Handed out: Thursday November 21, 2013
Due: Thursday December 10, 2013 (turnin on lectura by 11:59 PM)

1 Overview

Your assignment is to create a subset of the MyNotes web application. You are to create three Java servlets and one Java server page (JSP) as described below. You will be working in pairs on this assignment.

We have provided part of the code for MyNotes; you are to augment our code by finishing the following files.

`LoginServlet.java`	is a Java servlet that will allow the users to login to the MyNotes database.
`AddProfileInformation.java`	is a Java servlet that will allow user profile/member information to be added into the MyNotes database by a user.
`FindBoards.java`	is a Java servlet that will list the boards in the database than have more than a user-specified number of subscribers.
`AddCard.java`	is a Java servlet that will allow a user to create a card for a specified board.
`SharedAssignments.jsp`	is a Java Server Page that prints the names of users who are all responsible for the same card.

These files are provided but contain only method stubs. They serve to specify the expected output. This is a project that will require you to learn many new technologies so start this project as soon as possible.

2 Underlying Database

Your database should have the following tables as specified in Handout 19.

```
User
Creation
Board
Card
Subscribes
AssignedTo
```

Additionally, you will need to create three SQL scripts.

`create.sql`	Creates the tables listed above.
`populate.sql`	Populate the necessary tables for your demo.
`destroy.sql`	Drop all the tables.

3 Working Environment

Java servlets need a server to run them. You will use Tomcat. Below we describe how to get

3.1 Installing Tomcat and Setting Environment Variables

Tomcat is running on either the `oxford` or `cambridge` machines. You can use either machine. First, you will need to install an instance of Tomcat. The following link will guide you through the installation. Note that this takes place on `oxford` or `cambridge` rather than `lectura`.

http://faq.cs.arizona.edu/index.php?action=artikel&cat=9&id=86&artlang=en

You need to set the following environment variables (note that the directions below assume the bash shell; if you are using csh/tcsh then substitute export = with setenv) .

```
export JAVA_HOME=/usr/local/jdk

export CATALINA_HOME=/home/YourLecturaUserName/tomcat7/YourInstanceName

export ORACLE_HOME=/usr/local/oracle/app/oracle/product/11.2/client/oui/jlib/classes12.jar

export SERVLET_HOME=/usr/share/tomcat7/lib/servlet-api.jar

export JSP_HOME=${CATALINA_HOME}/lib/jsp-api.jar
```

You need to set the PATH variable to include the web applications. You will do option 1, below, if you have already defined an environment variable PATH . You will do option 2 if you have not already defined a PATH environment variable. Make sure you don't overwrite PATH!

Option 1:

```
export PATH=.:${PATH}:${CATALINA_HOME}/webapps/MyNotes/WEB-INF/classes/
```

Option 2:

```
export PATH=.:${CATALINA_HOME}/webapps/MyNotes/WEB-INF/classes/
```

You need to set the CLASSPATH variable to include the Oracle drivers and servlet-api.jar . Again, you will do option 1, below, if you have already defined an environment variable CLASSPATH . You will do option 2 if you have not already defined a CLASSPATH environment variable.

Option 1:

```
export CLASSPATH=.:${CLASSPATH}:${ORACLE_HOME}:${SERVLET_HOME}:${JSP_HOME}:${CATALINA_HOME}/webapps/MyNotes/WEB-INF/classes/
```

Option 2:

```
export CLASSPATH=.:${ORACLE_HOME}:${SERVLET_HOME}:${JSP_HOME}:${CATALINA_HOME}/webapps/MyNotes/WEB-INF/classes/
```

Once you have defined these environment variables, you should be able to compile your Java programs on oxford or cambridge using javac . You may want to put these commands in your /home/*YourLecturaUserName*/.profile file. Otherwise, you will to set these variables manually every time you start a new shell or log in to oxford/cambridge .

3.2 Getting Started for MyNotes

Now you need the stubs for the MyNotes files. The MyNotes application directory is located at http://www.cs.arizona.edu/classes/cs460/fall13/resources/hw8_MyNotes.tar .

The tar file will contain a directory called MyNotes. Place this directory in *YourTomcatPath*/webapps/.

