

Internet Technologies Project Report

Project name: Tabworld

Student name: Christian Krueger

Student id: 5147

Course: Internet Technologies

Academic year: 2007 – 2008

Table of Contents

Motivation.....	3
Functionalities.....	3
Roles.....	3
Login.....	3
Logout.....	3
Registration.....	3
Browsing tablatures.....	4
View a tablature.....	4
View the comments of a tablature.....	4
Add a tablature or comment.....	4
Edit a tab or comment.....	4
Rate a tablature.....	4
Download tabulature.....	5
Tabs management (edit/delete).....	5
User Management (add/edit/delete).....	5
Contact the administrator.....	5
Download the report.....	5
Technologies used.....	5
HTML, Javascript, CSS.....	5
JSP, Java Expression Language, JSTL.....	6
Servlet.....	6
JDBC.....	6
Java Beans.....	6
YUI.....	6
Thickbox.....	6
Database.....	6
Design.....	7
Database tables.....	7
ER diagram.....	7
Procedure flows of the functionalities.....	7
Main classes and methods.....	11
UserBean.....	11
RessourceBean.....	11
TabBean.....	12
GetTab (servlet).....	12
Major technical problems.....	14

Motivation

I am a motivated guitar player and I like to play my favorite songs. Therefore I look on the internet for the tablatures. It is sometimes very annoying browsing through the web pages, which offer tabs, because most of the tabs are throwaway. Because of this reason I decided to write my own web page offering high quality tabs. The project Tabworld should be an internet portal where motivated guitar players can share their own tablatures of any music song. By rating the several tablatures it should be easy to identify the good ones. The registered users of the web page can also comment and discuss about a tab. By also having managers of the internet portal, which can delete comments and tabs, the quality level can be kept high.

Functionalities

Roles

For this web application are defined 3 roles, which different rights:

- user: it can add tabs/comments and edit/delete its tabs/comments
- manager: it can add tabs/comments and edit/delete all tabs/comments
- admin: it can add tabs/comments, edit/delete all tabs/comments, add new accounts, delete accounts and edit the right of an account

Login

The username and password of an account are necessary to login. After a successful login the user can exploit the web page according to its role. Unlogged users can browse the tablatures and view the tab details, but cannot add/edit or delete contents.

Logout

By logging out the user is unlogged and has no additional rights. Its session gets invalidated.

Registration

A user can also create its own account by following the registration form. After a successful registration it has automatically the role user.

Browsing tablatures

Browsing the tablatures of the web page is allowed for all users. The tabs are presented as a list divided into pages. One page shows only an limited amount of tabs. To see more the user has to turn the pages. Browsing tablatures is also possible by getting the best tabs at the top of the list. So the order is descending according to the rating of the tabs. Normally the list is sorted alphabetically according to the title of the tab.

View a tablature

A tablature can be viewed by clicking on the title of a tab in the tab list. Then the content and the details are shown. A logged user can then add/view comments, rate the tab and download the tab as an html page.

View the comments of a tablature

The comments of a tablature are also organized as a list like the tabs. The comments are sorted according to the creation timestamp. By viewing the contents an edit/delete option is given according to the role of the user.

Add a tablature or comment

Only registered users can add content. A comment can be added by viewing a tablature. A tablature can be added from the menu. The form for adding is the same for both. It is possible to add a title and a content. The content can be inserted trough a rich text editor.

Edit a tab or comment

The form for editing a tab or comment is the same as for adding it. A comment can be edited from the comments view. A tab can be edited from the tabs management.

Rate a tablature

This feature is reserved for registered users. By viewing a tab it is possible to rate it. The rate scale goes from 1 to 5. A user can also change its rate for a tab.

Download tabulature

Registered users can download a tab in the tab view. The download is a html page, since the tab content is saved as html.

Tabs management (edit/delete)

From the tabs management it is possible to edit or delete tabs. Every logged user can manage its tabs. They are again presented in a list with page control. Manager and admins can manage all tabs. The editing form of a tab is again the same as for adding it.

User Management (add/edit/delete)

The user management is reserved for the admins. Starting from the list of all users it is possible to delete them change their right and also adding new users.

