Design and Implementation of an Internet based Calendar System

Winter semester 2000/2001

Thomas Jungmann & Reinhold Klapsing

University of Essen

Information systems and software engineering

Univ. Prof. Dr. Rony G. Flatscher

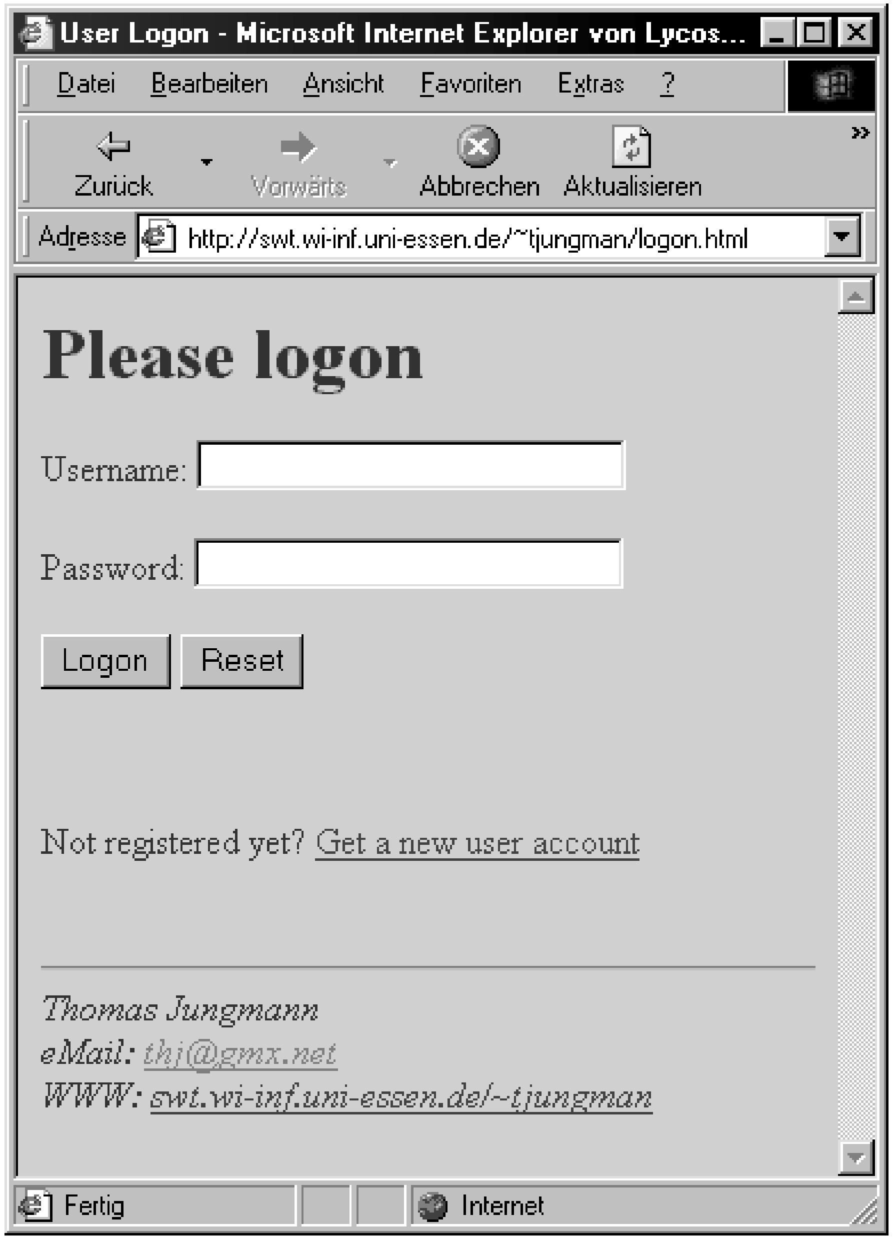
Introduction
Overview of this presentation

- Introduction
- System usage (‚walkthrough‘)
- Requirement analysis
- System architecture
- Database design
- Conclusion and future work

Introduction
Scope of work

- Within scope:
  - Design and implementation of an internet based calendar system
  - Access to a calendar with a web browser
- Out of scope:
  - Groupware-functionality like free/busy-time planning
  - Interoperation with other calendar systems

