Contents

1 About

This little manual shall introduce the basic concepts that are relevant to solve the Implementation phase of the lab. Only basic programming skills in Java and MySQL are necessary. If you only know other languages like C++ it should be very easy to start this new topic. It is a very similar syntax :-)

To simplify the implementation, we make use of a template engine called Freemarker[1]. This provides an easy to use syntax and clear separation between Java Application Code and HTML as the language for representation. We use only a small extract of all functions so do not get fear using this technology.

If you have any questions or problems do not hesitate to contact us.

Hint: The screenshots may contain older version numbers than in the text. Please always use the version as stated in the text.

2 Installation

We assume that you use one of the following operating systems: Mac OS, Windows, or Linux (ubuntu-based). For any other OS please check the installation and compatibility of the components on the internet.

[1] http://freemarker.org

First of all, we have to install and configure some components to be able to start programming. These are namely *Eclipse Java EE*, *Apache Tomcat* and *MySQL*.

The first step is to create a folder in which we later unzip and install our different components. A possible name is *swtdev* or something similar you can keep in mind. Now we will go on with downloading and installing the components.

2.1 Apache Tomcat

Apache Tomcat is our webserver that is responsible for the availability of our application on the internet. It is open source and can be downloaded for free.

a) Navigate to `https://tomcat.apache.org/download-90.cgi`. This is the download page for the current stable version.

b) Scroll down to the first appearing release (9.0.xx). There you have to distinguish between two files:

   Linux/Mac OS Select **zip** in section **Binary Distribution**. The download will start immediately.

   Windows Select **32-bit Windows zip** or **64-bit Windows zip** depending on your type of OS. The download will start immediately.

c) Unzip the downloaded file into the previously created folder. Make sure, that all files are stored in a subfolder and not directly in the main folder.

Since we use Tomcat only for development purposes, we do not have to install it as a service that is started automatically with the OS.

2.2 Eclipse Java EE

The main development will be done in the already known Java IDE *Eclipse*. The version for Java EE contains some special features for web development.

a) Navigate to `http://www.eclipse.org/downloads/packages/`. The actual release is shown on top. In the right corner of the table, you can select your operating system. Normally the system is chosen automatically.

b) In this table select **Eclipse IDE for Enterprise Java Developers**. Select one of the given mirrors on the next page and start downloading.

c) Extract the files into a subfolder as done with Tomcat before.

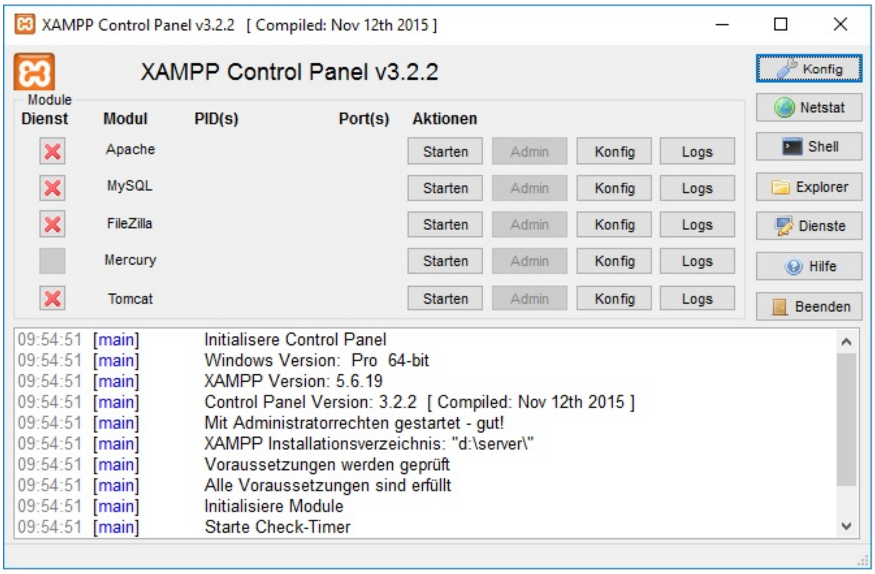
2.3 MySQL

For reasons of handling, we do not provide an installation guide for a stand-alone database server. Instead, we would like to propose to use XAMPP. This is a full web development framework that provides a MySQL server instance with an embedded administration tool called phpMyAdmin. As stated in the name it is based on PHP. Therefore Apache2 with PHP is also used in this framework.

