ADVANCE SOFTWARE ENGINEERING

REPORT ON PROJECT INCREMENT 2

Submitted by,

Manusha Reddy Bethelli

Sneha Lagandula

Dheeraj Reddy Bethi

Project ID: 11

IMPORT EXISTING SERVICES:

In the second increment, validations for all the log-in and registration forms have been created for student and administrator access. Database is created to store the details of students and to store the details of vacancies available at a particular place. Hence, no existing web services were used so far in the second increment.

However, for the further increments, we are to implement the following APIs:

User Feedback API: http://api.sandbox.freelancer.com/User/getUserFeedback.{xml|json}

Geo Location API: https://developers.google.com/maps/documentation/business/geolocation/

Direction Service API: https://developers.google.com/maps/documentation/directions/

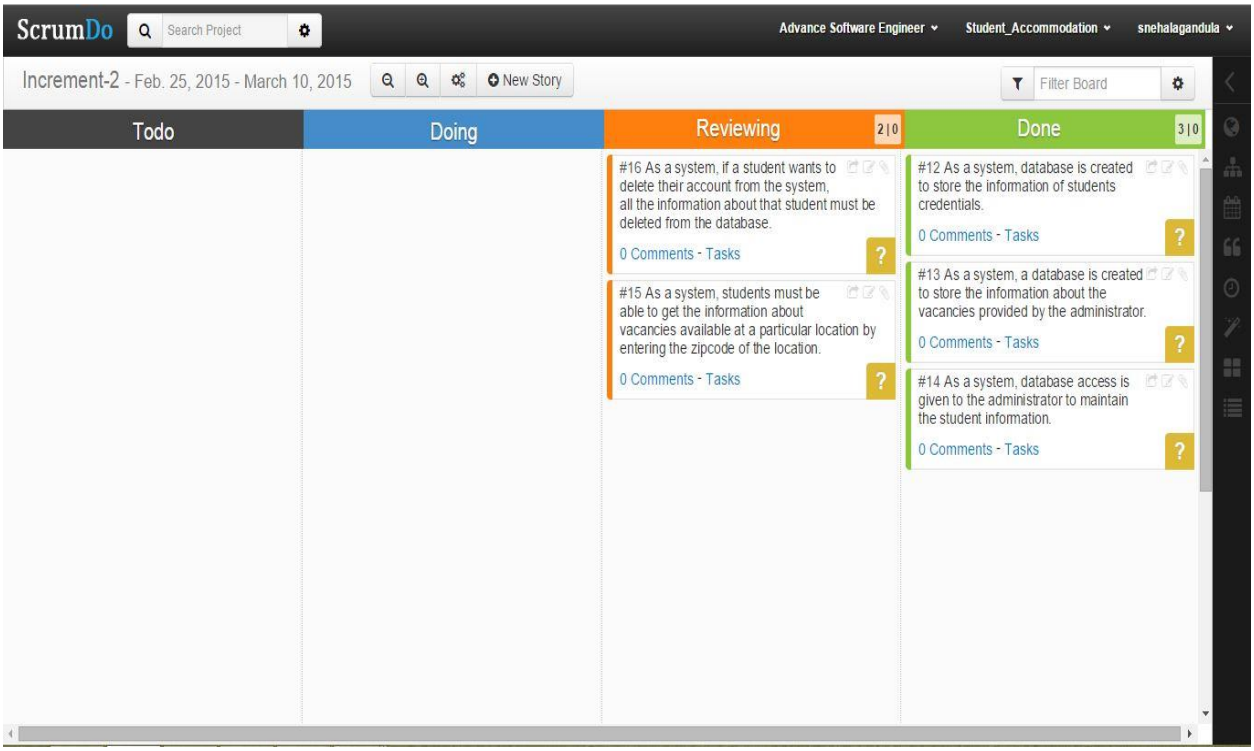
Facebook API: https://facebook.com/share/

