COMP 3095 – Java Web Application Development

Due Date: Monday, October 23rd, 2017 (8:00 am)
Team Size: 1 – 6 Team Members.

Problem Synopsis:
Having recently graduated from George Brown College, you've recently been offered and accepted a position at a lucrative start-up company. The company consists of a number of experienced engineers and developers, who believe a new graduate, such as you, will bring fresh and innovative development ideas to the organization.

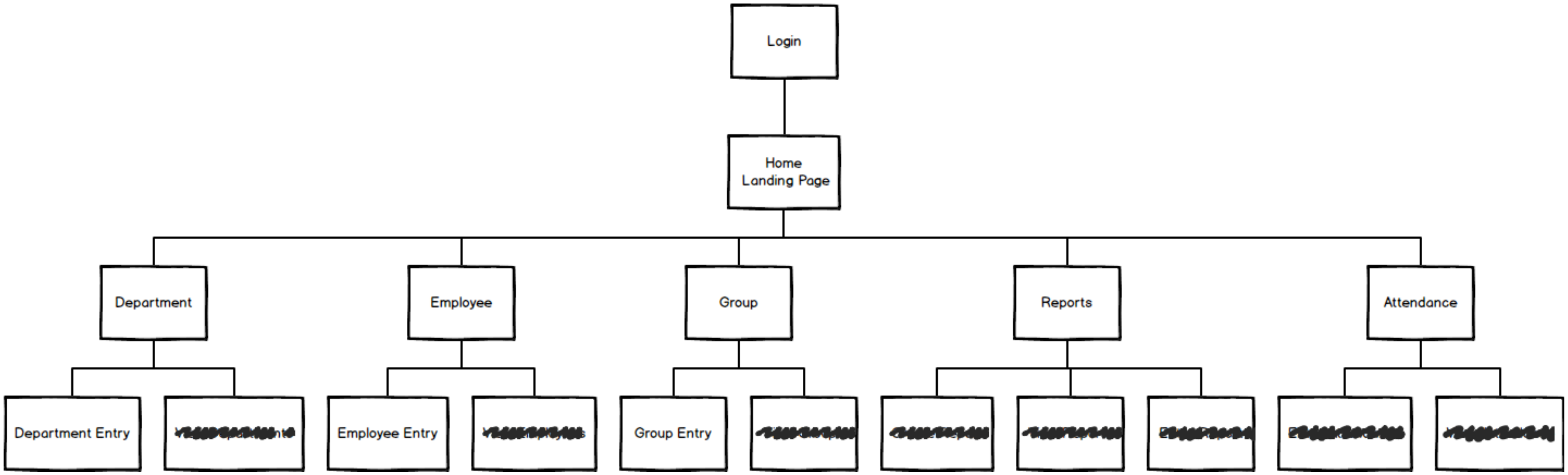
Thought the company you now work for is a startup, its employee base and size have been growing dramatically. Due to the growth in the company, the IT Director has identified the need for an internal web application that can manage and track employee resources more efficiently. An application that will fundamentally track employee attendance, give insight into employee/departmental/group performance and fundamentally act as a one stop shop for personnel reporting.

To start, your manager has been assigned the initial responsibility of assembling a capable team to construct the preliminary phase of the web application. The project has a limited budget, and as such, the IT Director, Management team and Architects all agree, that utilizing Tomcat (version 8), Java (version 8) and Eclipse would be the most attractive and cost effective strategy.

Your manager recalls during your interview process, that you had mentioned you know Java, Servlets and JSPs (at a later time). Giving your knowledge of these technologies, you immediately appeal as a great candidate for the project. Additionally, by selecting and adding you to the project initiative, your manager is presented with the opportunity to evaluate how you preform, while dually gauge your fit in future project assignments within the organization.

The initial project deliverables have been scoped-out for and your team ahead of time by the Business Analysts. Wireframes have been constructed and provided to assist and help further help elaborate on the project desirables that the solution must contain at a minimum.