a) Navigate to `https://www.apachefriends.org/de/index.html` where you can download XAMPP for your OS.

b) Follow the installation instructions on the webpage and finish the installation.

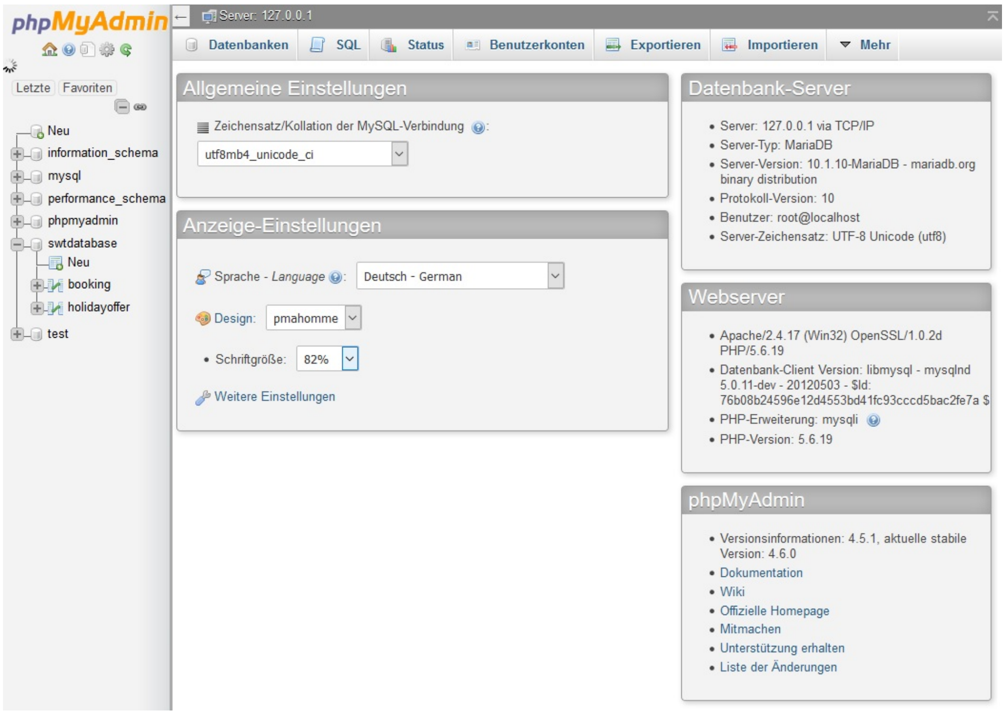
c) To start XAMPP open the folder you have installed it to and open the file "xampp-control" as **admin**. To proceed, start the modules **Apache and MySQL**.



```
09:54:51  [main]    Initialisere Control Panel
09:54:51  [main]    Windows Version:  Pro  64-bit
09:54:51  [main]    XAMPP Version: 5.6.19
09:54:51  [main]    Control Panel Version: 3.2.2  [ Compiled: Nov 12th 2015 ]
09:54:51  [main]    Mit Administratorrechten gestartet - gut!
09:54:51  [main]    XAMPP Installationsverzeichnis: "d:\server\"
09:54:51  [main]    Voraussetzungen werden geprüft
09:54:51  [main]    Alle Voraussetzungen sind erfüllt
09:54:51  [main]    Initialisiere Module
09:54:51  [main]    Starte Check-Timer
```

Figure 1: XAMPP Control Panel

d) After you have started XAMPP navigate to http://127.0.0.1/phpmyadmin. Optional: If you entered a MySQL password during the installation use these login information. If you do not get to the page shown in Figure 2, please make sure you did step c) properly.



