settings")
  dialogStage~initModality(.fx.Modality~WINDOW_MODAL)


  dialogStage~setScene(scene)

  -- Set the persons into the controller
  controller = .PersonOverviewController~new
 -- controller~setPersonData(personData)
  dialogStage~show

self~getSettings()

::method getSettings
expose primaryStage iAddress iPhoneNo iResturantName iCashboxId iCompanyId iBase64AesKey iPassword -
iUsername btnPrintSettings iSaveUserSettings conn driverMgr

siDir=.my.app~Settings.fxml


/* MySQL Connection  */
if (conn="CONN") then do
    conn=dbconstr()
end

if (conn~isClosed) then do
    conn=dbconstr()
end


statement = conn~createStatement
sqlkoma ="select * from pos.print_settings ;"
```

```
kunde=statement~executeQuery(sqlkoma)

rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


iAddress=siDir~iAddress
iPhoneNo=siDir~iPhoneNo
iResturantName=siDir~iResturantName
iCashboxId=siDir~iCashboxId
iCompanyId=siDir~iCompanyId
iBase64AesKey=siDir~iBase64AesKey
iPassword=siDir~iPassword
iUsername=siDir~iUsername

btnPrintSettings=siDir~btnPrintSettings ~~setOnAction(rp)
iSaveUserSettings=siDir~iSaveUserSettings ~~setOnAction(rp)

datam=.fx.FXCollections~observableArrayList


itemsicinde=.array~new

DO WHILE kunde~next
    iResturantName~setText(kunde~getString("name"))
    iAddress~setText(kunde~getString("address"))
    iPhoneNo~setText(kunde~getString("phoneno"))
    iCashboxId~setText(kunde~getString("cashboxid"))
    iCompanyId~setText(kunde~getString("companyid"))
    iBase64AesKey~setText(kunde~getString("aeskey"))
END

::method btnPrintSettings
expose btnPrintSettings primaryStage iAddress iPhoneNo iResturantName iCashboxId iCompanyId
iBase64AesKey iPassword iUsername btnPrintSettings iSaveUserSettings

siDir=.my.app~Settings.fxml

iAddress=siDir~iAddress
iPhoneNo=siDir~iPhoneNo
iResturantName=siDir~iResturantName
iCashboxId=siDir~iCashboxId
iCompanyId=siDir~iCompanyId
iBase64AesKey=siDir~iBase64AesKey
iPassword=siDir~iPassword
iUsername=siDir~iUsername
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")

btnPrintSettings=siDir~btnPrintSettings ~~setOnAction(rp)
iSaveUserSettings=siDir~iSaveUserSettings ~~setOnAction(rp)
/* MySQL Connection  */

if (conn="CONN") then do
    conn=dbconstr()
end

if (conn~isClosed) then do
    conn=dbconstr()
end

statement = conn~createStatement
sqlkoma ="delete from pos.print_settings;"
kunde=statement~executeUpdate(sqlkoma)
```

```
sqlkoma="INSERT INTO pos.print_settings(name, address, phoneno, cashboxid, companyid, aeskey) VALUES
('"iResturantName~getText()"', '"iAddress~getText()"','"iPhoneNo~getText()"',
'"iCashboxId~getText()"', '"iCompanyId~getText()"', '"iBase64AesKey~getText()"');"




kunde=statement~executeUpdate(sqlkoma)

btnPrintSettings~setText("DONE!")




::method iSaveUserSettings
expose iSaveUserSettings
--say geai

::method showPersonPrinterDialog
  expose primaryStage personData cartData dInvoiceComments dTotalPiece dCustomerIdText
use arg dTotalPiece,dCustomerIdText

poDir=.my.app~menu1.fxml
dInvoiceComments    =poDir~dInvoiceComments

dInvoiceComments~setText("QR Code saved!")

cartData=.my.app~cartData

/* MySQL Connection  */
if (conn="CONN") then do
    conn=dbconstr()
end

if (conn~isClosed) then do
    conn=dbconstr()
end


statement = conn~createStatement
sqlkoma ="select * from pos.print_settings ;"




kunde=statement~executeQuery(sqlkoma)

DO WHILE kunde~next
    iResturantName=kunde~getString("name")
    iAddress=kunde~getString("address")
    iPhoneNo=kunde~getString("phoneno")
    iCashboxId=kunde~getString("cashboxid")
    iCompanyId=kunde~getString("companyid")
    iBase64AesKey=kunde~getString("aeskey")
END

/* QR CODE  */

qrCodeText ="_R1-AT100_"iCashboxId"_Receipt-ID-21" -
"_"left(.dateTime~new,10)"-
08T"left(right(.dateTime~new,15),8)"_9,60_0,00_0,00_0,00_0,00_KCnacMHXtR4=" -
"_21650c9bb7c09a3a__YktlT0OSej8f2jOcQB67dMMC1DByqSIxCq2RF4Xo9Sa/5ugn1S0mwpKJXt" -
"6PntONG36FJ49rFALdn8bI6YNWCA=="

QR_CODE_IMAGE_PATH = "qrtest.png"
```

```
size=128
qrCodeWriter = .QRCodeWriter~new
bitMatrix = qrCodeWriter~encode(qrCodeText,.BarcodeFormat~QR_CODE, size, size)

--path=.Path
--fs=.FileSystems~getDefault()
--path=fs~getPath(QR_CODE_IMAGE_PATH)
path="qrtest.png"
qrFile = .bsf~new("java.io.File",path)
MIW=.MatrixToImageWriter
MIW~writeToFile(bitMatrix, "PNG", qrFile)

/*
filePath = "qrtest.png"

size = 128
fileType = "png"
qrFile = .bsf~new("java.io.File",filePath)
--createQRImage(qrFile, qrCodeText, size, fileType)

ench=.EncodeHintType
errc=.ErrorCorrectionLevel
hintMap = .bsf~new("java.util.Hashtable")
hintMap~put(ench~ERROR_CORRECTION, errc~L);
qrCodeWriter = .QRCodeWriter~new
byteMatrix = qrCodeWriter~encode(qrCodeText,.BarcodeFormat~QR_CODE, size, size, hintMap)

matrixWidth = byteMatrix~getWidth();
image = .bsf~new("java.awt.image.BufferedImage",matrixWidth,
matrixWidth,.BufferedImage~TYPE_INT_RGB);
image~createGraphics();
graphics = .Graphics2D
graphics = image~getGraphics();
graphics~setColor(.Color~WHITE);
graphics~fillRect(0, 0, matrixWidth, matrixWidth);
graphics~setColor(.Color~BLACK);

i=0
do while i < matrixWidth
   j=0
   do while j < matrixWidth
       if (byteMatrix~get(i, j)<> 0) then graphics~fillRect(i, j, 1, 1)
       j=j+1
   end
i=i+1
end

.ImageIO~write(image, fileType, qrFile)
*/

/* END QR CODE  */

  -- Load the fxml file and create a new stage for the popup
  Url=.bsf~new("java.net.URL", "file:PersonPrinterDialog.fxml")
  page  = .fx.FXMLLoader~load(Url)
  scene = .fx.Scene~new(page)

  -- AnchorPane page = (AnchorPane) loader.load();
  dialogStage = .bsf~new("javafx.stage.Stage")
  dialogStage~setTitle("Invoice Printing")
  dialogStage~initModality(.fx.Modality~WINDOW_MODAL)


  dialogStage~setScene(scene)

  -- Set the persons into the controller
  controller = .PersonPrinterDialogController~new(cartData, dialogStage,dTotalPiece,dCustomerIdText)
  controller~createAndLoadPrintData -- render the print data
```

```
  dialogStage~showAndWait


::method handle
expose btnPrint btnCancel dialogStage showInvoicesbtn showKundenbtn showItemsbtn  -
btnPrintSettings iSaveUserSettings
  use arg event

  tgtObjectName=event~getTarget~objectname

if tgtObjectName=btnPrintSettings  ~objectName then self~btnPrintSettings

if tgtObjectName=iSaveUserSettings  ~objectName then self~iSaveUserSettings


/* ============================================================================== */
/* for tutorial part 2: define the PersonOverviewController

   This Rexx class controls the interaction with the fxml form named "PersonOverview.fxml".

*/
::class PersonOverviewController
::attribute iInvoicetableView
::attribute firstNameColumn
::attribute lastNameColumn
::attribute cCustomerTable
::attribute dTableBill
::attribute personData
::attribute kundenData
::attribute itemData
::attribute invoiceData
::attribute cartData
::attribute primaryStage

::method init
  expose ItemsData primaryStage iInvoicetableView firstNameColumn lastNameColumn firstNameLabel
lastNameLabel streetLabel -
                     postalCodeLabel cityLabel birthdayLabel  -
                     btnNew dCustomerPageDashboard btnDelete showItemsbtn sItemPageButton itemsinit
itemsicinde dira -
             showInvoicesbtn showDashboardbtn showKundenbtn dFoodMenuButton1 dFoodMenuButton2
dFoodMenuButton3 -
             itemsinit  itemsicinde dira dFoodMenuButton1 dFoodMenuButton2 dFoodMenuButton3
dFoodMenuButton4 -
 dFoodMenuButton5 dFoodMenuButton6 okle delle dInvoiceComments -
 dFoodMenuButton7 dFoodMenuButton8 dFoodMenuButton9  dFoodMenuButton10 dFoodMenuButton11
dFoodMenuButton12 -
 dFoodMenuButton13 dFoodMenuButton14 dFoodMenuButton15  dFoodMenuButton16 dFoodMenuButton17
dFoodMenuButton18 -
 dFoodMenuButton19 dFoodMenuButton20 dPizzaItemButton1 dPizzaItemButton2 dPizzaItemButton3
dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20 cCustomerTable -
cartData personData itemData invoiceData kundenData dTableBill dTableBillColumn1 dTableBillColumn2 -
dTableBillColumn3 aCartItem ibisCheckBox butnext butprev butdown

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
iTableBillColumn1   =poDir~iTableBillColumn1
  iTableBillColumn2=poDir~iTableBillColumn2
```

```
  iTableBillColumn3 =poDir~iTableBillColumn3
--iInvoicetableView=poDir~iInvoicetableView
 cartData=.fx.FXCollections~observableArrayList

.my.app~cartData=cartData



   -- add self as the ChangeListener (we have its method "changed" implemented) and EventHandler
(method "handle" implemented)
  rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")

  /* set callback for creating TableView rows to which we add us as an event listener to get mouse
click events on the rows */
  --iInvoicetableView~setOnMouseClicked(rp) -- allows us to get the mouse clicked event

   /* set callback for filling the individual TableView cells */
 --
firstNameColumn~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("firstNameProperty")
, ,"javafx.util.Callback"))
 -- lastNameColumn
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("lastNameProperty"),  ,"javafx.util
.Callback"))

   /* for tutorial part 3: fill-in the Person's details */
 -- self~showPersonDetails(.nil)   -- clear the labels
   --  auto resize columns
  --iInvoicetableView~setColumnResizePolicy(iInvoicetableView~CONSTRAINED_RESIZE_POLICY)    --
TableView constant

   -- add self as the ChangeListener (we have its method "changed" implemented) and EventHandler
(method "handle" implemented)
--  iInvoicetableView~getSelectionModel ~selectedItemProperty ~addListener(rp)

   -- add us as the event handler to the buttons, the event handler (method "handle") will be able to
get at the pressed button
--  btnNew   ~setOnAction(rp)
  --dCustomerPageDashboard  ~setOnAction(rp)
 -- btnDelete~setOnAction(rp)

--  self~setItems      -- fill the TableView

showItemsbtn=poDir~showItemsbtn ~~setOnAction(rp)
showKundenbtn=poDir~showKundenbtn ~~setOnAction(rp)
showInvoicesbtn=poDir~showInvoicesbtn ~~setOnAction(rp)
showDashboardbtn=poDir~showDashboardbtn ~~setOnAction(rp)
okle=poDir~okle~~setOnAction(rp)
delle=poDir~delle~~setOnAction(rp)
dInvoiceComments=poDir~dInvoiceComments


--say iInvoiceEnd
/* MySQL Connection  */

if (conn="CONN") then do
conn=dbconstr()
end

if (conn~isClosed) then do
conn=dbconstr()
end


statement = conn~createStatement
sqlkoma ="SELECT DISTINCT ItemType FROM pos.item WHERE DiscontDate is null ;"


  poDir=.my.app~menu1.fxml
```

```
kunde=statement~executeQuery(sqlkoma)

ItemsData=.fx.FXCollections~observableArrayList

itemsicinde=.array~new
    dira=.directory~new
/*
     dira["firstName"] =p~firstName
     dira["lastName"]  =p~lastName
     dira["street"]    =p~street
     dira["postalCode"]=p~postalCode
     dira["city"]      =p~city
     dira["birthday"]  =p~birthday
     arr~append(dira)
 */
k=1
DO WHILE kunde~next


    datas=("gae,34")
    INTERPRET 'poDir~dFoodMenuButton'k'~setText(kunde~getString("ItemType"))'
    INTERPRET 'dFoodMenuButton'k'=poDir~dFoodMenuButton'k' ~~setOnAction(rp)'


    k=k+1
END

n=k
DO WHILE n<21
    INTERPRET 'poDir~dFoodMenuButton'n'~setVisible(.false)'

    n=n+1
END

z=1
DO WHILE z<k+1
    --INTERPRET 'poDir~dFoodMenuButton'z'~setStyle("-fx-padding: 75 90 25 20;")'
    --INTERPRET 'poDir~dFoodMenuButton'z'~setStyle("-fx-padding: 75 90 25 20;")'
    INTERPRET 'menukat=poDir~dFoodMenuButton'z'~getText'
    sqlkoman ="SELECT * FROM pos.item Where ItemType ='"menukat"';"

    itemsini=statement~executeQuery(sqlkoman)
    it=0
    --say itemsini~count
    DO WHILE itemsini~next
        it=it+1
        --ItemsData~add(.Item4Sale~new(itemsini~getString("ItemName"),
itemsini~getDouble("ItemPrice"), menukat))


ItemsData~add(.Item4Sale~new(itemsini~getString("ID"),itemsini~getString("ItemType"),itemsini~getStri
ng("ItemName"),itemsini~getString("ItemTax"), itemsini~getDouble("ItemPrice")))


--itemData~add(.Item4Sale~new(kunde~getString("ID"),kunde~getString("ItemType"),kunde~getString("Item
Name"),kunde~getString("ItemTax"),kunde~getString("ItemPrice")))

        itemsinit.z.it=itemsini~getString("ItemName")
        --say z it " = " itemsinit.z.it
        --say itemsini~getString("ItemName")
        --INTERPRET "dira"z"~"it"="itemsini~getString("ItemName")
        --say itemsinit.z.it
    END
    --say z "had" it "items in it."

    z=z+1
END
```

```
itemsicinde~append(dira)

::method okle
expose mainApp aCustomer dCustomerIdText dCustomerNameText dCustomerAddressText dCustomerNumberText
dTotalPiece
poDir=.my.app~menu1.fxml
dCustomerIdText=poDir~dCustomerIdText
dTotalPiece=poDir~dTotalPiece

if (dCustomerIdText~getText="Customer ID") then
do
        say "please first select a customer"
         lf="0a"x
          text="Error"lf-
             lf"No customer selected!" lf -
          lf"Please first open Customer page"lf -
         lf"select a Customer and click Create new order"lf

          if .my.app~bDialogsAlertsAvailable then
          do
             alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

             alert~setTitle("RexxPOS")
             alert~setHeaderText(.nil)
             alert~setContentText(text)
             alert~showAndWait
          end
          else  -- use BSF.CLS' .bsf.dialog utility class
          do
             -- .bsf.dialog is defined in BSF.CLS and uses swing
            .bsf.dialog~messageBox(text, "RexxPOS", "information")
          end
    end
    else
    do

    if (dTotalPiece~getText="") then
    do
        say "please add something in cart"
         lf="0a"x
          text="Error"lf-
             lf"No items added to cart! Total=0" lf -
          lf"Please click on some items"lf -
         lf"and add it into cart"lf

          if .my.app~bDialogsAlertsAvailable then
          do
             alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

             alert~setTitle("RexxPOS")
             alert~setHeaderText(.nil)
             alert~setContentText(text)
             alert~showAndWait
          end
          else  -- use BSF.CLS' .bsf.dialog utility class
          do
             -- .bsf.dialog is defined in BSF.CLS and uses swing
            .bsf.dialog~messageBox(text, "RexxPOS", "information")
          end
    end
    else
    do
        /* MySQL Connection  */
            if (conn="CONN") then do
                conn=dbconstr()
            end

            if (conn~isClosed) then do
```

```
                conn=dbconstr()
            end

            statement = conn~createStatement
            sqlkoma ="INSERT INTO pos.invoice(CustomerID, OrderComments,OrderAmount) VALUES
('"dCustomerIdText~getText"','  ', "dTotalPiece~getText")"




            kunde=statement~executeUpdate(sqlkoma)


                .my.app~mainApp~showPersonPrinterDialog(dTotalPiece~getText,dCustomerIdText~getText)

    end
end

::method delle
expose cartData dTableBill dTotalPiece

poDir=.my.app~menu1.fxml

dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece

cartData~clear
dTableBill~setItems(cartData)
dTotalPiece~setText("")

::method dPizzaItemButton1
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece

dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton1~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton1~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    do

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)


        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))
```

```
    end

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)

--controller = .CartController~new


--say poDir~dPizzaItemButton1~getText

::method dPizzaItemButton2
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton2~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton2~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    do

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    end

end
```

```
 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton3
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton3~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton3~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    do

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    end

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))
```

```
dTableBill~setItems(cartData)


::method dPizzaItemButton4
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3









getID=poDir~dPizzaItemButton4~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)

::method dPizzaItemButton5
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml
```

```
  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3



--cartData~add(.aCartItem~new(poDir~dPizzaItemButton5~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton5~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton6
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3



--cartData~add(.aCartItem~new(poDir~dPizzaItemButton6~getText,"1","12","20","12"))
```

```
getID=poDir~dPizzaItemButton6~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton7
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton7~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton7~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO
```

```
        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton8
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton8~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton8~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end
```

```
 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton9
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton9~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton9~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))
```

```
dTableBill~setItems(cartData)


::method dPizzaItemButton10
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3



--cartData~add(.aCartItem~new(poDir~dPizzaItemButton10~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton10~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton11
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3
```

```
  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3



--cartData~add(.aCartItem~new(poDir~dPizzaItemButton11~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton11~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton12
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3
```

```
--cartData~add(.aCartItem~new(poDir~dPizzaItemButton12~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton12~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton13
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton13~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton13~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData
```

```
    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton14
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton14~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton14~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END
```

```
end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton15
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton15~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton15~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
```

```
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton16
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3



--cartData~add(.aCartItem~new(poDir~dPizzaItemButton16~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton16~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton17
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3
```

```
  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3



--cartData~add(.aCartItem~new(poDir~dPizzaItemButton17~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton17~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton18
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3
```

```
--cartData~add(.aCartItem~new(poDir~dPizzaItemButton18~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton18~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton19
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3



--cartData~add(.aCartItem~new(poDir~dPizzaItemButton19~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton19~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)
```

```
do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))



    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)


::method dPizzaItemButton20
expose dFoodItemButton1 dTableBillColumn3  Item4Sale aCartItem  -
cartData personData dTotalPiece itemData ItemsData invoiceData kundenData dTableBill
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3



  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  poDir=.my.app~menu1.fxml

  dTableBill    =poDir~dTableBill
dTotalPiece =poDir~dTotalPiece
dTableBillColumn1=poDir~dTableBillColumn1
dTableBillColumn2=poDir~dTableBillColumn2
dTableBillColumn3=poDir~dTableBillColumn3




--cartData~add(.aCartItem~new(poDir~dPizzaItemButton20~getText,"1","12","20","12"))



getID=poDir~dPizzaItemButton20~getText
needle=pos(" ",getID)
subst=substr(getID,4,needle-4)

do u over ItemsData

    if (u~Item4SaleID == subst) then
    DO

        ki=dTotalPiece~getText() ; if ki="" then ki=0 ; vi=ki + u~Item4SalePrice ;
dTotalPiece~setText(vi)

        cartData~add(.aCartItem~new(u~Item4SaleName,u~Item4SalePrice,u~Item4SaleTax))
```

```
    END

end


 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))


dTableBill~setItems(cartData)



::method dFoodMenuButton1
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 -
dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice




kateg=poDir~dFoodMenuButton1~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'
        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton2
expose poDir itemsinit. itemsinit itemicinde dira ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
```

```
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)


kateg=poDir~dFoodMenuButton2~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'
        ax=ax+1
    END
end
nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("fa")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'
    nx=nx+1
END
::method dFoodMenuButton3
expose poDir itemsinit. itemsinit itemicinde dira ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)

do i over dira
INTERPRET "say dira~"i
INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira~"i")"
end
kateg=poDir~dFoodMenuButton3~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN

    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'
```

```
    nx=nx+1
END

::method dFoodMenuButton4
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton4~getText
--say kateg
ax=1
    do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

    ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton5
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
INTERPRET "say dira.1."i
INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end
```

```
kateg=poDir~dFoodMenuButton5~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton6
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton6~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton7
```

```
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton7~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton8
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton8~getText
--say kateg
ax=1
do i over ItemsData
```

```
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton9
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton9~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton10
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
```

```
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton10~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton11
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton11~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
```

```
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton12
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton12~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton13
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20
```

```
--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton13~getText
--say kateg
ax=1
do i over ItemsData
if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton14
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton14~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'
```

```
        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton15
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton15~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton16
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
```

```
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton16~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton17
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton17~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end
```

```
nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton18
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton18~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton19
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
```

```
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton19~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'

    nx=nx+1
END
::method dFoodMenuButton20
expose poDir itemsinit. itemsinit itemicinde dira dira. ItemsData dPizzaItemButton1 dPizzaItemButton2
dPizzaItemButton3 dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20

--use arg itemsinit,itemsicinde,dira
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
poDir=.my.app~menu1.fxml
poDir~dPizzaItemGrid~setVisible(.true)
--say i~Item4SaleName  EUR i~Item4SalePrice


do i over dira
    INTERPRET "say dira.1."i
    INTERPRET "poDir~dPizzaItemButton"i"~~setText(dira.1."i")"
end

kateg=poDir~dFoodMenuButton20~getText
--say kateg
ax=1
do i over ItemsData
    if (i~Item4SaleKat == kateg) THEN
    DO
        --say i~Item4SaleName  EUR i~Item4SalePrice
        INTERPRET "poDir~dPizzaItemButton"ax "~setText('ID='i~Item4SaleID i~Item4SaleName
i~Item4SalePrice'EUR')"
        INTERPRET "poDir~dPizzaItemButton"ax "~setVisible(.true)"
        INTERPRET 'dPizzaItemButton'ax'=poDir~dPizzaItemButton'ax' ~~setOnAction(rp)'

        ax=ax+1
    END
end

nx=ax
DO WHILE nx<21
    INTERPRET 'poDir~dPizzaItemButton'nx'~setText("")'
    INTERPRET 'poDir~dPizzaItemButton'nx'~setVisible(.false)'
```

```
    nx=nx+1
END

::method setItems
  expose iInvoicetableView
  iInvoicetableView~setItems(.my.app~mainApp~invoiceData) -- add current personData

/* for tutorial part 3: fill-in the Person's details */

::method showItems
  expose primaryStage dialogStage rootLayout personData showItemsbtn sItemPageButton showInvoicesbtn
showDashboardbtn showKundenbtn sItemTable -
sItemTableColumn1 sItemTableColumn2 sItemTableColumn3 sItemTableColumn4 sItemTableColumn5
sItemTypeAdd sItemNameAdd -
sItemAmountAdd sItemDiscountAdd sItemAddAction sItemDeleteAction Item4Sale itemData invoiceData
kundenData -
sIdSearchText sItemSubTypeText sItemTypeText
use arg rootLayout, primaryStage

primaryStage=showItemsbtn~getScene()~getWindow()
SettingsUrl=.bsf~new("java.net.URL", "file:item.fxml")
  page  = .fx.FXMLLoader~load(SettingsUrl)
  scene = .fx.Scene~new(page)
dialogStage = .bsf~new("javafx.stage.Stage")
--SettingsUrl=.bsf~new("java.net.URL", "file:item.fxml")
  --showItemspage  = .fx.FXMLLoader~load(SettingsUrl)
 dialogStage~setTitle("Items")
dialogStage~setScene(scene)
dialogStage~initModality(.fx.Modality~WINDOW_MODAL)
controller = .ItemsController~new
  dialogStage~show
dialogStage~setMaximized(.true)

self~getItems()

::method handleEditItem
expose sItemTable person

 skDir=.my.app~item.fxml
sItemTable=skDir~sItemTable
  selectedPerson = sItemTable~getSelectionModel~getSelectedItem
  if selectedPerson <> .nil then
  do
      person=BSFRexxProxy(selectedPerson) -- unbox Rexx object
      okClicked = .my.app~mainApp~showItemEditDialog(person)
      if okClicked=.true then
          self~showPersonDetails(person)
  end
  else
  do
      if .my.app~bDialogsAlertsAvailable then
      do
         alert=.fx.alert~new(.fx.Alert.Type~WARNING)   -- create a warning alert

         alert~setTitle("No Selection")
         alert~setHeaderText(.nil)
         alert~setContentText("Please select an item in the table.")
         alert~showAndWait
      end
      else  -- use BSF.CLS' .bsf.dialog utility class
      do
         -- .bsf.dialog is defined in BSF.CLS and uses swing
        .bsf.dialog~messageBox("Please select an item in the table.", "No Selection", "warning")
      end
  end
```

```
::method sItemAddAction
expose sItemTable sItemTypeAdd mainApp sItemNameAdd sItemDiscountAdd sItemAmountAdd ItemData
 siDir=.my.app~item.fxml
sItemTypeAdd=siDir~sItemTypeAdd~getText()
sItemNameAdd=siDir~sItemNameAdd~getText()
sItemDiscountAdd=siDir~sItemDiscountAdd~getText()
sItemAmountAdd=siDir~sItemAmountAdd~getText()
sItemTable=siDir~sItemTable

if sItemTypeAdd="" | sItemNameAdd="" | sItemDiscountAdd="" | sItemAmountAdd="" then
do
    say "please fill in all fields"
     lf="0a"x
      text="Error"lf-
         lf"All textfields are necessary!" lf -
      lf"Please fill the textfields"lf


      if .my.app~bDialogsAlertsAvailable then
      do
         alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

         alert~setTitle("RexxPOS")
         alert~setHeaderText(.nil)
         alert~setContentText(text)
         alert~showAndWait
      end
      else  -- use BSF.CLS' .bsf.dialog utility class
      do
         -- .bsf.dialog is defined in BSF.CLS and uses swing
        .bsf.dialog~messageBox(text, "RexxPOS", "information")
      end

end
else
do

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="INSERT INTO pos.item(ItemType, ItemName, ItemTax, ItemPrice) VALUES
('"sItemTypeAdd"','"sItemNameAdd"',"sItemDiscountAdd", "sItemAmountAdd")"




    kunde=statement~executeUpdate(sqlkoma)


--itemData~add(.Item4Sale~new(kunde~getString("ID"),kunde~getString("ItemType"),kunde~getString("Item
Name"),kunde~getString("ItemTax"),kunde~getString("ItemPrice")))

    --sItemTable~setItems(itemData)

    self~getItems()
end
::method sItemSubTypeText
expose primaryStage dialogStage rootLayout personData showItemsbtn sItemPageButton showInvoicesbtn
showDashboardbtn showKundenbtn sItemTable -
sItemTableColumn1 sItemTableColumn2 sItemTableColumn3 sItemTableColumn4 sItemTableColumn5
```

```
sItemTypeAdd sItemNameAdd -
sItemAmountAdd sItemDiscountAdd sItemAddAction sItemDeleteAction Item4Sale itemData invoiceData
kundenData -
sIdSearchText sItemTypeText sItemSubTypeText
use arg rootLayout, primaryStage

siDir=.my.app~item.fxml
itemData=.fx.FXCollections~observableArrayList
ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


sItemTable=siDir~sItemTable
sItemAddAction=siDir~sItemAddAction ~~setOnAction(rp)
sItemDeleteAction=siDir~sItemDeleteAction ~~setOnAction(rp)
sIdSearchText=siDir~sIdSearchText ~~setOnAction(rp)
sItemTypeText=siDir~sItemTypeText ~~setOnAction(rp)
sItemSubTypeText=siDir~sItemSubTypeText ~~setOnAction(rp)

datam=.fx.FXCollections~observableArrayList


itemsicinde=.array~new

searchID=sItemSubTypeText~getText()
 if searchID<>.nil then
  do

    /* MySQL Connection  */

    if (conn="CONN") then do
    conn=dbconstr()
    end

    if (conn~isClosed) then do
    conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="SELECT * FROM pos.item where DiscontDate is null and ItemName LIKE '%"searchID"%' WHERE
DiscontDate is null ;"




    kunde=statement~executeQuery(sqlkoma)



    k=1
    DO WHILE kunde~next









--datam~add(kunde~getString("ItemName"),kunde~getString("ItemName"),kunde~getString("ItemName"),kunde
~getString("ItemName"),kunde~getString("ItemName"))

itemData~add(.Item4Sale~new(kunde~getString("ID"),kunde~getString("ItemType"),kunde~getString("ItemNa
me"),kunde~getString("ItemTax"),kunde~getString("ItemPrice")))
    k=k+1
```