The Proposed Web Application Site Map



Page Summary

Title	Brief Description	Phase
Login	Page where admin user logs in. Username and password authentication level.	Phase 1
Home Page/Landing Page	Single page that appears in response to a successful admin login.	Phase 1
Department	Department landing page	Phase 1
• Department Entry	Form page to create and enter a new department.	Phase 1
• [redacted]	To be implemented in a future phase	
Employee	Employee landing page	Phase 1
• Employee Entry	Form page to create and enter a new employee.	Phase 1
• [redacted]	To be implemented in a future phase	
Group	Group landing page	Phase 1
• Group Entry	Form page to create and enter a new group.	Phase 1
• [redacted]	To be implemented in a future phase	
Reports	Reports landing page	Phase 1
• [redacted]	To be implemented in a future phase	
• [redacted]	To be implemented in a future phase	
• [redacted]	To be implemented in a future phase	
Employee Attendance	Employee landing page.	Phase 1
• [redacted]	To be implemented in a future phase	
• [redacted]	To be implemented in a future phase	

Login

Like any online application, a user must first login before they can actively start using it. Naturally, the proposed web application you will be constructing must too have a login page that will authenticate a user, utilizing a forwarded username / password pair.

When a user successfully completes login, the general recommendation is that your code creates the necessary object(s) and stores such objects so as to be accessible from any servlet on your site. A user object, for example, should contain the following properties a First name, Last name, Email, Role, Username, and Password

Login Servlet

The Business Analysts have provided a wireframe mockup of a potential login page below:

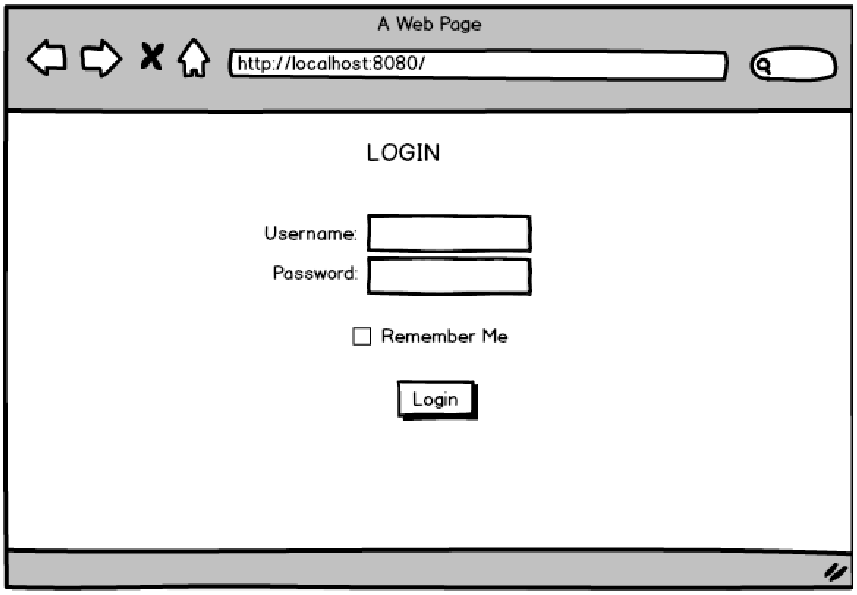


Figure 1: Login Page

The login servlet will prompt the user for the necessary credentials (username / password) to login. It will also contain a “Remember Me” check box that will allow the user to remain “logged in” even after they have closed their browser window. In summary the page will allow the user to submit their credentials and thus login.

The LoginServlet will also be responsible for displaying error messages in the case of the following two (2) events:

- Invalid username and/or password combination is provided
  - occurs when a user enters an invalid username/password combination.
- A User requests a page that requires a user to be successfully logged in first
  - occurs when a user explicitly enters the URL of a restricted page without first authenticating using the Login Servlet.