Zeichensatz/Kollation der MySQL-Verbindung: utf8mb4_unicode_ci

Sprache - *Language*: Deutsch - German

Design: pmahomme

Schriftgröße: 82%

Datenbank-Server

- Server: 127.0.0.1 via TCP/IP
- Server-Typ: MariaDB
- Server-Version: 10.1.10-MariaDB - mariadb.org binary distribution
- Protokoll-Version: 10
- Benutzer: root@localhost
- Server-Zeichensatz: UTF-8 Unicode (utf8)

Webserver

- Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.19
- Datenbank-Client Version: libmysql - mysqlnd 5.0.11-dev - 20120503 - $Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a $
- PHP-Erweiterung: mysqli
- PHP-Version: 5.6.19

phpMyAdmin

- Versionsinformationen: 4.5.1, aktuelle stabile Version: 4.6.0
- Dokumentation
- Wiki
- Offizielle Homepage
- Mitmachen
- Unterstützung erhalten
- Liste der Änderungen

Figure 2: phpMyAdmin main page

e) On the left side you can see a list of all databases and on the top the main functions are presented.

f) First of all you have to create a new database in which you can import the provided sample data. In the main navigation choose **Databases**. There you can enter a name for the new database.

g) Now choose the newly created database from the list on the left.

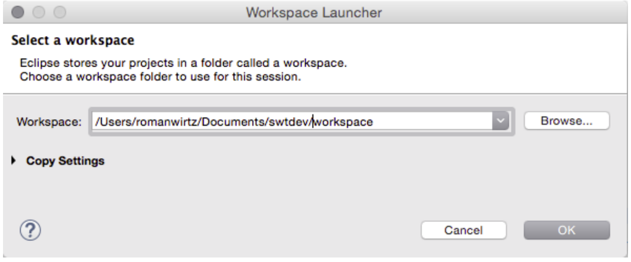
h) For importing data we use the function **Import**. Click on the item, and on the next page, you can select the file with the sample data which you downloaded from Moodle. Start the import with **Go**. The sample data should now be imported into your database.

3 Configuration & Data Import

Now it is time to start Eclipse. The following steps are independent of your OS.
Navigate to the subfolder and double-click on the file **eclipse** (If Eclipse is not starting properly, please make sure you have the latest java version installed).

3.1 Eclipse Configuration

a) First you have to select a workspace. It is recommended to put this folder in your development folder. The path given in Figure 3 depends on your filesystem. Then click Ok and Eclipse will start.

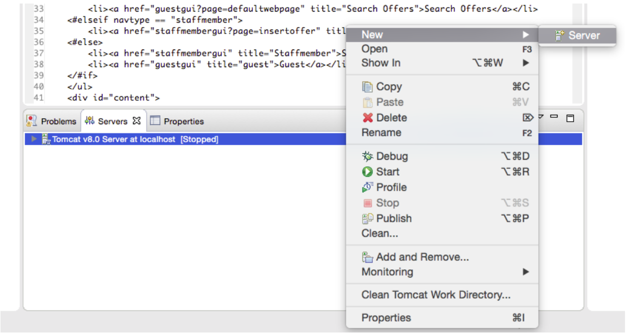


Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /Users/romanwirtz/Documents/swtdev/workspace

Figure 3: Select Workspace in Eclipse

b) Now we first have to define a new Tomcat Server based on our downloaded files. On the bottom there is a tab called **Servers**. Do a right-click in the empty space and select **New** → **Server**. This is shown in Figure 4.