Twitter API: https://twitter.com/share/

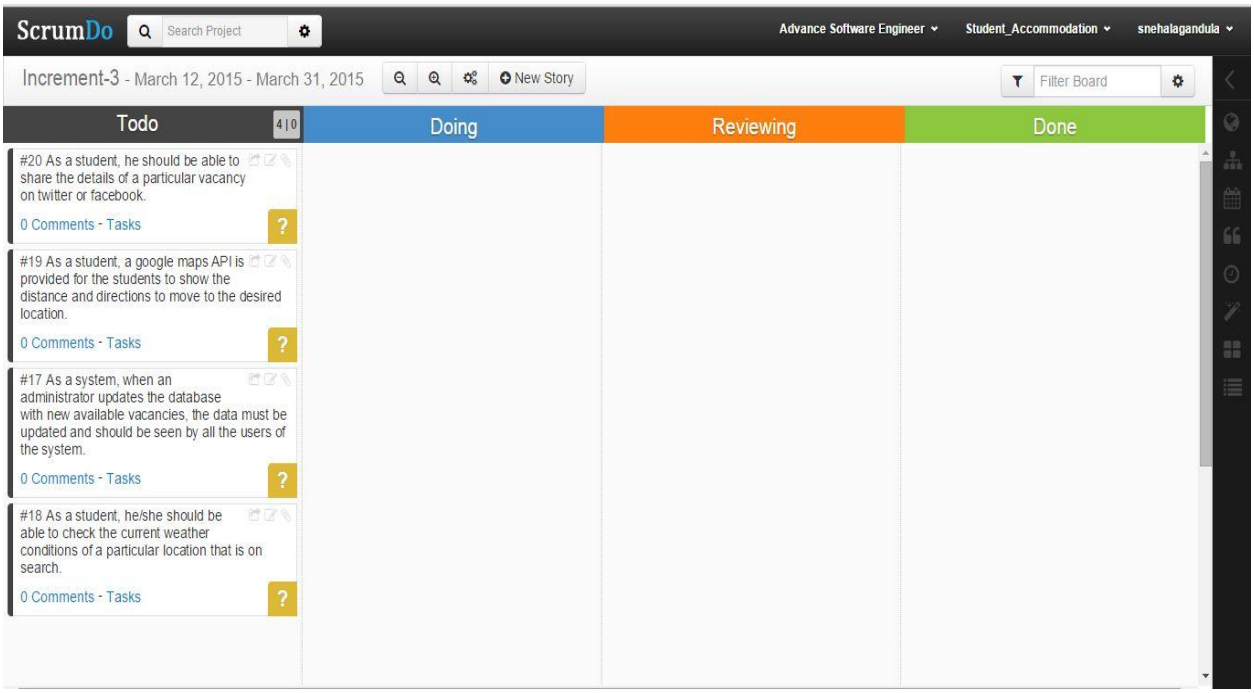
Source code for the so far developed application is uploaded in GitHub.

DETAIL DESING OF SERVICES:

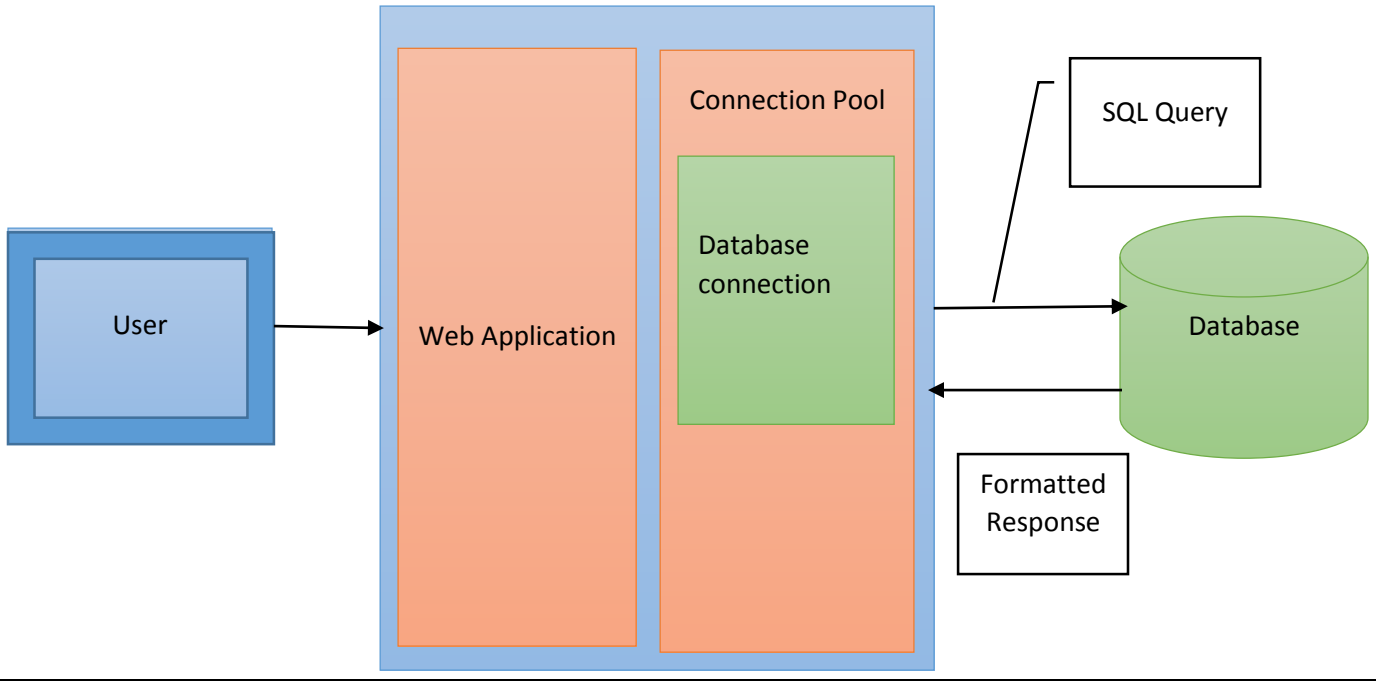
- **User stories for second increment**
  The second increment includes creation of database for the storage and retrieval of student information and house vacancies details.