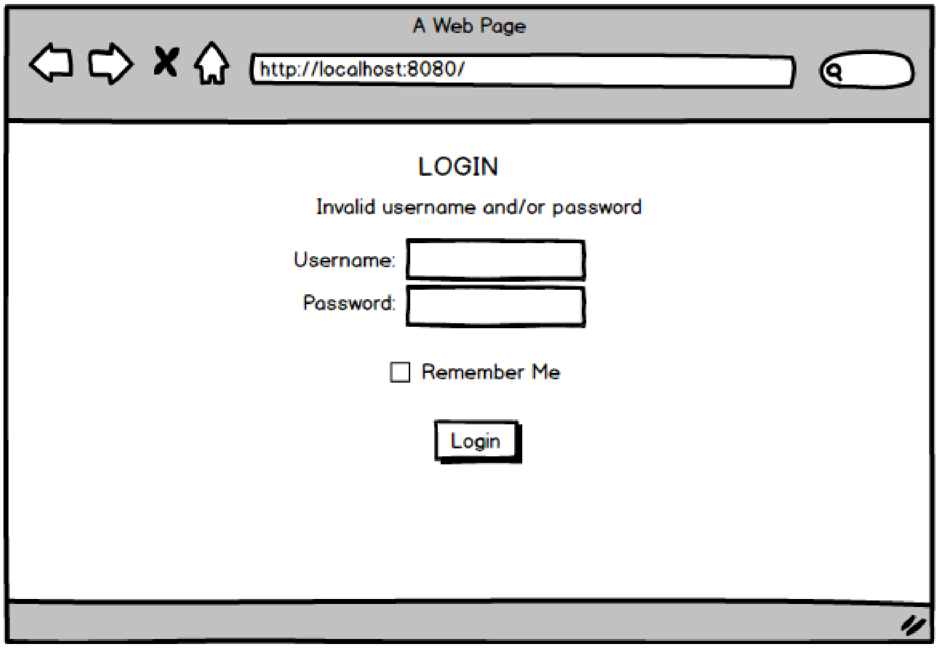
Note there is **no** registration for this web application, rather a user gains access to this application, via an explicit entry inserted into the applications User database. This inserts is typically performed by the application development team during the setup and configuration of the project. For simplicity, at present, the site will only recognize one valid user, a user with the following credentials, username=**admin**, password=**admin**.

Authenticate Servlet

The Authentication Servlet is primarily used for validating a forwarded username and password against a list of recognized users. This servlet will not generate any HTML. The servlet expects two parameters (a username and password). If any mandatory parameters are missing, the servlet should redirect the user back to the login servlet without displaying any error messages.

If the user authenticates successfully, the servlet should store the user's information in a session object and redirect the user to the Home / Landing Page.

Alternatively, if the user fails authentication, the servlet should redirect the user back to the login servlet with an appropriate error message, educating the user of the failure, similar to the wireframe below:



LOGIN

Invalid username and/or password

Figure 2: Login Error

If the third (optional) parameter (Remember Me) is set to true, the servlet should implement some facility for remembering the user even after the user's browser has been closed.

Department

All organizations have specialized areas within them that handle specific verticals of the companies work, called Departments. Departments such as accounting, Marketing, Planning, Application Development, IT, Support etc.… exist to house and oversee related and associated tasks. Your application must allow for the entry of the different departments within your company’s growing organization.

Department Entry

The Business Analysts have provided a wireframe mockup of a potential department entry page below:

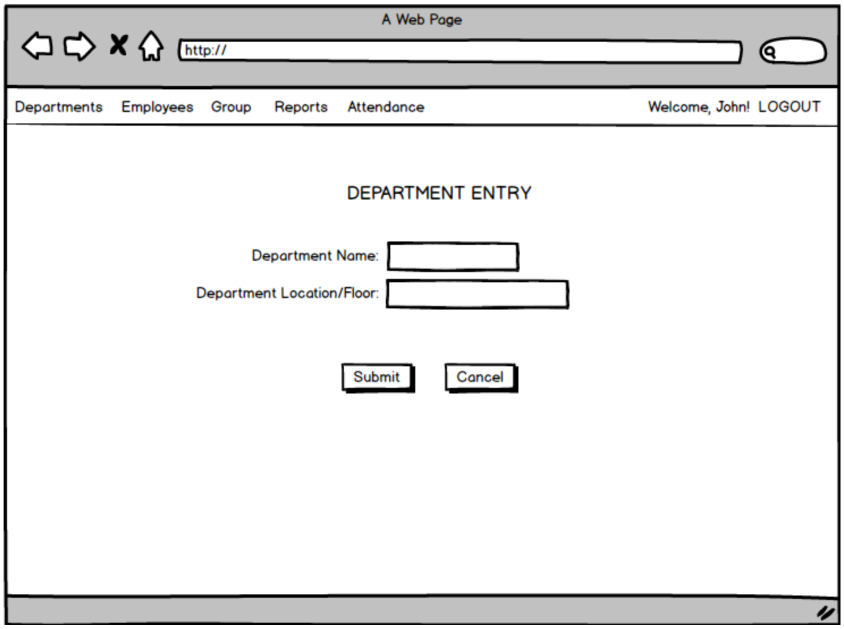


Figure 3: Department Entry

Your application should notify the user if a successful department has been created. Additionally, your application must validate the form data supplied for a departmental entry and educate the user if a failure occurs and why. Treat department name and department location/floor as mandatory fields.