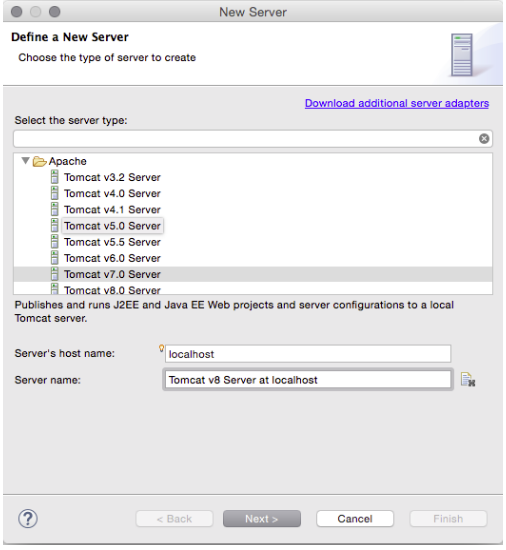


```
        <li><a href="guestgui?page=defaultwebpage" title="Search Offers">Search Offers</a></li>
    <#elseif navtype == "staffmember">
        <li><a href="staffmembergui?page=insertoffer" titl
    <#else>
        <li><a href="staffmembergui" title="Staffmember">S
        <li><a href="guestgui" title="guest">Guest</a></li
    </#if>
    </ul>
    <div id="content">
```

Tomcat v8.0 Server at localhost [Stopped]

Figure 4: Create new server: Step 1

In the next windows you have to select **Apache** → **Tomcat v9.0 Server**. The text fields do not have to be changed. Click **Next**. This is shown in Figure 5.



New Server

Define a New Server

Choose the type of server to create

Select the server type:

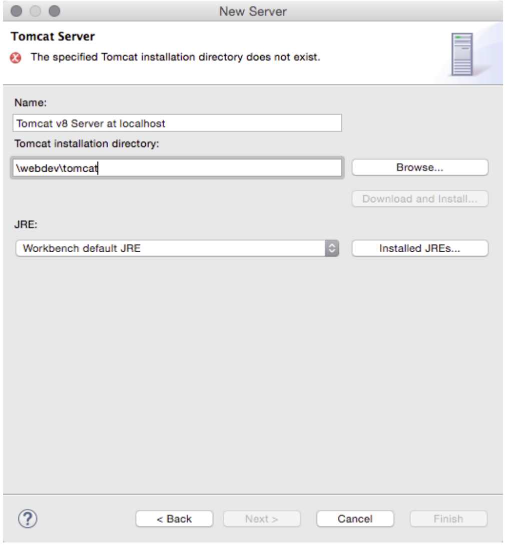
Publishes and runs J2EE and Java EE Web projects and server configurations to a local Tomcat server.

Server's host name: localhost

Server name: Tomcat v8 Server at localhost

Figure 5: Create new server: Step 2

Now you have to select the installation folder of your Tomcat installation which is equal to the subfolder where you unziped the downloaded files. If all works fine no error will be shown and you can click **Finish**. This is shown in Figure 6.



New Server

Tomcat Server

The specified Tomcat installation directory does not exist.

Name: Tomcat v8 Server at localhost

Tomcat installation directory: \webdev\tomcat

JRE: Workbench default JRE

Figure 6: Create new server: Step 3

In the Server tab your created server Server will be shown as **Tomcat v9.0 Server at localhost** [**Stopped**]. That means all works fine and your server is not yet started.

c) To make the first programming steps easier, we provide an Eclipse project in Moodle. This contains a small implementation extract of the well-know *Vacation Rental* example presented in the lecture. The functionalities of this application will not be explained again.

Download the zip-file from Moodle. Do **NOT** extract the file! Then open Eclipse and in the main menu select **File** → **Import** and in the new window select **General** → **Existing Project into Workspace**. Click next.
In the next window, you have to select the already downloaded file. Please mark all elements in the explorer to be imported. After that please finish this process.

Afterward, the imported project is shown in the Project Explorer as you should already be familiar with from other lectures that use Eclipse.