```
    END
    sItemTable~setItems(itemData)
    --siDir~sItemTable~setItems(datam)



end
if searchID=.nil then
do
    self~getItems()
end

::method sItemTypeText
expose primaryStage dialogStage rootLayout personData showItemsbtn sItemPageButton showInvoicesbtn
showDashboardbtn showKundenbtn sItemTable -
sItemTableColumn1 sItemTableColumn2 sItemTableColumn3 sItemTableColumn4 sItemTableColumn5
sItemTypeAdd sItemNameAdd -
sItemAmountAdd sItemDiscountAdd sItemAddAction sItemDeleteAction Item4Sale itemData invoiceData
kundenData -
sIdSearchText sItemTypeText sItemSubTypeText
use arg rootLayout, primaryStage

siDir=.my.app~item.fxml
itemData=.fx.FXCollections~observableArrayList
ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


sItemTable=siDir~sItemTable
sItemAddAction=siDir~sItemAddAction ~~setOnAction(rp)
sItemDeleteAction=siDir~sItemDeleteAction ~~setOnAction(rp)
sIdSearchText=siDir~sIdSearchText ~~setOnAction(rp)
sItemTypeText=siDir~sItemTypeText ~~setOnAction(rp)
sItemSubTypeText=siDir~sItemSubTypeText ~~setOnAction(rp)

datam=.fx.FXCollections~observableArrayList


itemsicinde=.array~new

searchID=sItemTypeText~getText()
 if searchID<>.nil then
  do

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="SELECT * FROM pos.item where DiscontDate is null and ItemType LIKE '%"searchID"%' ;"




    kunde=statement~executeQuery(sqlkoma)



    k=1
    DO WHILE kunde~next
```

```
--datam~add(kunde~getString("ItemName"),kunde~getString("ItemName"),kunde~getString("ItemName"),kunde
~getString("ItemName"),kunde~getString("ItemName"))

itemData~add(.Item4Sale~new(kunde~getString("ID"),kunde~getString("ItemType"),kunde~getString("ItemNa
me"),kunde~getString("ItemTax"),kunde~getString("ItemPrice")))
        k=k+1
    END
    sItemTable~setItems(itemData)
    --siDir~sItemTable~setItems(datam)



end
if searchID=.nil then
do
    self~getItems()
end

::method sIdSearchText
expose primaryStage dialogStage rootLayout personData showItemsbtn sItemPageButton showInvoicesbtn
showDashboardbtn showKundenbtn sItemTable -
sItemTableColumn1 sItemTableColumn2 sItemTableColumn3 sItemTableColumn4 sItemTableColumn5
sItemTypeAdd sItemNameAdd -
sItemAmountAdd sItemDiscountAdd sItemAddAction sItemDeleteAction Item4Sale itemData invoiceData
kundenData -
sIdSearchText sItemTypeText sItemSubTypeText
use arg rootLayout, primaryStage

siDir=.my.app~item.fxml
itemData=.fx.FXCollections~observableArrayList
ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


sItemTable=siDir~sItemTable
sItemAddAction=siDir~sItemAddAction ~~setOnAction(rp)
sItemDeleteAction=siDir~sItemDeleteAction ~~setOnAction(rp)
sIdSearchText=siDir~sIdSearchText ~~setOnAction(rp)
sItemTypeText=siDir~sItemTypeText ~~setOnAction(rp)
sItemSubTypeText=siDir~sItemSubTypeText ~~setOnAction(rp)

datam=.fx.FXCollections~observableArrayList


itemsicinde=.array~new

searchID=sIdSearchText~getText()
 if searchID<>.nil then
  do

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end
```

```
    statement = conn~createStatement
    sqlkoma ="SELECT * FROM pos.item where DiscontDate is null and ID LIKE '%"searchID"%' ;"




    kunde=statement~executeQuery(sqlkoma)



    k=1
    DO WHILE kunde~next









--datam~add(kunde~getString("ItemName"),kunde~getString("ItemName"),kunde~getString("ItemName"),kunde
~getString("ItemName"),kunde~getString("ItemName"))

itemData~add(.Item4Sale~new(kunde~getString("ID"),kunde~getString("ItemType"),kunde~getString("ItemNa
me"),kunde~getString("ItemTax"),kunde~getString("ItemPrice")))
        k=k+1
        END
        sItemTable~setItems(itemData)
        --siDir~sItemTable~setItems(datam)

        do u over itemData
        --say u~Item4SaleName
    end

end
if searchID=.nil then
do
    self~getItems()
end
::method getItems
 expose primaryStage dialogStage rootLayout personData showItemsbtn sItemPageButton showInvoicesbtn
showDashboardbtn showKundenbtn sItemTable -
sItemTableColumn1 sItemTableColumn2 sItemTableColumn3 sItemTableColumn4 sItemTableColumn5
sItemTypeAdd sItemNameAdd -
sItemAmountAdd sItemDiscountAdd sItemAddAction sItemDeleteAction Item4Sale itemData invoiceData
kundenData -
sIdSearchText sItemTypeText sItemSubTypeText
use arg rootLayout, primaryStage

/* MySQL Connection  */
if (conn="CONN") then do
    conn=dbconstr()
end

if (conn~isClosed) then do
    conn=dbconstr()
end


statement = conn~createStatement
sqlkoma ="SELECT * FROM pos.item where DiscontDate is null ;"


  siDir=.my.app~item.fxml

kunde=statement~executeQuery(sqlkoma)
```

```
itemData=.fx.FXCollections~observableArrayList
ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


sItemTable=siDir~sItemTable
sItemAddAction=siDir~sItemAddAction ~~setOnAction(rp)
sItemDeleteAction=siDir~sItemDeleteAction ~~setOnAction(rp)
sIdSearchText=siDir~sIdSearchText ~~setOnAction(rp)
sItemTypeText=siDir~sItemTypeText ~~setOnAction(rp)
sItemSubTypeText=siDir~sItemSubTypeText ~~setOnAction(rp)
rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")
sItemTable~setOnMouseClicked(rp)
datam=.fx.FXCollections~observableArrayList


itemsicinde=.array~new

k=1
DO WHILE kunde~next









--datam~add(kunde~getString("ItemName"),kunde~getString("ItemName"),kunde~getString("ItemName"),kunde
~getString("ItemName"),kunde~getString("ItemName"))

itemData~add(.Item4Sale~new(kunde~getString("ID"),kunde~getString("ItemType"),kunde~getString("ItemNa
me"),kunde~getString("ItemTax"),kunde~getString("ItemPrice")))
    k=k+1
END
sItemTable~setItems(itemData)
--siDir~sItemTable~setItems(datam)

::method sItemDeleteAction
expose primaryStage dialogStage rootLayout personData showItemsbtn sItemPageButton showInvoicesbtn
showDashboardbtn showKundenbtn sItemTable -
sItemTableColumn1 sItemTableColumn2 sItemTableColumn3 sItemTableColumn4 sItemTableColumn5
sItemTypeAdd sItemNameAdd -
sItemAmountAdd sItemDiscountAdd sItemAddAction sItemDeleteAction Item4Sale itemData invoiceData
kundenData
--use arg rootLayout, primaryStage
siDir=.my.app~item.fxml
sItemTable=siDir~sItemTable

selectedIndex=sItemTable~getSelectionModel~getSelectedIndex
if selectedIndex="-1"   then
do

say "please select an item"
 lf="0a"x
  text="Error"lf-
     lf"Nothing selected!" lf -
  lf"Please select the item to delete"lf


  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

     alert~setTitle("RexxPOS")
     alert~setHeaderText(.nil)
```

```
     alert~setContentText(text)
     alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end

end
else
do
delcust=BSFRexxProxy(sItemTable~getItems~get(selectedIndex))~Item4SaleID

/* MySQL Connection  */
if (conn="CONN") then do
    conn=dbconstr()
end

if (conn~isClosed) then do
    conn=dbconstr()
end


statement = conn~createStatement
sqlkoma ="UPDATE pos.item SET DiscontDate = CURRENT_TIMESTAMP WHERE ID ="delcust




kunde=statement~executeUpdate(sqlkoma)




self~getItems()
end
::method showInvoices
  expose primaryStage personData iInvoicetableView showItemsbtn sItemPageButton showInvoicesbtn
showDashboardbtn showKundenbtn -
     iCancelOrderButtonAction iPrintOrderButtonAction iRefreshButtonAction itemData invoiceData
kundenData iBillTextArea -
butnext butprev butdown ibisCheckBox

  -- Load the fxml file and create a new stage for the popup
  SettingsUrl=.bsf~new("java.net.URL", "file:menu2.fxml")
  page  = .fx.FXMLLoader~load(SettingsUrl)
  scene = .fx.Scene~new(page)

  -- AnchorPane page = (AnchorPane) loader.load();
  dialogStage = .bsf~new("javafx.stage.Stage")
  dialogStage~setTitle("Invoices")
  dialogStage~initModality(.fx.Modality~WINDOW_MODAL)
  --dialogStage~initModality(.fx.Modality~NONE)


  dialogStage~setScene(scene)
--primaryStage~setScene(scene)

  -- Set the persons into the controller
  controller = .InvoicesController~new
 -- controller~setPersonData(personData)
dialogStage~setMaximized(.true)

  dialogStage~show

self~getInvoices()
```

```
::method getInvoices
expose primaryStage personData iInvoicetableView showItemsbtn sItemPageButton showInvoicesbtn
showDashboardbtn showKundenbtn -
     iCancelOrderButtonAction iPrintOrderButtonAction iRefreshButtonAction itemData invoiceData
kundenData -
iInvoiceCountText iInvoiceTotal iBillTextArea ibisCheckBox butprev butnext butdown iInvoiceStart
iInvoiceEnd
--use arg invstart, invend
/* MySQL Connection  */
if (conn="CONN") then do
    conn=dbconstr()
end

if (conn~isClosed) then do
    conn=dbconstr()
end

sinDir=.my.app~menu2.fxml
today=.LocalDa~now()
--say "next " iInvoiceEnd~setValue(today~plusDays(1))

iInvoiceStart=sinDir~iInvoiceStart
iInvoiceEnd=sinDir~iInvoiceEnd

invstart=iInvoiceStart~getValue()
invend=iInvoiceEnd~getValue()

if invstart="" then invstart=today~plusDays(-30)
if invend="" then invend=today

statement = conn~createStatement
sqlkoma ="SELECT * FROM pos.invoice INNER JOIN pos.customer on invoice.CustomerID =customer.ID where
invoice.DiscontDate is null and  DATE (invoice.CreateTime) >= '"invstart~toString"' and DATE
(invoice.CreateTime) <= '"invend~toString"';"




kunde=statement~executeQuery(sqlkoma)


kunde=statement~executeQuery(sqlkoma)

ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


--iInvoicetableView=sinDir~iInvoicetableView




datam=.fx.FXCollections~observableArrayList


itemsicinde=.array~new

invoiceData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


iInvoicetableView=sinDir~iInvoicetableView

iInvoicetableView~setOnMouseClicked(rp)

--iInvoicetableView~getSelectionModel ~selectedItemProperty ~addListener(rp)
```

```
iInvoiceCountText=sinDir~iInvoiceCountText
iInvoiceTotal=sinDir~iInvoiceTotal
iBillTextArea=sinDir~iBillTextArea
iCancelOrderButtonAction=sinDir~iCancelOrderButtonAction ~~setOnAction(rp)
iPrintOrderButtonAction=sinDir~iPrintOrderButtonAction ~~setOnAction(rp)
butnext=sinDir~butnext ~~setOnAction(rp)
butprev=sinDir~butprev ~~setOnAction(rp)
butdown=sinDir~butdown ~~setOnAction(rp)
ibisCheckBox=sinDir~ibisCheckBox ~~setOnAction(rp)
iRefreshButtonAction=sinDir~iRefreshButtonAction ~~setOnAction(rp)

datam=.fx.FXCollections~observableArrayList


itemsicinde=.array~new
invtotal=0
k=1

DO WHILE kunde~next



--datam~add(kunde~getString("ItemName"),kunde~getString("ItemName"),kunde~getString("ItemName"),kunde
~getString("ItemName"),kunde~getString("ItemName"))

    --invoiceData~add(.anInvoice~new("2","3","a","21","21","23"))


    --lOrderDate = new SimpleDateFormat("yyyy/MM/dd").format(new Date(lOrderTimestamp.getTime()));

    --say .SimpleDateFormat~SimpleDateFormat("yyyy/MM/dd")~format(.LocalDa~now())
    --iInvoiceStart~setValue(.LocalDa~now())



invoiceData~add(.anInvoice~new(kunde~getString("ID"),kunde~getString("CustomerID"),"",left(kunde~getS
tring("CreateTime"),10),left(right(kunde~getString("CreateTime"),10),5),kunde~getString("OrderAmount"
)))


--invoiceData~add(.anInvoice~new(kunde~getString("ID"),kunde~getString("Customer_Name"),kunde~getStri
ng("OrderComments"),kunde~getString("OrderAmount"),kunde~getTimestamp("CreateTime")~getTime()))
    invtotal=invtotal+kunde~getInt("OrderAmount")
    k=k+1
END
--sItemTable~setItems(ItemsData)

iInvoiceCountText~setText(k-1)
iInvoiceTotal~setText(invtotal)


--sItemTable~setItems(ItemsData)
--iInvoicetableView~setItems(.my.app~mainApp~invoiceData)
--iInvoicetableView~setItems(invoiceData)
iInvoicetableView~setItems(invoiceData)




::method handlegetInvDetails
expose iInvoicetableView anInvoice iBillTextArea
--say cCustomerTable~getSelectionModel~getSelectedIndex
selectedIndex=iInvoicetableView~getSelectionModel~getSelectedIndex
if selectedIndex <> "-1"   then
do
delcust=BSFRexxProxy(iInvoicetableView~getItems~get(selectedIndex))~anInvoiceID
    lisp=.bsf4rexx~system.class ~getProperty('line.separator')
```

```
--say "getting invoice details with ID" delcust

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="SELECT customer.Customer_Name,customer.Customer_Address,customer.Postcode,
customer.Customer_No, invoice.CustomerID, invoice.OrderComments, invoice.OrderAmount,
invoice.CreateTime ,invoice.DiscontDate FROM pos.invoice  INNER JOIN pos.customer on
invoice.CustomerID =customer.ID WHERE invoice.ID ="delcust


    kunde=statement~executeQuery(sqlkoma)
    iBillTextArea~setText(lisp)
DO WHILE kunde~next

iCust=kunde~getString("customer.Customer_Name")
cusNo=kunde~getString("customer.Customer_No")
cusPLZ=kunde~getString("customer.Postcode")
cusAdr=kunde~getString("customer.Customer_Address")
iOrdA= kunde~getString("invoice.OrderAmount")
iCT= kunde~getString("invoice.CreateTime")
iC= kunde~getString("invoice.OrderComments")
cID=kunde~getString("invoice.CustomerID")

--say kunde~getString("invoice_details.ItemName")

    END

sqlkoma ="SELECT * FROM pos.invoice_details WHERE invoice_details.InvoiceID ="delcust

    kunde=statement~executeQuery(sqlkoma)
iOrdered= " Invoice Number: " delcust lisp "-"~COPIES(30) lisp  lisp  "Customer Name: " iCust lisp
"Customer Phone: " cusNo  lisp "Address: " cusAdr lisp "Postcode: " cusPLZ lisp "Customer ID:" cID
lisp lisp "Order Date:" iCT lisp "Order Comments: " iC lisp lisp   "-"~COPIES(23) lisp " Ordered
Items:" lisp "-"~COPIES(23)
iBillTextArea~appendText(iOrdered)
iOrderedd=" "
DO WHILE kunde~next

iOrderedd= kunde~getString("invoice_details.Quantity") "x "
kunde~getString("invoice_details.ItemName")
--iOrdered= iOrdered " Tax(%): " kunde~getString("invoice_details.ItemTax")
iOrderedd= iOrderedd " Price: " kunde~getString("invoice_details.ItemPrice") lisp
--iOrdered= iOrdered " Subtotal: "kunde~getString("invoice_details.TotalAmount")

--say kunde~getString("invoice_details.ItemName")

    END


iBillTextArea~setEditable(.false)
iBillTextArea~setWrapText(.true)

iBillTextArea~appendText(lisp iOrderedd lisp lisp "="~COPIES(20) lisp " Total: " iOrdA "euro (incl.
Tax)" lisp "="~COPIES(20))


end

::method iCancelOrderButtonAction
```

```
expose iInvoicetableView anInvoice
--say cCustomerTable~getSelectionModel~getSelectedIndex
selectedIndex=iInvoicetableView~getSelectionModel~getSelectedIndex

--say selectedIndex "selectedIndex"
--say delcust "delcust"


if selectedIndex="-1"   then
do

say "please select a customer in table"
 lf="0a"x
  text="Error"lf-
     lf"Nothing selected!" lf -
  lf"Please select the customer to delete"lf


  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

     alert~setTitle("RexxPOS")
     alert~setHeaderText(.nil)
     alert~setContentText(text)
     alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end

end
else
do
    say "deleting invoice with ID" delcust
    delcust=BSFRexxProxy(iInvoicetableView~getItems~get(selectedIndex))~anInvoiceID

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="UPDATE pos.invoice SET invoice.DiscontDate = CURRENT_TIMESTAMP WHERE invoice.ID
="delcust




    kunde=statement~executeUpdate(sqlkoma)






    self~getInvoices()
end


::method butnexthandle
```

```
expose iInvoiceEnd iInvoiceStart iInvoicetableView

sinDir=.my.app~menu2.fxml
iInvoiceStart=sinDir~iInvoiceStart
iInvoiceEnd=sinDir~iInvoiceEnd
--say iInvoiceStart~getValue()
today=iInvoiceStart~getValue()
--today=.LocalDa~now()
--say "next " iInvoiceEnd~setValue(today~plusDays(1))

--say "next " today~plusDays(1)~toString

iInvoiceStart~setValue(iInvoiceStart~getValue()~plusDays(1))
iInvoiceEnd~setValue(iInvoiceEnd~getValue()~plusDays(1))
self~getInvoices()

::method ibisCheckBoxhandle
--say "ibis"

::method butprevhandle
expose iInvoiceEnd iInvoiceStart iInvoicetableView

sinDir=.my.app~menu2.fxml
iInvoiceStart=sinDir~iInvoiceStart
iInvoiceEnd=sinDir~iInvoiceEnd
--say iInvoiceStart~getValue()
today=iInvoiceStart~getValue()
--today=.LocalDa~now()
--say "next " iInvoiceEnd~setValue(today~plusDays(1))

--say "next " today~plusDays(1)~toString

iInvoiceStart~setValue(iInvoiceStart~getValue()~plusDays(-1))
iInvoiceEnd~setValue(iInvoiceEnd~getValue()~plusDays(-1))
self~getInvoices()

::method butdownhandle
expose iInvoiceEnd iInvoiceStart iInvoicetableView invoiceData
say "exportin invoice data in file invoices.json"
filePath="invoices.json"

  arr=.array~new
  do p over invoiceData  -- iterate over ObservableList
     dir=.directory~new
     dir["Invoice_ID"] =p~anInvoiceID
     dir["Order_Comments"]  =p~anInvoiceOrderComments
     dir["Order_Amount"]    =p~anInvoiceOrderAmount
     dir["Order_Time"]=p~anInvoiceCreateTime
     dir["Order_Date"]      =p~anInvoiceDiscontDate
     dir["Customer_ID"]  =p~anInvoiceCustomerID

    arr~append(dir)
  end

  s=.stream~new(filepath)~~open("write replace")   -- open stream for replacement
  jsonStr=s~charout(.json~new~toJson(arr)) -- create and write JSON string to stream
  s~close                                  -- close stream
--  self~setPersonFilePath(filePath)     -- save this one as preference for next load


 lf="0a"x
  text="Invoice data has been successfully exported"lf-
     lf"to the file: invoices.json" lf -
  lf"in /bin directory"lf


  if .my.app~bDialogsAlertsAvailable then
  do
```

```
    alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

    alert~setTitle("RexxPOS")
    alert~setHeaderText(.nil)
    alert~setContentText(text)
    alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end



::method iPrintOrderButtonAction
expose iInvoicetableView anInvoice
--say cCustomerTable~getSelectionModel~getSelectedIndex
selectedIndex=iInvoicetableView~getSelectionModel~getSelectedIndex


if selectedIndex="-1"   then
do

say "please select an invoice in table"
 lf="0a"x
  text="Error"lf-
     lf"Nothing selected!" lf -
  lf"Please select the invoice to delete"lf


  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

     alert~setTitle("RexxPOS")
     alert~setHeaderText(.nil)
     alert~setContentText(text)
     alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end

end
else
do
    delcust=BSFRexxProxy(iInvoicetableView~getItems~get(selectedIndex))~anInvoiceID
    say "printing invoice with ID" delcust

end

::method iRefreshButtonAction
self~getInvoices()

::method showKunden
  expose primaryStage personData showItemsbtn sItemPageButton showInvoicesbtn showDashboardbtn
showKundenbtn cCustomerTable -
   cCustomerAdd cCustomerDelete cCustomerNewOrder itemData invoiceData kundenData -
cIdSearchText cMobileSearchText cNameSearchText dialogStage
 use arg primaryStage

  -- Load the fxml file and create a new stage for the popup
  SettingsUrl=.bsf~new("java.net.URL", "file:kunden.fxml")
  page  = .fx.FXMLLoader~load(SettingsUrl)
  scene = .fx.Scene~new(page)
```

```
  -- AnchorPane page = (AnchorPane) loader.load();
  dialogStage = .bsf~new("javafx.stage.Stage")
--dialogStage=primaryStage
dialogStage~setTitle("Customers")
  dialogStage~initModality(.fx.Modality~WINDOW_MODAL)
  --dialogStage~initModality(.fx.Modality~NONE)



  dialogStage~setScene(scene)
--primaryStage~setScene(scene)

  -- Set the persons into the controller
  controller = .KundenController~new
 -- controller~setPersonData(personData)
dialogStage~setMaximized(.true)

  dialogStage~show
--primaryStage~show
--say primaryStage
self~getCustomers()


::method handleEditCustomer
expose cCustomerTable
 skDir=.my.app~kunden.fxml
cCustomerTable=skDir~cCustomerTable

  selectedPerson = cCustomerTable~getSelectionModel~getSelectedItem
  if selectedPerson <> .nil then
  do
      person=BSFRexxProxy(selectedPerson) -- unbox Rexx object
      okClicked = .my.app~mainApp~showPersonEditDialog(person)
      if okClicked=.true then
          self~showPersonDetails(person)
  end
  else
  do
      if .my.app~bDialogsAlertsAvailable then
      do
         alert=.fx.alert~new(.fx.Alert.Type~WARNING)   -- create a warning alert

         alert~setTitle("No Selection")
         alert~setHeaderText(.nil)
         alert~setContentText("Please select a customer in the table.")
         alert~showAndWait
      end
      else  -- use BSF.CLS' .bsf.dialog utility class
      do
         -- .bsf.dialog is defined in BSF.CLS and uses swing
        .bsf.dialog~messageBox("Please select a customer in the table.", "No Selection", "warning")
      end
  end


::method cCustomerAdd
expose cCustomerNameAdd mainApp dialogStage cCustomerMobileAdd cCustomerAddressAdd
cCustomerPincodeAdd cCustomerTable aCustomer



 skDir=.my.app~kunden.fxml
cCustomerNameAdd=skDir~cCustomerNameAdd~getText()
cCustomerMobileAdd=skDir~cCustomerMobileAdd~getText()
cCustomerAddressAdd=skDir~cCustomerAddressAdd~getText()
cCustomerPincodeAdd=skDir~cCustomerPincodeAdd~getText()
--say adda cCustomerNameAdd
```

```
if cCustomerNameAdd="" | cCustomerMobileAdd="" | cCustomerAddressAdd="" | cCustomerPincodeAdd="" then
do
say "please fill in all fields"
 lf="0a"x
  text="Error"lf-
     lf"All textfields are necessary!" lf -
  lf"Please fill the textfields"lf


  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

     alert~setTitle("RexxPOS")
     alert~setHeaderText(.nil)
     alert~setContentText(text)
     alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end

end
else
do

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="INSERT INTO pos.customer(Customer_Name, Customer_No, Customer_Address, Postcode) VALUES
('"cCustomerNameAdd"',"cCustomerMobileAdd", '"cCustomerAddressAdd"', " cCustomerPincodeAdd ")"


      skDir=.my.app~kunden.fxml

    kunde=statement~executeUpdate(sqlkoma)

    -- INSERT INTO pos.customer(Customer_Name, Customer_No, Customer_Address, Postcode) VALUES
('"cCustomerNameAdd"',"cCustomerMobileAdd", '"cCustomerAddressAdd"', " cCustomerPincodeAdd ")

    self~getCustomers()
end

::method cCustomerDelete
expose cCustomerTable aCustomer
--say cCustomerTable~getSelectionModel~getSelectedIndex
selectedIndex=cCustomerTable~getSelectionModel~getSelectedIndex

if selectedIndex="-1"   then
do

say "please select a customer"
 lf="0a"x
  text="Error"lf-
     lf"Nothing selected!" lf -
  lf"Please select the customer to delete"lf
```

```
  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

     alert~setTitle("RexxPOS")
     alert~setHeaderText(.nil)
     alert~setContentText(text)
     alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end

end
else
do
    delcust=BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))~aCustomerID
    say "deleting customer with ID" delcust

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="UPDATE pos.customer SET Discont_Date = CURRENT_TIMESTAMP WHERE customer.ID ="delcust


      skDir=.my.app~kunden.fxml

    kunde=statement~executeUpdate(sqlkoma)




    self~getCustomers()
end
::method cMobileSearchText
expose primaryStage personData showItemsbtn sItemPageButton showInvoicesbtn showDashboardbtn
showKundenbtn cCustomerTable -
   cCustomerAdd cCustomerDelete cCustomerNewOrder itemData invoiceData kundenData -
cIdSearchText cMobileSearchText cNameSearchText

 skDir=.my.app~kunden.fxml



ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


cIdSearchText=skDir~cIdSearchText ~~setOnAction(rp)
cMobileSearchText=skDir~cMobileSearchText ~~setOnAction(rp)
cNameSearchText=skDir~cNameSearchText ~~setOnAction(rp)

cCustomerTable=skDir~cCustomerTable
searchID=cMobileSearchText~getText()
 if searchID<>.nil then
  do
```

```
    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="SELECT * FROM pos.customer where Discont_Date is null and Customer_No like
'%"searchID"%' ;"



    cCustomerAdd=skDir~cCustomerAdd ~~setOnAction(rp)
    cCustomerDelete=skDir~cCustomerDelete ~~setOnAction(rp)
    cCustomerNewOrder=skDir~cCustomerNewOrder ~~setOnAction(rp)
    kunde=statement~executeQuery(sqlkoma)
    --cCustomerTable~setColumnResizePolicy(personTable~CONSTRAINED_RESIZE_POLICY)
    datam=.fx.FXCollections~observableArrayList
    cCustomerTable~setOnMouseClicked(rp)

    itemsicinde=.array~new

    kundenData=.fx.FXCollections~observableArrayList
    k=1
    DO WHILE kunde~next


kundenData~add(.aCustomer~new(kunde~getString("ID"),kunde~getString("Customer_Name"),kunde~getString(
"Customer_Address"),kunde~getLong("Customer_No"),kunde~getInt("Postcode")))
        k=k+1
    END
    --sItemTable~setItems(ItemsData)

    --cCustomerTable~setItems(.my.app~mainApp~kundenData)
    cCustomerTable~setItems(kundenData)
end

if searchID=.nil then
do
    self~getCustomers()
end
::method cNameSearchText
expose primaryStage personData showItemsbtn sItemPageButton showInvoicesbtn showDashboardbtn
showKundenbtn cCustomerTable -
   cCustomerAdd cCustomerDelete cCustomerNewOrder itemData invoiceData kundenData -
cIdSearchText cMobileSearchText cNameSearchText

 skDir=.my.app~kunden.fxml



ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


cIdSearchText=skDir~cIdSearchText ~~setOnAction(rp)
cMobileSearchText=skDir~cMobileSearchText ~~setOnAction(rp)
cNameSearchText=skDir~cNameSearchText ~~setOnAction(rp)

cCustomerTable=skDir~cCustomerTable
searchID=cNameSearchText~getText()
 if searchID<>.nil then
  do
    /* MySQL Connection  */
```

```
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="SELECT * FROM pos.customer where Discont_Date is null and (Customer_Name LIKE
'%"searchID"%' or Customer_Address like '%"searchID"%') ;"



    cCustomerAdd=skDir~cCustomerAdd ~~setOnAction(rp)
    cCustomerDelete=skDir~cCustomerDelete ~~setOnAction(rp)
    cCustomerNewOrder=skDir~cCustomerNewOrder ~~setOnAction(rp)
    kunde=statement~executeQuery(sqlkoma)
    --cCustomerTable~setColumnResizePolicy(personTable~CONSTRAINED_RESIZE_POLICY)
    datam=.fx.FXCollections~observableArrayList
    cCustomerTable~setOnMouseClicked(rp)

    itemsicinde=.array~new

    kundenData=.fx.FXCollections~observableArrayList
    k=1
    DO WHILE kunde~next


kundenData~add(.aCustomer~new(kunde~getString("ID"),kunde~getString("Customer_Name"),kunde~getString(
"Customer_Address"),kunde~getLong("Customer_No"),kunde~getInt("Postcode")))
        k=k+1
    END
    --sItemTable~setItems(ItemsData)

    --cCustomerTable~setItems(.my.app~mainApp~kundenData)
    cCustomerTable~setItems(kundenData)
end

if searchID=.nil then
do
    self~getCustomers()
end

::method cIdSearchText
expose primaryStage personData showItemsbtn sItemPageButton showInvoicesbtn showDashboardbtn
showKundenbtn cCustomerTable -
   cCustomerAdd cCustomerDelete cCustomerNewOrder itemData invoiceData kundenData -
cIdSearchText cMobileSearchText cNameSearchText

 skDir=.my.app~kunden.fxml



ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


cIdSearchText=skDir~cIdSearchText ~~setOnAction(rp)
cMobileSearchText=skDir~cMobileSearchText ~~setOnAction(rp)
cNameSearchText=skDir~cNameSearchText ~~setOnAction(rp)

cCustomerTable=skDir~cCustomerTable
searchID=cIdSearchText~getText()
 if searchID<>.nil then
  do
    /* MySQL Connection  */
```