Contact the administrator

It is possible to contact the administrator by email.

Download the report.

It is possible to download the report as a pdf file from the menu.

Technologies used

As web server acts the apache tomcat on lamj.inf.unibz.it, which is for java server pages.

My web application is therefore based on java server pages. I use HTML, Javascript, CSS, JSP, Java Expression Language, JSTL (core and function library), Servlet, JDBC and Java Beans. With this technologies the web application becomes a compact structured application.

HTML, Javascript, CSS

HTML, Javascript and css files allows to make static html pages. I used a HTML page for closing the thickbox, where on the body onload event the thickbox gets closed and the whole page reloads. Sure in the JSP pages is also contained HTML code. I use javascript to validate the registration and user addition formular, for showing messages (alert) to the user. Also the thickbox and files of YUI are

javascripts. As content style sheets I reset the browsers default CSS by the one provided of YUI. Every page has in addition a separate CSS file if it has specific style needs.

JSP, Java Expression Language, JSTL

I use jsp files to create dynamic web content and mainly to use java beans, which contain all the data. Using Java Expression language (JEL) in combination with JSTL the Java code in a JSP site is reduced to a minimum and the jsp page is more structured and assessable. With JEL I can access easily the beans and with JSTL I can walk trough lists of users or tabs.

Servlet

The servlets mainly retrieve the data from the database and store it temporarily for the session or request in the beans. Moreover the servlets check if a user has the rights to perform this request or limit the request respectively. In the servlet I also set the HTTP header. For example the status 500 if an error occurred while operating with the database or the status 403 if the user has not the rights to call this servlet. I also provide the download of the tab as HTML file (since the content is saved from the rich text editor as HTML) and set an attachment to the HTTP header.

JDBC

With the Java Database Connectivity I get the connection to the postgres database. Moreover I use some classes, which handle the connection pool.

Java Beans

In the beans are stored the users, tabs, comments and messages, which I pass from servlets to jsp files.

YUI

Further, I have used the Yahoo! user interface library (YUI). I used it to structure the content of my web application, for styling the buttons and the menu and the rich text editor for the input of the tabatures and comments.

Thickbox

For the user management, the tab and comment addition/modification I used the Thickbox 3.1 by Cody Lindley. It is a user interface dialog that is placed on top of the web page. The layer has an iframe in the center and the content below is dimmed, in order to highlight the iframe.

Database

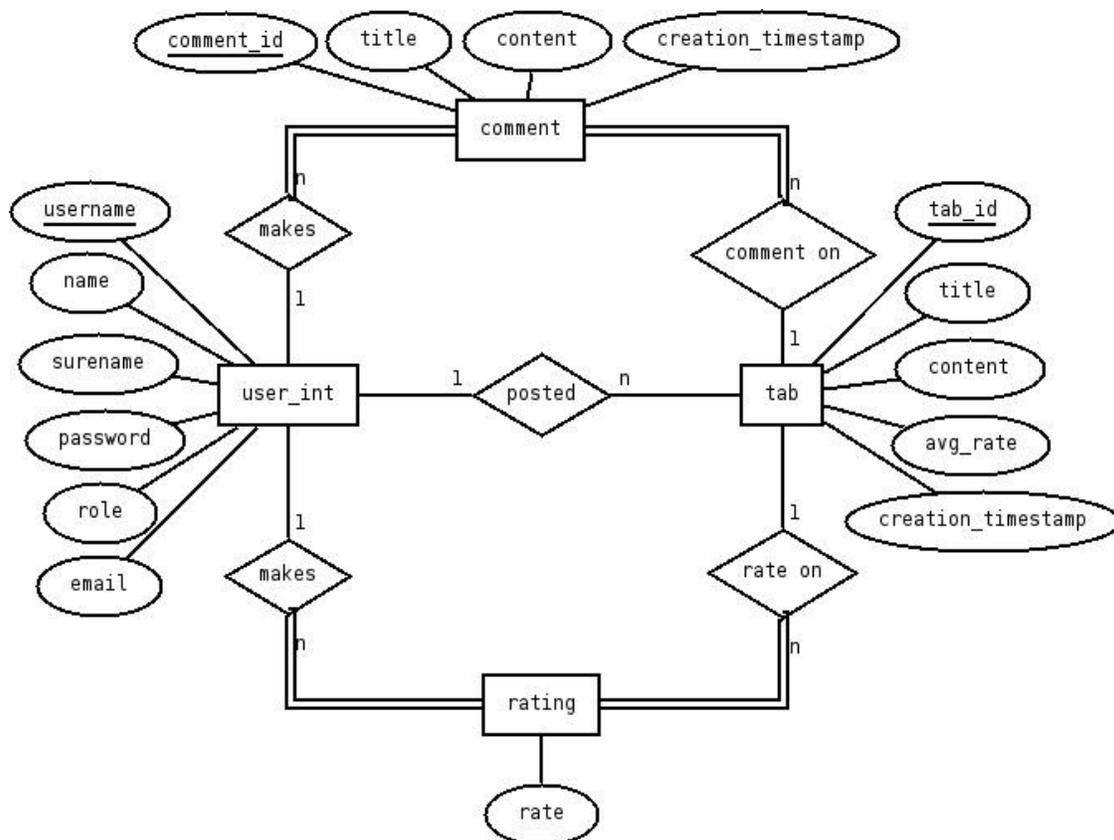
The database is postgres and hosts on alcor.inf.unibz.it. The database consists of four tables namely: user_int, tab, comment and rating.