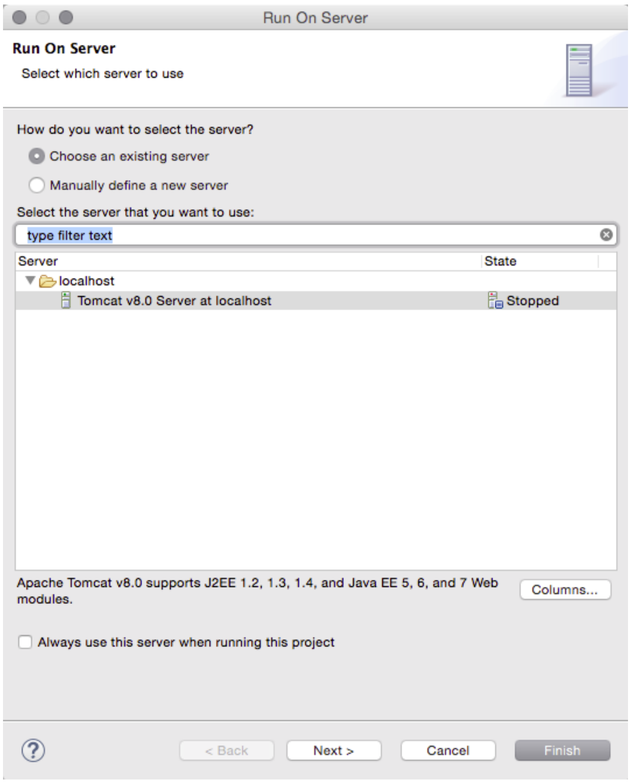
3.2 Database Preparation

The next step is to prepare the database. Therefore, you have to enter the login data in your Java code (only if you have created a password in step 2.3).

In the Project Explorer select the imported project and then **Java Resources** → **src** → **dbadapter** → **Configuration.java**. This class contains some static variables of type String where you can enter the data provided by your database server, i.e. username, password, and database name. Usually, the server will remain *localhost* with the standard port *3306*.

3.3 First start

If you have done all steps above it is time for the first look at our running application. Switch back to eclipse and in Project Explorer select **Java Resources** → **src** → **servlets** → **WelcomeServlet.java**.



Run On Server

Run On Server

Select which server to use

How do you want to select the server?

● Choose an existing server

○ Manually define a new server

Select the server that you want to use:

type filter text

Server	State
localhost	
Tomcat v8.0 Server at localhost	Stopped

Apache Tomcat v8.0 supports J2EE 1.2, 1.3, 1.4, and Java EE 5, 6, and 7 Web modules.