```
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="SELECT * FROM pos.customer where Discont_Date is null and ID like '%"searchID"%' ;"



    cCustomerAdd=skDir~cCustomerAdd ~~setOnAction(rp)
    cCustomerDelete=skDir~cCustomerDelete ~~setOnAction(rp)
    cCustomerNewOrder=skDir~cCustomerNewOrder ~~setOnAction(rp)
    kunde=statement~executeQuery(sqlkoma)
    --cCustomerTable~setColumnResizePolicy(personTable~CONSTRAINED_RESIZE_POLICY)
    datam=.fx.FXCollections~observableArrayList
    cCustomerTable~setOnMouseClicked(rp)

    itemsicinde=.array~new

    kundenData=.fx.FXCollections~observableArrayList
    k=1
    DO WHILE kunde~next

kundenData~add(.aCustomer~new(kunde~getString("ID"),kunde~getString("Customer_Name"),kunde~getString(
"Customer_Address"),kunde~getLong("Customer_No"),kunde~getInt("Postcode")))
        k=k+1
    END
    --sItemTable~setItems(ItemsData)

    --cCustomerTable~setItems(.my.app~mainApp~kundenData)
    cCustomerTable~setItems(kundenData)
end

if searchID=.nil then
do
    self~getCustomers()
end

::method getCustomers
 expose primaryStage personData showItemsbtn sItemPageButton showInvoicesbtn showDashboardbtn
showKundenbtn cCustomerTable -
   cCustomerAdd cCustomerDelete cCustomerNewOrder itemData invoiceData kundenData -
cIdSearchText cMobileSearchText cNameSearchText
 use arg primaryStage


/* MySQL Connection  */
if (conn="CONN") then do
    conn=dbconstr()
end

if (conn~isClosed) then do
    conn=dbconstr()
end


statement = conn~createStatement
sqlkoma ="SELECT * FROM pos.customer where Discont_Date is null ;"


  skDir=.my.app~kunden.fxml

kunde=statement~executeQuery(sqlkoma)
```

```
ItemsData=.fx.FXCollections~observableArrayList
rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")


cIdSearchText=skDir~cIdSearchText ~~setOnAction(rp)
cMobileSearchText=skDir~cMobileSearchText ~~setOnAction(rp)
cNameSearchText=skDir~cNameSearchText ~~setOnAction(rp)

cCustomerTable=skDir~cCustomerTable
cCustomerAdd=skDir~cCustomerAdd ~~setOnAction(rp)
cCustomerDelete=skDir~cCustomerDelete ~~setOnAction(rp)
cCustomerNewOrder=skDir~cCustomerNewOrder ~~setOnAction(rp)

--cCustomerTable~setColumnResizePolicy(personTable~CONSTRAINED_RESIZE_POLICY)
datam=.fx.FXCollections~observableArrayList

cCustomerTable~setOnMouseClicked(rp)

itemsicinde=.array~new

kundenData=.fx.FXCollections~observableArrayList
k=1
DO WHILE kunde~next

kundenData~add(.aCustomer~new(kunde~getString("ID"),kunde~getString("Customer_Name"),kunde~getString(
"Customer_Address"),kunde~getLong("Customer_No"),kunde~getInt("Postcode")))
k=k+1
END
--sItemTable~setItems(ItemsData)

--cCustomerTable~setItems(.my.app~mainApp~kundenData)
cCustomerTable~setItems(kundenData)


::method cCustomerNewOrder
expose cCustomerTable aCustomer dialogStage primaryStage
--say newor
--say cCustomerTable~getSelectionModel~getSelectedIndex
selectedIndex=cCustomerTable~getSelectionModel~getSelectedIndex
--say cCustomerTable~getItems~bsf.selectedPerson("remove", "int", selectedIndex)
--say cCustomerTable~getItems~get(selectedIndex)
--say BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))
custn=BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))~aCustomerID
--say "new order from customer" custn
-- say BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))~aCustomerID
--say BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))~aCustomerIDproperty
--say BSFRexxProxy(selectedPerson)
-- cCustomerTable~getItems()
--say aCustomer(selectedIndex)
--self~showDashboard(custn)

skDir=.my.app~menu1.fxml
dCustomerIdText=skDir~dCustomerIdText
dCustomerNameText=skDir~dCustomerNameText
dCustomerAddressText=skDir~dCustomerAddressText
dCustomerNumberText=skDir~dCustomerNumberText

dCustomerIdText~setText(custn)
dCustomerNameText~setText(BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))~aCustomerName)
dCustomerAddressText~setText(BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))~aCustomerAddres
s )
dCustomerNumberText~setText(BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))~aCustomerPostcod
e BSFRexxProxy(cCustomerTable~getItems~get(selectedIndex))~aCustomerNo)
dialogStage~hide

::method showDashboard
  expose primaryStage personData cartData showItemsbtn sItemPageButton showInvoicesbtn
```

```
showDashboardbtn -
  showKundenbtn dCustomerIdText custn dialogStage
use arg custn
--say custn
--say "bye"
--dialogStage~hide

  -- Load the fxml file and create a new stage for the popup
  SettingsUrl=.bsf~new("java.net.URL", "file:menu1.fxml")
  page  = .fx.FXMLLoader~load(SettingsUrl)
  scene = .fx.Scene~new(page)

  -- AnchorPane page = (AnchorPane) loader.load();
  dialogStage = .bsf~new("javafx.stage.Stage")
  dialogStage~setTitle("Dashboard")
  dialogStage~initModality(.fx.Modality~WINDOW_MODAL)
  --dialogStage~initModality(.fx.Modality~NONE)


  dialogStage~setScene(scene)
--primaryStage~setScene(scene)

  -- Set the persons into the controller
  controller = .DashboardController~new
 -- controller~setPersonData(personData)
  dialogStage~show
dialogStage~setMaximized(.true)



::method showPersonDetails
  expose firstNameLabel lastNameLabel streetLabel postalCodeLabel cityLabel birthdayLabel
  use arg p
/*
  if p=.nil then  -- reset details

do
     firstNameLabel  ~setText("")
     lastNameLabel   ~setText("")
     streetLabel     ~setText("")
     postalCodeLabel ~setText("")
     cityLabel       ~setText("")
     birthdayLabel   ~setText("")
  end
  else   -- set Person's details
  do
     firstNameLabel  ~setText(p~firstName)
     lastNameLabel   ~setText(p~lastName)
     streetLabel     ~setText(p~street)
     postalCodeLabel ~setText(p~postalCode)
     cityLabel       ~setText(p~city)
     birthdayLabel   ~setText(p~birthday)
  end
*/

::method handleDeletePerson   /* invoked when the user clicks the delete button   */
  expose iInvoicetableView

  selectedIndex=iInvoicetableView~getSelectionModel~getSelectedIndex

  if selectedIndex >= 0 then
  do
     -- IMPORTANT: we must use bsf.invokeStrict() to become able to give explicitly the argument's
type as otherwise
     --            it may be possible that the wrong "remove(Object)" method is used instead of
"remove(int)" (both
     --            methods behave differntly)
     iInvoicetableView~getItems~bsf.invokeStrict("remove", "int", selectedIndex) -- we want the
```

```
remove method with the primitive int argument!
  end
  else
  do
      if .my.app~bDialogsAlertsAvailable then
      do
         alert=.fx.alert~new(.fx.Alert.Type~WARNING)   -- create a warning alert

         alert~setTitle("No Selection")
         alert~setHeaderText(.nil)
         alert~setContentText("Please select a person in the table.")
         alert~showAndWait
      end
      else  -- use BSF.CLS' .bsf.dialog utility class
      do
         -- .bsf.dialog is defined in BSF.CLS and uses swing
        .bsf.dialog~messageBox("Please select a person in the table.", "No Selection", "warning")
      end
  end


/* for tutorial part 3: fill-in the Person's details */
::method handleNewPerson
  tempPerson=.Person~new

  okClicked = .my.app~mainApp~showPersonEditDialog(tempPerson)
  if okClicked then
     .my.app~mainApp~personData~add(tempPerson)


::method handleEditPerson
  expose cCustomerTable

  selectedPerson = cCustomerTable~getSelectionModel~getSelectedItem
  if selectedPerson <> .nil then
  do
      person=BSFRexxProxy(selectedPerson) -- unbox Rexx object
      okClicked = .my.app~mainApp~showPersonEditDialog(person)
      if okClicked=.true then
          self~showPersonDetails(person)
  end
  else
  do
      if .my.app~bDialogsAlertsAvailable then
      do
         alert=.fx.alert~new(.fx.Alert.Type~WARNING)   -- create a warning alert

         alert~setTitle("No Selection")
         alert~setHeaderText(.nil)
         alert~setContentText("Please select a person in the table.")
         alert~showAndWait
      end
      else  -- use BSF.CLS' .bsf.dialog utility class
      do
         -- .bsf.dialog is defined in BSF.CLS and uses swing
        .bsf.dialog~messageBox("Please select a person in the table.", "No Selection", "warning")
      end
  end


   -- JavaFX event handlers implemented in Rexx
::method changed unguarded    /* implements the interface "javafx.beans.value.ChangeListener" */
  use arg observable, oldValue, newValue
  self~showPersonDetails(newValue)  -- fill in the labels to show currently selected Person's details



::method handle   /* implements the interface "javafx.event.EventHandler"   */
```

```
  expose btnNew dCustomerPageDashboard btnDelete iInvoicetableView showItemsbtn sItemPageButton
showInvoicesbtn showKundenbtn showDashboardbtn dFoodMenuButton1 dFoodMenuButton2 dFoodMenuButton3
dFoodMenuButton4 dFoodMenuButton5 dFoodMenuButton6 -
 dFoodMenuButton7 dFoodMenuButton8 dFoodMenuButton9  dFoodMenuButton10 dFoodMenuButton11
dFoodMenuButton12 -
 dFoodMenuButton13 dFoodMenuButton14 dFoodMenuButton15  dFoodMenuButton16 dFoodMenuButton17
dFoodMenuButton18 -
 dFoodMenuButton19 dFoodMenuButton20 dPizzaItemButton1 dPizzaItemButton2 dPizzaItemButton3
dPizzaItemButton4 -
dPizzaItemButton5 dPizzaItemButton6 dPizzaItemButton7 dPizzaItemButton8 dPizzaItemButton9
dPizzaItemButton10 -
dPizzaItemButton11 dPizzaItemButton12 dPizzaItemButton13 dPizzaItemButton14 dPizzaItemButton15
dPizzaItemButton16 -
dPizzaItemButton17 dPizzaItemButton18 dPizzaItemButton19 dPizzaItemButton20 sItemAddAction
sItemDeleteAction -
cCustomerAdd cCustomerDelete cCustomerNewOrder iCancelOrderButtonAction iPrintOrderButtonAction
iRefreshButtonAction -
btnPrintSettings iSaveUserSettings cIdSearchText cMobileSearchText cNameSearchText -
sIdSearchText sItemTypeText sItemSubTypeText okle delle cCustomerTable sItemTable ibisCheckBox
butprev butnext butdown
  use arg event

  tgtObjectName=event~getTarget~objectname

 select
     when tgtObjectName=btnNew~objectName    then self~handleNewPerson
     when tgtObjectName=dCustomerPageDashboard~objectName   then self~handleEditPerson
     when tgtObjectName=btnDelete~objectName then self~handleDeletePerson
 when tgtObjectName=showItemsbtn~objectName then self~showItems
when tgtObjectName=showInvoicesbtn~objectName then self~showInvoices
 when tgtObjectName=sItemPageButton~objectName then self~showItems
when tgtObjectName=showKundenbtn~objectName then self~showKunden
when tgtObjectName=okle~objectName then self~okle
when tgtObjectName=delle~objectName then self~delle

when tgtObjectName=showDashboardbtn~objectName then self~showDashboard
when tgtObjectName=dFoodMenuButton1~objectName then self~dFoodMenuButton1
when tgtObjectName=dFoodMenuButton2~objectName then self~dFoodMenuButton2
when tgtObjectName=dFoodMenuButton3~objectName then self~dFoodMenuButton3
when tgtObjectName=dFoodMenuButton4~objectName then self~dFoodMenuButton4
when tgtObjectName=dFoodMenuButton5~objectName then self~dFoodMenuButton5
when tgtObjectName=dFoodMenuButton6~objectName then self~dFoodMenuButton6
when tgtObjectName=dFoodMenuButton7~objectName then self~dFoodMenuButton7
when tgtObjectName=dFoodMenuButton8~objectName then self~dFoodMenuButton8
when tgtObjectName=dFoodMenuButton9~objectName then self~dFoodMenuButton9
when tgtObjectName=dFoodMenuButton10~objectName then self~dFoodMenuButton10
when tgtObjectName=dFoodMenuButton11~objectName then self~dFoodMenuButton11
when tgtObjectName=dFoodMenuButton12~objectName then self~dFoodMenuButton12
when tgtObjectName=dFoodMenuButton13~objectName then self~dFoodMenuButton13
when tgtObjectName=dFoodMenuButton14~objectName then self~dFoodMenuButton14
when tgtObjectName=dFoodMenuButton15~objectName then self~dFoodMenuButton15
when tgtObjectName=dFoodMenuButton16~objectName then self~dFoodMenuButton16
when tgtObjectName=dFoodMenuButton17~objectName then self~dFoodMenuButton17
when tgtObjectName=dFoodMenuButton18~objectName then self~dFoodMenuButton18
when tgtObjectName=dFoodMenuButton19~objectName then self~dFoodMenuButton19
when tgtObjectName=dFoodMenuButton20~objectName then self~dFoodMenuButton20
when tgtObjectName=dPizzaItemButton1~objectName then self~dPizzaItemButton1
when tgtObjectName=dPizzaItemButton2~objectName then self~dPizzaItemButton2
when tgtObjectName=dPizzaItemButton3~objectName then self~dPizzaItemButton3
when tgtObjectName=dPizzaItemButton4~objectName then self~dPizzaItemButton4
when tgtObjectName=dPizzaItemButton5~objectName then self~dPizzaItemButton5
when tgtObjectName=dPizzaItemButton6~objectName then self~dPizzaItemButton6
when tgtObjectName=dPizzaItemButton7~objectName then self~dPizzaItemButton7
when tgtObjectName=dPizzaItemButton8~objectName then self~dPizzaItemButton8
when tgtObjectName=dPizzaItemButton9~objectName then self~dPizzaItemButton9
when tgtObjectName=dPizzaItemButton10~objectName then self~dPizzaItemButton10
when tgtObjectName=dPizzaItemButton11~objectName then self~dPizzaItemButton11
when tgtObjectName=dPizzaItemButton12~objectName then self~dPizzaItemButton12
```

```
when tgtObjectName=dPizzaItemButton13~objectName then self~dPizzaItemButton13
when tgtObjectName=dPizzaItemButton14~objectName then self~dPizzaItemButton14
when tgtObjectName=dPizzaItemButton15~objectName then self~dPizzaItemButton15
when tgtObjectName=dPizzaItemButton16~objectName then self~dPizzaItemButton16
when tgtObjectName=dPizzaItemButton17~objectName then self~dPizzaItemButton17
when tgtObjectName=dPizzaItemButton18~objectName then self~dPizzaItemButton18
when tgtObjectName=dPizzaItemButton19~objectName then self~dPizzaItemButton19
when tgtObjectName=dPizzaItemButton20~objectName then self~dPizzaItemButton20

when tgtObjectName=sItemAddAction~objectName then self~sItemAddAction
when tgtObjectName=sItemDeleteAction~objectName then self~sItemDeleteAction

when tgtObjectName=iRefreshButtonAction~objectName then self~iRefreshButtonAction
when tgtObjectName=iPrintOrderButtonAction~objectName then self~iPrintOrderButtonAction
when tgtObjectName=iCancelOrderButtonAction~objectName then self~iCancelOrderButtonAction
when tgtObjectName=cCustomerNewOrder~objectName then self~cCustomerNewOrder
when tgtObjectName=cCustomerDelete~objectName then self~cCustomerDelete
when tgtObjectName=cCustomerAdd~objectName then self~cCustomerAdd
when tgtObjectName=btnPrintSettings~objectName then self~btnPrintSettings
when tgtObjectName=iSaveUserSettings~objectName then self~iSaveUserSettings

when tgtObjectName=cIdSearchText~objectName then self~cIdSearchText
when tgtObjectName=cMobileSearchText~objectName then self~cMobileSearchText
when tgtObjectName=cNameSearchText~objectName then self~cNameSearchText

when tgtObjectName=sIdSearchText~objectName then self~sIdSearchText
when tgtObjectName=sItemTypeText~objectName then self~sItemTypeText
when tgtObjectName=sItemSubTypeText~objectName then self~sItemSubTypeText

when tgtObjectName=ibisCheckBox~objectName then self~ibisCheckBoxhandle
when tgtObjectName=butprev~objectName then self~butprevhandle
when tgtObjectName=butnext~objectName then self~butnexthandle

when tgtObjectName=butdown~objectName then self~butdownhandle

    otherwise
     do
        if event~getSource~objectName=cCustomerTable~objectName then -- TableView the source of the
event?
        do
            if event~getClickCount>1 then    -- if a double-click, then go into edit record mode
               self~handleEditCustomer
        end
if event~getSource~objectName=sItemTable~objectName then -- TableView the source of the event?
        do
            if event~getClickCount>1 then    -- if a double-click, then go into edit record mode
               self~handleEditItem
        end
if event~getSource~objectName=iInvoicetableView~objectName then -- TableView the source of the event?
        do
            if event~getClickCount>1 then    -- if a double-click, then go into edit record mode
              do
            --  self~handleEditInvoice
say
end
else
do
self~handlegetInvDetails
end
--end
        end

        else   -- a truly unknown/unexpected event!
        do
--say  pp(event~toString)
      --     .error~say(self"::handle, UNKNOWN event:" pp(event~toString))
        end
     end
```

```
  end



/* ============================================================================== */
/* implements "R javafx.util.Callback<P,R>(P o) for PropertyValueFactory */

/* This class allows instances that remember the message to be sent to person instances to
   return the property of the attribute that should be shown in the table cell.
*/
::class PropertyValueFactory
::method init
  expose  propName   --handler -- name of property getter method
  use strict arg propName -- , handler

::method call
  expose propName   -- handler
  use arg o          -- an observable value for the ooRexx person object boxed in a Java RexxProxy
object
  return BsfRexxProxy(o~getValue)~send(propName)


/* ============================================================================== */
/* for tutorial part 3: define the PersonEditDialogController */
/*
   This Rexx class controls the interaction with the fxml form named "PersonEditDialog.fxml".
*/
::class PersonEditDialogController
::attribute okClicked
::attribute dialogStage
::attribute person         -- the Person object to edit

::method init
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         btnOK btnCancel -
         okClicked

  okClicked=.false

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  pedDir=.my.app~PersonEditDialog.fxml
  firstNameField =pedDir~firstNameField
  lastNameField  =pedDir~lastNameField
  streetField    =pedDir~streetField
  postalCodeField=pedDir~postalCodeField
  cityField      =pedDir~cityField
  birthdayField  =pedDir~birthdayField

  btnOK          =pedDir~btnOK
  btnCancel      =pedDir~btnCancel

   -- add self as the EventHandler (method "handle" implemented)
  rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")
   -- add us as the event handler to the buttons, the event handler (method "handle") will be able to
get at the pressed button
  btnOK    ~setOnAction(rp)
  btnCancel~setOnAction(rp)


::method handle      -- method defined "javafx.event.EventHandler"
  expose btnOK btnCancel
  use arg event

  tgtObjectName=event~getTarget~objectname
  select
     when tgtObjectName=btnOK~objectName     then self~handleOK         -- say "BUTTON OK    was the
target!"
     when tgtObjectName=btnCancel~objectName then self~handleCancel     -- say "BUTTON CANCEL was the
```

```
target!"
     otherwise .error~say("UNKNOWN target:" pp(event~getTarget) pp(event~getTarget~tostring))
  end


::method setPerson
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         person
  use arg person  -- get person (assigning it to the attribute)

  firstNameField ~setText(person~aCustomerID)
  lastNameField  ~setText(person~aCustomerName)
  streetField    ~setText(person~aCustomerNo)
  postalCodeField~setText(person~aCustomerPostcode)
  --cityField      ~setText(person~city)
  birthdayField  ~setText(person~aCustomerAddress)
  --birthdayField  ~setPromptText("yyyy-mm-dd")

::method setItem
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         person
  use arg person  -- get person (assigning it to the attribute)

  firstNameField ~setText(person~aCustomerID)
  lastNameField  ~setText(person~aCustomerName)
  streetField    ~setText(person~aCustomerNo)
  postalCodeField~setText(person~aCustomerPostcode)
  --cityField      ~setText(person~city)
  birthdayField  ~setText(person~aCustomerAddress)
  --birthdayField  ~setPromptText("yyyy-mm-dd")


::method handleCancel
  expose dialogStage

  dialogStage~hide


::method handleOK
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         person okClicked dialogStage

  if (self~isInputValid) then
  do
      person~aCustomerID  = firstNameField ~getText
      person~aCustomerName   = lastNameField  ~getText
      person~aCustomerNo    = streetField    ~getText
      person~aCustomerPostcode = postalCodeField~getText
  --    person~city       = cityField      ~getText
      person~aCustomerAddress   = birthdayField  ~getText



if person~aCustomerID="" | person~aCustomerName="" | person~aCustomerAddress="" |
person~aCustomerPostcode="" | person~aCustomerNo="" then
do
say "please fill in all fields"
 lf="0a"x
  text="Error"lf-
     lf"All textfields are necessary!" lf -
  lf"Please fill the textfields"lf


  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

     alert~setTitle("RexxPOS")
```

```
    alert~setHeaderText(.nil)
    alert~setContentText(text)
    alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
    -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end

end
else
do

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="UPDATE pos.customer SET Customer_Name='"person~aCustomerName"',
Customer_No='"person~aCustomerAddress"', Customer_Address='"person~aCustomerNo"',
Postcode='"person~aCustomerPostcode"' WHERE ID='"person~aCustomerID"'"


      skDir=.my.app~kunden.fxml

    kunde=statement~executeUpdate(sqlkoma)

    -- INSERT INTO pos.customer(Customer_Name, Customer_No, Customer_Address, Postcode) VALUES
('"cCustomerNameAdd"',"cCustomerMobileAdd", '"cCustomerAddressAdd"', " cCustomerPincodeAdd ")

--self~getCustomers()
end

      okClicked = .true
      dialogStage~hide
  end

::method isInputValid
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         dialogStage


  errorMessage = ""
  LF = "0a"x     -- line feed character

  if (firstNameField~getText = .nil | firstNameField~getText~length = 0) then
     errorMessage = LF"No valid first name"

  if (lastNameField~getText = .nil | lastNameField~getText~length = 0) then
     errorMessage ||= LF"No valid last name"

  if (streetField~getText = .nil | streetField~getText~length = 0) then
     errorMessage ||= LF"No valid street"

  if (postalCodeField~getText = .nil | postalCodeField~getText~length = 0) then
     errorMessage ||= LF"No valid postal code"
  else
     if \ DataType(postalCodeField~getText, "Whole") then
         errorMessage ||= LF"No valid postal code (must be an integer)"
```

```
  if (birthdayField~getText = .nil | birthdayField~getText~length = 0) then
     errorMessage ||= LF"No valid birthday"


  if (errorMessage~length = 0) then
     return .true

  -- .bsf.dialog is defined in BSF.CLS and uses swing
  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~ERROR)  -- create a warning alert
     alert~setTitle("Invalid Fields")
     alert~setHeaderText("Please correct the listed errors!")
     alert~setContentText("Invalid values:" LF errorMessage)
     alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox("Please correct the listed invalid fields:" LF errorMessage, "Invalid
Fields", "error")
  end

   return .false

::class ItemEditDialogController
::attribute okClicked
::attribute dialogStage
::attribute person         -- the Person object to edit

::method init
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         btnOK btnCancel noprice plztax adrname namecat okClicked

  okClicked=.false

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
  pedDir=.my.app~PersonEditDialog.fxml
  firstNameField =pedDir~firstNameField
  lastNameField  =pedDir~lastNameField
  streetField    =pedDir~streetField
  postalCodeField=pedDir~postalCodeField
  cityField      =pedDir~cityField
  birthdayField  =pedDir~birthdayField

  btnOK          =pedDir~btnOK
  btnCancel      =pedDir~btnCancel

namecat          =pedDir~namecat
adrname          =pedDir~adrname
plztax          =pedDir~plztax
noprice=pedDir~noprice

namecat~setText("Category")
adrname~setText("Item Name")
plztax~setText("Tax")
noprice~setText("Price")


   -- add self as the EventHandler (method "handle" implemented)
  rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")
   -- add us as the event handler to the buttons, the event handler (method "handle") will be able to
get at the pressed button
  btnOK    ~setOnAction(rp)
  btnCancel~setOnAction(rp)


::method handle      -- method defined "javafx.event.EventHandler"
```

```
  expose btnOK btnCancel
  use arg event

  tgtObjectName=event~getTarget~objectname
  select
     when tgtObjectName=btnOK~objectName     then self~handleOK         -- say "BUTTON OK    was the
target!"
     when tgtObjectName=btnCancel~objectName then self~handleCancel     -- say "BUTTON CANCEL was the
target!"
     otherwise .error~say("UNKNOWN target:" pp(event~getTarget) pp(event~getTarget~tostring))
  end


::method setPerson
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         person
  use arg person  -- get person (assigning it to the attribute)

  firstNameField ~setText(person~Item4SaleID)
  lastNameField  ~setText(person~Item4SaleKat)
  streetField    ~setText(person~Item4SaleName)
  postalCodeField~setText(person~Item4SaleTax)
  --cityField      ~setText(person~city)
  birthdayField  ~setText(person~Item4SalePrice)
  --birthdayField  ~setPromptText("yyyy-mm-dd")

::method setItem
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         person
  use arg person  -- get person (assigning it to the attribute)

firstNameField ~setText(person~Item4SaleID)
  lastNameField  ~setText(person~Item4SaleKat)
  streetField    ~setText(person~Item4SaleName)
  postalCodeField~setText(person~Item4SaleTax)
  --cityField      ~setText(person~city)
  birthdayField  ~setText(person~Item4SalePrice)
  --birthdayField  ~setPromptText("yyyy-mm-dd")


::method handleCancel
  expose dialogStage

  dialogStage~hide


::method handleOK
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         person okClicked dialogStage

  if (self~isInputValid) then
  do
      person~Item4SaleID  = firstNameField ~getText
      person~Item4SaleKat   = lastNameField  ~getText
      person~Item4SaleName    = streetField    ~getText
      person~Item4SaleTax = postalCodeField~getText
  --    person~city       = cityField      ~getText
      person~Item4SalePrice   = birthdayField  ~getText



if person~Item4SaleID="" | person~Item4SaleKat="" | person~Item4SaleName="" | person~Item4SaleTax=""
| person~Item4SalePrice="" then
do
say "please fill in all fields"
 lf="0a"x
  text="Error"lf-
     lf"All textfields are necessary!" lf -
```

```
  lf"Please fill the textfields"lf


  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

     alert~setTitle("RexxPOS")
     alert~setHeaderText(.nil)
     alert~setContentText(text)
     alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end

end
else
do

    /* MySQL Connection  */
    if (conn="CONN") then do
        conn=dbconstr()
    end

    if (conn~isClosed) then do
        conn=dbconstr()
    end


    statement = conn~createStatement
    sqlkoma ="UPDATE pos.item SET ItemName='"person~Item4SaleName"',
ItemType='"person~Item4SaleKat"', ItemPrice='"person~Item4SalePrice"',
ItemTax='"person~Item4SaleTax"' WHERE ID='"person~Item4SaleID"'"



      skDir=.my.app~kunden.fxml

    kunde=statement~executeUpdate(sqlkoma)


end

      okClicked = .true
      dialogStage~hide
  end

::method isInputValid
  expose firstNameField lastNameField streetField postalCodeField cityField birthdayField -
         dialogStage


  errorMessage = ""
  LF = "0a"x     -- line feed character

  if (firstNameField~getText = .nil | firstNameField~getText~length = 0) then
     errorMessage = LF"No valid first name"

  if (lastNameField~getText = .nil | lastNameField~getText~length = 0) then
     errorMessage ||= LF"No valid last name"

  if (streetField~getText = .nil | streetField~getText~length = 0) then
     errorMessage ||= LF"No valid street"

  if (postalCodeField~getText = .nil | postalCodeField~getText~length = 0) then
```

```
    errorMessage ||= LF"No valid postal code"
  else
    if \ DataType(postalCodeField~getText, "Whole") then
        errorMessage ||= LF"No valid postal code (must be an integer)"


  if (birthdayField~getText = .nil | birthdayField~getText~length = 0) then
    errorMessage ||= LF"No valid birthday"


  if (errorMessage~length = 0) then
    return .true

  -- .bsf.dialog is defined in BSF.CLS and uses swing
  if .my.app~bDialogsAlertsAvailable then
  do
    alert=.fx.alert~new(.fx.Alert.Type~ERROR)  -- create a warning alert
    alert~setTitle("Invalid Fields")
    alert~setHeaderText("Please correct the listed errors!")
    alert~setContentText("Invalid values:" LF errorMessage)
    alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
    -- .bsf.dialog is defined in BSF.CLS and uses swing
   .bsf.dialog~messageBox("Please correct the listed invalid fields:" LF errorMessage, "Invalid
Fields", "error")
  end

   return .false




/* ============================================================================== */
/* for tutorial part 2: define The Model Class */

/* This class defines a person, its attributes (backed by JavaFX properties), the
   appropriate getter and setters.

