Assignment 2

COMP 3095 – Java Web Application Development

Due Date: Monday, December 5th, 2016 (4:00 pm)
Team Size: 2 – 4 Team Members (**group assignment mandatory as per syllabus**).

Problem statement:
Having submitted your first assignment, your manager is presently surprised in your capabilities. The company directors have taken note to your skill-set and have received excellent feedback from the clients on your very first project submission.

With the aid of your assignment 1, the clients have come back to your managers and business analysts (BA) with more requirements. These new requirements have been gathered by your BA's and after a number of meetings, have been conveniently tabulated into an updated business requirements document. The summary of these new requirements have been summarized below.

Your manager continues to be under direct pressure from the company Director to produce the final product that essentially meets all client requests. Your manager, fully recognizes your potential, has much more confidence in you given the success of the first prototype. With your mangers trust, your manager is now allowing you to lead the remaining portion of this project to completion.

Please note, for Assignment 2, you are required to build upon and extend the original assignment (Assignment 1). Again, you will continue to use, Java 8, Eclipse and Tomcat (version 8). Your assignment **must** contain the following items described below. Don't let your manager down and your initial hard work go to waste. Continue to build a strong and reliable reputation for yourself.

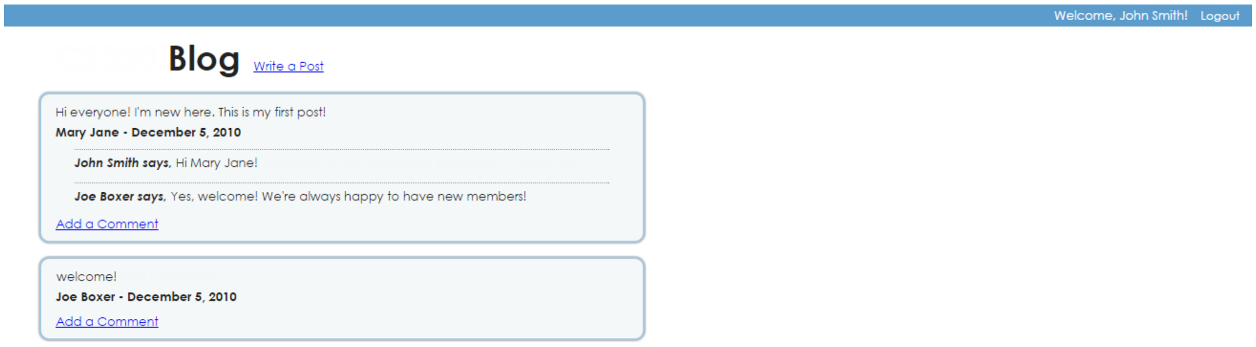
Welcome Page

In this assignment you will be implementing the following functionality:

- View all blogs
- Add new Blog Post
- Comment on a Blog Post

The functionality should utlize the **MVC programming paradigm** in conjuction with a relational database (MySQL). Begin by adding a Welcome page to your Assignment 1. The Welcome page is to display all Posts on the site with the newest post (ie. more recent timestamp) displayed first.

The sample wireframe below helps clarify:



Hi everyone! I'm new here. This is my first post!
Mary Jane - December 5, 2010

John Smith says, Hi Mary Jane!

Joe Boxer says, Yes, welcome! We're always happy to have new members!

welcome!
Joe Boxer - December 5, 2010

Figure 1: Blog Wireframe

Post Page

When a User clicks the 'Write a Post' link near the **top of the page**, they should be re-directed to a page similar to the following wirframe:

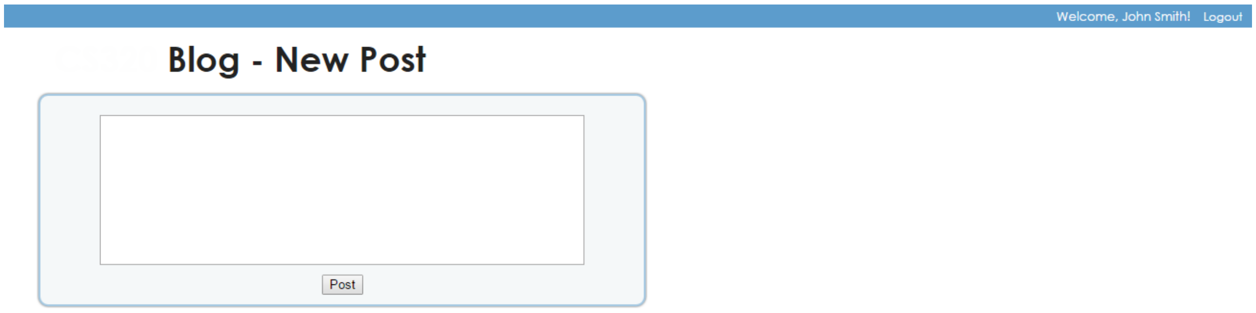


Figure 2: New Post Wireframe

The User will type their post in the provided textarea. Upon submitting the post, your application should validate the post. The post should be considered valid if the length of the text (excluding white-space) is greater than zero (0). If the post does not validate you should re-display the Form with an error message. If the post validates, your application is to store the post to a MySQL database, and redirect the User back to the main Welcome page. The main welcome page should now display the User's post along with all other posts on the site.

Your implementation should consist of a WelcomeController, Welcome View Page, NewPostController, and NewPost View Page. All data must be stored and retrieved from within you backend java code from you database table (ie: MySQL database).