System usage Screenshots



http://swt.wi-inf.uni-essen.de/~tjungman/logon.html

Please logon

Username:

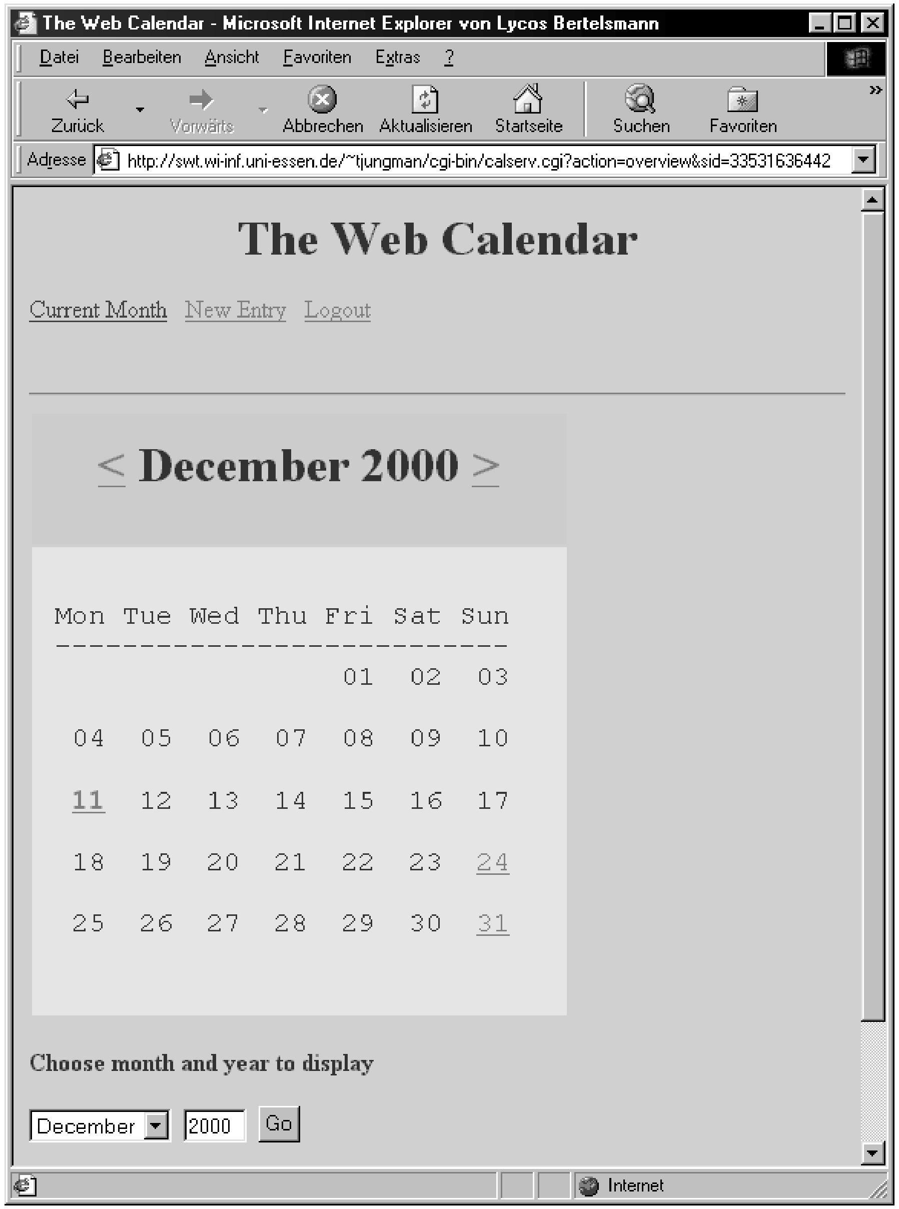
Password:

Logon | Reset

Not registered yet? Get a new user account

Thomas Jungmann
eMail: thj@gmx.net
WWW: swt.wi-inf.uni-essen.de/~tjungman

System usage Screenshots



The Web Calendar - Microsoft Internet Explorer von Lycos Bertelsmann
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten
Adresse
Internet

http://swt.wi-inf.uni-essen.de/~tjungman/cgi-bin/calserv.cgi?action=overview&sid=33531636442