   (Hint: as there is a clear pattern defining attributes and the needed setters and getters,
   one could apply ooRexx metaprogramming to define them in the class constructor. This is
   left as an excercise for the interested reader.)
*/
::class Item4Sale
::attribute Item4SaleID get
  expose Item4SaleID
  return Item4SaleID~get

::attribute Item4SaleID set
  expose Item4SaleID
 use arg val
  return Item4SaleID~set(val)

::attribute Item4SaleIDproperty get
  expose Item4SaleID
  return Item4SaleID

::attribute Item4SaleKat get
  expose Item4SaleKat
  return Item4SaleKat~get

::attribute Item4SaleKat set
  expose Item4SaleKat
 use arg val
  return Item4SaleKat~set(val)

::attribute Item4SaleKatproperty get
```

```
  expose Item4SaleKat
  return Item4SaleKat

::attribute Item4SaleName get
  expose Item4SaleName
  return Item4SaleName~get

::attribute Item4SaleName set
  expose Item4SaleName
 use arg val
  return Item4SaleName~set(val)

::attribute Item4SaleNameproperty get
  expose Item4SaleName
  return Item4SaleName

::attribute Item4SaleTax get
  expose Item4SaleTax
  return Item4SaleTax~get

::attribute Item4SaleTax set
  expose Item4SaleTax
 use arg val
  return Item4SaleTax~set(val)

::attribute Item4SaleTaxproperty get
  expose Item4SaleTax
  return Item4SaleTax

::attribute Item4SalePrice get
  expose Item4SalePrice
  return Item4SalePrice~get

::attribute Item4SalePrice set
  expose Item4SalePrice
 use arg val
  return Item4SalePrice~set(val)

::attribute Item4SalePriceproperty get
  expose Item4SalePrice
  return Item4SalePrice

::method init        -- constructor
  expose Item4SaleName Item4SalePrice Item4SaleKat Item4SaleTax Item4SaleID
  use arg strItem4SaleID="32", strKatName="test",strFirstName="Test Item",strItem4SaleTax="20",
strLastName="6"

Item4SaleID = .fx.SimpleStringProperty~new(strItem4SaleID)
Item4SaleKat = .fx.SimpleStringProperty~new(strKatName)
Item4SaleName= .fx.SimpleStringProperty~new(strFirstName)
Item4SaleTax = .fx.SimpleStringProperty~new(strItem4SaleTax)
  Item4SalePrice = .fx.SimpleStringProperty~new(strLastName)



::class aCustomer
::attribute aCustomerID get
  expose aCustomerID
  return aCustomerID~get

::attribute aCustomerID set
  expose aCustomerID
 use arg val
  return aCustomerID~set(val)

::attribute aCustomerIDproperty get
  expose aCustomerID
  return aCustomerID
```

```
::attribute aCustomerName get
  expose aCustomerName
  return aCustomerName~get

::attribute aCustomerName set
  expose aCustomerName
 use arg val
  return aCustomerName~set(val)

::attribute aCustomerNameproperty get
  expose aCustomerName
  return aCustomerName

::attribute aCustomerNo get
  expose aCustomerNo
  return aCustomerNo~get

::attribute aCustomerNo set
  expose aCustomerNo
 use arg val
  return aCustomerNo~set(val)

::attribute aCustomerNoproperty get
  expose aCustomerNo
  return aCustomerNo

::attribute aCustomerAddress get
  expose aCustomerAddress
  return aCustomerAddress~get

::attribute aCustomerAddress set
  expose aCustomerAddress
 use arg val
  return aCustomerAddress~set(val)

::attribute aCustomerAddressproperty get
  expose aCustomerAddress
  return aCustomerAddress

::attribute aCustomerPostcode get
  expose aCustomerPostcode
  return aCustomerPostcode~get

::attribute aCustomerPostcode set
  expose aCustomerPostcode
 use arg val
  return aCustomerPostcode~set(val)

::attribute aCustomerPostcodeproperty get
  expose aCustomerPostcode
  return aCustomerPostcode

::method init        -- constructor
  expose aCustomerID aCustomerName aCustomerNo aCustomerAddress aCustomerPostcode
  use arg  straCustomerID="6",straCustomerName="Test Item",straCustomerNo="0650",
straCustomerAddress="test", straCustomerPostcode="1020"

  aCustomerName= .fx.SimpleStringProperty~new(straCustomerName)
  aCustomerID = .fx.SimpleStringProperty~new(straCustomerID)
aCustomerAddress = .fx.SimpleStringProperty~new(straCustomerAddress)
aCustomerNo = .fx.SimpleStringProperty~new(straCustomerNo)
aCustomerPostcode = .fx.SimpleStringProperty~new(straCustomerPostcode)

::class aCartItem
::attribute aCartItemName get
  expose aCartItemName
  return aCartItemName~get
```

```
::attribute aCartItemName set
  expose aCartItemName
 use arg val
  return aCartItemName~set(val)

::attribute aCartItemNameProperty get
  expose aCartItemName
  return aCartItemName

::attribute aCartItemTax get
  expose aCartItemTax
  return aCartItemTax~get

::attribute aCartItemTax set
  expose aCartItemTax
 use arg val
  return aCartItemTax~set(val)

::attribute aCartItemTaxProperty get
  expose aCartItemTax
  return aCartItemTax

::attribute aCartItemPrice get
  expose aCartItemPrice
  return aCartItemPrice~get

::attribute aCartItemPrice set
  expose aCartItemPrice
 use arg val
  return aCartItemPrice~set(val)

::attribute aCartItemPriceProperty get
  expose aCartItemPrice
  return aCartItemPrice


::method init        -- constructor
  expose aCartItemName aCartItemPrice aCartItemTax
  use arg straCartItemName="32", straCartItemPrice="test",straCartItemTax="20"


aCartItemName= .fx.SimpleStringProperty~new(straCartItemName)
aCartItemTax = .fx.SimpleStringProperty~new(straCartItemTax)
aCartItemPrice = .fx.SimpleStringProperty~new(straCartItemPrice)


::class anInvoice
::attribute anInvoiceID get
  expose anInvoiceID
  return anInvoiceID~get

::attribute anInvoiceID set
  expose anInvoiceID
 use arg val
  return anInvoiceID~set(val)

::attribute anInvoiceIDproperty get
  expose anInvoiceID
  return anInvoiceID

::attribute anInvoiceOrderComments get
  expose anInvoiceOrderComments
  return anInvoiceOrderComments~get

::attribute anInvoiceOrderComments set
  expose anInvoiceOrderComments
 use arg val
```

```
  return anInvoiceOrderComments~set(val)

::attribute anInvoiceOrderCommentsproperty get
  expose anInvoiceOrderComments
  return anInvoiceOrderComments

::attribute anInvoiceOrderAmount get
  expose anInvoiceOrderAmount
  return anInvoiceOrderAmount~get

::attribute anInvoiceOrderAmount set
  expose anInvoiceOrderAmount
 use arg val
  return anInvoiceOrderAmount~set(val)

::attribute anInvoiceOrderAmountproperty get
  expose anInvoiceOrderAmount
  return anInvoiceOrderAmount

::attribute anInvoiceCreateTime get
  expose anInvoiceCreateTime
  return anInvoiceCreateTime~get

::attribute anInvoiceCreateTime set
  expose anInvoiceCreateTime
 use arg val
  return anInvoiceCreateTime~set(val)

::attribute anInvoiceCreateTimeproperty get
  expose anInvoiceCreateTime
  return anInvoiceCreateTime

::attribute anInvoiceDiscontDate get
  expose anInvoiceDiscontDate
  return anInvoiceDiscontDate~get

::attribute anInvoiceDiscontDate set
  expose anInvoiceDiscontDate
 use arg val
  return anInvoiceDiscontDate~set(val)

::attribute anInvoiceDiscontDateproperty get
  expose anInvoiceDiscontDate
  return anInvoiceDiscontDate


::attribute anInvoiceCustomerID get
  expose anInvoiceCustomerID
  return anInvoiceCustomerID~get

::attribute anInvoiceCustomerID set
  expose anInvoiceCustomerID
 use arg val
  return anInvoiceCustomerID~set(val)

::attribute anInvoiceCustomerIDproperty get
  expose anInvoiceCustomerID
  return anInvoiceCustomerID

::method init        -- constructor
  expose anInvoiceDiscontDate anInvoiceOrderComments anInvoiceOrderAmount anInvoiceID
anInvoiceCustomerID anInvoiceCreateTime
  use arg stranInvoiceID="7", stranInvoiceCustomerID="6",
stranInvoiceOrderComments="test",stranInvoiceDiscontDate="1020",stranInvoiceCreateTime="12",stranInvo
iceOrderAmount="0650"

  anInvoiceID= .fx.SimpleStringProperty~new(stranInvoiceID)
  anInvoiceCustomerID = .fx.SimpleStringProperty~new(stranInvoiceCustomerID)
```

```
anInvoiceOrderComments = .fx.SimpleStringProperty~new(stranInvoiceOrderComments)
anInvoiceOrderAmount = .fx.SimpleStringProperty~new(stranInvoiceOrderAmount)
anInvoiceDiscontDate = .fx.SimpleStringProperty~new(stranInvoiceDiscontDate)
anInvoiceCreateTime= .fx.SimpleStringProperty~new(stranInvoiceCreateTime)


::class Person

   -- the following attribute definitions follow a pattern that could be exploited by creating
   -- them dynamically when this class constructor runs:
   -- - the attribute's storage is a JavaFX property
   -- - the getter method returns the value stored in the property
   -- - the setter method sets the value stored in the property
   ------------------------------------------------------------------------------------------
::attribute firstName get
  expose firstName
  return firstName~get

::attribute firstName set
  expose firstName
  use arg val
  return firstName~set(val)

::attribute firstNameProperty get
  expose firstName
  return firstName

   ------------------------------------------------------------------------------------------
::attribute lastName get
  expose lastName
  return lastName~get

::attribute lastName set
  expose lastName
  use arg val
  return lastName~set(val)

::attribute lastNameProperty get
  expose lastName
  return lastName

   ------------------------------------------------------------------------------------------
::attribute street get
  expose street
  return street~get

::attribute street set
  expose street
  use arg val
  return street~set(val)

::attribute lastStreetProperty get
  expose street
  return street

   ------------------------------------------------------------------------------------------
::attribute postalCode get
  expose postalCode
  return postalCode~get

::attribute postalCode set
  expose postalCode
  use arg val
  return postalCode~set(val)

::attribute postalCodeProperty get
  expose postalCode
  return postalCode
```

```
   ------------------------------------------------------------------------------------------
::attribute city get
  expose city
  return city~get

::attribute city set
  expose city
  use arg val
  return city~set(val)

::attribute cityProperty get
  expose city
  return city

   ------------------------------------------------------------------------------------------
::attribute birthday get
   expose birthday
   return birthday~get

::attribute birthday set
   expose birthday
  use arg val
   return birthday~set(val)

::attribute birthdayProperty get
   expose birthday
   return birthday

   ------------------------------------------------------------------------------------------
::method init        -- constructor
  expose firstName lastName street postalCode city birthday
  use arg strFirstName="<enter firstName>", strLastName="<enter lastName>", -
          strStreet="some unknown street", strPostalCode=(random(1000,9999)), -
          strCity="Some City",
strBirthday=(random(1950,.dateTime~new~year)"-"random(1,12)~right(2,0)"-"random(1,28)~right(2,0))

  firstName= .fx.SimpleStringProperty~new(strFirstName)
  lastName = .fx.SimpleStringProperty~new(strLastName)

  street    = .fx.SimpleStringProperty~new(strStreet)
  postalCode= .fx.SimpleIntegerProperty~new(strPostalCode)
  city      = .fx.SimpleStringProperty~new(strCity)
  birthday  = .fx.SimpleStringProperty~new(strBirthday)


/* ============================================================================== */
/* for tutorial part 5: define the RootLayoutController

   This Rexx class controls the interaction with the fxml form named "RootLayout.fxml".

*/
/* tutorial, part 5 */
::class RootLayoutController

::method init        -- constructor
  expose mainApp -
         menuNew menuOpen menuPrint menuSave menuSaveAs menuAbout menuExit menuSettings showItemsbtn
-
sItemPageButton showInvoicesbtn btnPrintSettings iSaveUserSettings
use arg mainApp

   -- add self as the ChangeListener (we have its method "changed" implemented) and EventHandler
(method "handle" implemented)
  rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them us as for event
handling (cf. method "handle")
```

```
  rlDir=.my.app~RootLayout.fxml
  menuNew   =rlDir~menuNew   ~~setOnAction(rp)
  menuOpen  =rlDir~menuOpen  ~~setOnAction(rp)
  menuPrint =rlDir~menuPrint ~~setOnAction(rp)  -- demonstrate JavaFX printing as of JavaFX 8
  menuSave  =rlDir~menuSave  ~~setOnAction(rp)
  menuSaveAs=rlDir~menuSaveAs~~setOnAction(rp)
  menuAbout =rlDir~menuAbout ~~setOnAction(rp)
  menuExit  =rlDir~menuExit  ~~setOnAction(rp)

--btnPrintSettings =rlDir~btnPrintSettings  ~~setOnAction(rp)

--iSaveUserSettings =rlDir~iSaveUserSettings  ~~setOnAction(rp)


 menuSettings=rlDir~menuSettings ~~setOnAction(rp)

::method handle   /* implements the interface "javafx.event"   */
  expose menuNew menuOpen menuPrint menuSave menuSaveAs menuAbout menuExit mainApp menuSettings
showItemsbtn sItemPageButton showInvoicesbtn showKundenbtn btnPrintSettings iSaveUserSettings
  use arg event, slotDir

-- say self"::handle, tid="pp(bsfGetTID())  -- debug statement

  tgtObjectName=event~getTarget~objectname

  select
     when tgtObjectName=menuNew   ~objectName then self~handleNew
     when tgtObjectName=menuOpen  ~objectName then self~handleOpen
     when tgtObjectName=menuSave  ~objectName then self~handleSave
     when tgtObjectName=menuSaveAs~objectName then self~handleSaveAs
     when tgtObjectName=menuAbout ~objectName then self~handleAbout
     when tgtObjectName=menuExit  ~objectName then self~handleExit
      -- tutorial, part 6:
     when tgtObjectName=showItemsbtn~objectName then mainApp~showItems
     when tgtObjectName=showInvoicesbtn~objectName then mainApp~showInvoices
when tgtObjectName=btnPrintSettings~objectName then self~btnPrintSettings
when tgtObjectName=iSaveUserSettings~objectName then self~iSaveUserSettings

     when tgtObjectName=sItemPageButton~objectName then mainApp~showItems


 when tgtObjectName=menuSettings~objectName then mainApp~showSettings

 -- demonstrate JavaFX printing as of JavaFX 8
     when tgtObjectName=menuPrint ~objectName then mainApp~showPersonPrinterDialog

     otherwise .error~say("UNKNOWN target:" pp(event~getTarget) pp(event~getTarget~tostring))
  end


/* create an empty address book */
::method btnPrintSettings
say bingo

::method iSaveUserSettings
say bingo
::method handleNew
  expose mainApp

  mainApp~personData~clear
  mainApp~personFilePath=.nil
  mainApp~primaryStage~title="RexxPOS"

/* load person data from JSON file */
::method handleOpen
  expose mainApp

  fileChooser=.bsf~new("javafx.stage.FileChooser")
```

```
   -- crate an array of file extensions (only one in this case)
  jarrExtensions=bsf.createJavaArrayOf("java.lang.String", "*.json")
  extFilter=  .bsf~new("javafx.stage.FileChooser$ExtensionFilter", "JSON files (*.json)",
jarrExtensions)
  fileChooser~getExtensionFilters~add(extFilter)
  fileChooser~setInitialDirectory(.bsf~new("java.io.File", "."))  -- set current directory

  -- Show open file dialog
  file = fileChooser~showOpenDialog(mainApp~primaryStage)
  if file<>.nil then
  do
     filePath=file~getPath
     mainApp~loadPersonDataFromFile(filePath)   -- supply the full path, not the Java file object
     mainApp~primaryStage~title="RexxPOS -" filePath
  end


/* Saves the file to the person file that is currently open. If there is no
 * open file, the "save as" dialog is shown. */
 ::method handleSave
  expose mainApp

  if mainApp~personFilePath<>.nil then
     mainApp~savePersonDataToFile(mainApp~personFilePath)
  else
     self~handleSaveAs


 /* Opens a FileChooser to let the user select a file to save to. */
::method handleSaveAs
  expose mainApp

  fileChooser=.bsf~new("javafx.stage.FileChooser")
   -- crate an array of file extensions (only one in this case)
  jarrExtensions=bsf.createJavaArrayOf("java.lang.String", "*.json")
  extFilter=  .bsf~new("javafx.stage.FileChooser$ExtensionFilter", "JSON files (*.json)",
jarrExtensions)
  fileChooser~getExtensionFilters~add(extFilter)

  filePath=mainApp~personFilePath
  if filePath=.nil then
     fileChooser~setInitialDirectory(.bsf~new("java.io.File", "."))  -- set current directory
  else
     fileChooser~setInitialDirectory(.bsf~new("java.io.File", filespec("Directory",filePath)))  --
set directory

  -- Show save file dialog
  file = fileChooser~showSaveDialog(mainApp~primaryStage)
  if file<>.nil then
  do
     filePath=file~getPath
     if filePath~right(5)<>".json" then   -- Make sure it has the correct extension
        filePath||=".json"
     mainApp~savePersonDataToFile( filePath )
  end


/* Opens an about dialog. */
::method handleAbout
  expose mainApp

  lf="0a"x
  text="RexxPOS www.lieferex.at/rexxpos"lf-
     lf"E-mail: h0553559@gmail.com" lf -
  lf"Thanks a lot to Prof. Flatscher"lf -
lf"for his helps"lf
```

```
  if .my.app~bDialogsAlertsAvailable then
  do
     alert=.fx.alert~new(.fx.Alert.Type~information)  -- create an information alert

     alert~setTitle("RexxPOS")
     alert~setHeaderText(.nil)
     alert~setContentText(text)
     alert~showAndWait
  end
  else  -- use BSF.CLS' .bsf.dialog utility class
  do
     -- .bsf.dialog is defined in BSF.CLS and uses swing
    .bsf.dialog~messageBox(text, "RexxPOS", "information")
  end


 /* Closes the application. */
::method handleSettings
expose mainApp



::method handleExit

  bsf.loadClass("javafx.application.Platform")~exit   -- unload JavaFX, but let Rexx continue in main
thread
--"exit"
  -- .java.lang.System~exit(0)   -- the Java System class is always available by its environment name

/* ============================================================================== */
/*
*/
::class InvoicesController
::attribute iInvoicetableView
::method init        -- constructor
  expose primaryStage iInvoicetableView firstNameColumn lastNameColumn firstNameLabel lastNameLabel
streetLabel -
                     postalCodeLabel cityLabel birthdayLabel  -
                     btnNew dCustomerPageDashboard btnDelete showItemsbtn sItemPageButton -
            showInvoicesbtn showDashboardbtn showKundenbtn personData -
iTableBillColumn1 iTableBillColumn2 iTableBillColumn3 iTableBillColumn4 iTableBillColumn5
iTableBillColumn6 -
kundenData invoiceData itemData ibisCheckBox butprev butnext butdown

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them us as for event
handling (cf. method "handle")
  bsDir=.my.app~menu2.fxml
  rp=BSFCreateRexxProxy(self, ,"javafx.event.EventHandler")
   -- add us as the event handler to the buttons, the event handler (method "handle") will be able to
get at the pressed button
showKundenbtn=bsDir~showKundenbtn ~~setOnAction(rp)
iInvoiceStart=bsDir~iInvoiceStart
iInvoiceEnd=bsDir~iInvoiceEnd
--say iInvoiceStart~getValue

iInvoiceStart~setValue(.LocalDa~now()~plusDays(-30))
iInvoiceEnd~setValue(.LocalDa~now())

ibisCheckBox=bsDir~ibisCheckBox ~~setOnAction(rp)
butprev=bsDir~butprev ~~setOnAction(rp)
butnext=bsDir~butnext ~~setOnAction(rp)
butdown=bsDir~butdown ~~setOnAction(rp)


iInvoiceEnd~setDisable(.false)
--say iInvoiceStart~getValue
```

```
--bsDir~iInvoiceStart~setText("01.01.2018")
  --barChart=bsDir~barChart
  --xAxis   =bsDir~xAxis

  -- create and set the monthNames
  -- Get an array with the English month names.
 -- dfsClz=bsf.loadClass("java.text.DateFormatSymbols")
  --locClz=bsf.loadClass("java.util.Locale")
  --months = dfsClz~getInstance(locClz~ENGLISH)~getMonths
  --  Convert it to a list and add it to an ObservableList of months.
  --monthNames=.fx.FXCollections~observableArrayList
~~addAll(bsf.loadClass("java.util.Arrays")~asList(months))
 -- barChart~title="Birthday Month Distributions"
  --xAxis~setCategories(monthNames)

iInvoicetableView=bsDir~iInvoicetableView
iTableBillColumn1=bsDir~iTableBillColumn1
iTableBillColumn2=bsDir~iTableBillColumn2
iTableBillColumn3=bsDir~iTableBillColumn3
iTableBillColumn4=bsDir~iTableBillColumn4
iTableBillColumn5=bsDir~iTableBillColumn5
iTableBillColumn6=bsDir~iTableBillColumn6
iInvoicetableView~setItems(.my.app~mainApp~invoiceData)
--iInvoicetableView~setItems(invoiceData)


iTableBillColumn1~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("anInvoiceIDProper
ty"), ,"javafx.util.Callback"))
iTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("anInvoiceCustomerIDProperty"),  ,"
javafx.util.Callback"))
iTableBillColumn3
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("anInvoiceOrderAmountProperty"),  ,
"javafx.util.Callback"))

iTableBillColumn4
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("anInvoiceDiscontDateProperty"),  ,
"javafx.util.Callback"))
iTableBillColumn5
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("anInvoiceCreateTimeProperty"),  ,"
javafx.util.Callback"))
--iTableBillColumn6
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("anInvoiceOrderCommentsProperty"),
,"javafx.util.Callback"))


/*  Sets the persons to show the statistics for. */
::method handle   /* implements the interface "javafx.event"   */
  expose menuNew menuOpen menuPrint menuSave menuSaveAs menuAbout menuExit mainApp menuSettings
showItemsbtn sItemPageButton showInvoicesbtn showKundenbtn  ibisCheckBox butprev butnext butdown
  use arg event, slotDir

-- say self"::handle, tid="pp(bsfGetTID())  -- debug statement

  tgtObjectName=event~getTarget~objectname

  select
     when tgtObjectName=menuNew   ~objectName then self~handleNew
     when tgtObjectName=menuOpen  ~objectName then self~handleOpen
     when tgtObjectName=menuSave  ~objectName then self~handleSave
     when tgtObjectName=menuSaveAs~objectName then self~handleSaveAs
     when tgtObjectName=menuAbout ~objectName then self~handleAbout
     when tgtObjectName=menuExit  ~objectName then self~handleExit

when tgtObjectName=btnPrintSettings  ~objectName then self~btnPrintSettings

when tgtObjectName=iSaveUserSettings  ~objectName then self~iSaveUserSettings
```

```
 -- tutorial, part 6:
when tgtObjectName=showKundenbtn~objectName then self~showKunden

     when tgtObjectName=showItemsbtn~objectName then mainApp~showItems
     when tgtObjectName=showInvoicesbtn~objectName then mainApp~showInvoices


     when tgtObjectName=sItemPageButton~objectName then mainApp~showItems


 when tgtObjectName=menuSettings~objectName then mainApp~showSettings

 -- demonstrate JavaFX printing as of JavaFX 8
     when tgtObjectName=menuPrint ~objectName then mainApp~showPersonPrinterDialog

     otherwise .error~say("UNKNOWN target:" pp(event~getTarget) pp(event~getTarget~tostring))
  end


::method  setPersonData
  expose barChart

  use arg persons
  monthCounter=bsf.createJavaArray("int.class", 12)
  do p over persons
     parse value p~birthDay with "-" month "-"
     monthCounter[month]+=1
  end
  series=self~createMonthDataSeries(monthCounter)
  series~name="Month"
 -- barChart~getData~add(series)


/* Creates a XYChart.Data object for each month. All month data is then returned as a series. */
::method createMonthDataSeries
  expose monthNames
  use arg monthCounter

  series=.bsf~new("javafx.scene.chart.XYChart$Series")
  seriesData=series~getData
  do i=1 to monthCounter~size
      -- force number by boxing montCounter value to int
     monthData=.bsf~new("javafx.scene.chart.XYChart$Data", monthNames~get(i-1),
box("i",monthCounter[i]))
     seriesData~add(monthData)
  end
  return series

/* ============================================================================== */
/*
*/
::class DashboardController

::method init        -- constructor
  expose monthNames barChart xAxis

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them us as for event
handling (cf. method "handle")
  bsDir=.my.app~Settings.fxml
  --barChart=bsDir~barChart
  --xAxis   =bsDir~xAxis

  -- create and set the monthNames
  -- Get an array with the English month names.
 -- dfsClz=bsf.loadClass("java.text.DateFormatSymbols")
  --locClz=bsf.loadClass("java.util.Locale")
  --months = dfsClz~getInstance(locClz~ENGLISH)~getMonths
  --  Convert it to a list and add it to an ObservableList of months.
```

```
  --monthNames=.fx.FXCollections~observableArrayList
~~addAll(bsf.loadClass("java.util.Arrays")~asList(months))
 -- barChart~title="Birthday Month Distributions"
  --xAxis~setCategories(monthNames)


/*  Sets the persons to show the statistics for. */
::method  setPersonData
  expose barChart

  use arg persons
  monthCounter=bsf.createJavaArray("int.class", 12)
  do p over persons
     parse value p~birthDay with "-" month "-"
     monthCounter[month]+=1
  end
  series=self~createMonthDataSeries(monthCounter)
  series~name="Month"
 -- barChart~getData~add(series)


/* Creates a XYChart.Data object for each month. All month data is then returned as a series. */
::method createMonthDataSeries
  expose monthNames
/*  use arg monthCounter

  series=.bsf~new("javafx.scene.chart.XYChart$Series")
  seriesData=series~getData
  do i=1 to monthCounter~size
      -- force number by boxing montCounter value to int
     monthData=.bsf~new("javafx.scene.chart.XYChart$Data", monthNames~get(i-1),
box("i",monthCounter[i]))
     seriesData~add(monthData)
  end
*/
return series


/* ============================================================================== */
/*




*/
::class KundenController
::attribute cCustomerTable
::method init        -- constructor
 expose primaryStage iInvoicetableView firstNameColumn lastNameColumn firstNameLabel lastNameLabel
streetLabel -
                    postalCodeLabel cityLabel birthdayLabel  -
                    btnNew dCustomerPageDashboard btnDelete showItemsbtn sItemPageButton -
            showInvoicesbtn showDashboardbtn showKundenbtn cCustomerTable personData cartData-
cCustomerTableColumn1 cCustomerTableColumn2 cCustomerTableColumn3 cCustomerTableColumn4
cCustomerTableColumn5

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them us as for event
handling (cf. method "handle")
 -- bsDir=.my.app~kunden.fxml
  --barChart=bsDir~barChart
  --xAxis   =bsDir~xAxis

  -- create and set the monthNames
  -- Get an array with the English month names.
 -- dfsClz=bsf.loadClass("java.text.DateFormatSymbols")
  --locClz=bsf.loadClass("java.util.Locale")
  --months = dfsClz~getInstance(locClz~ENGLISH)~getMonths
```

```
  --  Convert it to a list and add it to an ObservableList of months.
  --monthNames=.fx.FXCollections~observableArrayList
~~addAll(bsf.loadClass("java.util.Arrays")~asList(months))
 -- barChart~title="Birthday Month Distributions"
  --xAxis~setCategories(monthNames)
skDir=.my.app~kunden.fxml
cCustomerTable=skDir~cCustomerTable
cCustomerTableColumn1=skDir~cCustomerTableColumn1
cCustomerTableColumn2=skDir~cCustomerTableColumn2
cCustomerTableColumn3=skDir~cCustomerTableColumn3
cCustomerTableColumn4=skDir~cCustomerTableColumn4
cCustomerTableColumn5=skDir~cCustomerTableColumn5
cCustomerTable~setItems(.my.app~mainApp~kundenData)


cCustomerTableColumn1~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCustomerIDPr
operty"), ,"javafx.util.Callback"))
  cCustomerTableColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCustomerNameProperty"),  ,"javafx
.util.Callback"))
cCustomerTableColumn3
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCustomerNoProperty"),  ,"javafx.u
til.Callback"))
cCustomerTableColumn4
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCustomerAddressProperty"),  ,"jav
afx.util.Callback"))
cCustomerTableColumn5
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCustomerPostcodeProperty"),  ,"ja
vafx.util.Callback"))


/*  Sets the persons to show the statistics for. */
::method  setPersonData
  expose barChart cCustomerTable

  use arg persons
  monthCounter=bsf.createJavaArray("int.class", 12)
  do p over persons
     parse value p~birthDay with "-" month "-"
     monthCounter[month]+=1
  end
  series=self~createMonthDataSeries(monthCounter)
  series~name="Month"
 -- barChart~getData~add(series)


/* Creates a XYChart.Data object for each month. All month data is then returned as a series. */
::method createMonthDataSeries
  expose monthNames
  use arg monthCounter
/*
  series=.bsf~new("javafx.scene.chart.XYChart$Series")
  seriesData=series~getData
  do i=1 to monthCounter~size
      -- force number by boxing montCounter value to int
     monthData=.bsf~new("javafx.scene.chart.XYChart$Data", monthNames~get(i-1),
box("i",monthCounter[i]))
     seriesData~add(monthData)
  end
*/
return series


/* ============================================================================== */
/*
```

```
*/
::class CartController
::attribute sItemTable
::method init        -- constructor
   expose primaryStage iInvoicetableView firstNameColumn lastNameColumn firstNameLabel lastNameLabel
streetLabel -
                     postalCodeLabel cityLabel birthdayLabel aCartItem cartData -
                     btnNew dCustomerPageDashboard btnDelete showItemsbtn sItemPageButton -
            showInvoicesbtn showDashboardbtn showKundenbtn sItemTable dTableBill -
dTableBillColumn1 dTableBillColumn2 dTableBillColumn3

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them us as for event
handling (cf. method "handle")
  --bsDir=.my.app~item.fxml
  --barChart=bsDir~barChart
  --xAxis   =bsDir~xAxis

  -- create and set the monthNames
  -- Get an array with the English month names.
 -- dfsClz=bsf.loadClass("java.text.DateFormatSymbols")
  --locClz=bsf.loadClass("java.util.Locale")
  --months = dfsClz~getInstance(locClz~ENGLISH)~getMonths
  --  Convert it to a list and add it to an ObservableList of months.
  --monthNames=.fx.FXCollections~observableArrayList
~~addAll(bsf.loadClass("java.util.Arrays")~asList(months))
 -- barChart~title="Birthday Month Distributions"
  --xAxis~setCategories(monthNames)

siDir=.my.app~menu1.fxml
dTableBill=siDir~dTableBill
dTableBillColumn1=siDir~dTableBillColumn1
dTableBillColumn2=siDir~dTableBillColumn2
dTableBillColumn3=siDir~dTableBillColumn3
dTableBill~setItems(.my.app~mainApp~cartData)

 dTableBillColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemNameProperty"),  ,"javafx
.util.Callback"))
 dTableBillColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemPriceProperty"),  ,"javaf
x.util.Callback"))
dTableBillColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("aCartItemTaxPrope
rty"), ,"javafx.util.Callback"))