Design

Database tables

- The table `user_int` is for storing the the users of the web page. A user has a name, a surname, a password, an email, an username and a role.
- The table `tab` is for storing the tabs of the web page. A tab has an id, a title, a content, a creation timestamp, a poster and an average rate, which is calculated from all the ratings on this tab.
- The table `comment` is for storing the comments of the tabs. A comment has a title, a content, a creation timestamp and a poster.
- The table `rating` is for storing the ratings of the users on a tab. A rating has a critic, a tab and a rate number.

ER diagram

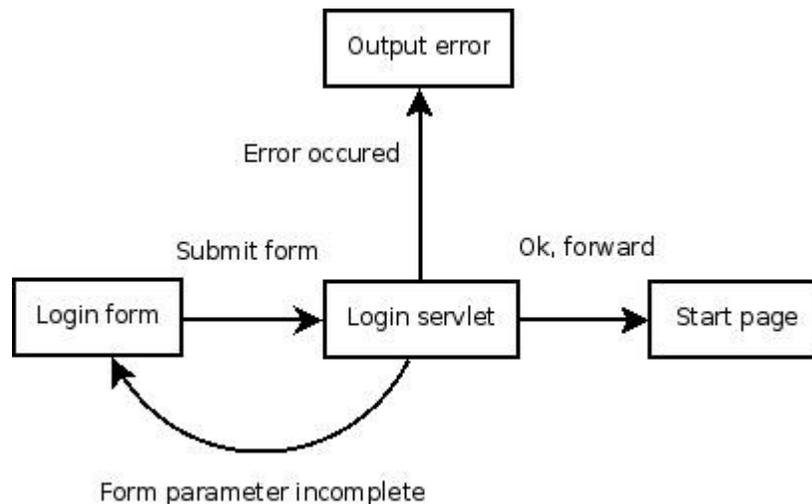


Procedure flows of the functionalities

The sequences (server side) of the actions on the web page are based on the MVC-concept.

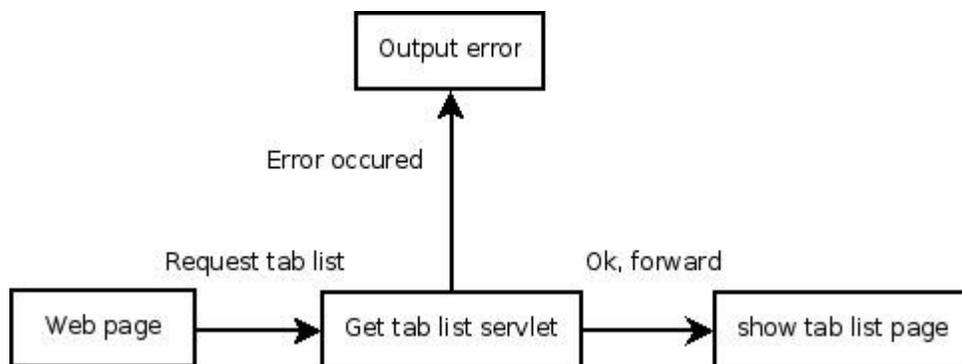
By requesting a page where data has to be retrieved from the database or sometimes also from the beans first a servlet is called. The servlet puts the data into beans or modifies them. On an error, the status is set to 500. If the user has insufficient rights to call this servlet the status code 403 is sent. If the parameters don't fulfill the operation of the servlet (for example the login process), the servlet forwards to the according jsp page and sets the message bean. The jsp pages uses this messages and embeds it into the HTML code in order that the end user gets informed (for example by alert) that something is wrong. If the servlet processes successful, normally I forward to another jsp page, which presents the requested data.