The Web Calendar

Current Month New Entry Logout

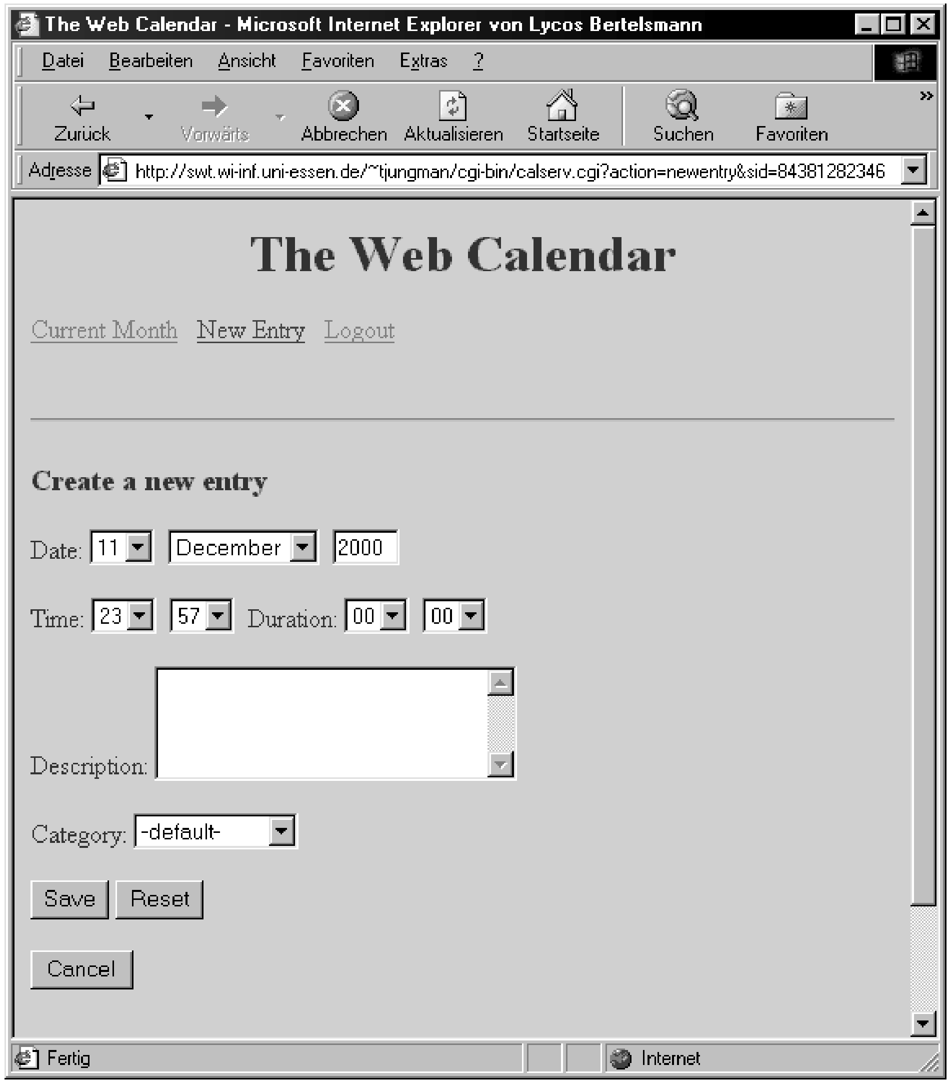
< December 2000 >

```
Mon Tue Wed Thu Fri Sat Sun
---------------------------
                 01  02  03

 04  05  06  07  08  09  10

 11  12  13  14  15  16  17

 18  19  20  21  22  23  24

 25  26  27  28  29  30  31
```

Choose month and year to display

December 2000 Go

System usage Screenshots


The Web Calendar - Microsoft Internet Explorer von Lycos Bertelsmann
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten
Adresse
Fertig
Internet

http://swt.wi-inf.uni-essen.de/~tjungman/cgi-bin/calserv.cgi?action=newentry&sid=84381282346