□ Always use this server when running this project

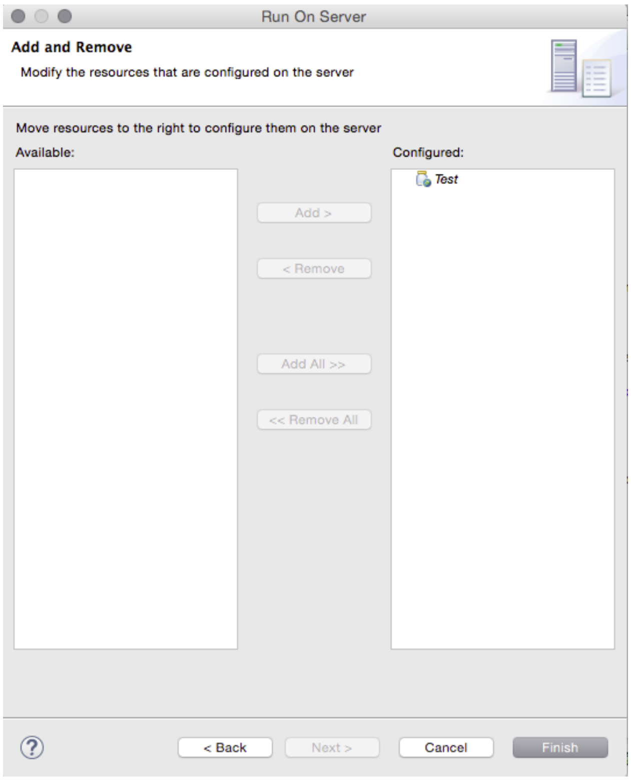
Figure 7: Application start: Step 1

Now start this class as you know from other Java programming lectures by clicking the green arrow in the top toolbar. The next screen is shown in Figure 7. Maybe you have to choose **Run on server** in a window opened before. Select your local server in click **Next**. In the next window you have to add your application to the server. This is done by marking it in the left field and then clicking **Add**. In some cases the application will be added automatically. *Test* will be replaced with your application name. Next the application appears under *Configured*. To start the application click **Finish**. This step is shown in Figure 8. In Eclipse a new window will be opened which is a very small browser. It does not support all recent standards, so open your own browser and navigate to `http://localhost/VR`. VR has to be replaced with the name of your project.

If errors occur: Switch back to Eclipse and have a look into the *Console*. As in every other Java application, all exceptions are printed there. Check all the previous steps maybe undo some steps and repeat them. If you are not able to locate the error copy the printed exceptions into an e-mail and explain what went wrong and when. It is also possible and sometimes easier to ask one of the supervisors in the labs or tutorials.

4 Programming

We hope that you have already looked at the example and that you can understand the main concepts of the Java files. In this section, we will explain how the different files are connected and how you can adapt and extend the web application for your own needs.



Run On Server

Add and Remove

Modify the resources that are configured on the server

Move resources to the right to configure them on the server

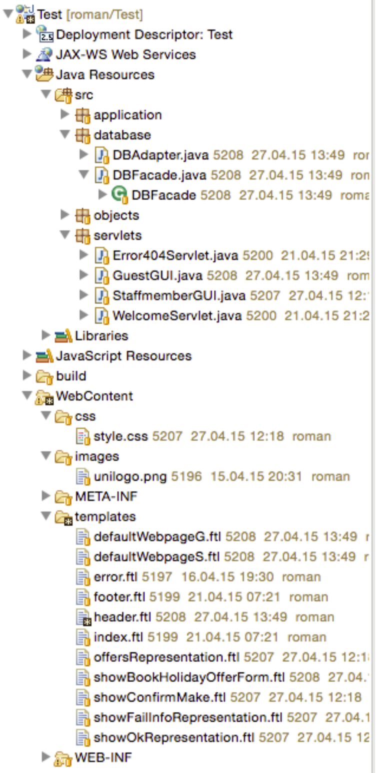
Available:

Configured:

Test

Figure 8: Application start: Step 2

4.1 Structure



```
▼ Test [roman/Test]
  ▶ Deployment Descriptor: Test
  ▶ JAX-WS Web Services
  ▼ Java Resources
    ▼ src
      ▶ application
      ▼ database
        ▶ DBAdapter.java 5208  27.04.15 13:49  ror
        ▼ DBFacade.java 5208  27.04.15 13:49  ron
          ▶ DBFacade 5208  27.04.15 13:49  roma
      ▶ objects
      ▼ servlets
        ▶ Error404Servlet.java 5200  21.04.15 21:2
        ▶ GuestGUI.java 5208  27.04.15 13:49  rom
        ▶ StaffmemberGUI.java 5207  27.04.15 12:
        ▶ WelcomeServlet.java 5200  21.04.15 21:2
    ▶ Libraries
  ▶ JavaScript Resources
  ▶ build
  ▼ WebContent
    ▼ css
        style.css 5207  27.04.15 12:18  roman
    ▼ images
        unilogo.png 5196  15.04.15 20:31  roman
    ▶ META-INF
    ▼ templates
        defaultWebpageG.ftl 5208  27.04.15 13:49  r
        defaultWebpageS.ftl 5208  27.04.15 13:49  r
        error.ftl 5197  16.04.15 19:30  roman
        footer.ftl 5199  21.04.15 07:21  roman
        header.ftl 5208  27.04.15 13:49  roman
        index.ftl 5199  21.04.15 07:21  roman
        offersRepresentation.ftl 5207  27.04.15 12:1
        showBookHolidayOfferForm.ftl 5208  27.04.
        showConfirmMake.ftl 5207  27.04.15 12:18
        showFailInfoRepresentation.ftl 5207  27.04.1
        showOkRepresentation.ftl 5207  27.04.15 12
    ▶ WEB-INF
```