In all cases, the Business Analysts have decided to leave all design decisions to the developers to ultimately decide and convey.

Employee

Companies are built by/on/with employees. The application you are required to build must also allow for the entry of employee profiles.

Employee Entry

The Business Analysts have provided a wireframe mockup of a potential employee entry page below:

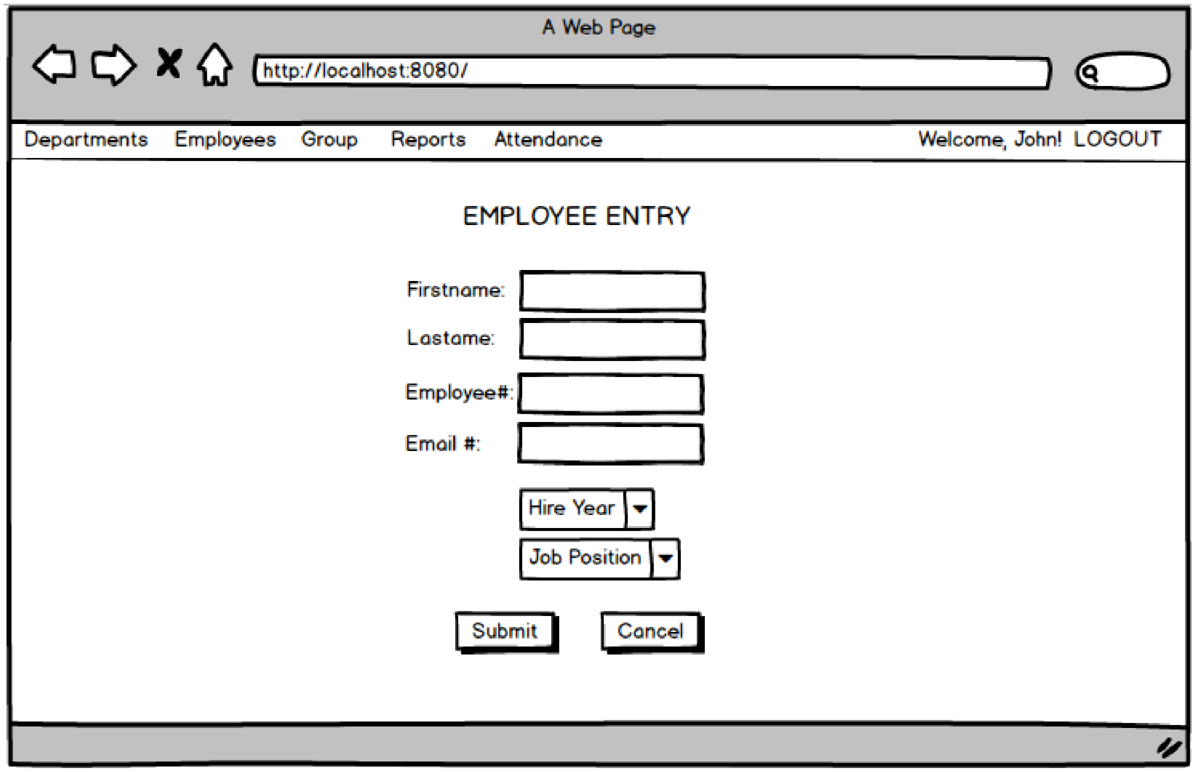
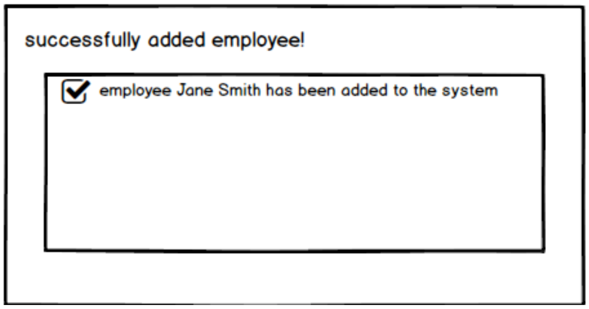


Figure 4: Employee Entry Page

If the employee data supplied by the user is valid and all fields have been vetted successfully, your application is required to notify the user of the successful employee entry into the system.