The Web Calendar

Current Month New Entry Logout

Create a new entry

Date: 11 December 2000

Time: 23 57 Duration: 00 00

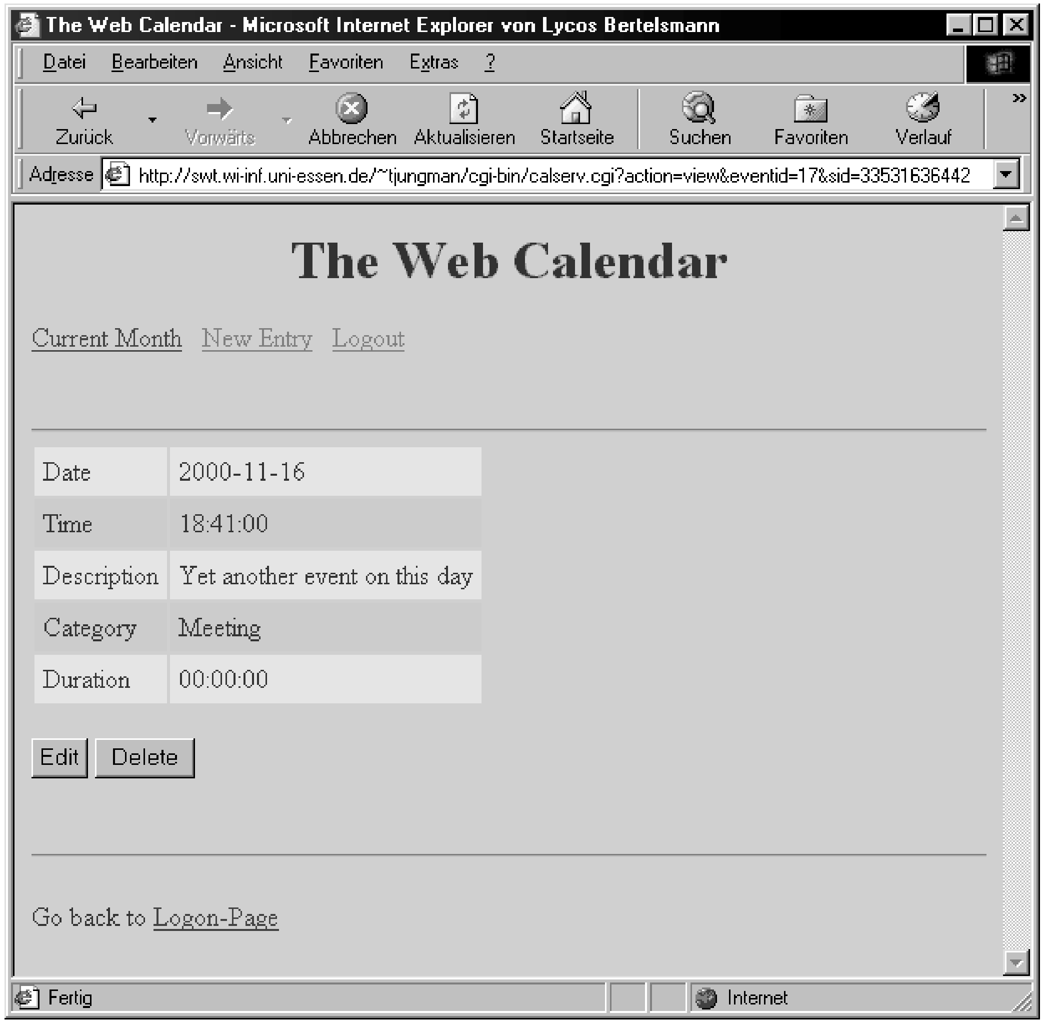
Description:

Category: -default-

Save Reset

Cancel

System usage Screenshots


The Web Calendar - Microsoft Internet Explorer von Lycos Bertelsmann
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten Verlauf
Adresse

http://swt.wi-inf.uni-essen.de/~tjungman/cgi-bin/calserv.cgi?action=view&eventid=17&sid=33531636442

The Web Calendar

Current Month New Entry Logout

Date	2000-11-16
Time	18:41:00
Description	Yet another event on this day
Category	Meeting
Duration	00:00:00

Go back to Logon-Page

Introduction
Standards

- IETF Working Group „**Cal**endaring and **Sched**uling“ (calsched)
- Main work:
  - RFC 2445 Internet Calendaring and Scheduling Core Object Specifications (iCalendar)
    - Specifies the objects and data types (MIME-Type text/calendar)
  - RFC 2446 iCalendar Transport-Independent Interoperability Protocol (iTIP)
    - Interoperation of calendar systems using iCalendar Objects
  - Several other RFCs and Internet drafts as well, but all concerning interoperation between calendar systems