Comment Page

Users of your Blog may want to comment on posts made by other Users. To do so you must first implement "Comment" functionality. Begin by displaying a hyperlink below every post that reads 'Add a Comment'. Use the Welcome page wireframe (above) as a reference for the layout of the link.

When a User clicks on an **'Add a Comment'** link, they should be re-directed to an **Add A Comment** page.

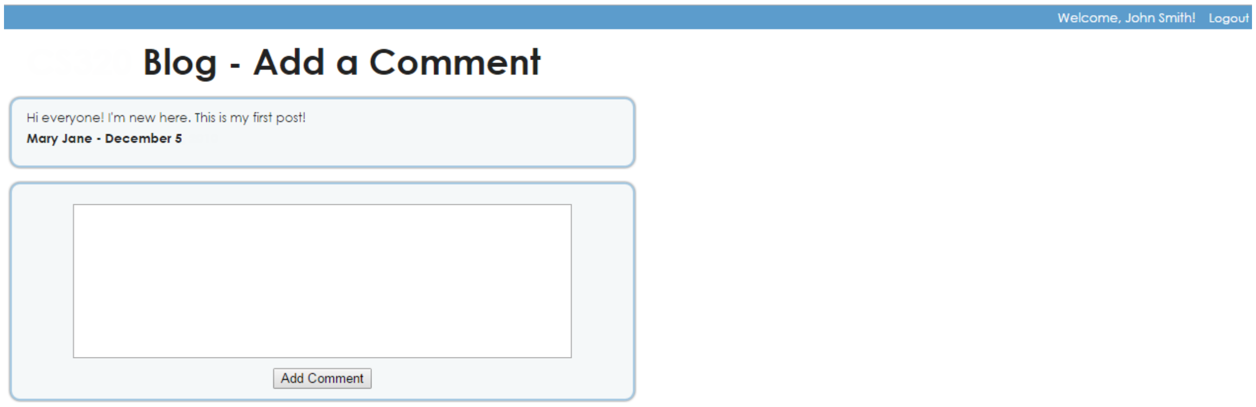


Figure 3: Add a Comment Wireframe

Notice that the page displays the Blog Post that the User is commenting on, as well as a Form with a textarea where the User can enter their comment text. Upon submitting the Form, your application should validate the comment. If the length of the text (excluding white-space) is greater than zero (0), then the comment should be considered valid. If the comment does not validate, re-display the Form with an error message. If the comment validates add it to a MySQL database and re-direct the User back to the main Welcome page. The main Welcome page should now display the newly added comment below the Blog Post that was commented on.

Your 'Comment' implementation should at a minimum consist of a CommentController and an appropriate view. Once again, all data should be retrieved and stored from a MySQL database.

Data Persistence

You are required to store all Posts and Comments in a MySQL Database (read/write to a file). Please note, you are also required to provide the appropriate sql seed scripts if needed (no sql script handed in, means no credit).

Specifically, for this second assignment, you must provide a single **assign2.sql** script that both creates and populates the database/schema for your assignment. Please note, again these script should create the complete database schema and database tables and seed them appripriately.

Please utilzie the following for your data persistence configurtions:

- Utilize username: **root**, password: **admin to** create connections with your database**.**
- Utilzie a schema name of **comp3095**

Important Assignment Guidelines:

1. The project name and project navigation must be strictly followed as described above.
2. You must email your assignment to your manager (Professor Santilli) via blackboard.
3. Within the body of the email, clarify course code, team name, team members and student numbers. Title the email accordingly **COMP 3095 – Assignment 2**.

   Example:
   COMP 3095 - Assignment 2

   Course: COMP 3095
   Team Name: The Hackers
   Team Members: John Smith - 1234567
   Sally Jones - 7654321
   Jane Wilson - 2342342
   Bilbo Baggins - 353453

4. When emailing, cc' a copy to yourself for backup and time verification.
5. The uploaded compressed file must be in **.rar** or **.zip** format.
6. The contents of the compressed file must include:
   1. **.war** file of the project (with **source code** included, no source, means no credit)
   2. **scripts** folder (only if applicable) – .sql scripts located inside.
7. The .zip/.rar file naming convention as follows:

   COMP3095_**YOUR_TEAM_NAME.**rar or COMP3095_**YOUR_TEAM_NAME.**zip

   Example: COMP3095_The_Hackers**.**zip

   *where **YOUR_TEAM_NAME** should be replaced with your team name in the company.

8. Each java file (.java) should include a header.

```
//*************************************************************************************
******************
* Project:           < project name ... >
* Assignment:        < assignment # >
* Author(s):         < author name ...>
* Student Number: < student number ... >
* Date:
* Description:        <describe the java file and its purpose briefly only – 1 or 2 lines>
**************************************************************************************
*****************//
```

9. Your code should be <u>modular</u> and should show no signs or dry (don't repeat yourself) code.
10. You are required to devise and use a form of persistent data storage.
11. You must use MySQL as your data storage for persistence.
12. The necessary **assign2.sql** file must be provided and included within the uploaded compressed file. There should be only **<u>one</u>** .sql file, for schema creation, table creation, and seed data (if necessary).

13. Test your deployments.
14. Your manager should need only to deploy your application (run .sql script if applicable) and be able to use and evaluate your application immediately (navigating to the pages described). Your manager will be somewhat disappointed if otherwise.
15. Be cautious **DO NOT** share your application with others. Complete failures will be assigned if code is shared. All assignments will be reviewed and analyzed strictly within these regards.
16. Late assignments are assigned a penalty of 10% per day.

Marking Scheme:

Functionality	70%
Coding Style	15%
Design / Usability	15%