Assignment 3

Database Integration

Points: 100

Due 03/29/16 EOD

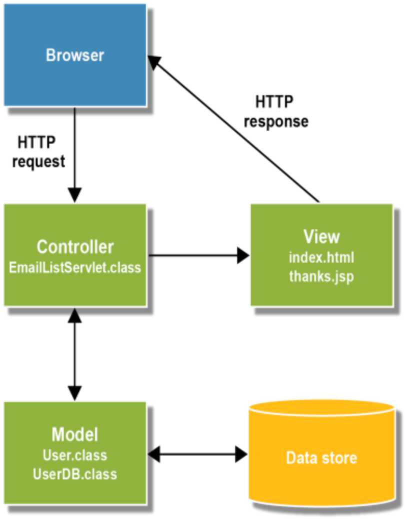
This assignment is to be done by a team of two students. Your first step of the assignment is to ensure that you have documented your team membership to course staff in the manner your instructor has specified. There is a fair amount of work to be done, but there are two people working the assignment together. You should establish a sensible way to share your project development work. You **MUST NOT** make your assignment code available publicly; as this would have to be treated as a violation of UNCC academic integrity policy. So do not use a repository service such as GitHub where all projects are publicly viewable – any such repository must be private.

This assignment is intended to familiarize you with using databases to persist the data in your web application. The assignment involves the use of servlets for the controller and JavaBeans for the business logic and sessions to maintain state. This assignment is an extension to the previous assignment. To be able to meet the requirements for this step successfully you are encouraged to resolve any issues or missing requirements from Assignment 2. To start, you should coordinate with your team member(s) on which submission your will use as a starting point for this solution.

Assignment Description

In this assignment you will replace the hard-coded "database" with an actual database, which entails some modifications to your previous JSP/Servlet MVC web application, according to the following specifications:

1. All structure, design, and content requirements from previous assignments are mandatory, unless explicitly updated in this assignment description.
2. Use JavaBeans to implement the business layer of the application (**model**).
3. Use JSP pages to present the **view** to the browser.
4. Use Servlet pages to **control** the flow of the application.
5. Functionality that does not follow the assignment specifications will not receive credit.



JavaBean Creation

Create JavaBeans for the following Model element, with the specified instance variables and additional methods. Bean class names MUST use the specified names.

Reported

- StudyCode
- Date
- NumberOfParticipants [number of participants that reported this study]
- Status [Under Review | Approved | Disapproved]

Database Creation

In addition to the standard WAR file for this assignment, you will submit a text file (plain text, NOT Word or RTF or PDF or any other fancy document format). This file will contain all of the SQL statements that you use to (1) create, and (2) populate your database. Your script will look very similar to the first part of the create_databases.sql script used to create the murach database used in the textbook exercises. (See the example table creation and data insert statements there for the murach database User and Download tables). Your database script file must be called:

nbad3_create_db.sql

Create a database in MySQL to hold your application data. With this assignment you are provided with a **template** database creation script “nbad3_create_db_start.sql” that you can use to start out. All of the columns listed here should be in your table definitions, but you may need to modify them to reflect the naming scheme and data types you used in your JavaBeans (Model) from the previous assignments.

Database Population

1. Your database may be populated however you like. Here are three options:
   - Create an insert script for the three tables and run against the database in your SQL client.
   - Create a populateDatabase method in one of your DB helper classes. You can use some of the code you wrote to create the JavaBeans in assignment 2, but use your “add” methods” to save them to the database. An unlinked admin page can be created with a button to run this code.
   - Manually enter data in MySQLWorkbench (not recommended, but will work).
2. Your database should be populated with data according to the same requirements from the previous assignments.
3. Within the same script file for database and table creation (nbad3_create_db.sql), add statements at the end of the script to insert your data items into all tables.
4. Running the nbad3_create_db.sql script should re-create / reset your database to its initial state with initial data.

Code Changes/Additions

- UserDB – Refactor class “UserDB” to add/update users records in the database:
  - User getUser(String email)
  - validateUser(Email,Password)
  - addUser(User usr)
  - updateParticipations()
  - updateUserStudies()
- StudyDB – Refactor class “StudyDB” to add/update studies in the database:
  - getStudy(String SCode) - returns a Study object for the provided studyCode
  - List/Collection<Study> getStudies(String criteria [“Open”, “Closed”, “All”) - returns a set of all the studies in the MySQL “database”
  - List/Collection<Study> getStudiesFor(String email) –returns a set of all the studies in the MySQL “database” for the passed-in email.
  - addStudy(Study study) – add a new study to the MySQL database – Study table.
  - updateStudy(String SCode, Study study)
- Create a new AnswerDB class to add/retrieve answers from the database:
  - AddAnswer(String SCode , Answer ans) – adding a new answer to the MySQL database – Answer table.
  - List/Collection<Answer> getAnswersFor(String SCode) – returns a set of all the answers in the MySQL “database” for the passed-in study code.
  - List/Collection<Answer> getAnswersFor(String email) – returns a set of all the answers in the MySQL “database” for the passed-in participant email.

Add as many methods as you want to the above two classes. Feel free to share your ideas with regard to adding new methods with your classmates at the general discussion forum.

HTML5 Validation/Responsiveness/Cross-browser testing:

Validate your HTML5 on http://html5.validator.nu/. Also, make sure your app works (and looks right) in both Google chrome and Firefox. Check the document in the moodle page on how to conduct the HTML5 validation (Resource Links >> How to? HTML 5 Validation).
Add responsiveness to our web pages by following the tutorial in the moodle page (Resource Links>> How to make your website responsive?).

Deploying your database into openshift

In week 6, we posted a tutorial on how to deploy your web project that integrates a MySQL database into openshift. Please refer to that tutorial for deploying your project.

Extra Credit (20 points):

You will receive up to 20 extra points on this assignment if you implement the email feature for the Recommend and Contact pages.

Assignment Submissions

What to submit using Moodle (Email submissions will NOT be accepted):

1. **firstname_assignment3.war** - An archive of the entire web application (project) stored in a standard WAR file. The WAR file will be imported into Netbeans for grading. **You must ensure that the java source files are included as part of the archive**. Refer to the "How to create a war file in Netbeans" document available on the course's Moodle page under the Resources section.
2. **nbad3_create_db.sql –** SQL script file to create and populate the tables for this project.
3. **info.pdf** – PDF document with the following assignment information :
    a. Indicate the assignment 2 submission that was used as a starting point for this solution
    b. Provide the credential used to login into your project for user with a participant type and a user with an admin type.
    c. Explanation of status and stopping point, if incomplete.
    d. Explanation of additional features, if any.
    e. Discuss the easy and challenging parts of the assignment. How did you overcome all or some of the challenges?
    f. Discuss division of labor specifying who did what and why this is a fair and equal split.
4. **Openshift link** – Upload your project into openshift and post the openshift link in the online text section of the Moodle submission page.

Demonstration

You will be contacted by a course staff member to setup a time for your team to demo your work for this assignment. Students should be prepared to answer questions posed by the grader about their work. Failing to demonstrate the assignment to the grader may result in no credit for all team members.