Introduction
Definition of terms (1/2)

- Calendar
  - A collection of calendar events associated with a specific user
- Calendar Event
  - An entry in a calendar that represents an event for a specific user
- Calendar User
  - An entity that uses a calendaring system

Introduction
Definition of terms (2/2)

- **Calendar User Agent**
  - The client application that a Calendar User utilizes to access and manipulate a calendar (the web browser)
- **Calendar Service**
  - The collection of programs that receive and interpret the Calendar Users commands and also generate and format the output for the user
- **Calendar Store**
  - The database that stores the calendars

Requirement analysis Overview

- 2 steps of requirements analysis
  - Step 1: Technical considerations must not dominate users needs
    => no technical terms/solutions in mind, only **Users View** in ‚plain english'
  - Step 2: search for appropriate technical solution for these needs from the **Software Designers** point of view and refinement of needs

Requirement analysis

Users view

- Easy access to the calendar from everywhere, no special software is needed (e.g. Internet Café Scenario)

Programmers view

- Calendar User Agent must be a standard web browser. Communication over HTTP, HTML and CGI only
- Web server must support CGI as well (‚Apache' will be used, because it is available on many platforms)

Requirement analysis

Users view

- System must be able to work with multiple users

Programmers view

- Probably large amount of data (incl. meta-data for admin. purposes) => use of a powerful database recommended
- MySQL will be used (relieable, available for many platforms, ANSI SQL 92 Standard used)
- Session management needed to distinguish users (HTTP is stateless)

Requirement analysis

Users view

- Calendar data is private, need for confidentiality => User must be authenticated