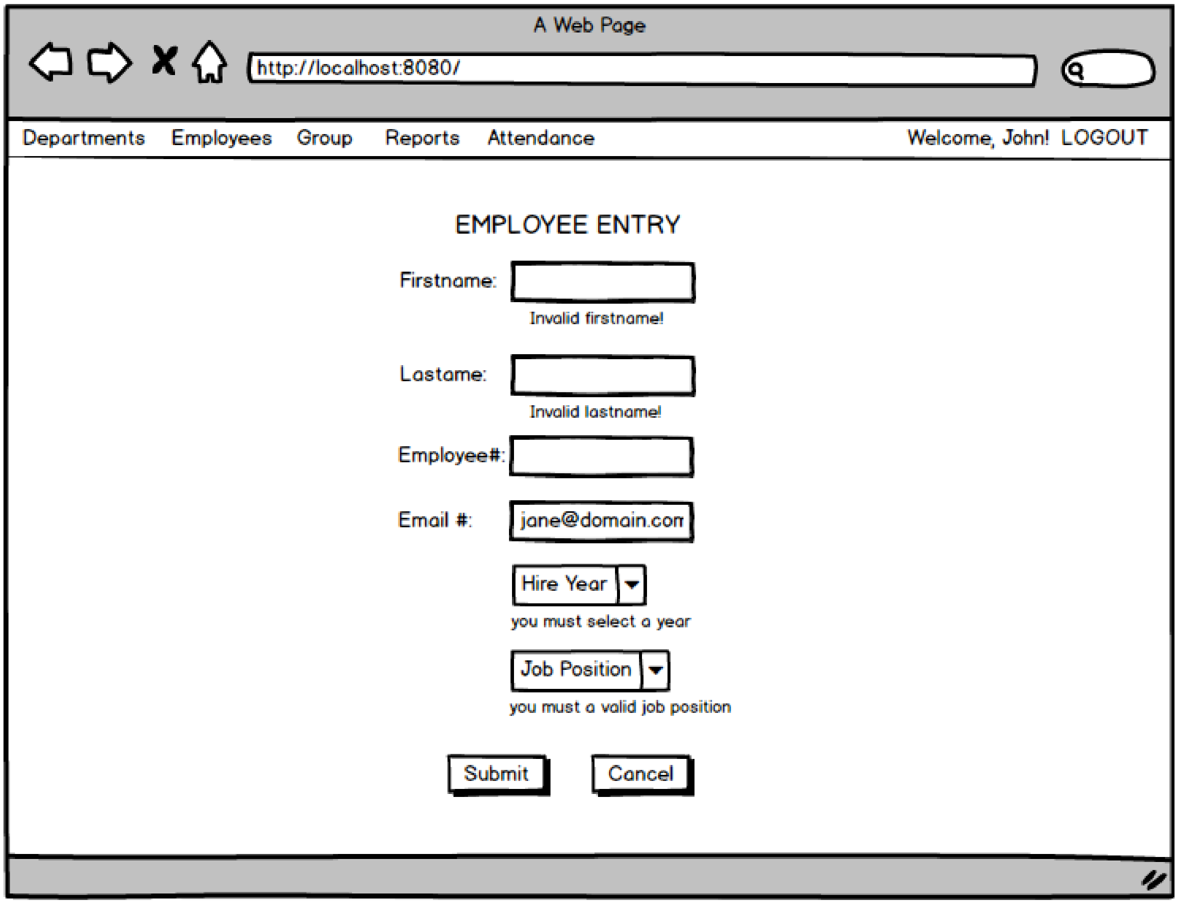
successfully added employee!

☑ employee Jane Smith has been added to the system

Figure 5: Example Successful Entry

It is expected that the user fills out all fields correctly. However, if the user submits the form with empty fields or with incorrect data, your web application should redisplay the form with the appropriate errors educating the user of the error or errors.

The following wireframe is an example of a registration page that has been submitted **incorrectly**.



EMPLOYEE ENTRY

Firstname:
Invalid firstname!

Lastame:
Invalid lastname!