Figure 9: Structure overview

The sample project is structured in three different (important) parts. This structure is shown in Figure 9. All other packages and files not explained in the below sections may not be touched. Otherwise your application can be broken.

We recommend that you do backups of your already working parts. This can be easily done using your SVN repository. See the introduction slides for

4.1.1 web.xml / Deployment Descriptor

The deployment descriptor is written as an XML file and contains all necessary information for the server to deploy the application.
Here you only have to change/add the servlet mappings. They describe what Java file is responsible for a given request.
Example: When the user calls the request */index*, his request will be forwarded to the Java file Index.java.

```
<servlet-mapping>
    <servlet-name>indexservlet</servlet-name>
    <url-pattern>/index</url-pattern>
</servlet-mapping>
<servlet>
    <servlet-name>indexservlet</servlet-name>
    <servlet-class>Index</servlet-class>
</servlet>
```

First the URL and the servlet name are declared. After the class is mapped to the servlet.

4.1.2 Programming Code

All needed Java files are contained in the *src* folder. The package structure gives you a good overview of the content.
The concepts of how the code is structured in the files were adapted from the lecture so that it should be easy for you to understand. If you like to change the database configuration you can find the corresponding configuration file in *database/Configuration.java*.

The files are connected as shown in the sequence diagrams of D2/D3:

a) Servlet calls VRApplication.

b) VRApplication calls DBAdapter.

c) DBAdapter calls DBFacade.

d) DBFacade connects to your MySQL database.

Values are returned in reversed order.

4.1.3 Template Engine

The concept for the template engine was already introduced in the lecture. We already prepared the project for you with all the necessary libraries.
You can find all template files in *WebContent/templates/*.ftl*. Additionally, we separated the *Cascading Style Sheets* which are responsible for the look of the website from the HTML code. This makes it easier for you to adapt the defined design to your app. If you like to change some colors or images you can find the files in *WebContent/css* and *WebContent/images*.

Links are contained in the header.ftl. Here you can edit the links for the top navigation. We distinguish between links for guests and staff members. Therefore we introduced a new variable called *navtype*. You have to add your link between the condition brackets of the navigation type in header.ftl. The type for the navigation is attached to the request in the servlet files. If you like to add global links please add them bevor the brackets.

5 Appendix

5.1 Ask questions

If there are any questions please feel free to ask them in your lab or to open a new thread in the Moodle discussion board. Most of the questions occur several times so the board is a good collection of possible questions and corresponding answers.
It is also possible to write us an email but please keep in mind that the response may take a while.

5.2 Feedback

We are not perfect. So if you find any errors in our code or this manual please inform us.
We also like suggestions to improve our work in the next years so please feel free to send them to one of us by mail. We will discuss these points and if possible add them in a future course for Software Engineering.

5.3 Helpful links

- http://tomcat.apache.org - Website of the Apache Tomcat project with a download section and a helpful documentation.
- http://eclipse.org - Website of the Eclipse project.
- http://freemarker.org - Webpage for the used template engine.
- http://www.w3schools.com - Webpage with a nice collection of tutorials especially for web engineering topics.
- http://mysql.org - SQL database server.
- http://apachefriends.org - Development framework with database and administration tool.
- https://www.phpmyadmin.net - Browser-based administration tool for MySQL databases.

List of Figures