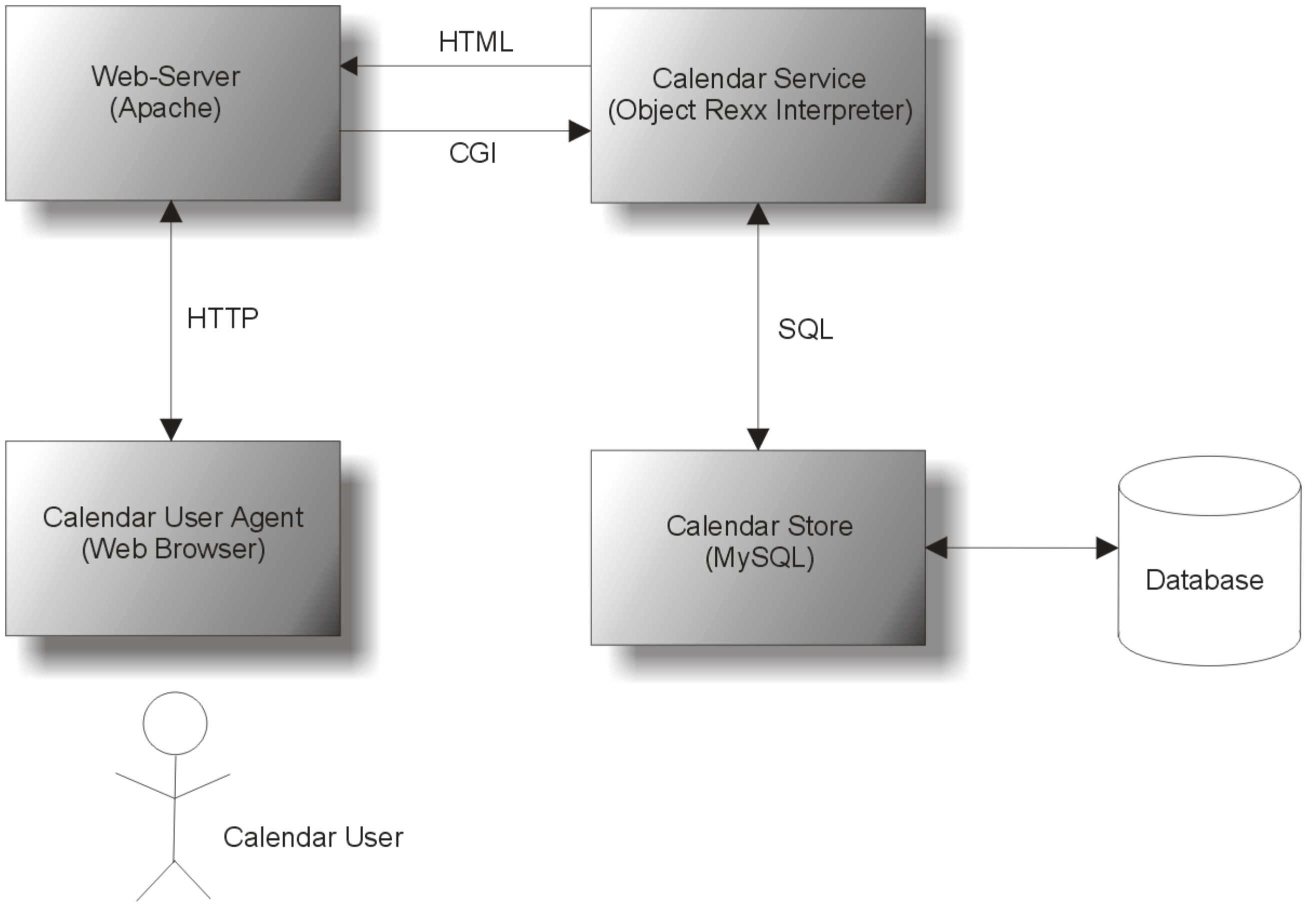
Programmers view

- Password check during logon
- Sessions must timeout after certain time

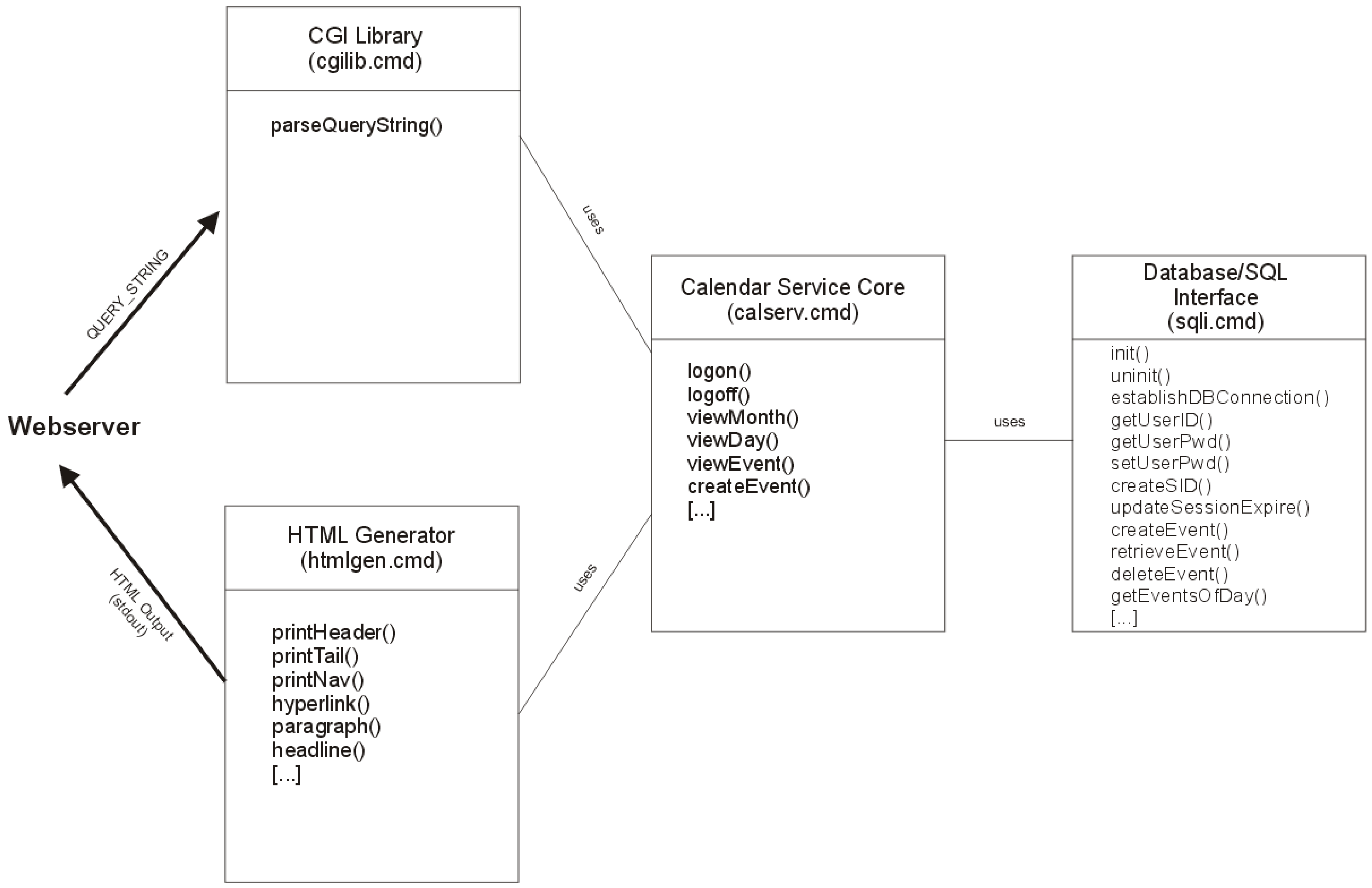
Requirement analysis

- Other (technical) requirements
  - Programming language Object Rexx
    - Scripting language with powerful string parsing functions preferable (because of HTML/CGI)
    - Available on many platforms
    - Interface to many databases available (Rexx/SQL)
- Session management
  - Can be achieved by
    - HTML Hidden Form fields