::class ItemsController
::attribute sItemTable
::method init        -- constructor
   expose primaryStage iInvoicetableView firstNameColumn lastNameColumn firstNameLabel lastNameLabel
streetLabel -
                     postalCodeLabel cityLabel birthdayLabel  -
                     btnNew dCustomerPageDashboard btnDelete showItemsbtn sItemPageButton -
            showInvoicesbtn showDashboardbtn showKundenbtn sItemTable

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them us as for event
handling (cf. method "handle")
  --bsDir=.my.app~item.fxml
  --barChart=bsDir~barChart
  --xAxis   =bsDir~xAxis

  -- create and set the monthNames
  -- Get an array with the English month names.
 -- dfsClz=bsf.loadClass("java.text.DateFormatSymbols")
  --locClz=bsf.loadClass("java.util.Locale")
  --months = dfsClz~getInstance(locClz~ENGLISH)~getMonths
```

```
  --  Convert it to a list and add it to an ObservableList of months.
  --monthNames=.fx.FXCollections~observableArrayList
~~addAll(bsf.loadClass("java.util.Arrays")~asList(months))
 -- barChart~title="Birthday Month Distributions"
  --xAxis~setCategories(monthNames)

siDir=.my.app~item.fxml
sItemTable=siDir~sItemTable
sItemTableColumn1=siDir~sItemTableColumn1
sItemTableColumn2=siDir~sItemTableColumn2
sItemTableColumn3=siDir~sItemTableColumn3
sItemTableColumn4=siDir~sItemTableColumn4
sItemTableColumn5=siDir~sItemTableColumn5
sItemTable~setItems(.my.app~mainApp~itemData)

 sItemTableColumn1
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("Item4SaleIDProperty"),  ,"javafx.u
til.Callback"))
 sItemTableColumn2
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("Item4SaleKatProperty"),  ,"javafx.
util.Callback"))
sItemTableColumn3~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("Item4SaleNameProp
erty"), ,"javafx.util.Callback"))

sItemTableColumn4
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("Item4SaleTaxProperty"),  ,"javafx.
util.Callback"))

 sItemTableColumn5
~setCellValueFactory(BsfCreateRexxProxy(.PropertyValueFactory~new("Item4SalePriceProperty"),  ,"javaf
x.util.Callback"))


/*  Sets the persons to show the statistics for. */
::method  setPersonData
  expose barChart

  use arg persons
  monthCounter=bsf.createJavaArray("int.class", 12)
  do p over persons
     parse value p~birthDay with "-" month "-"
     monthCounter[month]+=1
  end
  series=self~createMonthDataSeries(monthCounter)
  series~name="Month"
 -- barChart~getData~add(series)


/* Creates a XYChart.Data object for each month. All month data is then returned as a series. */
::method createMonthDataSeries
  expose monthNames
  use arg monthCounter
/*
  series=.bsf~new("javafx.scene.chart.XYChart$Series")
  seriesData=series~getData
  do i=1 to monthCounter~size
      -- force number by boxing montCounter value to int
     monthData=.bsf~new("javafx.scene.chart.XYChart$Data", monthNames~get(i-1),
box("i",monthCounter[i]))
     seriesData~add(monthData)
  end
*/
return series

/* ============================================================================== */
/*
```

```
*/
::class SettingsController

::method init        -- constructor
 expose primaryStage iInvoicetableView firstNameColumn lastNameColumn firstNameLabel lastNameLabel
streetLabel -
                     postalCodeLabel cityLabel birthdayLabel  -
                     btnNew dCustomerPageDashboard btnDelete showItemsbtn sItemPageButton -
            showInvoicesbtn showDashboardbtn showKundenbtn btnPrintSettings iSaveUserSettings

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them us as for event
handling (cf. method "handle")
  siDir=.my.app~Settings.fxml
--btnPrintSettings=siDir~btnPrintSettings ~~setOnAction(rp)
--iSaveUserSettings=siDir~iSaveUserSettings ~~setOnAction(rp)
  --barChart=bsDir~barChart
  --xAxis   =bsDir~xAxis

  -- create and set the monthNames
  -- Get an array with the English month names.
 -- dfsClz=bsf.loadClass("java.text.DateFormatSymbols")
  --locClz=bsf.loadClass("java.util.Locale")
  --months = dfsClz~getInstance(locClz~ENGLISH)~getMonths
  --  Convert it to a list and add it to an ObservableList of months.
  --monthNames=.fx.FXCollections~observableArrayList
~~addAll(bsf.loadClass("java.util.Arrays")~asList(months))
 -- barChart~title="Birthday Month Distributions"
  --xAxis~setCategories(monthNames)


/*  Sets the persons to show the statistics for. */
::method  setPersonData
  expose barChart

  use arg persons
  monthCounter=bsf.createJavaArray("int.class", 12)
  do p over persons
     parse value p~birthDay with "-" month "-"
     monthCounter[month]+=1
  end
  series=self~createMonthDataSeries(monthCounter)
  series~name="Month"
 -- barChart~getData~add(series)


/* Creates a XYChart.Data object for each month. All month data is then returned as a series. */
::method createMonthDataSeries
  expose monthNames
  use arg monthCounter
/*
  series=.bsf~new("javafx.scene.chart.XYChart$Series")
  seriesData=series~getData
  do i=1 to monthCounter~size
      -- force number by boxing montCounter value to int
     monthData=.bsf~new("javafx.scene.chart.XYChart$Data", monthNames~get(i-1),
box("i",monthCounter[i]))
     seriesData~add(monthData)
  end
  */
return series


/* ================================================================================== */
/*
```

```
   This Rexx class controls the interaction with the fxml form named "PersonPrinterDialog.fxml".

*/
::class PersonPrinterDialogController
::method init
  expose  webViewControl webEngine lblHint btnPrint btnCancel personData cartData -
webEngine worker.state dialogStage currPrinter dTableBill dTotalPiece dCustomerIdText
  use arg cartData, dialogStage,dTotalP,dCustomerId

  -- fetch the directory containing the JavaFX fx:id JavaFX objects and assign them to attributes
poDir=.my.app~menu.fxml
--dTotalPiece=poDir~dTotalPiece
--dTotal=dTotalPiece~getText
--dCustomerIdText=poDir~dCustomerIdText~~getText
ppdDir=.my.app~personPrinterDialog.fxml
  webViewControl =ppdDir~webViewControl
  worker.state   =bsf.import("javafx.concurrent.Worker$State") -- load the Enumeration class to check
whether loading succeeded

  lblHint        =ppdDir~lblHint
  btnPrint       =ppdDir~btnPrint
  btnPrint ~setDisable(.true) -- make sure button is disabled (will be enabled when loading html-data
is done)
  btnCancel      =ppdDir~btnCancel


   -- add self as the ChangeListener (we have its method "changed" implemented) and EventHandler
(method "handle" implemented)
  rp=BSFCreateRexxProxy(self, ,"javafx.beans.value.ChangeListener", "javafx.event.EventHandler")

   -- add self as the ChangeListener (we have its method "changed" implemented) and EventHandler
(method "handle" implemented)
  webEngine =webViewControl~getEngine
  webEngine ~getLoadWorker ~stateProperty ~addListener(rp)

   -- add us as the event handler to the buttons, the event handler (we have its method "handle"
implemented)
  btnPrint ~setOnAction(rp)
  btnCancel~setOnAction(rp)

   -- define printer to print to
  currPrinter=bsf.importClass("javafx.print.Printer")~getDefaultPrinter


  -- carry out the creation of text and loading of it in a separate thread
  -- note: loading must be done in the JavaFX thread !
::method createAndLoadPrintData
  expose webEngine personData cartData dTableBill

  bSaveToFile=.true     -- set to .false to do it in memory only

/* MySQL Connection  */
if (conn="CONN") then do
    conn=dbconstr()
end

if (conn~isClosed) then do
    conn=dbconstr()
end


statement = conn~createStatement
sqlkoma ="select * from pos.print_settings ;"




kunde=statement~executeQuery(sqlkoma)
```

```
DO WHILE kunde~next
    iResturantName=kunde~getString("name")
    iAddress=kunde~getString("address")
    iPhoneNo=kunde~getString("phoneno")
    iCashboxId=kunde~getString("cashboxid")
    iCompanyId=kunde~getString("companyid")
    iBase64AesKey=kunde~getString("aeskey")
END
totsum=0
totmwst10=0
totmwst20=0
nettob=0
  mb=.mutableBuffer~new
  nl="0d0a"x   -- CRLF characters

  mb~~append("<!DOCTYPE html>")
  mb ~~append(nl) ~~append("<html>")

  mb ~~append(nl) ~~append("<head>")
  mb ~~append(nl) ~~append("  <title>Invoice Printing</title>")
   -- this defines the (print) CSS to use
  --mb ~~append(nl) ~~append('  <link rel="stylesheet" href="DarkThemePrint.css" type="text/css"
/>' )
  mb ~~append(nl) ~~append("</head>")

  mb ~~append(nl) ~~append("<body>")
 mb~~append(nl) ~~append('<div
align="center"><b>'iResturantName'</b><br/>'iAddress'<br/>'iPhoneNo'<br/><br/><div
align="center">Date:  'left(.dateTime~new,10)'  Time:
'left(right(.dateTime~new,15),5)'<br/></div><br/>----------------------------------------------------
-----<br/>')
  do p over cartData  -- iterate over ObservableList, create xhtml rendering

   --  mb~~append(nl) ~~append('  <div')

     mb~~append(nl) ~~append('    <span>')   ~~append(esc(p~aCartItemName))   ~~append('</span>, ')
     mb~~append(nl) ~~append('    <span>EUR ')  ~~append(esc(p~aCartItemPrice))  ~~append('.-</span>
<br/>')
   --  mb~~append(nl) ~~append('    <span >%')   ~~append(esc(p~aCartItemTax))
~~append('MwSt</span>  <br/>')
--INTERPRET 'if (p~aCartItemTax="20") then totmwst20 = totmwst20 + value(p~aCartItemPrice) * 0,2'
--INTERPRET 'if (p~aCartItemTax="10") then totmwst10 = totmwst10 + value(p~aCartItemPrice) * 0,1'
--INTERPRET 'if (p~aCartItemTax="20") then totmwst20 = totmwst20 + p~aCartItemPrice * 0,2'
--INTERPRET 'if (p~aCartItemTax="10") then totmwst10 = totmwst10 + p~aCartItemPrice * 0,1'

     totsum=totsum+p~aCartItemPrice
--INTERPRET 'mwst = totsum / 5'
--nettob=totsum-mwst
     --mb~~append(nl) ~~append('  </div>')

     mb~append(nl)
  end
--say totmwst20
INTERPRET 'nettob=value(totsum)-value(totmwst20)-value(totmwst10)'
INTERPRET 'mwst=value(totmwst20)+value(totmwst10)'
--INTERPRET 'nettob=totsum-totmwst20-totmwst10'
--INTERPRET 'mwst=totmwst20+totmwst10'
mb ~~append(nl) ~~append("------------------------------------------------------------<br/>")
  mb~~append(nl) ~~append('<span>')   ~~append(esc('TOTAL SUM: EUR 'totsum ))   ~~append('</span>.-
<br/>')
--mb~~append(nl) ~~append('<span>')   ~~append(esc('MwST: 'mwst ))   ~~append('</span>.-<br/>')

mb ~~append(nl) ~~append("------------------------------------------------------------<br/>")

  mb~~append(nl) ~~append('<br/><br/><img src="qrtest.png"/> <br/><br/>Thank you for your purchase!
</div>')
  mb ~~append(nl) ~~append("</body>")
```

```
  mb ~~append(nl) ~~append("</html>")

 -- say "mb~string:" pp(mb)           -- show user generated HTML

  currDir=directory()               -- get current directory from Rexx
 -- say "Employing:" pp(webEngine~getUserAgent)
  if bSaveToFile=.true then         -- save the html data in a file (e.g. for debugging)
  do
     fn="Invoice_printout.html" -- define file name to use

     stream=.stream~new(fn)~~open("replace")
     stream~~charout(mb~string)~~close
    say "generated HTML data saved as:" pp(fn)

        -- on Windows the path does not start with a forward slash
     if currDir~left(1)<>"/" then strUrlFn=.bsf~new("java.net.URL", "file:///"currDir"/"fn)~toString
-- Windows should start with "file:///"!
                             else strUrlFn=.bsf~new("java.net.URL", "file:"currDir"/"fn)~toString
-- Unix
     say "strUrlFn:" pp(strUrlFn)
     webEngine~load(strUrlFn)       -- load data from file
-- webEngine~load(.bsf~new("java.net.URL", "file:///"currDir"/testTable.html")~toString)
  end
  else  -- load HTML from string
  do
  --   sn="DarkThemePrint.css"        -- name of style sheet file
--if currDir~left(1)<>"/" then  strUrlStyleSheet=.bsf~new("java.net.URL",
"file:///"currDir"/"sn)~toString  -- define location of CSS to use
  --                          else  strUrlStyleSheet=.bsf~new("java.net.URL",
"file:"currDir"/"sn)~toString  -- define location of CSS to use

--     webEngine~setUserStyleSheetLocation(strUrlStyleSheet)
     webEngine~loadContent(mb~string)    -- now let the webEngine load the data into the WebView
  end


::routine esc  -- escape '&', '<', '>' with their SGML entities
  parse arg str
  return str~changeStr('&','&quot;') ~changeStr('<','&lt;') ~changeStr('>','&gt;')


::method changed unguarded    -- javafx.beans.value.ChangeListener (for webWorker)
  expose btnPrint lblHint worker.state  webEngine
  use arg obsValue, oldState, newState

  str=.dateTime~new "-> loadWorker status="pp(newState~toString)

  --.traceOutput~say("... loadWorker status changed:" str)

  if newState~toString="FAILED" then
  do
    lblHint~setText("loadWorker FAILED!") -- inform user via label
    worker=webEngine~getLoadWorker
    .error~say("FAILED! loadWorker~message:  " pp(worker~message))
    exc=worker~exception
    .error~say("        loadWorker~exception:" pp(exc~toString))
    .error~say("        stacktrace of the exception:")
    exc~printStackTrace
    return
  end

  lblHint~setText(str)
      -- using the ooRexx object name which is the same for both Enum values, if the Enum values are
the same
  if newState~objectName=worker.state~succeeded~objectName then -- text has been fully loaded
  do
     if .my.app~bDialogsAlertsAvailable then    -- on Java 1.8 or newer, all what is needed from
javafx.print.* available!
```

```
         btnPrint~setDisable(.false) -- enable print button
      else
      do
          -- .bsf.dialog is defined in BSF.CLS and uses swing
         .bsf.dialog~messageBox("You need Java 1.8/8 or newer to use the JavaFX printing feature!",
"No JavaFX-Printer Support", "error")
      end
  end

::method handle   /* implements the interface "javafx.event.EventHandler"   */
  expose btnPrint btnCancel dialogStage showInvoicesbtn showKundenbtn showItemsbtn
  use arg event

  tgtObjectName=event~getTarget~objectname
  if tgtObjectName=btnPrint~objectName then self~handlePrinting
                                       else dialogStage~close     -- close the dialog stage
(btnCancel was pressed)

::method handlePrinting
  expose  webViewControl webEngine lblHint btnPrint dialogStage currPrinter

  if currPrinter=.nil then    -- no default printer, hence no printer installed at all!
  do
     alert=.fx.alert~new(.fx.Alert.Type~ERROR)  -- create a warning alert
     alert~setTitle("No Default Printer Found")
     alert~setHeaderText("Printer Missing")
     alert~setContentText("No printer found, please install one on this system!")
     alert~showAndWait
     return
  end

  lblHint~text=pp(currPrinter~toString)

/*
--- >
  paper      =bsf.loadClass("javafx.print.Paper")~a4
  orientation=bsf.loadClass("javafx.print.PageOrientation")~portrait
  margins    =bsf.loadClass("javafx.print.Printer$MarginType")~default
  pageLayout=defPrinter~createPageLayout(paper,orientation,margins)

/* the following is meant for printing a GUI-node from screen to printer, hence the transforming, cf.
   <https://carlfx.wordpress.com/2013/07/15/introduction-by-example-javafx-8-printing/>: */

  -- transform screen viewport to print viewport dimensions
  boundsInParent=webViewControl~getBoundsInParent
  scaleX=pageLayout~getPrintableWidth  / boundsInParent~width
  scaleY=pageLayout~getPrintableHeight / boundsInParent~height

say "webViewControl="pp(webViewControl)
say "webViewControl~toString="pp(webViewControl~toString)
  webViewControl~getTransforms~add(.bsf~new("javafx.scene.transform.Scale", scaleX, scaleY) )
--- <

  job=bsf.loadClass("javafx.print.PrinterJob")~createPrinterJob
  if job<>.nil, job~printPage(webViewControl) then job~endJob
*/

  job=bsf.loadClass("javafx.print.PrinterJob")~createPrinterJob

  if job<>.nil then
  do
      -- allow user to switch printer and set page properties
     if job~showPrintDialog(dialogStage)=.false then  -- user cancelled printing
     do
        lblHint~text="Printing cancelled by user"
        job~endJob
        return
     end
```

```
     jobPrinter=job~getPrinter      -- get selected printer
     if jobPrinter~objectName<>currPrinter~objectname then
     do
        lblHint~text="Printing to:" pp(jobPrinter~toString)
        currPrinter=jobPrinter
     end

     webEngine~print(job)  -- using the WebView's WebEngine to print, which is able to do multiple
page printings
     job~endJob
  end



/* ============================================================================== */
/* for tutorial part 3: interacting with the user */
   /* Instead of a class we define a Rexx routine that carries out the checking */
::Routine validString    -- returns .true, if date is valid, .false else
  parse arg date
  signal on syntax
  .DateTime~fromStandardDate(date, "-")
  return .true
syntax:     -- Rexx condition raised
  return .false



/*
      ----------------------- Apache Version 2.0 license -------------------------
         Copyright 2018 H. Kaya Aydin

         Licensed under the Apache License, Version 2.0 (the "License");
         you may not use this file except in compliance with the License.
         You may obtain a copy of the License at