Increment 3:

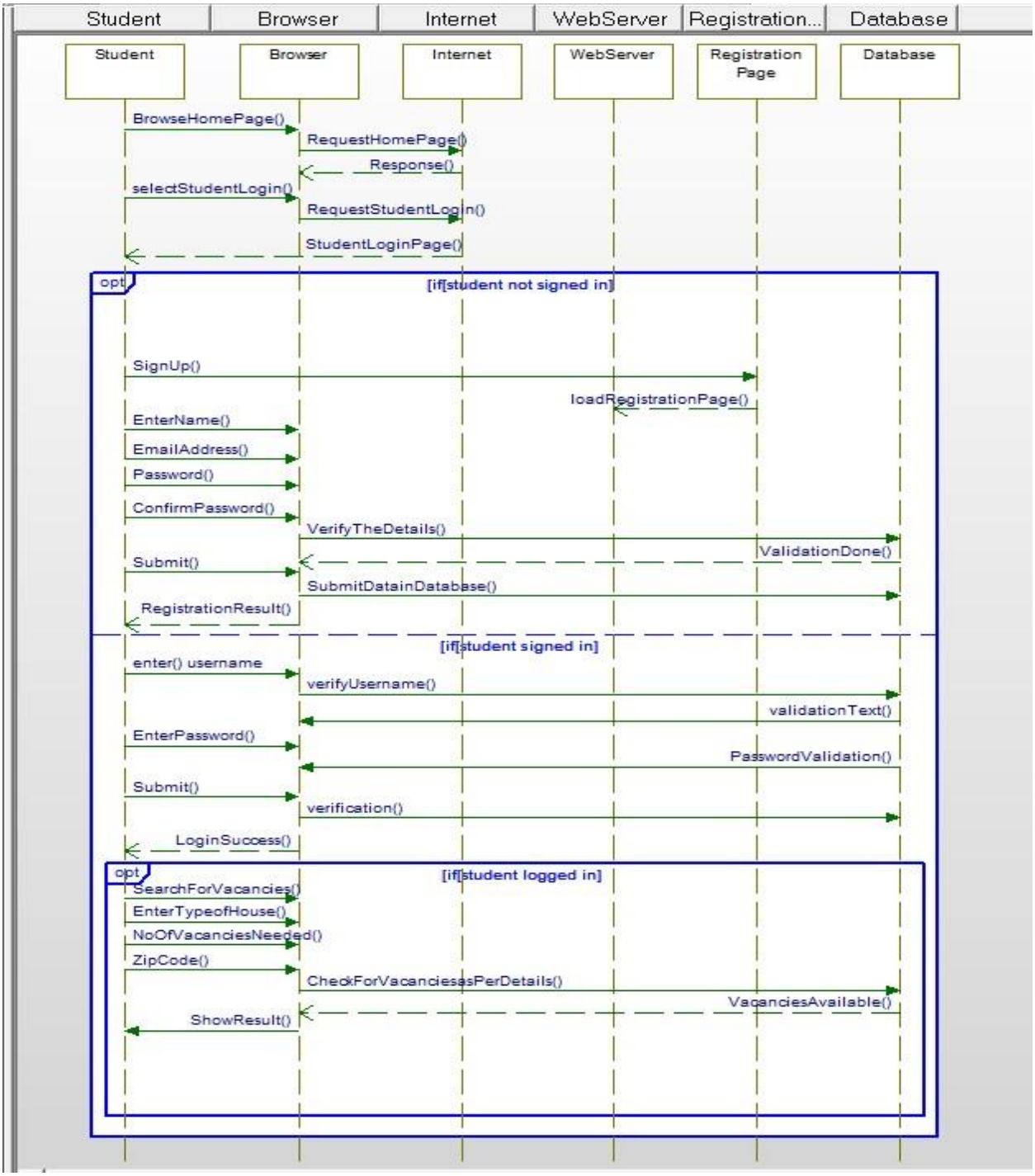


- **Design of Mobile Client Interface**

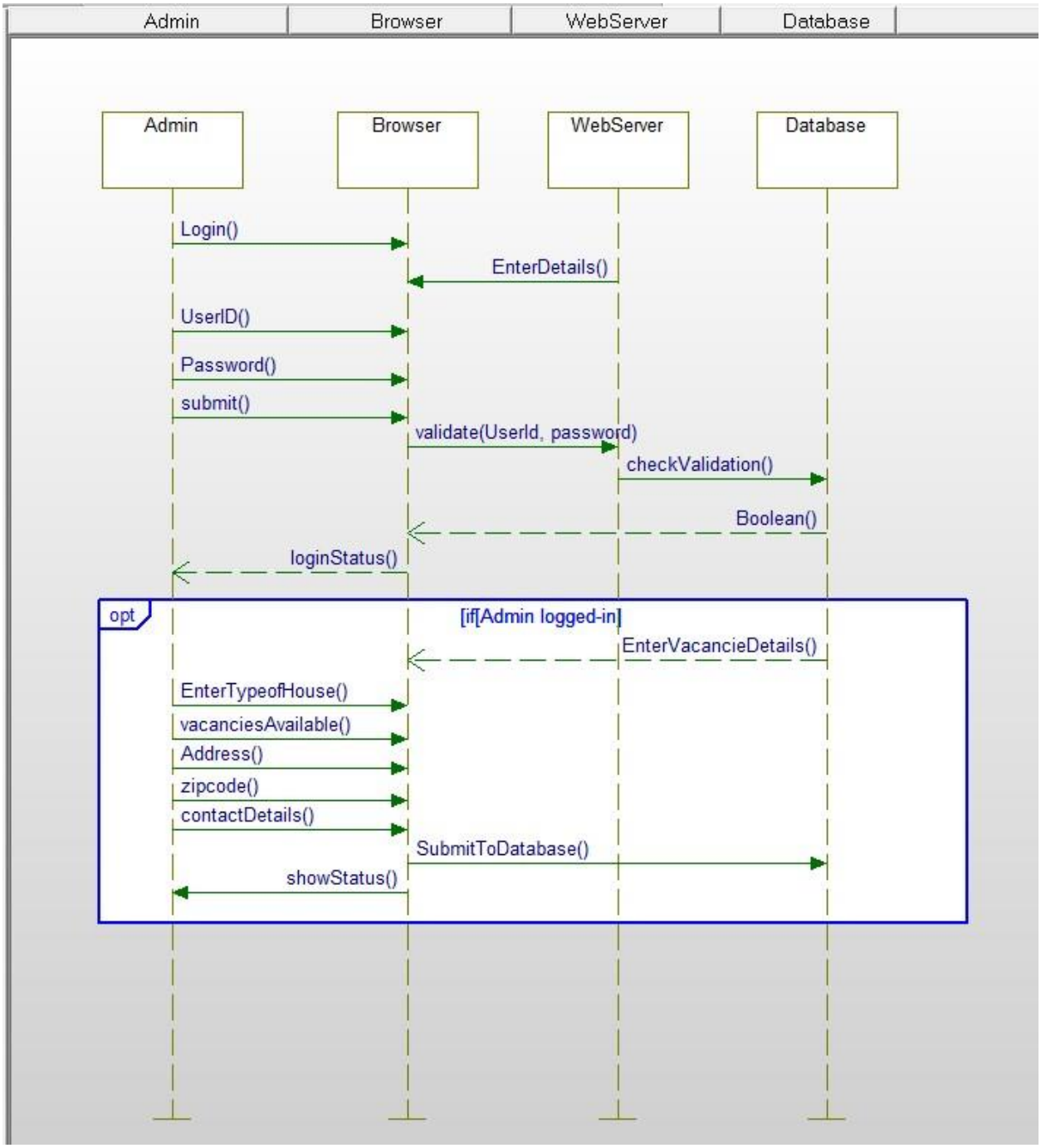


- **Sequence Diagram**

  A sequence diagram is an interaction diagram that provides the collaboration of objects