    - CGI PATH_INFO Mechanism
    - RFC 2109, HTTP State Management Mechanism: „Cookies“

System architecture



System architecture: Diagramm of classes



System architecture Interfaces

- Self-Initializing through constructor:

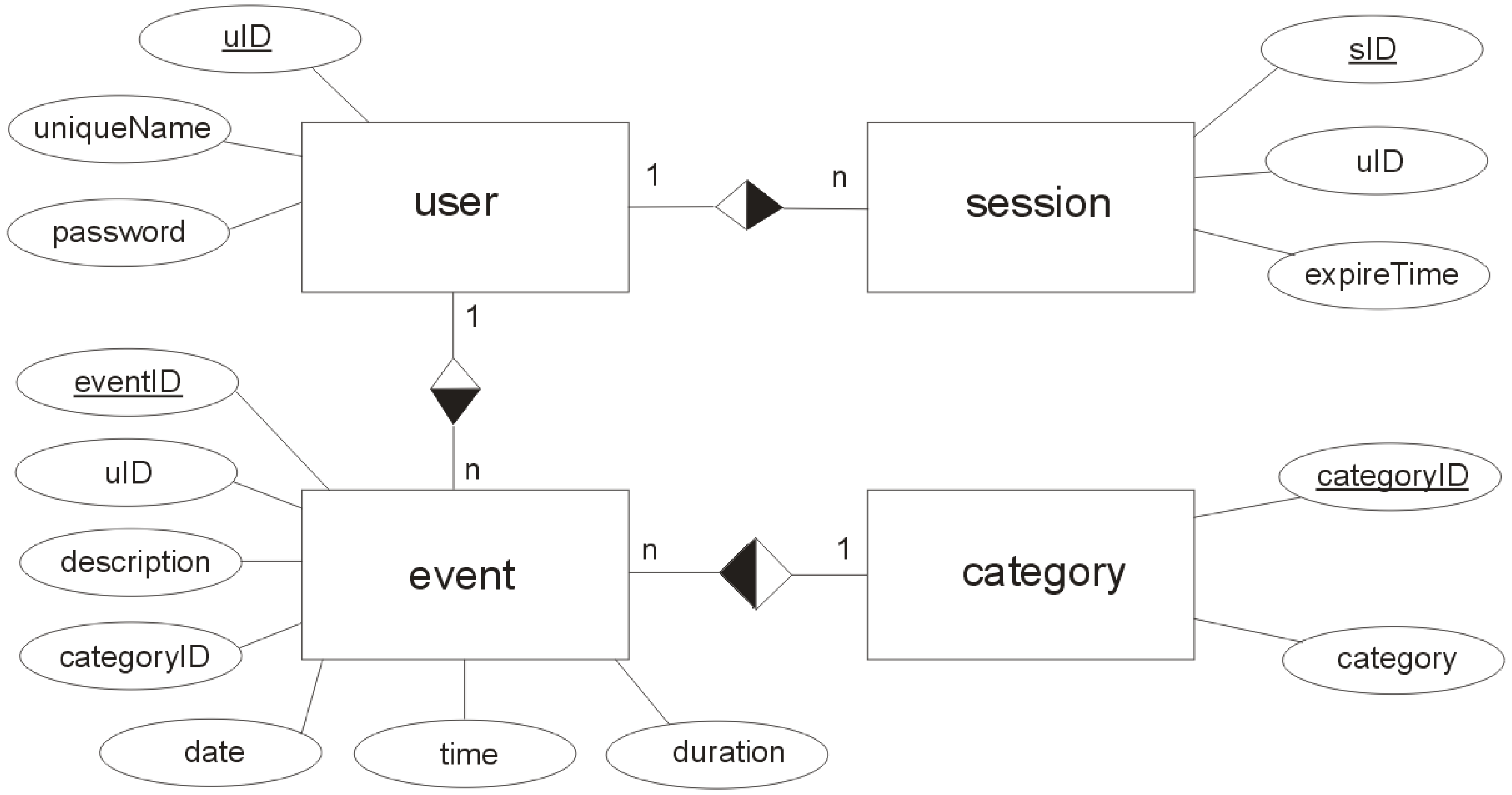
```
::METHOD INIT         /* Constructor */
       if rxFuncQuery("SQLLoadFuncs") then do
           call rxFuncAdd "SQLLoadFuncs","rexxsql","SQLLoadFuncs"
           call SQLLoadFuncs
       end
       self~establishDBConnection()
```