- Scenario type 1: Login



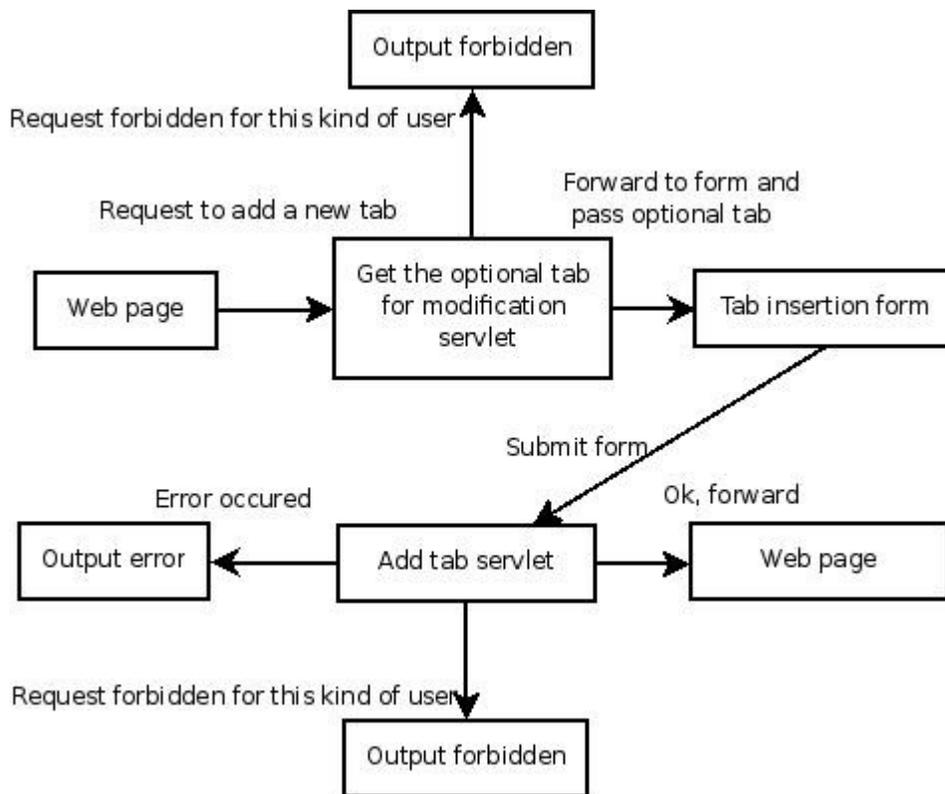
The scenario of this kind is used for the login, the user addition and the registration.