depending on a particular time sequence. The following is the sequence diagram that shows the flow of processes that operates one after the other in a certain order. The sequence diagram is to show the flow of student registration and login form with certain validations.



Sequence Diagram for AdminLogin:



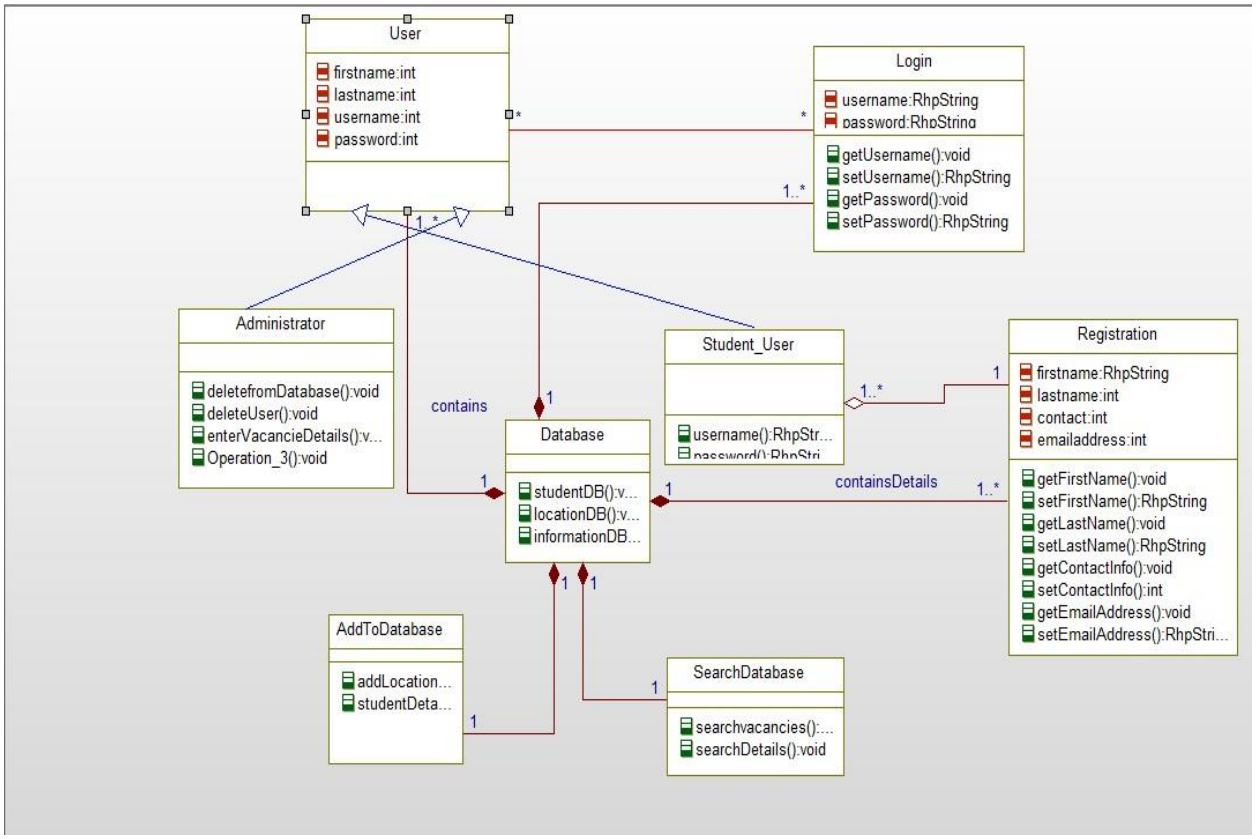
- **Class Diagram**

A class diagram identifies the structure of the application by describing the relationship between the objects and classes of the system.