             http://www.apache.org/licenses/LICENSE-2.0

         Unless required by applicable law or agreed to in writing, software
         distributed under the License is distributed on an "AS IS" BASIS,
         WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
         See the License for the specific language governing permissions and
         limitations under the License.
      ----------------------------------------------------------------------------
*/
```

Source code of the file "**menu1.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.net.URL?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ScrollPane?>
<?import javafx.scene.control.Tab?>
<?import javafx.scene.control.TabPane?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextArea?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.layout.VBox?>
<?language rexx?>

<VBox id="" alignment="TOP_CENTER" prefHeight="-1.0" prefWidth="-1.0"
xmlns="http://javafx.com/javafx/8.0.141" xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <AnchorPane id="AnchorPane" VBox.vgrow="ALWAYS">
      <children>
        <VBox id="VBox" alignment="CENTER" spacing="5.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
          <children>

            <AnchorPane id="dark-blue2" minHeight="-1.0" minWidth="-1.0" VBox.vgrow="ALWAYS">
              <children>
                <VBox id="VBox" alignment="CENTER" spacing="5.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
                  <children>
                    <HBox alignment="TOP_CENTER" spacing="2.0">
                      <children>
                                    <HBox alignment="TOP_CENTER" prefHeight="-1.0" prefWidth="-1.0"
spacing="2.0">
                                       <children>
                                          <Button id="custom-button-imageButton"
fx:id="showDashboardbtn" minHeight="40.0" minWidth="70.0" mnemonicParsing="false" text="Dashboard"
visible="false">
                            <graphic>
                               <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                  <image>
                                     <Image url="@dashboard.png" />
                                  </image>
                               </ImageView>
                            </graphic>
                         </Button>
                <Button id="custom-button-imageButton" fx:id="showKundenbtn" minHeight="40.0"
minWidth="70.0" mnemonicParsing="false" text="Customer page">
                            <graphic>
                               <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                  <image>
                                     <Image url="@customer.png" />
                                  </image>
                               </ImageView>
                            </graphic></Button>
                <Button id="custom-button-imageButton" fx:id="showItemsbtn" minHeight="40.0"
minWidth="70.0" mnemonicParsing="false" text="Item Page">
                            <graphic>
```

```
                                <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                    <image>
                                        <Image url="@img3.png" />
                                    </image>
                                </ImageView>
                            </graphic></Button>
                <Button id="custom-button-imageButton" fx:id="showInvoicesbtn" minHeight="40.0"
minWidth="70.0" mnemonicParsing="false" text="Invoices">
                            <graphic>
                                <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                    <image>
                                        <Image url="@img2.png" />
                                    </image>
                                </ImageView>
                            </graphic></Button>
                                        </children>
                                    </HBox>
                        </children>
                    </HBox>
                    <HBox VBox.vgrow="ALWAYS">
                        <children>
                            <VBox alignment="TOP_CENTER" fillWidth="false" minWidth="50.0" prefWidth="-
1.0" spacing="5.0" HBox.hgrow="SOMETIMES">
                                        <children>
                                            <TabPane side="LEFT" tabClosingPolicy="ALL_TABS"
tabMinHeight="50.0">
                                                <tabs>
                                                    <Tab closable="false" text="Order Tab" />
                                                </tabs>
                                            </TabPane>
                                        </children></VBox>
                        <VBox alignment="TOP_CENTER" fillWidth="false" maxWidth="310.0" minWidth="-
1.0" prefHeight="-1.0" prefWidth="-1.0" spacing="5.0" HBox.hgrow="ALWAYS">
                            <children>
                                <HBox maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308">
                                    <children>
                                        <TextField id="custom-textField-tf1" fx:id="dCustomerIdText"
maxWidth="1.7976931348623157E308" minHeight="25.0" prefHeight="25.0" prefWidth="129.0" text="Customer
ID" HBox.hgrow="ALWAYS" />
                                        <TextField id="custom-textField-tf1" fx:id="dCustomerNameText"
maxWidth="1.7976931348623157E308" minHeight="25.0" prefHeight="25.0" prefWidth="150.0" text="Name"
HBox.hgrow="ALWAYS" />
                                    </children>
                                </HBox>
                                <HBox maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308">
                                    <children>
                                        <TextArea fx:id="dCustomerAddressText" maxHeight="74.0"
prefHeight="74.0" prefWidth="121.0" text="Address" wrapText="true" />
                                                <VBox prefHeight="74.0" prefWidth="151.0">
                                                    <children>
                                                        <HBox maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308">
                                                            <children>
                                                                <TextArea fx:id="dCustomerNumberText"
maxHeight="104.0" prefHeight="74.0" prefWidth="155.0" text="Phone" wrapText="true" />
                                                            </children>
                                                        </HBox>
                                                    </children>
                                                </VBox>
                                    </children>
                                </HBox>
                                <HBox id="HBox" spacing="5.0" VBox.vgrow="ALWAYS">
                                    <children>
                                        <ScrollPane id="ScrollPane" prefViewportHeight="360.0"
```

```
prefViewportWidth="300.0" HBox.hgrow="ALWAYS">
                                    <content>
                                      <TableView id="custom-table-table1" fx:id="dTableBill"
editable="true" maxHeight="-1.0" maxWidth="-1.0" minWidth="282.0" prefHeight="400.0"
prefWidth="282.0">
                                        <columns>
                                          <TableColumn fx:id="dTableBillColumn1" editable="false"
maxWidth="5000.0" minWidth="80.0" prefWidth="130.0" text="Item" />
                                                                                <TableColumn
fx:id="dTableBillColumn2" editable="true" minWidth="0.0" prefWidth="82.0" text="Price" />
                                                                    <TableColumn fx:id="dTableBillColumn3"
editable="true" minWidth="30.0" prefWidth="52.0" text="Tax" />

                                        </columns>
                                      </TableView>
                                    </content>
                                  </ScrollPane>
                                </children>
                              </HBox>
                              <VBox id="VBox" alignment="CENTER" spacing="5.0" VBox.vgrow="ALWAYS">
                                <children>
                                  <GridPane maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" prefHeight="119.0" prefWidth="346.0" vgap="5.0">
                                    <children>
                                      <TextField minHeight="20.0" prefWidth="200.0" text="TOTAL"
GridPane.columnIndex="2" />
                                      <TextField fx:id="dTotalPiece" editable="false" prefWidth="200.0"
GridPane.columnIndex="3" GridPane.rowIndex="0" />
                                      <TextField id="custom-textField-tf1" fx:id="dInvoiceComments"
minHeight="20.0" prefHeight="23.0" prefWidth="168.0" promptText="" GridPane.columnIndex="1"
GridPane.columnSpan="3" GridPane.rowIndex="1" />
                                      <Button id="custom-button-imageButton" fx:id="delle"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="24.0"
mnemonicParsing="false" prefWidth="55.0" text="Del" GridPane.columnIndex="1" />
                                                                        <Button id="custom-button-imageButton"
fx:id="okle" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="24.0"
mnemonicParsing="false" prefHeight="57.0" prefWidth="110.0" text="OK" GridPane.rowSpan="2" />
                                    </children>
                                    <columnConstraints>
                                      <ColumnConstraints halignment="CENTER" hgrow="ALWAYS"
minWidth="10.0" prefWidth="100.0" />
                                      <ColumnConstraints halignment="CENTER" hgrow="ALWAYS"
minWidth="10.0" prefWidth="100.0" />
                                      <ColumnConstraints halignment="CENTER" hgrow="ALWAYS"
minWidth="10.0" prefWidth="100.0" />
                                      <ColumnConstraints halignment="CENTER" hgrow="ALWAYS"
minWidth="10.0" prefWidth="100.0" />
                                    </columnConstraints>
                                    <rowConstraints>
                                      <RowConstraints valignment="CENTER" vgrow="ALWAYS" />
                                      <RowConstraints valignment="CENTER" vgrow="ALWAYS" />
                                    </rowConstraints>
                                  </GridPane>
                                </children>
                              </VBox>
                              <VBox>
                                <children>
                                  <HBox spacing="0.0">
                                    <children>
                                      <GridPane hgap="2.0" vgap="5.0" visible="false"
HBox.hgrow="ALWAYS">
                                        <children>
                                          <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton2" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="2" textFill="WHITE" GridPane.columnIndex="1" GridPane.rowIndex="0" />
                                          <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton4" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="4" textFill="WHITE" GridPane.columnIndex="0" GridPane.rowIndex="1" />
```

```
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton3" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="3" textFill="WHITE" GridPane.columnIndex="2" GridPane.rowIndex="0" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton5" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="5" textFill="WHITE" GridPane.columnIndex="1" GridPane.rowIndex="1" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton6" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="6" textFill="WHITE" GridPane.columnIndex="2" GridPane.rowIndex="1" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorResetButton" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="C" textFill="WHITE" GridPane.columnIndex="0" GridPane.rowIndex="3" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton7" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="7" textFill="WHITE" GridPane.columnIndex="0" GridPane.rowIndex="2" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton8" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="8" textFill="WHITE" GridPane.columnIndex="1" GridPane.rowIndex="2" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton9" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="9" textFill="WHITE" GridPane.columnIndex="2" GridPane.rowIndex="2" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton1" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="1" textFill="WHITE" GridPane.columnIndex="0" GridPane.rowIndex="0" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorNumberButton0" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefWidth="49.0" text="0" textFill="WHITE" GridPane.columnIndex="1" GridPane.rowIndex="3" />
                                        <Button id="custom-button-skyDarkBlueCalculator"
fx:id="dCalculatorCommaButton" minHeight="-1.0" minWidth="-1.0" mnemonicParsing="false"
prefHeight="31.0" prefWidth="49.0" text="." textFill="WHITE" GridPane.columnIndex="2"
GridPane.rowIndex="3" />
                                    </children>
                                    <columnConstraints>
                                      <ColumnConstraints halignment="CENTER" hgrow="ALWAYS"
minWidth="-1.0" prefWidth="-1.0" />
                                      <ColumnConstraints halignment="CENTER" hgrow="ALWAYS"
minWidth="-1.0" prefWidth="-1.0" />
                                      <ColumnConstraints halignment="CENTER" hgrow="ALWAYS"
minWidth="-1.0" prefWidth="-1.0" />
                                    </columnConstraints>
                                    <padding>
                                      <Insets right="10.0" />
                                    </padding>
                                    <rowConstraints>
                                      <RowConstraints minHeight="-1.0" prefHeight="-1.0"
valignment="CENTER" vgrow="ALWAYS" />
                                      <RowConstraints minHeight="-1.0" prefHeight="-1.0"
valignment="CENTER" vgrow="ALWAYS" />
                                      <RowConstraints minHeight="-1.0" prefHeight="-1.0"
valignment="CENTER" vgrow="ALWAYS" />
                                      <RowConstraints minHeight="-1.0" prefHeight="-1.0"
valignment="CENTER" vgrow="ALWAYS" />
                                    </rowConstraints>
                                  </GridPane>
                                  <GridPane maxWidth="120.0" visible="false" HBox.hgrow="ALWAYS">
                                    <children>
                                      <Button id="custom-button-imageButton"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="24.0" minWidth="-1.0"
mnemonicParsing="false" prefHeight="41.0" prefWidth="28.0" text="+" GridPane.rowIndex="1" />
                                      <Button id="custom-button-imageButton"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="24.0" minWidth="-1.0"
mnemonicParsing="false" prefHeight="41.0" prefWidth="123.0" text="Bal" GridPane.rowIndex="2" />
                                      <Button id="custom-button-imageButton"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="24.0"
mnemonicParsing="false" prefWidth="55.0" GridPane.rowIndex="3">
                                        <graphic><ImageView fitHeight="37.0" fitWidth="59.0"
pickOnBounds="true" preserveRatio="true"><image><Image url="@img1.png"
/></image></ImageView></graphic></Button>
```

```
                                    <Button id="custom-button-imageButton"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="24.0"
mnemonicParsing="false" prefHeight="53.0" prefWidth="49.0" text="-" GridPane.columnIndex="1"
GridPane.rowIndex="1" />
                                    <TextField id="custom-textField-tf1"
fx:id="dCalculatorScreenText" minHeight="20.0" prefHeight="48.0" prefWidth="100.0"
GridPane.columnSpan="2" />
                                    <Button id="custom-button-imageButton"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" mnemonicParsing="false"
prefHeight="40.0" prefWidth="-1.0" text="Enter" wrapText="true" GridPane.columnIndex="1"
GridPane.rowIndex="2" GridPane.rowSpan="2" />
                                  </children>
                                  <columnConstraints>
                                    <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES"
minWidth="10.0" prefWidth="100.0" />
                                    <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES"
minWidth="10.0" prefWidth="100.0" />
                                  </columnConstraints>
                                  <rowConstraints>
                                    <RowConstraints minHeight="25.0" prefHeight="25.0"
vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="25.0" prefHeight="25.0"
vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="25.0" prefHeight="25.0"
vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="25.0" prefHeight="25.0"
vgrow="SOMETIMES" />
                                  </rowConstraints>
                                </GridPane>
                              </children>
                            </HBox>
                          </children>
                        </VBox>
                      </children>
                    </VBox>
                    <VBox spacing="0.0" HBox.hgrow="ALWAYS">
                      <children>
                        <TextField id="custom-textField-tf1" fx:id="sucheingabe" minHeight="25.0"
prefWidth="200.0" />
                        <HBox prefHeight="280.0" prefWidth="200.0" VBox.vgrow="ALWAYS">
                                      <children>
                                         <GridPane fx:id="dPizzaItemGrid"
gridLinesVisible="false" hgap="1.0" maxHeight="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="200.0" prefWidth="945.0" vgap="1.0" visible="false">
                                           <children>
                                              <Button id="custom-button-lightPink"
fx:id="dPizzaItemButton2" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK"
wrapText="true" GridPane.columnIndex="1" />
                                              <Button id="custom-button-yellow"
fx:id="dPizzaItemButton1" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK" wrapText="true"
GridPane.columnIndex="0">
                                                 <GridPane.margin>
                                                    <Insets />
                                                 </GridPane.margin>
                                              </Button>
                                              <Button id="custom-button-moonLightPink"
fx:id="dPizzaItemButton3" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK"
wrapText="true" GridPane.columnIndex="2" />
                                              <Button id="custom-button-pinkDarkligght"
fx:id="dPizzaItemButton9" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="1"
GridPane.rowIndex="1" />
                                              <Button id="custom-button-parrot"
fx:id="dPizzaItemButton8" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="0"
```

```
GridPane.rowIndex="1" />
                                                <Button id="custom-button-darkCreem"
fx:id="dPizzaItemButton10" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="2"
GridPane.rowIndex="1" />
                                                <Button id="custom-button-skyBlue"
fx:id="dPizzaItemButton4" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK"
wrapText="true" GridPane.columnIndex="3" />
                                                <Button id="custom-button-sceenColor"
fx:id="dPizzaItemButton11" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="3"
GridPane.rowIndex="1" />
                                                <Button id="custom-button-skyBlue"
fx:id="dPizzaItemButton12" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="4"
GridPane.rowIndex="1" />
                                                <Button id="custom-button-yellow"
fx:id="dPizzaItemButton5" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="4" />
                                                <Button id="custom-button-skyDarkBlue"
fx:id="dPizzaItemButton6" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="5" />
                                                <Button id="custom-button-pinkDarkligght"
fx:id="dPizzaItemButton13" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="5"
GridPane.rowIndex="1" />
                                                <Button id="custom-button-parrot"
fx:id="dPizzaItemButton7" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="6" />
                                                <Button id="custom-button-lightPink"
fx:id="dPizzaItemButton14" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK"
wrapText="true" GridPane.columnIndex="6" GridPane.rowIndex="1" />
                                                <Button id="custom-button-lightPink"
fx:id="dPizzaItemButton15" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK"
wrapText="true" GridPane.rowIndex="2" />
                                                <Button id="custom-button-yellow"
fx:id="dPizzaItemButton16" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK" wrapText="true"
GridPane.columnIndex="1" GridPane.rowIndex="2" />
                                                <Button id="custom-button-darkCreem"
fx:id="dPizzaItemButton19" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="4"
GridPane.rowIndex="2" />
                                                <Button id="custom-button-skyBlue"
fx:id="dPizzaItemButton17" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK"
wrapText="true" GridPane.columnIndex="2" GridPane.rowIndex="2" />
                                                <Button id="custom-button-skyBlue"
fx:id="dPizzaItemButton20" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="5"
GridPane.rowIndex="2" />
                                                <Button id="custom-button-skyDarkBlue"
fx:id="dPizzaItemButton18" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" GridPane.columnIndex="3"
GridPane.rowIndex="2" />
                                            </children>
                                            <columnConstraints>
                                                <ColumnConstraints halignment="CENTER"
hgrow="ALWAYS" minWidth="-1.0" prefWidth="-1.0" />
                                                <ColumnConstraints halignment="CENTER"
hgrow="ALWAYS" minWidth="-1.0" prefWidth="-1.0" />
                                                <ColumnConstraints halignment="CENTER"
hgrow="ALWAYS" minWidth="-1.0" prefWidth="-1.0" />
                                                <ColumnConstraints halignment="CENTER"
hgrow="ALWAYS" minWidth="-1.0" prefWidth="-1.0" />
```

```
                                                <ColumnConstraints halignment="CENTER"
hgrow="ALWAYS" minWidth="-1.0" prefWidth="-1.0" />
                                                <ColumnConstraints halignment="CENTER"
hgrow="ALWAYS" minWidth="-1.0" prefWidth="-1.0" />
                                                <ColumnConstraints halignment="CENTER"
hgrow="ALWAYS" minWidth="-1.0" prefWidth="-1.0" />
                                             </columnConstraints>
                                             <rowConstraints>
                                                <RowConstraints minHeight="60.0"
prefHeight="30.0" valignment="CENTER" vgrow="SOMETIMES" />
                                                <RowConstraints minHeight="60.0"
prefHeight="30.0" valignment="CENTER" vgrow="SOMETIMES" />
                                                <RowConstraints minHeight="60.0"
prefHeight="30.0" valignment="CENTER" vgrow="SOMETIMES" />
                                             </rowConstraints>
                                          </GridPane>
                                       </children></HBox>
                                 <HBox id="HBox" alignment="CENTER" fillHeight="false" maxHeight="-
Infinity" spacing="0.0" VBox.vgrow="ALWAYS">
                                    <children>
                                       <GridPane gridLinesVisible="false" hgap="1.0" vgap="1.0"
GridPane.halignment="CENTER" GridPane.hgrow="ALWAYS" GridPane.valignment="CENTER"
GridPane.vgrow="ALWAYS" HBox.hgrow="ALWAYS">
                                          <children>
                                             <Button id="custom-button-darkCreem" fx:id="dFoodMenuButton1"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="0"
GridPane.rowIndex="0" />
                                             <Button id="custom-button-yellow" fx:id="dFoodMenuButton2"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="1"
GridPane.rowIndex="0" />
                                             <Button id="custom-button-lightPink" fx:id="dFoodMenuButton9"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK" wrapText="true"
GridPane.columnIndex="3" GridPane.rowIndex="1" />
                                             <Button id="custom-button-yellow" fx:id="dFoodMenuButton8"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK" wrapText="true"
GridPane.columnIndex="2" GridPane.rowIndex="1" />
                                             <Button id="custom-button-lightPink" fx:id="dFoodMenuButton3"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="2" />
                                             <Button id="custom-button-moonLightPink"
fx:id="dFoodMenuButton10" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK"
wrapText="true" GridPane.columnIndex="4" GridPane.rowIndex="1" />
                                             <Button id="custom-button-pinkDarkligght"
fx:id="dFoodMenuButton16" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" wrapText="true"
GridPane.rowIndex="3" />
                                             <Button id="custom-button-parrot" fx:id="dFoodMenuButton15"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="4"
GridPane.rowIndex="2" />
                                             <Button id="custom-button-darkCreem" fx:id="dFoodMenuButton17"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="1"
GridPane.rowIndex="3" />
                                             <Button id="custom-button-skyBlue" fx:id="dFoodMenuButton4"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" wrapText="true" GridPane.columnIndex="3" />
                                             <Button id="custom-button-skyBlue" fx:id="dFoodMenuButton11"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK" wrapText="true" GridPane.rowIndex="2"
/>
                                             <Button id="custom-button-skyBlue" fx:id="dFoodMenuButton18"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
```

```
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="2"
GridPane.rowIndex="3" />
                                    <Button id="custom-button-skyBlue" fx:id="dFoodMenuButton19"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="3"
GridPane.rowIndex="3" />
                                    <Button id="custom-button-yellow" fx:id="dFoodMenuButton12"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="1"
GridPane.rowIndex="2" />
                                    <Button id="custom-button-skyDarkBlue" fx:id="dFoodMenuButton5"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textAlignment="CENTER" wrapText="true" GridPane.columnIndex="4" />
                                    <Button id="custom-button-skyDarkBlue" fx:id="dFoodMenuButton6"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.rowIndex="1" />
                                    <Button id="custom-button-skyDarkBlue" fx:id="dFoodMenuButton13"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="2"
GridPane.rowIndex="2" />
                                    <Button id="custom-button-pinkDarkligght"
fx:id="dFoodMenuButton20" maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"
minHeight="-1.0" minWidth="65.0" mnemonicParsing="false" textFill="BLACK" wrapText="true"
GridPane.columnIndex="4" GridPane.rowIndex="3" />
                                    <Button id="custom-button-darkCreem" fx:id="dFoodMenuButton7"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textAlignment="CENTER" textFill="BLACK" wrapText="true"
GridPane.columnIndex="1" GridPane.rowIndex="1" />
                                    <Button id="custom-button-parrot" fx:id="dFoodMenuButton14"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="-1.0" minWidth="65.0"
mnemonicParsing="false" textFill="BLACK" wrapText="true" GridPane.columnIndex="3"
GridPane.rowIndex="2" />
                                </children>
                                <columnConstraints>
                                  <ColumnConstraints halignment="CENTER" hgrow="ALWAYS" minWidth="-
1.0" prefWidth="-1.0" />
                                  <ColumnConstraints halignment="CENTER" hgrow="ALWAYS" minWidth="-
1.0" prefWidth="-1.0" />
                                  <ColumnConstraints halignment="CENTER" hgrow="ALWAYS" minWidth="-
1.0" prefWidth="-1.0" />
                                  <ColumnConstraints halignment="CENTER" hgrow="ALWAYS" minWidth="-
1.0" prefWidth="-1.0" />
                                  <ColumnConstraints halignment="CENTER" hgrow="ALWAYS" minWidth="-
1.0" prefWidth="-1.0" />
                                </columnConstraints>
                                <rowConstraints>
                                                  <RowConstraints minHeight="60.0"
prefHeight="30.0" valignment="CENTER" vgrow="ALWAYS" />
                                  <RowConstraints fillHeight="true" minHeight="60.0"
prefHeight="30.0" valignment="CENTER" vgrow="ALWAYS" />
                                  <RowConstraints minHeight="60.0" prefHeight="30.0"
valignment="CENTER" vgrow="SOMETIMES" />
                                  <RowConstraints minHeight="60.0" prefHeight="30.0"
valignment="CENTER" vgrow="SOMETIMES" />
                                </rowConstraints>
                              </GridPane>
                            </children>
                          </HBox>
                        </children>
                        <padding>
                          <Insets left="5.0" />
                        </padding>
                      </VBox>
                    </children>
                  </HBox>
                </children>
              </VBox>
            </children>
```

```
            </AnchorPane>
          </children>
        </VBox>
      </children>
    </AnchorPane>
  </children>
  <stylesheets>
    <URL value="@button.css" />
  </stylesheets>
<fx:script source="put_FXID_objects_into.my.app.rex" />
</VBox>
```

Source code of the file "**menu2.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.net.*?>

<?import javafx.collections.*?>
<?import javafx.geometry.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.image.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.paint.*?>
<?import javafx.scene.text.*?>
<?import java.lang.String?>
<?import java.net.URL?>
<?import javafx.collections.FXCollections?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.CheckBox?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.DatePicker?>
<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.control.ScrollPane?>
<?import javafx.scene.control.SeparatorMenuItem?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextArea?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.paint.Color?>
<?import javafx.scene.text.Text?>
<?language rexx?>


<VBox id="" alignment="TOP_CENTER" maxHeight="1.7976931348623157E308" prefHeight="-1.0"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <AnchorPane id="" maxHeight="1.7976931348623157E308" maxWidth="-1.0" prefHeight="-1.0"
prefWidth="-1.0" VBox.vgrow="ALWAYS">


   <children>
        <VBox id="VBox" alignment="CENTER" maxHeight="1.7976931348623157E308" spacing="5.0"
AnchorPane.bottomAnchor="0.0" AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="0.0">
          <children>

            <HBox alignment="TOP_CENTER" prefHeight="42.0" prefWidth="898.0" spacing="2.0">
              <children>
                        <Button id="custom-button-imageButton" minHeight="40.0" minWidth="70.0"
mnemonicParsing="false" fx:id="showDashboardbtn"  visible="false" text="back to Dashboard">
                           <graphic>
                              <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                 <image>
                                    <Image url="@dashboard.png" />
                                 </image>
                              </ImageView>
```

```
                        </graphic>
                     </Button>
                  <Button id="custom-button-imageButton" fx:id="showKundenbtn"  visible="false"
minHeight="40.0" minWidth="70.0" mnemonicParsing="false"  text="Customer page">
                        <graphic>
                           <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                              <image>
                                 <Image url="@customer.png" />
                              </image>
                           </ImageView>
                        </graphic></Button>
                  <Button id="custom-button-imageButton" minHeight="40.0" minWidth="70.0"
visible="false" mnemonicParsing="false"  fx:id="showItemsbtn" text="Item Page">
                        <graphic>
                           <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                              <image>
                                 <Image url="@img3.png" />
                              </image>
                           </ImageView>
                        </graphic></Button>
                  <Button id="custom-button-imageButton" minHeight="40.0" fx:id="showInvoicesbtn"
visible="false" minWidth="70.0" mnemonicParsing="false"  text="Invoices">
                        <graphic>
                           <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                              <image>
                                 <Image url="@img2.png" />
                              </image>
                           </ImageView>
                        </graphic></Button>
               </children>
            </HBox>
            <VBox id="VBox" alignment="CENTER" maxHeight="1.7976931348623157E308"
VBox.vgrow="ALWAYS">
               <children>
                  <AnchorPane id="custom-anchor-gray" maxHeight="1.7976931348623157E308" minHeight="-
1.0" VBox.vgrow="ALWAYS">
                     <children>
                        <VBox id="VBox" alignment="CENTER" maxHeight="1.7976931348623157E308"
prefHeight="564.0" prefWidth="898.0" spacing="5.0" AnchorPane.bottomAnchor="11.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
                           <children>
                              <HBox spacing="5.0">
                                 <children>
                                    <VBox alignment="TOP_LEFT" fillWidth="false" spacing="5.0"
HBox.hgrow="SOMETIMES">
                                       <children>
                                          <VBox id="custom-vbox-black" prefHeight="207.0" prefWidth="226.0"
VBox.vgrow="ALWAYS">
                                             <children>
                                                <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Select
Date(s):">
                                                   <fill>
                                                      <Color blue="0.000" green="0.478" red="1.000" fx:id="x11" />
                                                   </fill>
                                                </Text>
                                                <VBox prefHeight="150.0" prefWidth="100.0" spacing="5.0">
                                                   <children>
                                                      <HBox alignment="CENTER" spacing="3.0">
                                                         <children>
                                                            <HBox alignment="CENTER" minWidth="52.0">
                                                               <children>
                                                                  <Text fill="$x11" strokeType="OUTSIDE"
strokeWidth="0.0" text="from:" />
                                                               </children>
                                                            </HBox>
```

```
                                        <DatePicker fx:id="iInvoiceStart" minHeight="30.0"
minWidth="120.0"  />
                                    </children>
                                  </HBox>
                                  <HBox alignment="CENTER" spacing="3.0">
                                    <children>
                                      <HBox alignment="CENTER" minWidth="52.0">
                                        <children>
 <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0" text=" to:" />

                                        </children>
                                      </HBox>
                                          <DatePicker fx:id="iInvoiceEnd" disable="true"
minHeight="30.0" minWidth="120.0"  />
                                    </children>
                                  </HBox>
                                  <HBox alignment="CENTER" spacing="3.0">
                                    <children>
                                      <Button fx:id="butprev" mnemonicParsing="false" text="">
                                        <graphic>
                                          <ImageView fitHeight="50.0" fitWidth="50.0"
pickOnBounds="true" preserveRatio="true">
                                            <image>
                                              <Image url="@arrow_back.png" />
                                            </image>
                                          </ImageView>
                                        </graphic>
                                      </Button>
                                      <Button fx:id="butnext" mnemonicParsing="false"  text="">
                                        <graphic>
                                          <ImageView fitHeight="50.0" fitWidth="50.0"
pickOnBounds="true" preserveRatio="true">
                                            <image>
                                              <Image url="@arrow.png" />
                                            </image>
                                          </ImageView>
                                        </graphic>
                                      </Button>
                                                                <Button fx:id="butdown"
layoutX="104.0" layoutY="10.0" mnemonicParsing="false" text="">
                                                                   <graphic>
                                                                      <ImageView fitHeight="50.0"
fitWidth="50.0"  pickOnBounds="true" preserveRatio="true">
                                                                         <image>
                                                                            <Image url="@on.png" />
                                                                         </image>
                                                                      </ImageView>
                                                                   </graphic>
                                                                </Button>
                                    </children>
                                  </HBox>
                                </children>
                              </VBox>
                            </children>
                            <padding>
                              <Insets left="5.0" right="5.0" fx:id="x1" />
                            </padding>
                          </VBox>
                          <VBox id="custom-vbox-black" prefHeight="138.0" prefWidth="226.0"
VBox.vgrow="ALWAYS"  visible="false">
                            <children>
                              <Text strokeType="OUTSIDE" strokeWidth="0.0" text="Auswahl:">
                                <fill>
                                  <Color blue="0.000" green="0.480" red="1.000" fx:id="x3" />
                                </fill>
                              </Text>
                              <GridPane>
                                <children>
```

```
                                        <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
text="User:" GridPane.columnIndex="0" GridPane.rowIndex="0" />
                                        <Text fill="$x11" scaleX="1.0" scaleY="1.0"
strokeType="OUTSIDE" strokeWidth="0.0" text="Ebene:" GridPane.columnIndex="0" GridPane.rowIndex="1"
/>
                                        <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
text="Objekt:" GridPane.columnIndex="0" GridPane.rowIndex="2" />
                                        <ComboBox minHeight="25.0" minWidth="90.0" prefWidth="105.0"
GridPane.columnIndex="1" GridPane.rowIndex="0">
                                          <items>
                                            <FXCollections fx:factory="observableArrayList">
                                              <String fx:value="Item 1" />
                                              <String fx:value="Item 2" />
                                              <String fx:value="Item 3" />
                                            </FXCollections>
                                          </items>
                                        </ComboBox>
                                        <ComboBox minHeight="25.0" prefWidth="105.0"
GridPane.columnIndex="1" GridPane.rowIndex="2">
                                          <items>
                                            <FXCollections fx:factory="observableArrayList">
                                              <String fx:value="Item 1" />
                                              <String fx:value="Item 2" />
                                              <String fx:value="Item 3" />
                                            </FXCollections>
                                          </items>
                                        </ComboBox>
                                        <ComboBox minHeight="25.0" prefWidth="105.0"
GridPane.columnIndex="1" GridPane.rowIndex="1">
                                          <items>
                                            <FXCollections fx:factory="observableArrayList">
                                              <String fx:value="Item 1" />
                                              <String fx:value="Item 2" />
                                              <String fx:value="Item 3" />
                                            </FXCollections>
                                          </items>
                                        </ComboBox>
                                      </children>
                                      <columnConstraints>
                                        <ColumnConstraints hgrow="SOMETIMES" minWidth="75.0"
prefWidth="-1.0" />
                                        <ColumnConstraints hgrow="SOMETIMES" minWidth="-Infinity"
prefWidth="100.0" />
                                      </columnConstraints>
                                      <rowConstraints>
                                        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                      </rowConstraints>
                                    </GridPane>
                                  </children>
                                  <padding>
                                    <Insets bottom="5.0" left="5.0" />
                                  </padding>
                                </VBox>
                                <VBox id="custom-vbox-black" padding="$x1" prefHeight="169.0"
prefWidth="226.0" VBox.vgrow="ALWAYS">
                                  <children>
                                    <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
text="Total:" visible="false" />
                                    <GridPane prefHeight="150.0" prefWidth="220.0" visible="false">
                                      <children>
                                        <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
text="Bar:"  visible="false" GridPane.columnIndex="0" GridPane.rowIndex="0" />
                                        <Text fill="$x11" scaleX="1.0" scaleY="1.0"
```

```
strokeType="OUTSIDE"  visible="false" strokeWidth="0.0" text="Kredit:" GridPane.columnIndex="0"
GridPane.rowIndex="1" />
                                        <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
visible="false" text="Gutschein:" GridPane.columnIndex="0" GridPane.rowIndex="2" />
                                        <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
visible="false" text="Ges.:" GridPane.columnIndex="0" GridPane.rowIndex="3" />
                                        <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
visible="false" text="Offene Tische:" GridPane.columnIndex="0" GridPane.rowIndex="4" />
                                        <TextField fx:id="iBarTotal" minWidth="90.0" prefWidth="90.0"
GridPane.columnIndex="1" GridPane.rowIndex="0" />
                                        <TextField fx:id="iCreditTotal"  visible="false"
minWidth="90.0" prefWidth="90.0" GridPane.columnIndex="1" GridPane.rowIndex="1" />
                                        <TextField fx:id="iVoucherTotal"  visible="false"
minWidth="90.0" prefWidth="90.0" GridPane.columnIndex="1" GridPane.rowIndex="2" />
                                        <TextField minWidth="90.0"  visible="false" prefWidth="90.0"
GridPane.columnIndex="1" GridPane.rowIndex="4" />
                                        <TextField minWidth="90.0"  visible="false" prefWidth="90.0"
GridPane.columnIndex="1" GridPane.rowIndex="3" />
                                      </children>
                                      <columnConstraints>
                                        <ColumnConstraints hgrow="SOMETIMES" minWidth="-1.0"
prefWidth="100.0" />
                                        <ColumnConstraints hgrow="SOMETIMES" minWidth="-1.0"
prefWidth="-1.0" />
                                      </columnConstraints>
                                      <rowConstraints>
                                        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                        <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                      </rowConstraints>
                                    </GridPane>
                                  </children>
                                </VBox>
                              </children>
                              <HBox.margin>
                                <Insets />
                              </HBox.margin>
                            </VBox>
                            <VBox id="custom-vbox-black" alignment="TOP_CENTER" fillWidth="true"
maxWidth="1.7976931348623157E308" padding="$x1" spacing="5.0" HBox.hgrow="ALWAYS">
                              <children>
                                <HBox id="HBox" alignment="TOP_LEFT"
maxWidth="1.7976931348623157E308" spacing="5.0" VBox.vgrow="ALWAYS">
                                  <children>
                                    <Text fill="$x3" strokeType="OUTSIDE" strokeWidth="0.0"
text="Showing invoices for given Date(s):" HBox.hgrow="ALWAYS" />
                                  </children>
                                </HBox>
                                <HBox maxWidth="1.7976931348623157E308" VBox.vgrow="ALWAYS">
                                  <children>
                                    <GridPane hgap="5.0" HBox.hgrow="ALWAYS">
                                      <children>
                                        <HBox id="custom-hbox-red" alignment="CENTER_LEFT"
minWidth="79.0" padding="$x1" prefHeight="-1.0" prefWidth="-1.0" spacing="0.0"
GridPane.columnIndex="0" GridPane.rowIndex="0">
                                          <children>
                                            <CheckBox fx:id="iStornoCheckBox" mnemonicParsing="false"
text="Show also Cancelled invoices" />
                                          </children>
                                        </HBox>
                                        <HBox id="custom-hbox-darkGray"  visible="false"
```

```
alignment="CENTER_LEFT" minWidth="79.0" prefHeight="-1.0" prefWidth="-1.0" GridPane.columnIndex="1"
GridPane.rowIndex="0">
                                        <children>
                                          <CheckBox fx:id="iBarCheckBox" mnemonicParsing="false"
visible="false" text="Bar" />
                                        </children>
                                        <padding>
                                          <Insets left="5.0" />
                                        </padding>
                                      </HBox>
                                      <HBox  visible="false" id="custom-hbox-skyBlue"
alignment="CENTER_LEFT" minWidth="79.0" prefHeight="-1.0" prefWidth="-1.0" GridPane.columnIndex="4"
GridPane.rowIndex="0">
                                        <children>
                                          <CheckBox fx:id="iVoucherCheckBox"
mnemonicParsing="false"  text="Gutschein" />
                                        </children>
                                        <padding>
                                          <Insets left="5.0" />
                                        </padding>
                                      </HBox>
                                      <HBox visible="false" id="custom-hbox-skyBlue"
alignment="CENTER_LEFT" minWidth="79.0" prefHeight="-1.0" prefWidth="-1.0" GridPane.columnIndex="2"
GridPane.rowIndex="0">
                                        <children>
                                          <CheckBox fx:id="iCreditCheckBox" mnemonicParsing="false"
onAction="slotDir=arg(arg()); call iCreditCheckBoxAction" text="Kredit" />
                                        </children>
                                        <padding>
                                          <Insets left="5.0" />
                                        </padding>
                                      </HBox>
                                    </children>
                                    <columnConstraints>
                                      <ColumnConstraints fillWidth="true" hgrow="SOMETIMES"
maxWidth="-1.0" minWidth="-1.0" prefWidth="-1.0" />
                                      <ColumnConstraints hgrow="SOMETIMES" minWidth="-1.0"
prefWidth="-1.0" />
                                      <ColumnConstraints hgrow="SOMETIMES" minWidth="-1.0"
prefWidth="-1.0" />
                                      <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0"
prefWidth="60.0" />
                                      <ColumnConstraints hgrow="SOMETIMES" minWidth="-1.0"
prefWidth="-1.0" />
                                    </columnConstraints>
                                    <rowConstraints>
                                      <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                    </rowConstraints>
                                  </GridPane>
                                </children>
                              </HBox>
                              <ScrollPane id="ScrollPane" fitToWidth="true" hbarPolicy="ALWAYS"
maxWidth="-1.0" prefViewportHeight="420.0" prefViewportWidth="400.0" vbarPolicy="ALWAYS">
                                <content>
                                  <TableView fx:id="iInvoicetableView"
maxWidth="1.7976931348623157E308"  prefHeight="480.0">
                                    <columns>
                                      <TableColumn fx:id="iTableBillColumn1" minWidth="-Infinity"
prefWidth="50.0" text="Id" />
                                      <TableColumn fx:id="iTableBillColumn2" minWidth="-Infinity"
prefWidth="175.0" text="Customer ID" />
<TableColumn fx:id="iTableBillColumn3" minWidth="-Infinity" prefWidth="150.0" text="Total Amount" />
                                      <TableColumn fx:id="iTableBillColumn4" minWidth="-Infinity"
prefWidth="120.0" text="Order Date" />
                                      <TableColumn fx:id="iTableBillColumn5" minWidth="-Infinity"
prefWidth="100.0" text="Order Time" />
```

```
<TableColumn fx:id="iTableBillColumn6" minWidth="-Infinity" prefWidth="175.0" text="Invoice Comments"
/>
                                    </columns>
                                  </TableView>
                                </content>
                              </ScrollPane>
                              <GridPane hgap="5.0" vgap="0.0">
                                <children>
                                  <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
text="Total Orders:" GridPane.columnIndex="0" GridPane.rowIndex="0" />
                                  <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
text="Total Amount:" GridPane.columnIndex="2" GridPane.rowIndex="0" />
                                  <TextField fx:id="iInvoiceCountText" minWidth="85.0"
GridPane.columnIndex="1" GridPane.rowIndex="0" />
                                  <TextField fx:id="iInvoiceTotal" minWidth="90.0" prefWidth="90.0"
GridPane.columnIndex="3" GridPane.rowIndex="0" />
                                </children>
                                <columnConstraints>
                                  <ColumnConstraints hgrow="SOMETIMES" minWidth="65.0"
prefWidth="65.0" />
                                  <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0"
prefWidth="100.0" />
                                  <ColumnConstraints hgrow="SOMETIMES" minWidth="65.0"
prefWidth="65.0" />
                                  <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0"
prefWidth="90.0" />
                                </columnConstraints>
                                <rowConstraints>
                                  <RowConstraints minHeight="10.0" prefHeight="30.0"
vgrow="SOMETIMES" />
                                </rowConstraints>
                              </GridPane>
                            </children>
                          </VBox>
                          <VBox id="custom-vbox-black" maxWidth="-Infinity" padding="$x1"
prefHeight="529.0" prefWidth="321.0" spacing="5.0" HBox.hgrow="ALWAYS">
                            <children>
                              <Text fill="$x11" strokeType="OUTSIDE" strokeWidth="0.0"
text="Details:" />
                              <HBox prefHeight="494.0" prefWidth="289.0">
                                                  <children>
                                                     <TextArea id="custom-hbox-white"
fx:id="iBillTextArea" prefHeight="468.0" prefWidth="277.0" text="Invoice" />
                                                  </children></HBox>
                            </children>
                          </VBox>
                        </children>
                      </HBox>
                      <HBox alignment="TOP_CENTER" maxHeight="1.7976931348623157E308" prefHeight="-
1.0" prefWidth="-1.0" spacing="5.0" VBox.vgrow="ALWAYS">
                        <children>
                          <Button id="dark-blue2" maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" minHeight="20.0" minWidth="135.0" mnemonicParsing="false"
fx:id="iCancelOrderButtonAction" text="Cancel order" HBox.hgrow="ALWAYS" />
                          <Button id="dark-blue2" maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" minHeight="20.0" minWidth="135.0" mnemonicParsing="false"
fx:id="iPrintOrderButtonAction" text="Print Invoice" HBox.hgrow="ALWAYS" />
                          <Button id="dark-blue2" maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" minHeight="20.0" minWidth="135.0" mnemonicParsing="false"
fx:id="iRefreshButtonAction" text="Refresh" HBox.hgrow="ALWAYS" />
                          <Button id="custom-button-imageButton"  fx:id="ibisCheckBox"
maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="20.0"
minWidth="135.0" mnemonicParsing="false" visible="false" text="Button" HBox.hgrow="ALWAYS" />
                          <Button id="custom-button-imageButton" maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" minHeight="20.0" minWidth="135.0" mnemonicParsing="false"
visible="false" text="Button" HBox.hgrow="ALWAYS" />
                          <Button id="custom-button-imageButton" maxHeight="1.7976931348623157E308"
```