Employee#:

Email #: jane@domain.com

Hire Year
you must select a year

Job Position
you must a valid job position

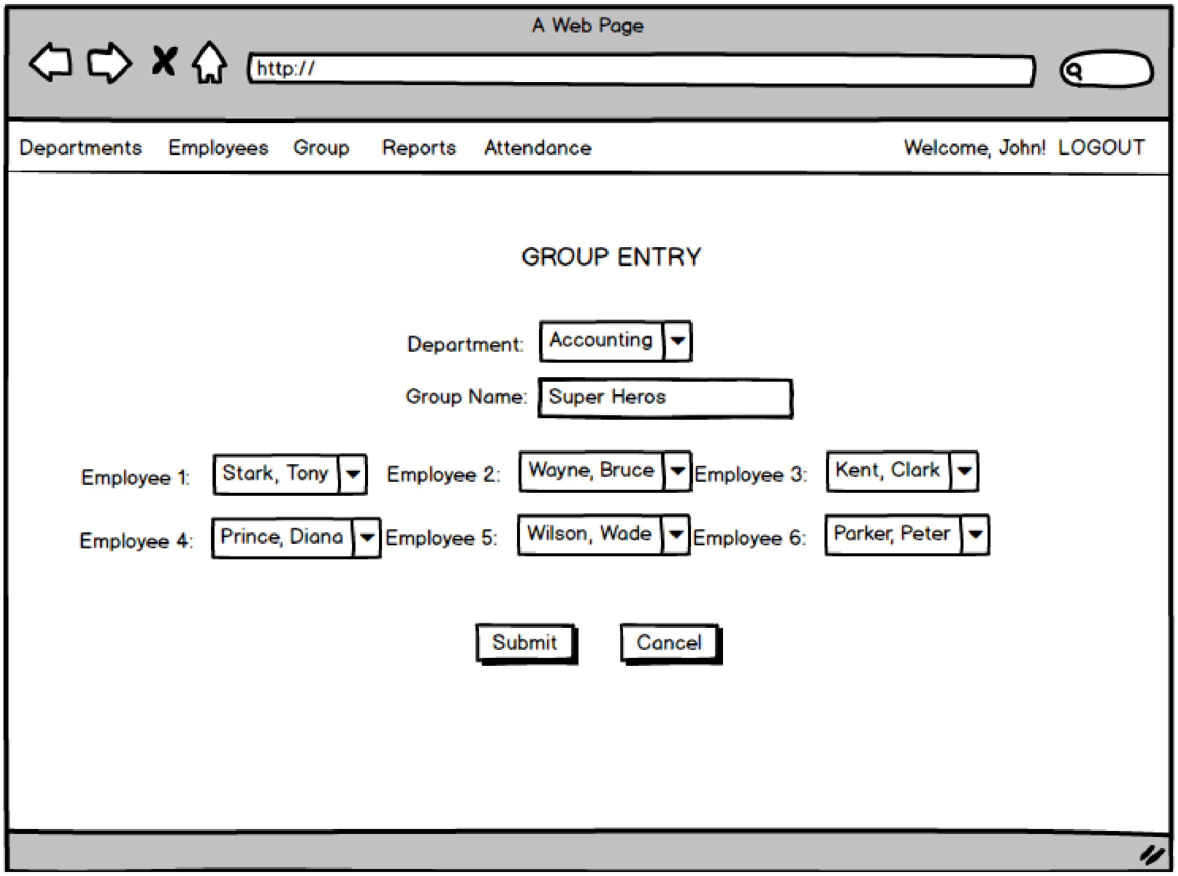
Figure 6: Failed Employee Entry

Group

Departments can be further divided into groups. A group can be thought synonymous to a team. A department can have many groups, your application must allow for the entry and configuration of a group.

Group Entry

The Business Analysts have provided a wireframe mockup of a potential group entry page below



GROUP ENTRY

Department: Accounting

Group Name: Super Heros

Employee 1: Stark, Tony
Employee 2: Wayne, Bruce
Employee 3: Kent, Clark

Employee 4: Prince, Diana
Employee 5: Wilson, Wade
Employee 6: Parker, Peter

Groups can have 1-6 members in them. The members of the group must be from the same department and must be associated with a group name.

Your application should notify the user if a successful group has been created. Additionally, your application must validate the form data supplied for a group entry and educate the user if a failure occurs and why.

In all cases, the Business Analysts have decided to leave all design decisions to the developers to ultimately decide and convey.