- Scenario type 2: Get the tab list



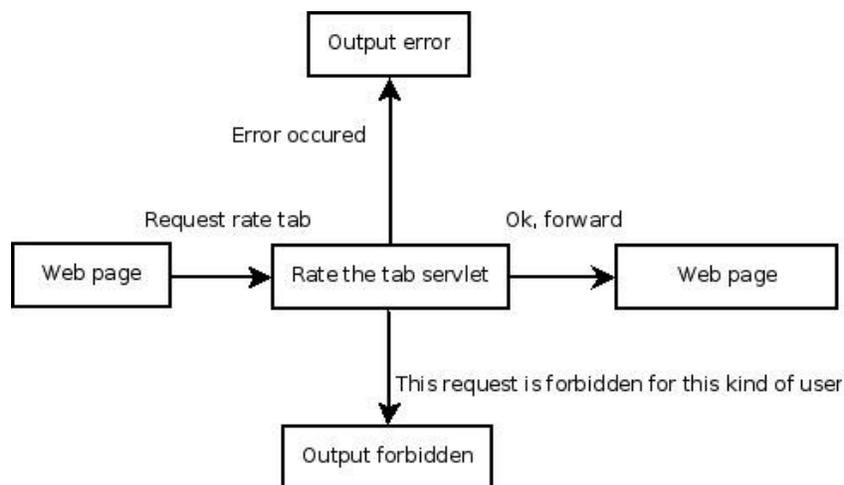
This scenario is used for getting the tab list, getting the comments of a tab and for showing a tab.

- Scenario type 3: Add a new tab



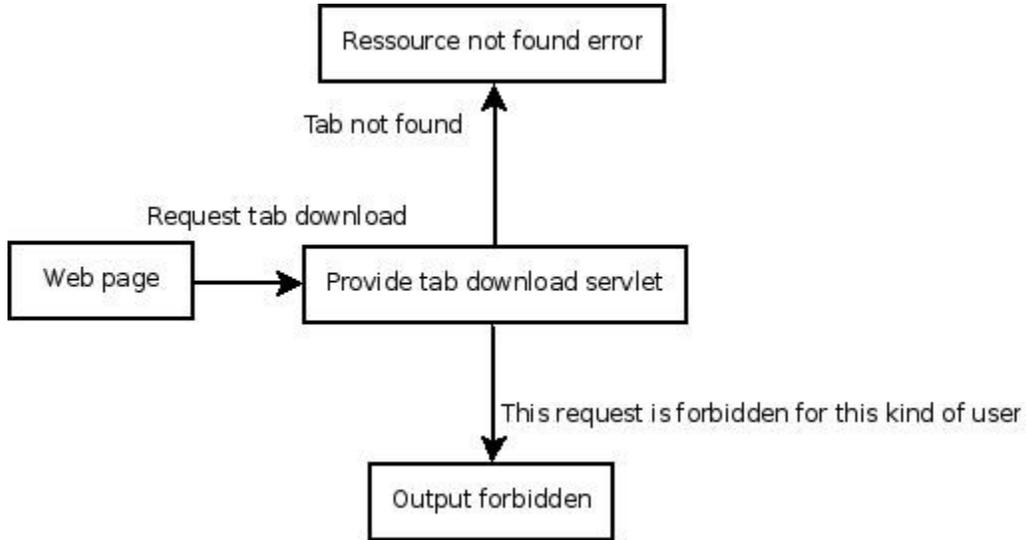
The modification and addition of a tab is very similar, therefore they use the same procedure and the same servltes. If the tab is to be modiflicated it gets the content and puts the content in the tab addition form. Then the add tab servlet updates the content of the tab. By adding a new tab, the servlet for getting the tab to modifcate simply forwards to the form by doing nothing else. So this schema is used for the tab and comment addition/modification and for the right modification trough an admin.

- Scenario type 4: Rate the tab



This scenario is similar to scenario of type 2. But its also possible that the kind of request is forbidden for some user. This schema follow the tab rating, the tabs deletion, the user deletion and the comment deletion.

- Scenario type 5: Download tab



Here the servlet produces the output, which is a file and contains simply the tab content.

.