```
maxWidth="1.7976931348623157E308" minHeight="20.0" minWidth="135.0" mnemonicParsing="false"
visible="false" text="Button" HBox.hgrow="ALWAYS" />
                          </children>
                        </HBox>
                      </children>
                      <padding>
                        <Insets left="5.0" right="5.0" top="1.0" fx:id="x1" />
                      </padding>
                    </VBox>
                  </children>
                </AnchorPane>
              </children>
            </VBox>
          </children>
        </VBox>
      </children>
    </AnchorPane>
  </children>
  <stylesheets>
    <URL value="@button.css" />
  </stylesheets>
    <fx:script source="put_FXID_objects_into.my.app.rex" />
</VBox>
```

Source code of the file "**Login.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.text.*?>
<?import javafx.scene.paint.*?>
<?import javafx.scene.image.*?>
<?import javafx.geometry.*?>
<?import java.lang.*?>
<?import java.net.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?language rexx?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="400.0" prefWidth="600.0" style="-fx-background-color: #DDDDDD;"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">



<children>
      <AnchorPane prefHeight="106.0" prefWidth="600.0" style="-fx-background-color: #4B4B4B;" />
      <AnchorPane layoutX="145.0" layoutY="49.0" prefHeight="288.0" prefWidth="312.0" style="-fx-
background-color: white; -fx-background-radius: 12px;">
         <children>
            <ImageView fitHeight="72.0" fitWidth="76.0" layoutX="120.0" layoutY="14.0"
pickOnBounds="true" preserveRatio="true">
               <image>
                  <Image url="@User.png" />
               </image>
            </ImageView>
            <TextField fx:id="iUsername" layoutX="61.0" layoutY="116.0" prefHeight="25.0"
prefWidth="191.0" />
            <PasswordField fx:id="iPassword" layoutX="61.0" layoutY="172.0" prefHeight="25.0"
prefWidth="191.0" />
            <Label layoutX="61.0" layoutY="99.0" text="Username:" />
            <Label layoutX="61.0" layoutY="155.0" text="Password:" />
            <Button layoutX="112.0" layoutY="219.0" mnemonicParsing="false"  prefHeight="32.0"
prefWidth="89.0" style="-fx-background-color: #3196DE;" text="Login" textAlignment="CENTER"
fx:id="loginbutton" textFill="WHITE">
               <font>
                  <Font name="System Bold" size="12.0" />
               </font>
            </Button>
         </children>
      </AnchorPane>
   </children>
  <stylesheets>
    <URL value="@button.css" />
  </stylesheets>
    <fx:script source="put_FXID_objects_into.my.app.rex" />
</AnchorPane>
```

Source code of the file "**kunden.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.net.URL?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.control.ScrollPane?>
<?import javafx.scene.control.SeparatorMenuItem?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?language rexx?>

<VBox id="" alignment="TOP_CENTER" prefHeight="-1.0" prefWidth="-1.0"
xmlns="http://javafx.com/javafx/8.0.141" xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <AnchorPane id="AnchorPane" VBox.vgrow="ALWAYS">



      <children>
        <VBox id="VBox" alignment="CENTER" spacing="5.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
          <children>

            <AnchorPane id="dark-blue2" minHeight="-1.0" minWidth="-1.0" VBox.vgrow="ALWAYS">
              <children>
                <VBox id="VBox" alignment="CENTER" spacing="5.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
                  <children>
                    <HBox alignment="TOP_CENTER" spacing="2.0">
                      <children>
                                    <HBox alignment="TOP_CENTER" prefHeight="-1.0" prefWidth="-1.0"
spacing="2.0">
                                       <children>
                                         <Button id="custom-button-imageButton"  visible="false"
minHeight="40.0" minWidth="70.0" mnemonicParsing="false" fx:id="showDashboardbtn" text="back to
Dashboard">
                            <graphic>
                               <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                  <image>
                                     <Image url="@dashboard.png" />
                                  </image>
                               </ImageView>
                            </graphic>
                         </Button>
                <Button id="custom-button-imageButton" fx:id="showKundenbtn"  visible="false"
minHeight="40.0" minWidth="70.0" mnemonicParsing="false"  text="Customer page">
                            <graphic>
                               <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                  <image>
                                     <Image url="@customer.png" />
                                  </image>
                               </ImageView>
                            </graphic></Button>
                <Button id="custom-button-imageButton" minHeight="40.0"  visible="false"
```

```
minWidth="70.0" mnemonicParsing="false"  fx:id="showItemsbtn" text="Item Page">
                           <graphic>
                              <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                 <image>
                                    <Image url="@img3.png" />
                                 </image>
                              </ImageView>
                           </graphic></Button>
                <Button id="custom-button-imageButton" minHeight="40.0" fx:id="showInvoicesbtn"
visible="false" minWidth="70.0" mnemonicParsing="false"  text="Invoices">
                           <graphic>
                              <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                 <image>
                                    <Image url="@img2.png" />
                                 </image>
                              </ImageView>
                           </graphic></Button>
                                       </children>
                                    </HBox>
                     </children>
                  </HBox>
                  <HBox VBox.vgrow="ALWAYS">
                    <children>
                      <VBox alignment="TOP_CENTER" fillWidth="false" maxWidth="310.0" minWidth="-
1.0" prefHeight="-1.0" prefWidth="-1.0" spacing="5.0" HBox.hgrow="ALWAYS">
                         <children>
                          <HBox maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" />
                          <HBox maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" />
                          <HBox id="HBox" spacing="5.0" VBox.vgrow="ALWAYS" />
                          <VBox id="VBox" alignment="CENTER" spacing="5.0" VBox.vgrow="ALWAYS" />
                          <VBox>
                            <children>
                              <HBox spacing="0.0" />
                            </children>
                          </VBox>
                         </children>
                      </VBox>
                      <VBox spacing="0.0" HBox.hgrow="ALWAYS">
                        <children>
                                       <HBox layoutX="15.0" layoutY="10.0" prefHeight="27.0">
                                          <children>
                                             <Label prefHeight="25.0" prefWidth="490.0"
text="Customer Search: (fill something in textfield and press ENTER to search)">
                                                <padding>
                                                   <Insets left="5.0" />
                                                </padding>
                                             </Label>
                                          </children>
                                       </HBox>
                                    <HBox prefHeight="33.0" VBox.vgrow="ALWAYS">
                                          <children>
                                             <Label prefHeight="25.0" prefWidth="101.0"
text="Customer ID:">
                                                <padding>
                                                   <Insets left="5.0" />
                                                </padding>
                                             </Label>
                                    <TextField id="custom-textField-tf1" fx:id="cIdSearchText"
minHeight="25.0"  prefHeight="25.0" prefWidth="130.0" />
                                             <Label prefHeight="25.0" prefWidth="114.0"
text="Telephone:" textAlignment="CENTER">
                                                <padding>
                                                   <Insets left="30.0" />
                                                </padding>
```

```
                                        </Label>
                                        <TextField id="custom-textField-tf1"
fx:id="cMobileSearchText" layoutX="89.0" layoutY="10.0" minHeight="25.0"   prefHeight="25.0"
prefWidth="130.0" />
                                        <Label prefHeight="25.0" prefWidth="297.0"
text="Customer Name or Address:" textAlignment="RIGHT">
                                           <padding>
                                              <Insets left="30.0" />
                                           </padding>
                                        </Label>
                                        <TextField id="custom-textField-tf1"
fx:id="cNameSearchText" layoutX="89.0" layoutY="10.0" minHeight="25.0"    prefHeight="25.0"
prefWidth="199.0" />
                                     </children>
                                  </HBox>
                               <HBox id="HBox" fillHeight="false" maxHeight="-Infinity"
prefHeight="550.0" spacing="0.0" VBox.vgrow="ALWAYS">
                                     <children>
                                        <ScrollPane id="ScrollPane" prefHeight="600.0"
prefViewportHeight="360.0" prefViewportWidth="300.0" prefWidth="1100.0">
                                           <content>
                                              <TableView id="custom-table-table1"
fx:id="cCustomerTable" editable="true" maxHeight="-1.0" maxWidth="-1.0" prefHeight="600.0"
prefWidth="1070.0">
                                                 <columns>
                                                    <TableColumn
fx:id="cCustomerTableColumn1" editable="false" maxWidth="5000.0" minWidth="63.0" prefWidth="63.0"
text="No." />
                                                    <TableColumn
fx:id="cCustomerTableColumn2" editable="true" maxWidth="5000.0" minWidth="250.0" prefWidth="270.0"
text="Name" />
                                                    <TableColumn
fx:id="cCustomerTableColumn3" editable="true" minWidth="357.0" prefWidth="374.0" text="Address" />
                                                    <TableColumn
fx:id="cCustomerTableColumn5" prefWidth="141.0" text="Postcode" />
                                                    <TableColumn
fx:id="cCustomerTableColumn4" editable="true" minWidth="205.0" prefWidth="205.0" text="Telephone" />
                                                 </columns>
 <columnResizePolicy>
                                                    <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
                                                 </columnResizePolicy>
                                              </TableView>
                                           </content>
                                           <HBox.margin>
                                              <Insets bottom="15.0" left="15.0" right="15.0"
top="15.0" />
                                           </HBox.margin>
                                           <padding>
                                              <Insets bottom="15.0" left="15.0" right="15.0"
top="15.0" />
                                           </padding>
                                        </ScrollPane>
                                     </children>
                                     <padding>
                                        <Insets bottom="5.0" top="5.0" />
                                     </padding>
                                  </HBox>
                                  <HBox prefHeight="33.0">
                                     <children>
                                        <Label prefHeight="25.0" prefWidth="115.0"
text="Customer Name:">
                                           <padding>
                                              <Insets left="5.0" right="5.0" />
                                           </padding>
                                        </Label>
                                        <TextField id="custom-textField-tf1"
fx:id="cCustomerNameAdd" minHeight="25.0" prefHeight="25.0" prefWidth="130.0" promptText="Customer
Name">
```

```
                                                <HBox.margin>
                                                    <Insets left="5.0" right="5.0" />
                                                </HBox.margin>
                                            </TextField>
                                            <Label prefHeight="25.0" prefWidth="115.0"
text="Telephone:" textAlignment="CENTER">
                                                <padding>
                                                    <Insets left="25.0" right="5.0" />
                                                </padding>
                                            </Label>
                                            <TextField id="custom-textField-tf1"
fx:id="cCustomerMobileAdd" layoutX="89.0" layoutY="10.0" minHeight="25.0" prefHeight="25.0"
prefWidth="130.0" promptText="Telephone">
                                                <padding>
                                                    <Insets left="5.0" right="5.0" />
                                                </padding>
                                            </TextField>
                                            <Label prefHeight="25.0" prefWidth="115.0"
text="Address:" textAlignment="RIGHT">
                                                <padding>
                                                    <Insets left="30.0" right="5.0" />
                                                </padding>
                                            </Label>
                                            <TextField id="custom-textField-tf1"
fx:id="cCustomerAddressAdd" layoutX="89.0" layoutY="10.0" minHeight="25.0" prefHeight="25.0"
prefWidth="278.0" promptText="Address" />
                                            <Label prefHeight="25.0" prefWidth="70.0" text="
Postcode:" />
                                            <TextField id="custom-textField-tf1"
fx:id="cCustomerPincodeAdd" prefHeight="34.0" prefWidth="150.0" promptText="Postcode" />
                                        </children>
                                        <padding>
                                            <Insets bottom="15.0" top="5.0" />
                                        </padding>
                                    </HBox>
                                    <HBox alignment="TOP_CENTER" prefHeight="46.0"
prefWidth="1000.0" spacing="5.0">
                                        <children>
                                            <Button id="custom-button-imageButton"
mnemonicParsing="false"  fx:id="cCustomerAdd"  prefHeight="30.0" prefWidth="210.0" text="Add new
customer" HBox.hgrow="ALWAYS" />
                                            <Button id="custom-button-imageButton"
mnemonicParsing="false"   fx:id="cCustomerDelete" prefHeight="30.0" prefWidth="210.0" text="Delete
customer" HBox.hgrow="ALWAYS" />
                                            <Button id="custom-button-imageButton"
mnemonicParsing="false" prefHeight="30.0" prefWidth="210.0" text="Recent orders"
fx:id="crecentOrders" HBox.hgrow="ALWAYS"  visible="false" />
                                            <Button id="custom-button-imageButton"
mnemonicParsing="false"  fx:id="cCustomerNewOrder"  prefHeight="30.0" prefWidth="210.0" text="create
New order" HBox.hgrow="ALWAYS" />
                                        </children>
                                        <padding>
                                            <Insets bottom="20.0" right="5.0" />
                                        </padding>
                                    </HBox>
                                </children>
                                <padding>
                                  <Insets left="5.0" />
                                </padding>
                            </VBox>
                        </children>
                    </HBox>
                </children>
            </VBox>
        </children>
    </AnchorPane>
  </children>
</VBox>
```

```
      </children>
    </AnchorPane>
  </children>
  <stylesheets>
    <URL value="@button.css" />
  </stylesheets>
    <fx:script source="put_FXID_objects_into.my.app.rex" />
</VBox>
```

Source code of the file "**item.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import java.lang.*?>
<?import java.net.*?>
<?import javafx.geometry.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.image.*?>
<?import javafx.scene.layout.*?>
<?import java.net.URL?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.control.ScrollPane?>
<?import javafx.scene.control.SeparatorMenuItem?>
<?import javafx.scene.control.Tab?>
<?import javafx.scene.control.TabPane?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?language rexx?>

<VBox id="" alignment="TOP_CENTER" prefHeight="-1.0" prefWidth="-1.0"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <AnchorPane id="AnchorPane" VBox.vgrow="ALWAYS">

<children>
        <VBox id="VBox" alignment="CENTER" spacing="5.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
          <children>

            <AnchorPane id="dark-blue2" minHeight="-1.0" minWidth="-1.0" VBox.vgrow="ALWAYS">
              <children>
                <VBox id="VBox" alignment="CENTER" spacing="5.0" AnchorPane.bottomAnchor="0.0"
AnchorPane.leftAnchor="0.0" AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0">
                  <children>
                    <HBox alignment="TOP_CENTER" spacing="2.0">
                      <children>
                                    <HBox alignment="TOP_CENTER" prefHeight="-1.0" prefWidth="-1.0"
spacing="2.0">
                                       <children>
                                          <Button id="custom-button-imageButton" minHeight="40.0"
minWidth="70.0" mnemonicParsing="false" fx:id="showDashboardbtn"  visible="false" text="back to
Dashboard">
                           <graphic>
                              <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                 <image>
                                    <Image url="@dashboard.png" />
                                 </image>
                              </ImageView>
                           </graphic>
                        </Button>
                <Button id="custom-button-imageButton" fx:id="showKundenbtn" visible="false"
minHeight="40.0" minWidth="70.0" mnemonicParsing="false"  text="Customer page">
                           <graphic>
                              <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
```

```
                                <image>
                                    <Image url="@customer.png" />
                                </image>
                            </ImageView>
                        </graphic></Button>
                <Button id="custom-button-imageButton" minHeight="40.0" minWidth="70.0"
mnemonicParsing="false"  visible="false" fx:id="showItemsbtn" text="Item Page">
                        <graphic>
                            <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                <image>
                                    <Image url="@img3.png" />
                                </image>
                            </ImageView>
                        </graphic></Button>
                <Button id="custom-button-imageButton" minHeight="40.0" fx:id="showInvoicesbtn"
visible="false" minWidth="70.0" mnemonicParsing="false"  text="Invoices">
                        <graphic>
                            <ImageView fitHeight="50.0" fitWidth="50.0" pickOnBounds="true"
preserveRatio="true">
                                <image>
                                    <Image url="@img2.png" />
                                </image>
                            </ImageView>
                        </graphic></Button>
                                    </children>
                                </HBox>
                    </children>
                </HBox>
                <HBox VBox.vgrow="ALWAYS">
                    <children>
                      <VBox alignment="TOP_CENTER" fillWidth="false" maxWidth="310.0" minWidth="-
1.0" prefHeight="-1.0" prefWidth="-1.0" spacing="5.0" HBox.hgrow="ALWAYS">
                        <children>
                          <HBox maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" />
                          <HBox maxHeight="1.7976931348623157E308"
maxWidth="1.7976931348623157E308" />
                          <HBox id="HBox" spacing="5.0" VBox.vgrow="ALWAYS" />
                          <VBox id="VBox" alignment="CENTER" spacing="5.0" VBox.vgrow="ALWAYS" />
                          <VBox>
                            <children>
                              <HBox spacing="0.0" />
                            </children>
                          </VBox>
                        </children>
                      </VBox>
                      <VBox spacing="0.0" HBox.hgrow="ALWAYS">
                        <children>
                                        <HBox layoutX="15.0" layoutY="10.0" prefHeight="27.0">
                                            <children>
                                                <Label prefHeight="25.0" prefWidth="490.0" text="Item
Search: (fill something in textfield and press ENTER to search)">
                                                    <padding>
                                                        <Insets left="5.0" />
                                                    </padding>
                                                </Label>
                                            </children>
                                        </HBox>
                                    <HBox prefHeight="33.0" VBox.vgrow="ALWAYS">
                                            <children>
                                                <Label prefHeight="25.0" prefWidth="101.0" text="Item
ID:">
                                                    <padding>
                                                        <Insets left="5.0" />
                                                    </padding>
                                                </Label>
                                    <TextField id="custom-textField-tf1" fx:id="sIdSearchText"
```

```
minHeight="25.0"  prefHeight="25.0" prefWidth="130.0" />
                                                <Label prefHeight="25.0" prefWidth="164.0" text="Item
Category:" textAlignment="CENTER">
                                                   <padding>
                                                      <Insets left="30.0" />
                                                   </padding>
                                                </Label>
                                                <TextField id="custom-textField-tf1"
fx:id="sItemTypeText" layoutX="89.0" layoutY="10.0" minHeight="25.0" prefHeight="25.0"
prefWidth="130.0" />
                                                <Label prefHeight="25.0" prefWidth="157.0" text="Item
Name:" textAlignment="RIGHT">
                                                   <padding>
                                                      <Insets left="30.0" />
                                                   </padding>
                                                </Label>
                                                <TextField id="custom-textField-tf1"
fx:id="sItemSubTypeText" layoutX="89.0" layoutY="10.0" minHeight="25.0"  prefHeight="25.0"
prefWidth="199.0" />
                                             </children>
                                          </HBox>
                                    <HBox id="HBox" fillHeight="false" maxHeight="-Infinity"
prefHeight="550.0" spacing="0.0" VBox.vgrow="ALWAYS">
                                             <children>
                                                <ScrollPane id="ScrollPane" prefHeight="600.0"
prefViewportHeight="360.0" prefViewportWidth="300.0" prefWidth="1100.0">
                                                   <content>
                                                      <TableView id="custom-table-table1"
fx:id="sItemTable" editable="true" maxHeight="-1.0" maxWidth="-1.0" prefHeight="600.0"
prefWidth="1070.0">
                                                         <columns>
                                                            <TableColumn fx:id="sItemTableColumn1"
editable="false" maxWidth="5000.0" minWidth="100.0" prefWidth="76.0" text="Id" />
                                                            <TableColumn fx:id="sItemTableColumn2"
editable="true" maxWidth="5000.0" minWidth="150.0" prefWidth="200.0" text="Item Category" />
                                                            <TableColumn fx:id="sItemTableColumn3"
editable="true" minWidth="300.0" prefWidth="400.0" text="Item Name" />
                                                            <TableColumn fx:id="sItemTableColumn4"
editable="true" minWidth="100.0" prefWidth="200.0" text="Tax [%]" />
                                                            <TableColumn fx:id="sItemTableColumn5"
editable="true" minWidth="100.0" prefWidth="200.0" text="Price" />
                                                         </columns>
                                                      </TableView>
                                                   </content>
                                                   <HBox.margin>
                                                      <Insets bottom="15.0" left="15.0" right="15.0"
top="15.0" />
                                                   </HBox.margin>
                                                   <padding>
                                                      <Insets bottom="15.0" left="15.0" right="15.0"
top="15.0" />
                                                   </padding>
                                                </ScrollPane>
                                             </children>
                                             <padding>
                                                <Insets bottom="5.0" top="5.0" />
                                             </padding>
                                    </HBox>
                                          <HBox prefHeight="33.0">
                                             <children>
                                                <Label prefHeight="25.0" prefWidth="115.0" text="Item
Category:">
                                                   <padding>
                                                      <Insets left="5.0" right="5.0" />
                                                   </padding>
                                                </Label>
                                                <TextField id="custom-textField-tf1"
fx:id="sItemTypeAdd" minHeight="25.0" prefHeight="25.0" prefWidth="130.0" promptText="Item Type">
```

```
                                                <HBox.margin>
                                                    <Insets left="5.0" right="5.0" />
                                                </HBox.margin>
                                            </TextField>
                                            <Label prefHeight="25.0" prefWidth="115.0" text="Item
Name:" textAlignment="CENTER">
                                                <padding>
                                                    <Insets left="25.0" right="5.0" />
                                                </padding>
                                            </Label>
                                            <TextField id="custom-textField-tf1"
fx:id="sItemNameAdd" layoutX="89.0" layoutY="10.0" minHeight="25.0" prefHeight="25.0"
prefWidth="130.0" promptText="Item Name">
                                                <padding>
                                                    <Insets left="5.0" right="5.0" />
                                                </padding>
                                            </TextField>
                                            <Label prefHeight="25.0" prefWidth="115.0" text="Tax
(%) :" textAlignment="RIGHT">
                                                <padding>
                                                    <Insets left="30.0" right="5.0" />
                                                </padding>
                                            </Label>
                                            <TextField id="custom-textField-tf1"
fx:id="sItemDiscountAdd" layoutX="89.0" layoutY="10.0" minHeight="25.0" prefHeight="25.0"
prefWidth="108.0" promptText="20" />
                                            <Label prefHeight="25.0" prefWidth="115.0"
text="Price" textAlignment="RIGHT">
                                                <padding>
                                                    <Insets left="30.0" right="5.0" />
                                                </padding>
                                            </Label>
                                            <TextField id="custom-textField-tf1"
fx:id="sItemAmountAdd" minHeight="25.0" prefHeight="25.0" prefWidth="108.0" promptText="Price" />
                                        </children>
                                        <padding>
                                            <Insets bottom="15.0" top="5.0" />
                                        </padding>
                                    </HBox>
                                    <HBox alignment="TOP_CENTER" prefHeight="46.0"
prefWidth="1000.0" spacing="5.0">
                                        <children>
                                            <Button id="custom-button-imageButton"
mnemonicParsing="false" prefHeight="30.0" prefWidth="210.0" fx:id="sItemAddAction" text="Add new
item" HBox.hgrow="ALWAYS" />
                                            <Button id="custom-button-imageButton"
mnemonicParsing="false" prefHeight="30.0" prefWidth="210.0" fx:id="sItemDeleteAction" text="Delete
item" HBox.hgrow="ALWAYS" />
                                        </children>
                                        <padding>
                                            <Insets bottom="20.0" right="5.0" />
                                        </padding>
                                    </HBox>
                                </children>
                                <padding>
                                  <Insets left="5.0" />
                                </padding>
                            </VBox>
                        </children>
                    </HBox>
                </children>
            </VBox>
        </children>
    </AnchorPane>
</children>
</VBox>
</children>
</AnchorPane>
```

```
  </children>
  <stylesheets>
    <URL value="@button.css" />
  </stylesheets>
    <fx:script source="put_FXID_objects_into.my.app.rex" />
</VBox>
```

Source code of the file "**Invoice_printout.html**"

```
<!DOCTYPE html>
<html>
<head>
  <title>Invoice Printing</title>
</head>
<body>
<div align="center"><b>Hello Rexx Company</b><br/>Rexx Street<br/>00431111111<br/><br/><div
align="left">                                  Date: 2018-01-
14T14:41:44.487362<br/></div><br/>------------------------------------------------------------<br/>
    <span>pasta pronto</span>,
    <span>EUR 8.0.-</span>  <br/>

    <span>pasta spinaci</span>,
    <span>EUR 9.0.-</span>  <br/>

------------------------------------------------------------<br/>
<span>TOTAL SUM: EUR 17.0</span>.-<br/>
------------------------------------------------------------<br/>
<br/><br/><img src="qrtest.png"/> <br/><br/>Thank you for your purchase!</div>
</body>
</html>
```

Source code of the file "**put_FXID_objects_into.my.app.rex**"

```
/*    author:     Rony G. Flatscher
      date:       2016-11-26
      purpose:    generic ooRexx program which stores the GLOBAL_SCOPE entries in .local as a
directory
                  with the name of the FXML-file it got invoked from;

      explanation: this program is invoked by the JavaFX FXMLLoader when loading a FXML document and
                   instantiating the JavaFX controls using the RexxScript support, which means that
                   a proper Rexx interpreter instance gets created in which this program executes;
                   to share data with other Rexx interpreter instances we use .environment (shared
                   among all Rexx interpreter instances) rather than .local (unique per Rexx
interpreter
                   instance);

                   FXMLLoader will create a new RexxScriptEngine for each FXML document it processes!

                  FXMLLoader will put all JavaFX objects with a fx:id attribute into the
ScriptContext's
                   Bindings for the global scope, such that we can fetch these objects from there and
                  save it in a directory named after the FXML location (file name) that defines them
for
                   later retrieval by other Rexx programs

      changed:    2017-02-09, rgf: remove usage of .jsr223 as it is not needed anymore
*/

parse source . . thisProg
thisProg=filespec("Name", thisProg)

     -- make sure we have a directory with the environment symbol ".my.app" available to everyone
which
     -- is to receive information about those JavaFX objects that have a unique "fx:id" value;
if \.environment~hasEntry("my.app") then
   .environment~setEntry("my.app", .directory~new)

bDebug=(.my.app~bDebug=.true)  -- set debug mode

if bDebug then say .dateTime~new " ==> ---> arrived in Rexx program '"thisProg"' ..."

-- use arg slotDir  -- fetch the slotDir argument (BSF4ooRexx adds this as the last argument at the
Java side)
slotDir=arg(arg())  -- fetch the slotDir argument (BSF4ooRexx adds this as the last argument at the
Java side)
scriptContext=slotDir~scriptContext   -- get the slotDir (the last) argument, get the entry
"SCRIPTCONTEXT"

if bDebug then
do
   tab="09"x
     -- make sure we have access to 'rgf_util2.rex' (supplied via BSF4ooRexx)
   .context~package~addPackage(.package~new('rgf_util2.rex'))
   call dump2 slotDir, "slotDir"
   if slotDir~hasEntry("REXXCOMPILEDSCRIPT") then
      rse=slotDir~RexxCompiledScript~getRexxScriptEngine
   else
      rse=slotDir~RexxScriptEngine

   if slotDir~hasEntry("RexxCompiledScript") then
       say tab "---> fetched RexxScriptEngine via slotDir's 'RexxCompiledScript' entry," -
" then '~getRexxScriptEngine'"
   else
       say tab "---> fetched RexxScriptEngine via slotDir's 'RexxScriptEngine' entry"

   say tab "---> RexxEngine:" pp(rse) "--->" "BSFEngine:" pp(rse~getBSFRexxEngine) -
"---> rii_id:" pp(rse~getBSFRexxEngine~get_rii_id) "<---"
```

```
   if slotDir~hasEntry("RexxCompiledScript") then
       rse=rse~toString "(via RexxCompiledScript entry)"
end


  -- get and save all entries in GLOBAL_SCOPE in .my.app, entry name is the FXML file name
  -- (FXMLLoader places all JavaFX objects with a fx:id into the GLOBAL_SCOPE Bindings)
engine_scope=100
global_scope=200
url= scriptContext~getAttribute("location", global_scope)
if bDebug then say "location:" pp(url~toString) "getFile:" pp(filespec("name",url~getFile))
-- make sure we get the file-name only
fxmlFileName=filespec("name",url~getFile)      -- get filename, make sure we get unqualified file-
name
dir2obj =.directory~new
-- all GLOBAL_SCOPE entries, mapping uppercase attribute names to Java-values
.my.app~setEntry(fxmlFileName, dir2obj)

bindings=scriptContext~getBindings(global_scope)   -- get the Bindings for the global scope
keys=bindings~keySet~makearray -- get the kay values as a Rexx array
do key over keys
   val=bindings~get(key)       -- fetch the key's value
   dir2obj ~setEntry(key,val)  -- save it in our directory
end

if bDebug then
do
      /* show the currently defined attributes in all ScriptContext's scopes   */
   say "getting all attributes from all ScriptContext's scopes..."
   dir=.directory~new
   dir[100]="ENGINE_SCOPE"
   dir[200]="GLOBAL_SCOPE"
   arr=.array~of(100,200)
   do sc over arr
       say "ScriptContext scope:" pp(sc) "("dir~entry(sc)"), available attributes:"
       bin=scriptContext~getBindings(sc)
       if bin=.nil then iterate -- inexistent scope
       keys=bin~keySet          -- get kay values
       it=keys~makearray        -- get the keys as a Rexx array
       do key over it~sortWith(.CaselessComparator~new) -- sort keys (attributes) caselessly
          val=bin~get(key)      -- fetch the key's value
          str=""
          if val~isA(.bsf) then str="~toString:" pp(val~toString)
          say "  " pp(key)~left(35,".") pp(val) str
       end
       say "-"~copies(79)
   end
end

if bDebug then
do
   say .dateTime~new " <== <--- returning from program '"thisProg"'."
   say
end


/*
      ------------------------ Apache Version 2.0 license -------------------------
         Copyright 2016-2017 Rony G. Flatscher

         Licensed under the Apache License, Version 2.0 (the "License");
         you may not use this file except in compliance with the License.
         You may obtain a copy of the License at

             http://www.apache.org/licenses/LICENSE-2.0

         Unless required by applicable law or agreed to in writing, software
         distributed under the License is distributed on an "AS IS" BASIS,
```

```
         WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
         See the License for the specific language governing permissions and
         limitations under the License.
      ------------------------------------------------------------------------
*/
```