Reports

The Business Analysts have identified the need for a Reports section, but have decided to leave the implementation for a later phase in the project. For this phase however, a simple Reporting Landing page is sufficient.

Please review section titled "**Future Implementation**" for details on implementing this for Phase I

Attendance

The Business Analysts have identified the need for an Attendance section, but have decided to leave the implementation for a later phase. For this phase however, a simple Attendance Landing page is sufficient.

Please review section titled "**Future Implementation**" for details on implementing this for Phase I

Database and Data Persistence

In order for your application to be deemed a success, you will be required to implement some form of data persistence to store the applications information. Data such as, user and departmental data, employee profiles and group configurations must all be stored.

To assist with the creation of your data store, the persistence team has constructed an initial sql script (**database.sql**) to assist in the creation of a User Table. This script is not complete, as the schema design will be entirely the development team (aka your) responsibility, but the initial script will at the very least, give you guidance to start the creation of your schema design.

Please note, you will be **required** to further enhance this script and **submit** a completed sql that will ultimately construct and seed your backend database for your proposed solution.

Additionally to assist with the understanding of the internal relationships of the entities in your organization / project, the Business Analysts have also provided a brief ERD which should help clarify the relationship between Departments, Employees and Groups.

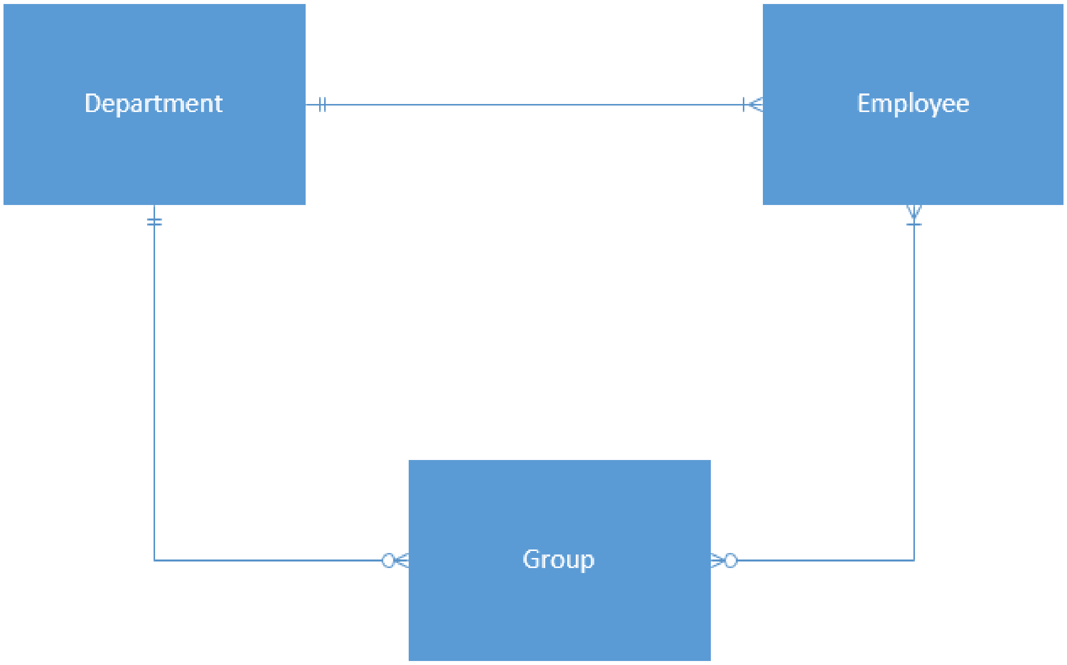


Figure 7: ERD Schema Design

As mentioned in the project synopsis, the database to be utilized is MySQL.