3.3 Building Java Classes

Before you can run the web application, you must make the project. In the *YourTomcatPath*/webapps/MyNotes/WEB-INF/classes directory, you will find a Makefile . Go to that directory and execute make . Once you start developing your solution you can assure yourself that you have removed all classes' files by typing make clean . Every time you modify a .java file you should execute make .

3.4 Starting/Stopping Tomcat

The command to start Tomcat is *YourTomcatPath*/bin/startup.sh . The command to stop Tomcat is *YourTomcatPath*/bin/shutdown.sh .

To access your application, type the following URL into your favorite browser after substituting your unique port number. That is, type in http://localhost:*YourUniquePortNumber*/MyNotes

Keep in mind each time that you make changes to your web application and re-make, you should stop and restart Tomcat again. **This is very important!**

You can read other provided servlet examples to understand how to create your servlets and JSP page. If you followed the instructions for setting up Tomcat and copied the examples directory, you can type http://localhost:*YourUniquePortNumber*/examples/ in your browser to see some examples.

Also, remember to modify the OracleConnect.java with your username and password for your Oracle account. This file is located in *YourTomcatPath*/webapps/MyNotes/WEB-INF/classes/MyNotes/utils .

4 Your Assignment (100 points)

You are to implement the MyNotes login, add profile information, add card, and find boards functionality in the MyNotes system. You will also implement a JSP report to display all the users who have share responsibilities for a card with another user.

Notice that you are given two packages: classes/MyNotes/utils and classes/MyNotes/servlets .

You only need to modify one file in MyNotes/utils, OracleConnect.java. In that file, you need to change the password/user name to your *Oracle* user name/password (different from your lectura password!). All servlets must reference this.

Modify LoginServlet.java, AddProfileInformation.java, AddCard.java, and FindBoards.java in the package MyNotes/servlets described below. You also need to modify MyNotes/JSP/SharedAssignment.jsp as described below.

You must validate all input from users. Do not trust the user to enter the correct data type or length of data in each field. Handle these errors gracefully by redisplaying the same page, along with an error message (formatting is up to you). The exception to this rule is whether or not date entries make sense.

Your task is to implement the following functionality.

4.1 Login/Register to MyNotes (**LoginServlet.java** and **AddProfileInformation.java**)(25 points)

Your servlet should prompt the user to enter the e-mail and username of a registered MyNotes user to login. If the e-mail and username the visitor provided is authorized by MyNotes (by checking the User table), the user is allowed to access MyNotes.

However, if the e-mail and username does not identify a registered user, the user is not allowed to log into MyNotes, but is allowed to retry the login process, by redisplaying the prompt.

If the e-mail does not exist, display the page with the following error message.

```
Error:  e-mail does not exist.
```

If the e-mail exists but the username is wrong, display the page with the following error message.

```
Error:  Enter the correct username.
```

One method stub of invalid user login can be found in LoginServlet.java but you must add some more specific error messages.

You can use the session JSP implicit object to check whether the user is logged in. See slide T-71 for an example in JSP. To get the session in a servlet you use request.getSession() .

Alternatively, the user can create their account by using the "Add Profile Information to MyNotes" button on the login page.

If the e-mail already exists in the database, then redisplay the same page with the error message:

```
Error:  User already exists!
```

If the user is successfully added, display the new user's information with a message to this effect.

```
New user added!
```

After successfully logging in, a menu is displayed with three buttons: "Add a Card", "Find Boards by Number of Subscribers", and "Who is assigned to the same card?". Each of these link to another page.

4.2 Add Card Information to MyNotes (`AddCard.java`) (15 points)

Your servlet has a form for the user to enter the board name, task name, day, month, and year of a card. This activity should be added to the database for that user.

4.3 Find Boards by Number of Subscribers (`FindBoards.java`) (30 points)

Your servlet prompts the user to enter an acceptable number of subscribers. Clicking the "Find" button should display the boards in the database with that many subscribers or less.

If no board satisfies the specified search criterion, redisplay the same page, but along with the the message

```
There is no board with few enough subscribers for you.
Please select another amount and try again.
```

If there exist boards that satisfy the search criterion, redisplay the same page, and include the names of the boards. There is a button to return to the menu.