- Advantages of separate interfaces:
  - easy adaption in changing environments
  - easy reuse of code in similar applications

System architecture
Main Program

```
call initVars
html~printHeader('The Web Calendar')
select
    when cgi.action = 'logon'       then call logon         --check username and password
                                                            --and create session
    when cgi.action = 'newuser'     then call newuser       --create a new user account
    when cgi.action = 'logout'      then call logout        --invalidate session
    when cgi.action = 'overview'    then call overview      --display overview of this month
    when cgi.action = 'goto'        then call gotoMonth     --navigate to specific month
    when cgi.action = 'gotoform'    then call gotoForm      --show HTML-Form for gotoMonth
    when cgi.action = 'viewday'     then call viewDay       --display all events of day
    when cgi.action = 'view'        then call viewEvent     --display event details
    when cgi.action = 'create'      then call createEvent   --add event to database
    when cgi.action = 'newentry'    then call newentryForm  --show HTML-Form for createEvent
    when cgi.action = 'Edit'        then call editEventForm --show HTML-Form for updateEvent
    when cgi.action = 'update'      then call updateEvent   --accept modifications for event
    when cgi.action = 'Delete'      then call deleteEvent   --delete event permanently
    otherwise call abort 'Unknown CGI-Action'
end
html~printTail
DROP db            --drop references to interfaces
DROP html
exit 0
```

Database design
Entity relationship model



Conclusion and future work

- Interoperation
  - Calendar system is standalone, no interoperations with other systems, no interoperation between users (free/busy schedule)
  - IETF has already released standards for data types and protocols for interoperation
- Conceptual improvements
  - Interface model has not been implemented strictly
  - Exchanging HTML for WML is not easy to do, as there is HTML specific code mixed into the core script

Conclusion and future work

- Object orientation
  - System was written in Object Rexx, but little concepts of the object oriented paradigm were actually used. To much procedural thinking
  - ‚Real' OO-Design also possible, e.g. Events have methods to create, alter or delete themselves, User objects have methods to check their passwords, etc.
- Security
  - Based only on passwords and session timeout.
  - Unencrypted, so sniffing attacks possible
  - Even worse: CGI-GET-Method used for data transmission => cache-logfiles store all information
  - Improvement: use of POST-Method
  - Even better: use of Secure Socket Layer SSL

Summary

- **INTERNET CALENDAR SYSTEM:**
  - Can be used from everywhere, even with a WAP-capable cellular phone
  - All components are freely available (MySQL only for non-comercial use)
  - Distributed system: Web server and Database server can be placed on different machines
  - Easy to use intuitive user interface
  - Year 2000 compliant ;)