Source code of the file "**Settings.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.layout.*?>
<?import javafx.scene.text.*?>
<?import javafx.scene.paint.*?>
<?import javafx.scene.image.*?>
<?import javafx.geometry.*?>
<?import java.lang.*?>
<?import java.net.*?>
<?import java.util.*?>
<?import javafx.scene.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?language rexx?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="400.0" prefWidth="600.0" style="-fx-background-color: rgb(50, 50, 50);"
xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1">


   <children>
      <TabPane layoutX="-1.0" layoutY="42.0" prefHeight="357.0" prefWidth="600.0"
tabClosingPolicy="UNAVAILABLE">
        <tabs>
          <Tab text="Print Settings">
            <content>
              <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="328.0" prefWidth="572.0"
style="-fx-background-color: rgb(245,245,245);">
                     <children>
                        <AnchorPane layoutX="51.0" layoutY="30.0" prefHeight="234.0"
prefWidth="498.0" style="-fx-border-color: BLACK; -fx-border-radius: 10px;;">
                           <children>
                              <GridPane hgap="4.0" layoutX="23.0" layoutY="29.0" vgap="5.0">
                                 <children>
                                    <Label layoutX="27.0" layoutY="50.0" text="Company Name:" />
                                    <Label layoutX="27.0" layoutY="108.0" text="Phone No.:"
GridPane.rowIndex="2" />
                                    <Label layoutX="27.0" layoutY="80.0" text="Address:"
GridPane.rowIndex="1" />
                                    <TextField fx:id="iAddress" layoutX="128.0" layoutY="75.0"
prefHeight="25.0" prefWidth="355.0" GridPane.columnIndex="1" GridPane.rowIndex="1" />
                                    <TextField fx:id="iPhoneNo" layoutX="128.0" layoutY="104.0"
prefHeight="25.0" prefWidth="355.0" GridPane.columnIndex="1" GridPane.rowIndex="2" />
                                    <TextField fx:id="iResturantName" layoutX="128.0" layoutY="46.0"
prefHeight="25.0" prefWidth="355.0" GridPane.columnIndex="1" />
                                    <Label text="CashBox Id:" GridPane.rowIndex="3" />
                                    <Label text="Company Id:" GridPane.rowIndex="4" />
                                    <TextField fx:id="iCashboxId" prefHeight="25.0" prefWidth="355.0"
GridPane.columnIndex="1" GridPane.rowIndex="3" />
                                    <TextField fx:id="iCompanyId" prefHeight="25.0" prefWidth="355.0"
GridPane.columnIndex="1" GridPane.rowIndex="4" />
                                    <Label text="Base64AesKey:" GridPane.rowIndex="5" />
                                    <TextField fx:id="iBase64AesKey" prefHeight="25.0"
prefWidth="355.0" GridPane.columnIndex="1" GridPane.rowIndex="5" />
                                 </children>
                                 <columnConstraints>
                                    <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" />
                                    <ColumnConstraints hgrow="NEVER" minWidth="10.0" />
                                 </columnConstraints>
                                 <rowConstraints>
                                    <RowConstraints minHeight="10.0" vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="10.0" vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="10.0" vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="10.0" vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="10.0" vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="10.0" vgrow="SOMETIMES" />
```

```
                                 </rowConstraints>
                              </GridPane>
                           </children>
                        </AnchorPane>
                        <Button fx:id="btnPrintSettings" layoutX="253.0" layoutY="280.0"
mnemonicParsing="false"   prefHeight="25.0" prefWidth="119.0" style="-fx-background-color: rgb(50,
50, 50);" text="Save" textFill="WHITE" />
                     </children>
                  </AnchorPane>
               </content>
            </Tab>
              <Tab text="User Settings">
                 <content>
                    <AnchorPane minHeight="0.0" minWidth="0.0" prefHeight="328.0" prefWidth="572.0"
style="-fx-background-color: rgb(245,245,245);">
                     <children>
                        <AnchorPane layoutX="51.0" layoutY="59.0" prefHeight="129.0"
prefWidth="498.0" style="-fx-border-color: BLACK; -fx-border-radius: 10px;;">
                           <children>
                              <GridPane hgap="4.0" layoutX="45.0" layoutY="35.0" vgap="5.0">
                                 <children>
                                    <Label layoutX="27.0" layoutY="50.0" text="Username:" />
                                    <Label layoutX="27.0" layoutY="80.0" text="Password:"
GridPane.rowIndex="1" />
                                    <TextField fx:id="iUsername" layoutX="128.0" layoutY="46.0"
prefHeight="25.0" prefWidth="355.0" GridPane.columnIndex="1" />
                                    <PasswordField fx:id="iPassword" GridPane.columnIndex="1"
GridPane.rowIndex="1" />
                                 </children>
                                 <columnConstraints>
                                    <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" />
                                    <ColumnConstraints hgrow="NEVER" minWidth="10.0" />
                                 </columnConstraints>
                                 <rowConstraints>
                                    <RowConstraints minHeight="10.0" vgrow="SOMETIMES" />
                                    <RowConstraints minHeight="10.0" vgrow="SOMETIMES" />
                                 </rowConstraints>
                              </GridPane>
                           </children>
                        </AnchorPane>
                        <Button layoutX="253.0" layoutY="280.0" mnemonicParsing="false"
fx:id="iSaveUserSettings"  prefHeight="25.0" prefWidth="119.0" style="-fx-background-color: rgb(50,
50, 50);" text="Save" textFill="WHITE" />
                     </children>
                  </AnchorPane>
               </content>
            </Tab>
        </tabs>
      </TabPane>
      <Label layoutX="14.0" layoutY="8.0" prefHeight="17.0" prefWidth="170.0" text="Settings"
textFill="WHITE">
         <font>
            <Font name="System Bold" size="18.0" />
         </font>
      </Label>
   </children>
    <fx:script source="put_FXID_objects_into.my.app.rex" />
</AnchorPane>
```

Source code of the file "**RootLayout.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.input.KeyCodeCombination?>
<?import javafx.scene.layout.BorderPane?>

<!-- comment: the following process instruction (PI) defines the Java script engine named 'rexx'
              to be used for the code in event attributes like 'onAction' -->

<?language rexx?>



<BorderPane fx:id="rootPane"  maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity"
minWidth="-Infinity" prefHeight="400.0" prefWidth="700.0"  xmlns="http://javafx.com/javafx/8.0.65"
xmlns:fx="http://javafx.com/fxml/1">
   <top>
      <MenuBar BorderPane.alignment="CENTER">
        <menus>
          <Menu  text="_File">
            <items>
                  <MenuItem fx:id="menuNew" visible="false" text="_New">
                     <accelerator>
                        <KeyCodeCombination alt="DOWN" code="N" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                     </accelerator>
                  </MenuItem>
                  <MenuItem fx:id="menuOpen" visible="false" text="_Open...">
                     <accelerator>
                        <KeyCodeCombination alt="DOWN" code="O" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                     </accelerator>
                  </MenuItem>
                  <MenuItem fx:id="menuPrint" visible="false" text="_Print">
                     <accelerator>
                        <KeyCodeCombination alt="DOWN" code="P" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                     </accelerator>
                  </MenuItem>
                  <MenuItem fx:id="menuSave" visible="false" text="_Save">
                     <accelerator>
                        <KeyCodeCombination alt="DOWN" code="S" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                     </accelerator>
                  </MenuItem>
                  <MenuItem fx:id="menuSaveAs" visible="false" text="Save _as...">
                     <accelerator>
                        <KeyCodeCombination alt="DOWN" code="A" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                     </accelerator>
                  </MenuItem>
              <MenuItem fx:id="MenuExit" text="E_xit">
                     <accelerator>
                        <KeyCodeCombination alt="DOWN" code="X" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                     </accelerator></MenuItem>
            </items>
               <accelerator>
                  <KeyCodeCombination alt="DOWN" code="F" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                </accelerator>
          </Menu>
          <Menu text="S_ettings">
            <items>
```

```
              <MenuItem fx:id="menuSettings" text="Edit S_ettings">
                     <accelerator>
                        <KeyCodeCombination alt="DOWN" code="T" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                     </accelerator></MenuItem>
            </items>
               <accelerator>
                  <KeyCodeCombination alt="DOWN" code="T" control="UP" meta="UP" shift="UP"
shortcut="UP" />
               </accelerator>
          </Menu>
          <Menu text="_Help">
            <items>
              <MenuItem fx:id="menuAbout" text="A_bout">
                     <accelerator>
                        <KeyCodeCombination alt="DOWN" code="B" control="UP" meta="UP" shift="UP"
shortcut="UP" />
                     </accelerator></MenuItem>
            </items>
               <accelerator>
                  <KeyCodeCombination alt="DOWN" code="H" control="UP" meta="UP" shift="UP"
shortcut="UP" />
               </accelerator>
          </Menu>
        </menus>
      </MenuBar>
   </top>

    <!-- comment: Rexx program that stores all fx:id objects in .local~rootLayout.fxml directory -->
    <fx:script source="put_FXID_objects_into.my.app.rex" />
</BorderPane>
```

Source code of the file "**PersonPrinterDialog.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.web.WebView?>

<!-- comment: the following process instruction (PI) defines the Java script engine named 'rexx'
              to be used for the code in event attributes like 'onAction' -->

<?language rexx?>

<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity"
prefHeight="600.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/8.0.65"
xmlns:fx="http://javafx.com/fxml/1">
   <children>
      <WebView fx:id="webViewControl" layoutX="10.0" layoutY="14.0" prefHeight="514.0"
prefWidth="580.0"  AnchorPane.bottomAnchor="76.0" AnchorPane.leftAnchor="10.0"
AnchorPane.rightAnchor="10.0" AnchorPane.topAnchor="10.0" />
      <Label fx:id="lblHint" alignment="CENTER" layoutX="10.0" layoutY="518.0" prefHeight="23.0"
prefWidth="580.0" styleClass="label-rgf" textAlignment="CENTER" AnchorPane.bottomAnchor="49.0"
AnchorPane.leftAnchor="10.0" AnchorPane.rightAnchor="10.0" />
      <HBox alignment="CENTER" layoutX="173.0" layoutY="552.0" prefHeight="39.0" prefWidth="223.0"
AnchorPane.bottomAnchor="10.0" AnchorPane.leftAnchor="200.0" AnchorPane.rightAnchor="200.0">
         <children>
            <Button fx:id="btnPrint" defaultButton="true" mnemonicParsing="false" prefWidth="100.0"
text="Print ..." />
            <Button fx:id="btnCancel" cancelButton="true" mnemonicParsing="false" prefWidth="100.0"
text="Cancel" />
         </children>
         <padding>
            <Insets bottom="3.0" left="3.0" right="3.0" top="3.0" />
         </padding>
      </HBox>
   </children>

    <!-- comment: Rexx program that stores all fx:id objects in .local~PersonPrinterDialog.fxml
directory -->
    <fx:script source="put_FXID_objects_into.my.app.rex" />
</AnchorPane>
```

Source code of the file "**PersonEditDialog.fxml**"

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.RowConstraints?>

<!-- comment: the following process instruction (PI) defines the Java script engine named 'rexx'
              to be used for the code in event attributes like 'onAction' -->

<?language rexx?>

<AnchorPane minHeight="-Infinity" minWidth="-Infinity" prefHeight="250.0" prefWidth="350.0"
styleClass="background" xmlns="http://javafx.com/javafx/8.0.65" xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <GridPane hgap="10.0" minHeight="-Infinity" minWidth="-Infinity" prefHeight="165.0"
prefWidth="300.0" AnchorPane.bottomAnchor="75.0" AnchorPane.leftAnchor="20.0"
AnchorPane.rightAnchor="20.0" AnchorPane.topAnchor="20.0">
      <children>
        <Label text="ID" GridPane.columnIndex="0" GridPane.rowIndex="0"  visible="false" />
        <Label fx:id="namecat" text="Name" GridPane.columnIndex="0" GridPane.rowIndex="1" />
        <Label fx:id="adrname" text="Address" GridPane.columnIndex="0" GridPane.rowIndex="2" />
        <Label fx:id="plztax" text="Postal Code" GridPane.columnIndex="0" GridPane.rowIndex="3" />
        <Label text="City" GridPane.columnIndex="0" GridPane.rowIndex="4" visible="false" />
        <Label fx:id="noprice" text="Phone number" GridPane.columnIndex="0" GridPane.rowIndex="5" />
        <TextField fx:id="firstNameField" prefWidth="200.0" GridPane.columnIndex="1"  visible="false"
GridPane.rowIndex="0"  />
        <TextField fx:id="lastNameField" prefWidth="200.0" GridPane.columnIndex="1"
GridPane.rowIndex="1" />
        <TextField fx:id="streetField" prefWidth="200.0" GridPane.columnIndex="1"
GridPane.rowIndex="2" />
        <TextField fx:id="postalCodeField" prefWidth="200.0" GridPane.columnIndex="1"
GridPane.rowIndex="3" />
        <TextField fx:id="cityField" prefWidth="200.0" GridPane.columnIndex="1"  visible="false"
GridPane.rowIndex="4" />
        <TextField fx:id="birthdayField" prefWidth="200.0" GridPane.columnIndex="1"
GridPane.rowIndex="5" />
      </children>
      <columnConstraints>
        <ColumnConstraints fillWidth="true" hgrow="SOMETIMES" maxWidth="80.0" minWidth="80.0"
prefWidth="80.0" />
        <ColumnConstraints fillWidth="true" hgrow="SOMETIMES" maxWidth="230.0" minWidth="10.0"
prefWidth="216.0" />
      </columnConstraints>
      <rowConstraints>
        <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
        <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
        <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
        <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
        <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
        <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
      </rowConstraints>
    </GridPane>
    <HBox id="HBox" alignment="CENTER" minWidth="-1.0" prefHeight="33.0" prefWidth="220.0"
spacing="5.0" AnchorPane.bottomAnchor="10.0" AnchorPane.rightAnchor="10.0">
      <children>
        <Button fx:id="btnOK" defaultButton="true" mnemonicParsing="false" prefWidth="80.0" text="OK"
/>
        <Button fx:id="btnCancel" cancelButton="true" mnemonicParsing="false" prefWidth="93.0"
text="Cancel" />
      </children>
    </HBox>
```

```
  </children>
    <!-- comment: Rexx program that stores all fx:id objects in .local~personEditDialog.fxml
directory -->
    <fx:script source="put_FXID_objects_into.my.app.rex" />
</AnchorPane>
```

Source code of the file "**button.css**"

```
.root {
-fx-base: rgb(50, 50, 50);
-fx-background: rgb(50, 50, 50);
-fx-control-inner-background:  rgb(50, 50, 50);
}

.tab {
-fx-background-color: linear-gradient(to top, -fx-base, derive(-fx-base,30%));
}

.menu-bar {
-fx-background-color: linear-gradient(to bottom, -fx-base, derive(-fx-base,30%));
}

.tool-bar:horizontal {
-fx-background-color:
linear-gradient(to bottom, derive(-fx-base,+50%), derive(-fx-base,-40%), derive(-fx-base,-20%));
}

.button {
-fx-background-color: transparent;
}

.button:hover {
-fx-background-color: -fx-shadow-highlight-color, -fx-outer-border, -fx-inner-border, -fx-body-color;
-fx-color: -fx-hover-base;
}

.table-view {
-fx-table-cell-border-color:derive(-fx-base,+10%);
-fx-table-header-border-color:derive(-fx-base,+20%);
}

.split-pane:horizontal > * > .split-pane-divider {
-fx-border-color: transparent -fx-base transparent -fx-base;
-fx-background-color: transparent, derive(-fx-base,20%);
-fx-background-insets: 0, 0 1 0 1;
}

.my-gridpane {
-fx-background-color: radial-gradient(radius 100%, derive(-fx-base,20%), derive(-fx-base,-20%));
}

.separator-label {
-fx-text-fill: orange;
}

#bevel-grey:hover {
    -fx-background-color:
        linear-gradient(#f0ff35, #a9ff00),
        radial-gradient(center 50% -40%, radius 200%, #b8ee36 45%, #80c800 50%);
    -fx-background-radius: 6, 5;
    -fx-background-insets: 0, 1;
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.4) , 5, 0.0 , 0 , 1 );
    -fx-text-fill: #395306;
}
#round-red {
    -fx-background-color: linear-gradient(#ff5400, #be1d00);
    -fx-background-radius: 30;
    -fx-background-insets: 0;
    -fx-text-fill: white;
}
#bevel-grey {
    -fx-background-color:
        linear-gradient(#f2f2f2, #d6d6d6),
        linear-gradient(#fcfcfc 0%, #d9d9d9 20%, #d6d6d6 100%),
```

```
        linear-gradient(#dddddd 0%, #f6f6f6 50%);
    -fx-background-radius: 8,7,6;
    -fx-background-insets: 0,1,2;
    -fx-text-fill: black;
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );
}
#glass-grey {
    -fx-background-color:
        #c3c4c4,
        linear-gradient(#d6d6d6 50%, white 100%),
        radial-gradient(center 50% -40%, radius 200%, #e6e6e6 45%, rgba(230,230,230,0) 50%);
    -fx-background-radius: 30;
    -fx-background-insets: 0,1,1;
    -fx-text-fill: black;
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 3, 0.0 , 0 , 1 );
}
#shiny-orange {
    -fx-background-color:
        linear-gradient(#ffd65b, #e68400),
        linear-gradient(#ffef84, #f2ba44),
        linear-gradient(#ffea6a, #efaa22),
        linear-gradient(#ffe657 0%, #f8c202 50%, #eea10b 100%),
        linear-gradient(from 0% 0% to 15% 50%, rgba(255,255,255,0.9), rgba(255,255,255,0));
    -fx-background-radius: 30;
    -fx-background-insets: 0,1,2,3,0;
    -fx-text-fill: #654b00;
    -fx-font-weight: bold;
    -fx-font-size: 14px;
    -fx-padding: 15 30 15 30;
}
#dark-blue {
    -fx-background-color:
        #090a0c,
        linear-gradient(#38424b 0%, #1f2429 20%, #191d22 100%),
        linear-gradient(#20262b, #191d22),
        radial-gradient(center 50% 0%, radius 100%, rgba(114,131,148,0.9), rgba(255,255,255,0));
    -fx-background-radius: 5,4,3,5;
    -fx-background-insets: 0,1,2,0;
    -fx-text-fill: white;
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );
    -fx-font-family: "Arial";
    -fx-text-fill: linear-gradient(white, #d0d0d0);
    -fx-font-size: 12px;
    -fx-padding: 10 20 10 20;
}
#dark-blue Text {
    -fx-effect: dropshadow( one-pass-box , rgba(0,0,0,0.9) , 1, 0.0 , 0 , 1 );
}
#dark-blue:hover {
    -fx-effect: dropshadow( one-pass-box , rgba(0,0,0,0.9) , 1, 0.0 , 0 , 1 );
}
#record-sales {
    -fx-padding: 8 15 15 15;
    -fx-background-insets: 0,0 0 5 0, 0 0 6 0, 0 0 7 0;
    -fx-background-radius: 8;
    -fx-background-color:
        linear-gradient(from 0% 93% to 0% 100%, #a34313 0%, #903b12 100%),
        #9d4024,
        #d86e3a,
        radial-gradient(center 50% 50%, radius 100%, #d86e3a, #c54e2c);
    -fx-effect: dropshadow( gaussian , rgba(0,0,0,0.75) , 4,0,0,1 );
    -fx-font-weight: bold;
    -fx-font-size: 1.1em;
}
#record-sales:hover {
    -fx-background-color:
        linear-gradient(from 0% 93% to 0% 100%, #a34313 0%, #903b12 100%),
        #9d4024,
```

```
        #d86e3a,
        radial-gradient(center 50% 50%, radius 100%, #ea7f4b, #c54e2c);
}
#record-sales:pressed {
    -fx-padding: 10 15 13 15;
    -fx-background-insets: 2 0 0 0,2 0 3 0, 2 0 4 0, 2 0 5 0;
}
#record-sales Text {
    -fx-fill: white;
    -fx-effect: dropshadow( gaussian , #a30000 , 0,0,0,2 );
}
#dark-blue:hover {
    -fx-background-color:
        #000000,
        linear-gradient(#7ebcea, #2f4b8f),
        linear-gradient(#426ab7, #263e75),
        linear-gradient(#395cab, #223768);
    -fx-background-insets: 0,1,2,3;
    -fx-background-radius: 3,2,2,2;
    -fx-padding: 12 30 12 30;
    -fx-text-fill: white;
    -fx-font-size: 12px;
}
#dark-blue:hover Text {
    -fx-effect: dropshadow( one-pass-box , rgba(0,0,0,0.8) , 0, 0.0 , 0 , 1);
}
#shiny-orange:hover {
    -fx-background-color:
        #ecebe9,
        rgba(0,0,0,0.05),
        linear-gradient(#dcca8a, #c7a740),
        linear-gradient(#f9f2d6 0%, #f4e5bc 20%, #e6c75d 80%, #e2c045 100%),
        linear-gradient(#f6ebbe, #e6c34d);
    -fx-background-insets: 0,9 9 8 9,9,10,11;
    -fx-background-radius: 50;
    -fx-padding: 15 30 15 30;
    -fx-font-weight: bold;
    -fx-font-size: 14px;
    -fx-text-fill: #311c09;
    -fx-effect: innershadow( three-pass-box , rgba(0,0,0,0.1) , 2, 0.0 , 0 , 1);
}
#shiny-orange:hover Text {
    -fx-effect: dropshadow( one-pass-box , rgba(255,255,255,0.5) , 0, 0.0 , 0 , 1);
}
#iphone-toolbar {
    -fx-background-color: linear-gradient(#98a8bd 0%, #8195af 25%, #6d86a4 100%);
}
#iphone {
    -fx-background-color:
        #a6b5c9,
        linear-gradient(#303842 0%, #3e5577 20%, #375074 100%),
        linear-gradient(#768aa5 0%, #849cbb 5%, #5877a2 50%, #486a9a 51%, #4a6c9b 100%);
    -fx-background-insets: 0 0 -1 0,0,1;
    -fx-background-radius: 5,5,4;
    -fx-padding: 7 30 7 30;
    -fx-text-fill: #242d35;
    -fx-font-family: "Helvetica";
    -fx-font-size: 12px;
    -fx-text-fill: white;
}
#iphone Text {
    -fx-effect: dropshadow( one-pass-box , rgba(0,0,0,0.8) , 0, 0.0 , 0 , -1 );
}
#ipad-dark-grey {
    -fx-background-color:
        linear-gradient(#686868 0%, #232723 25%, #373837 75%, #757575 100%),
        linear-gradient(#020b02, #3a3a3a),
        linear-gradient(#9d9e9d 0%, #6b6a6b 20%, #343534 80%, #242424 100%),
```

```
        linear-gradient(#8a8a8a 0%, #6b6a6b 20%, #343534 80%, #262626 100%),
        linear-gradient(#777777 0%, #606060 50%, #505250 51%, #2a2b2a 100%);
    -fx-background-insets: 0,1,4,5,6;
    -fx-background-radius: 9,8,5,4,3;
    -fx-padding: 15 30 15 30;
    -fx-font-family: "Helvetica";
    -fx-font-size: 18px;
    -fx-font-weight: bold;
    -fx-text-fill: white;
    -fx-effect: dropshadow( three-pass-box , rgba(255,255,255,0.2) , 1, 0.0 , 0 , 1);
}
#ipad-dark-grey Text {
    -fx-effect: dropshadow( one-pass-box , black , 0, 0.0 , 0 , -1 );
}
#ipad-grey {
    -fx-background-color:
        linear-gradient(#686868 0%, #232723 25%, #373837 75%, #757575 100%),
        linear-gradient(#020b02, #3a3a3a),
        linear-gradient(#b9b9b9 0%, #c2c2c2 20%, #afafaf 80%, #c8c8c8 100%),
        linear-gradient(#f5f5f5 0%, #dbdbdb 50%, #cacaca 51%, #d7d7d7 100%);
    -fx-background-insets: 0,1,4,5;
    -fx-background-radius: 9,8,5,4;
    -fx-padding: 15 30 15 30;
    -fx-font-family: "Helvetica";
    -fx-font-size: 18px;
    -fx-font-weight: bold;
    -fx-text-fill: #333333;
    -fx-effect: dropshadow( three-pass-box , rgba(255,255,255,0.2) , 1, 0.0 , 0 , 1);
}
#ipad-grey Text {
    -fx-effect: dropshadow( one-pass-box , white , 0, 0.0 , 0 , 1 );
}
#ipad-grey:hover {
      -fx-background-color:
        linear-gradient(#686868 0%, #232723 25%, #373837 75%, #757575 100%),
        linear-gradient(#020b02, #3a3a3a),
        linear-gradient(#9d9e9d 0%, #6b6a6b 20%, #343534 80%, #242424 100%),
        linear-gradient(#8a8a8a 0%, #6b6a6b 20%, #343534 80%, #262626 100%),
        linear-gradient(#777777 0%, #606060 50%, #505250 51%, #2a2b2a 100%);
    -fx-background-insets: 0,1,4,5,6;
    -fx-background-radius: 9,8,5,4,3;
    -fx-padding: 15 30 15 30;
    -fx-font-family: "Helvetica";
    -fx-font-size: 18px;
    -fx-font-weight: bold;
    -fx-text-fill: white;
    -fx-effect: dropshadow( three-pass-box , rgba(255,255,255,0.2) , 1, 0.0 , 0 , 1);
}
#lion:hover {
    -fx-background-color:
        rgba(0,0,0,0.08),
        linear-gradient(#5a61af, #51536d),
        linear-gradient(#e4fbff 0%,#cee6fb 10%, #a5d3fb 50%, #88c6fb 51%, #d5faff 100%);
    -fx-background-insets: 0 0 -1 0,0,1;
    -fx-background-radius: 5,5,4;
    -fx-padding: 3 30 3 30;
    -fx-text-fill: #242d35;
    -fx-font-size: 14px;
}
#lion {
    -fx-background-color:
        rgba(0,0,0,0.08),
        linear-gradient(#9a9a9a, #909090),
        linear-gradient(white 0%, #f3f3f3 50%, #ececec 51%, #f2f2f2 100%);
    -fx-background-insets: 0 0 -1 0,0,1;
    -fx-background-radius: 5,5,4;
    -fx-padding: 3 30 3 30;
    -fx-text-fill: #242d35;
```

```
    -fx-font-size: 14px;
}
#windows7:hover{
    -fx-background-color:
        #3c7fb1,
        linear-gradient(#fafdfe, #e8f5fc),
        linear-gradient(#eaf6fd 0%, #d9f0fc 49%, #bee6fd 50%, #a7d9f5 100%);
    -fx-background-insets: 0,1,2;
    -fx-background-radius: 3,2,1;
    -fx-padding: 3 30 3 30;
    -fx-text-fill: black;
    -fx-font-size: 14px;
}
#windows7 {
    -fx-background-color:
        #707070,
        linear-gradient(#fcfcfc, #f3f3f3),
        linear-gradient(#f2f2f2 0%, #ebebeb 49%, #dddddd 50%, #cfcfcf 100%);
    -fx-background-insets: 0,1,2;
    -fx-background-radius: 3,2,1;
    -fx-padding: 3 30 3 30;
    -fx-text-fill: black;
    -fx-font-size: 14px;
}



#custom-button-yellow {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: yellow;
}
#custom-button-darkCreem {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #F7D358;
}
#custom-button-lightPink {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #F78181;
}
#custom-button-moonLightPink {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #F6CECE;
}
#custom-button-skyBlue {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #81DAF5;
}
#custom-button-skyDarkBlue {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #58ACFA;
}
#custom-button-sceenColor {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #F3E2A9;
}
#custom-button-parrot {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #9AFE2E;
}
#custom-button-pinkDarkligght {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #F5BCA9;

}

#custom-button-imageButton {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: -fx-shadow-highlight-color, -fx-outer-border, -fx-inner-border, -fx-body-
```

```
color;
-fx-color: -fx-hover-base;
}

#custom-button-imageButton:hover {
    -fx-font: bold 11pt "Serif";
    -fx-background-color: #58ACFA, -fx-shadow-highlight-color, -fx-outer-border, -fx-inner-border,
-fx-body-color;
-fx-color: -fx-hover-base;
}

#custom-button-skyDarkBlueCalculator {
    -fx-font: bold 15pt "Serif";
    -fx-background-color: #58ACFA;
}

#custom-table-table1 {
    -fx-background-color: white;
}
#custom-textField-tf1 {
    -fx-background-color: white;
    -fx-font: 10pt "Serif";
    -fx-text-fill: black;
}
#custom-hbox-white {
    -fx-background-color: white;
}
#custom-hbox-red{
    -fx-background-color: #FE2E64;
}

#custom-hbox-darkGray{
    -fx-background-color: #848484;
}
#custom-hbox-skyBlue{
    -fx-background-color: #2E9AFE;
}

#custom-vbox-black{
    -fx-background-color: #2E2E2E;
}
#custom-anchor-gray{
    -fx-background-color: #848484;
}

#dark-blue2 {
    -fx-background-color:
        #090a0c,
        linear-gradient(#38424b 0%, #1f2429 20%, #191d22 100%),
        linear-gradient(#20262b, #191d22),
        radial-gradient(center 50% 0%, radius 100%, rgba(114,131,148,0.9), rgba(255,255,255,0));
    -fx-background-radius: 5,4,3,5;
    -fx-background-insets: 0,1,2,0;
    -fx-text-fill: white;
    -fx-effect: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );
    -fx-font-family: "Arial";
    -fx-text-fill: linear-gradient(white, #d0d0d0);
    -fx-font-size: 12px;
}
```