In case of any out of range errors, display an error page indicating that the request could not be carried out. This error page must have a button which allows the user to return to the add profile information main page. The error message should be:

```
Error:  Request could not be carried out.
```

4.4 Who is Assigned to the Same Card (`SharedAssignment.jsp`) (30 points)

`SharedAssignment.jsp` should display the names of all cards and the users that are assigned to it. The sample format is given in the provided servlets. There is a button to return to the menu.

5 Debugging a Servlet

Web applications are (unfortunately) notorious for being very difficult to debug. The simplest way to debug applications is to use gratuitous and meaningful `System.out.println` calls. However, since you are viewing the output of your code in a browser as a web page you need to know where the output from a `System.out.println` goes.

Under your main Tomcat directory, one of the directories you will see is a `logs` directory. In this directory, Tomcat keeps several files of logging information. The one that you should mainly be concerned with is `catalina.out` . This is where the output from your `System.out.println` will print along with any exceptions that occur while your program is running. This log is cumulative. Between starts/stops of Tomcat we suggest that you remove this file. Then, when you start Tomcat again it will recreate `catalina.out`, only with the newest logging information.

6 Extra Credit (Up to 40 Extra Credit Points)

You may implement any additional functionality in the MyNotes system to enable more functionality for the webpage. Also feel free to add more "cool" features. You may add additional tables and webpages. However, none of the new features should contradict any of the specifications given above. Make sure your basic assignment works before expanding. Points will be awarded based on the complexity of your additions and their correctness. Significant work must be done. A maximum of 15 of the extra credit points will be awarded for the *professional look* of your solution. A maximum of 25 of the extra credit points will be awarded for *new functionality* .

7 Helpful References

The following are some links to helpful webpages about servlets. HTML, and JSP.

The servlet API is not part of the standard Java API. Here is a link to it:

`http://java.sun.com/products/servlet/2.2/javadoc/index.html`

Here are some very helpful pages about servlets and JSPs. they contain lots of explanation and examples.

`http://www.coreservlets.com`

`http://www.moreservlets.com`

Here is a decent page that has to do with HTML tags. There are tons of similar pages out there so if you don't like this, we are sure you can find some that you do like.

`http://www.w3.org/Markup/Guide/`

8 Turnin and Grading

You need to turn in the following files: `index.html` (the first page), `LoginServlet.java`, `AddCard.java`, `AddProfileInformation.java`, `FindBoards.java`, `SharedAssignment.jsp`, `create.sql`, `destroy.sql`, `populate.sql` and a `README_hw8` file that describes anything else we should know about your program. If you implement the extra credit be sure to include this information in the `README_hw8` file. Once you have tested your program and are sure you have everything running the way you like it, issue the `turnin` command from `lectura` for the turnin folder `460.hw8` .

We will be grading your assignments using Tomcat on `oxford/cambridge` . Make sure that everything runs there before turning in your files. You should include any additional files (if you did any extra credit work) and explain what you implemented (i.e., if you did only the regular assignment or did extra credit) in the `README_hw8` file. Be sure to brag about your project in this file. If you feel that a feature you provided deserves extra attention, please record this information here. If your extra credit addition requires other data or tables, please turn in a separate file `extra.sql` for these and mention this in `README_hw8` .

If your extra credit requires you to turn in more files than those listed above, indicate this in the `README_hw8` . You should add all new servlets to the `MyNotes.servlets` package and all new JSPs to the directory that already contains JSP files. If you add more servlets for the extra credit part be sure to turn in your modified `WEB-INF/web.xml` file. Also, make sure that the `Makefile` compiles all files required to run your web application. If you have to change the `Makefile` for extra credit, also turn it in. If you do not follow these directions, you may not receive extra credit.

Grading will be different for this project. You will be scheduling a demo within a given time slot (a schedule sheet for signing up will be handed out later in class). In this demo, you will show us your MyNotes application running in your directory and demonstrate its functionality as required for this assignment. Do not change the files after you have turned them in, as we will check the timestamps. Both members of the pair must be present at the demo.