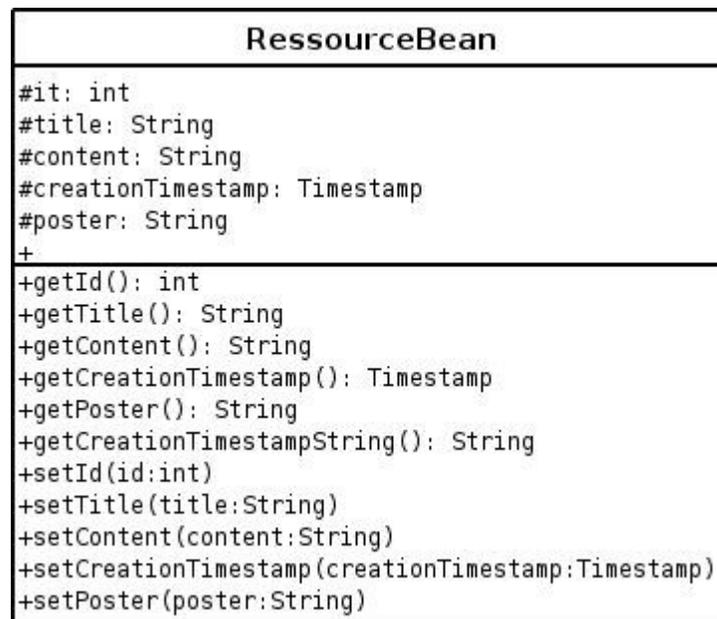
Main classes and methods

The main classes of the web application are the UserBean, the RessourceBean, TabBean and the servlet for getting the tablist and single tab.

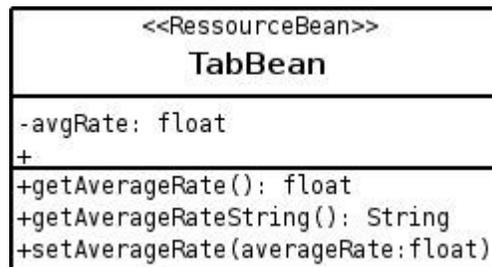
UserBean

UserBean
-username: String -name: String -surename: String -email: String -role: String -loggedIn: boolean +
+getUsername(): String +getName(): String +getSurename(): String +getEmail(): String +getRole(): String +getLoggedIn(): boolean +setUsername(username:String) +setName(name:String) +setSurename(surename:String) +setEmail(email:String) +setRole(role:String) +setLoggedIn(loggedIn:boolean)

RessourceBean



TabBean



GetTab (servlet)

Here is shown the main part of this servlet.

```
private String getTabQuery = "SELECT tab.* FROM tab WHERE tab_id = ?";
private String getRatingQuery = "SELECT rate FROM rating WHERE critic LIKE ? AND
tab = ?";
```

```
id = request.getParameter("id");
```

```
try {
    // Get the database connection.
    reg = (MyRegistrationBean) session.getAttribute("dbRegistration");
    if (reg == null) {
        reg = new MyRegistrationBean();
        session.setAttribute("dbRegistration", reg);
    }
}
```

```
}
connection = reg.getConnection();

// Get tab.
statement = connection.prepareStatement(getTabQuery);
statement.setInt(1, Integer.parseInt(id));
rs = statement.executeQuery();
while (rs.next()) {
    tab.setTitle(rs.getString("title"));
    tab.setContent(rs.getString("content"));
    tab.setAvgRate(rs.getFloat("avg_rate"));
    tab.setCreationTimestamp(rs.getTimestamp("creation_timestamp"));
    tab.setPoster(rs.getString("poster"));
    tab.setId(rs.getInt("tab_id"));
}
session.setAttribute("tab", tab);

rs.close();
statement.close();

// Get rating of the user, if it is logged in.
statement = connection.prepareStatement(getRatingQuery);
statement.setString(1, user.getUsername());
statement.setInt(2, tab.getId());
rs = statement.executeQuery();
while (rs.next()) {
    rating.setRate(rs.getInt(1));
}

session.setAttribute("rating", rating);

rs.close();
statement.close();
connection.close();
} catch (Exception e) {
    // Error occurred during retrieval.
    response.setStatus(500);
    return;
}

dispatcher = request.getRequestDispatcher("./showtab.jsp?View=tab");
dispatcher.forward(request, response);
```

Major technical problems

While working with YUI, the text editor and the radio buttons I had some problems. First I could not set the initial font of the rich text editor, because the tabs should be saved in the font Courier, which is a monospace font and fine for presenting the strings of the guitar and the numbers where to put the fingers. Then the radio buttons, in principal all the buttons from YUI, require some experience for fully understanding and adjusting their style.

Moreover, initially I worked with the database HSQLDB, which was sometimes annoying, because I changed the folder names or moved them and after inserting data and not flushing it, the data was not stored. For some reason I had really problems using prepared statements with HSQLDB, so I changed to postgres, which I am already used to.