Future Implementation

The Business Analysts have provided a wireframe mockup pages to be implemented in future phases of the project

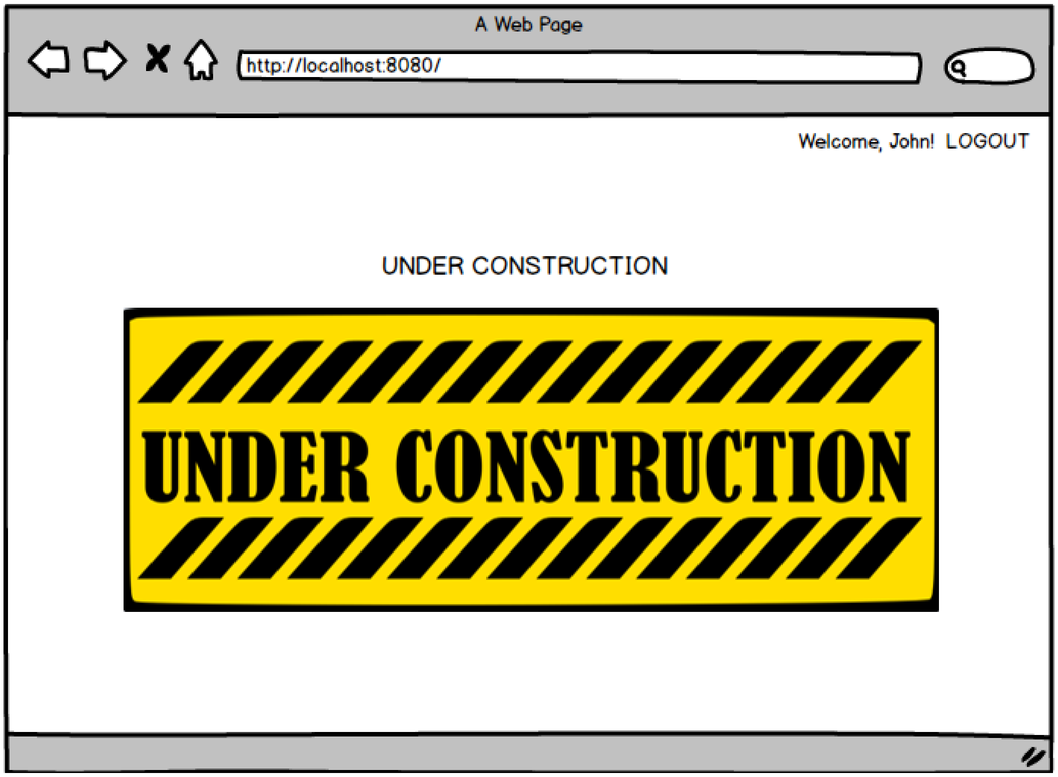


Figure 8: Welcome Page

The actual content of these page will not be implemented until Phase 2 (aka second assignment), however for Phase I, it should simply consist of a trivial page "Under Construction" image.

Validation Requirements

All pages under the Login page should not be visible/accessible to users who either have not have not authenticated or failed to authenticate. If a non-authenticated user attempts to access such pages, they should be redirected to the login servlet with an appropriate error message displayed.

You should also notice the "Login Status" bar located at the top of each page. It displays the message "**Welcome, FIRSTNAME**!", where the user's first name is replaced accordingly. Next to the welcome message is a link (LOGOUT) that, when clicked, sends the user back to the original login page and effectively logs the user out of your site. This "Logout" feature should occur in the login servlet. Note, not only should the current session be terminated when the user logs out, but if the user had requested enabled the site to "Remember Me" feature, the facility that was implemented to remember them should also be removed.

Lastly, it should be noted that the HTML content for this first assignment is largely created dynamically, and is produced via servlets. Additionally, it will be the under the developers full discretion, and intuitive design to craft appropriate and meaningful URLs to reach the various parts of the application.

Further validation clarifications

- Both first and last names must contain only alphabets.
- e-mail address must be of valid email form/structure (ie. must match the pattern of an e-mail address).
- The username must only contain alphabets.

Assignment Submission Guidelines:

1. You must upload your assignment to your manager (Professor Santilli) via **blackboard**.
2. All members in the project team must be cc'd on the final assignment submission. Failure to do so will result in a mark of zero for those member not cc'd on the assignment submission email.
3. Within the body of the email, clarify course code, team name, team members and student numbers. Title the email accordingly COMP 3095 – Assignment 1.

   Example:

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

4. When uploading, 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, no source code, no marks)
   2. **script** folder – .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 modular and should show no signs of dry (don't repeat yourself) code.
10. You are required to devise and use a form of persistent data storage.
11. The technologies used for this assignment are Java 8, Tomcat 8 and MySQL.
12. Your manager should need only to deploy your application (possibly run some .sql scripts) and be able to use and evaluate your application immediately (navigating to the pages described).
13. 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.
14. Late assignments are assigned a penalty of 10% per day.