The following class diagram describes a student or an administrator log-in to the website.

The system has two types of users, student user and the administrator. The student logs-in to the account only if the student is already registered in to the system. The student or admin logs-in to the system by entering the username and password, the system then validates the provided log-in details, and if successful, the system connects the user to the account for further access. If not successful, the system allows repeated attempts to the user and prevents the entry until the log-in is successful.

Database is created to store the information about the vacancies available at a certain location. When a student registers for the application all the details of the students are stored in a database for authorization. When a user logs in to the system, the entered details are checked in the database, if the details are availed in the database, further access is given to the system.



IMPLEMENTATION:

- **Implementation of User Interface**

  The second increment covers the storage and retrieval of student data from the database.

  When a student registers to the system, all the information about the student that is provided in registration form will be stored in to the database. After a successful registration, student can log-in to the system for further access. When the student logs in to the system, his/her credentials must be validated from the database. If the student is already registered for the application, and if the credentials provided by the student matches its respective fields in database, then a further access will be provided to the user.

- **Implementation of Test Cases**
  - **General Test Scenarios**
    - All the mandatory fields must be represented using asterisk symbol
    - Error should be displayed at a correct position
    - All the error or incorrect data should be displayed using same CSS styles
    - Reset functionality should be set, if user wants to clear the values in all the fields.
    - Application unavailable or crash should redirect to a new page that shows an error.
    - All fields on page should have the same alignment.
    - Scroll bar should be available when necessary
    - All pages must have their respective titles
    - Before performing any delete or update operation, confirmation message should be displayed.
    - Using all the parameters on the page, user should be able to filter the page.
    - Check whether the size of the window is resizable.

  - **Database Testing Test Scenarios:**
    - Table should have primary key in a column
    - Upon a page submit, check whether the data saved into database is correct.
    - Database name should be given depending on the type of the page created.
    - Length of the filed that is shown on the web page should be same as its length in database field.
    - Database fields should be designed with correct data type and data length.
    - Check the datatypes and its properties of their fields in database.
    - Verify the data in the database.
    - Logical names for the database are given according the database name.
    - For primary key column in database, null values should not be allowed.
    - All the required data tables must be created.

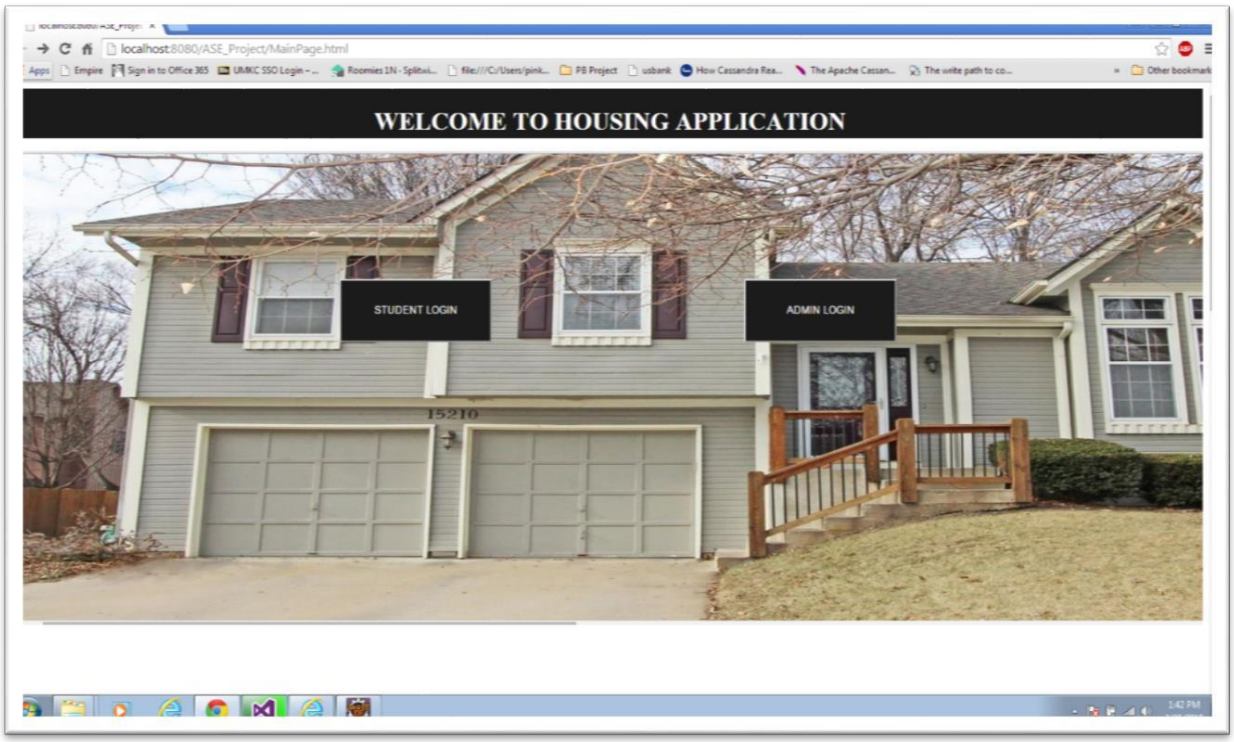
DEPLOYMENT:

- ScrumDo Link:
  http://www.scrumdo.com/projects/project/student_accommodation/iteration/122911/board#

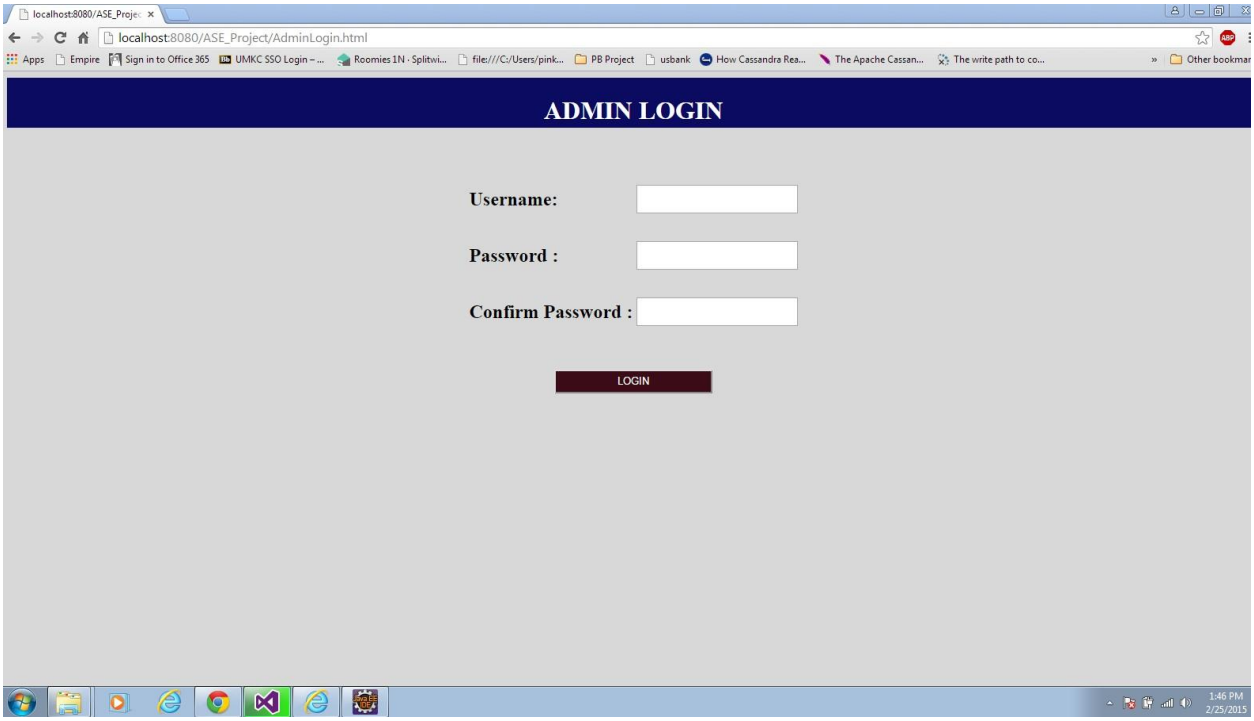
- GitHub (Source Code):
  https://github.com/dheerajreddy8/ASE-Project

REPORT:

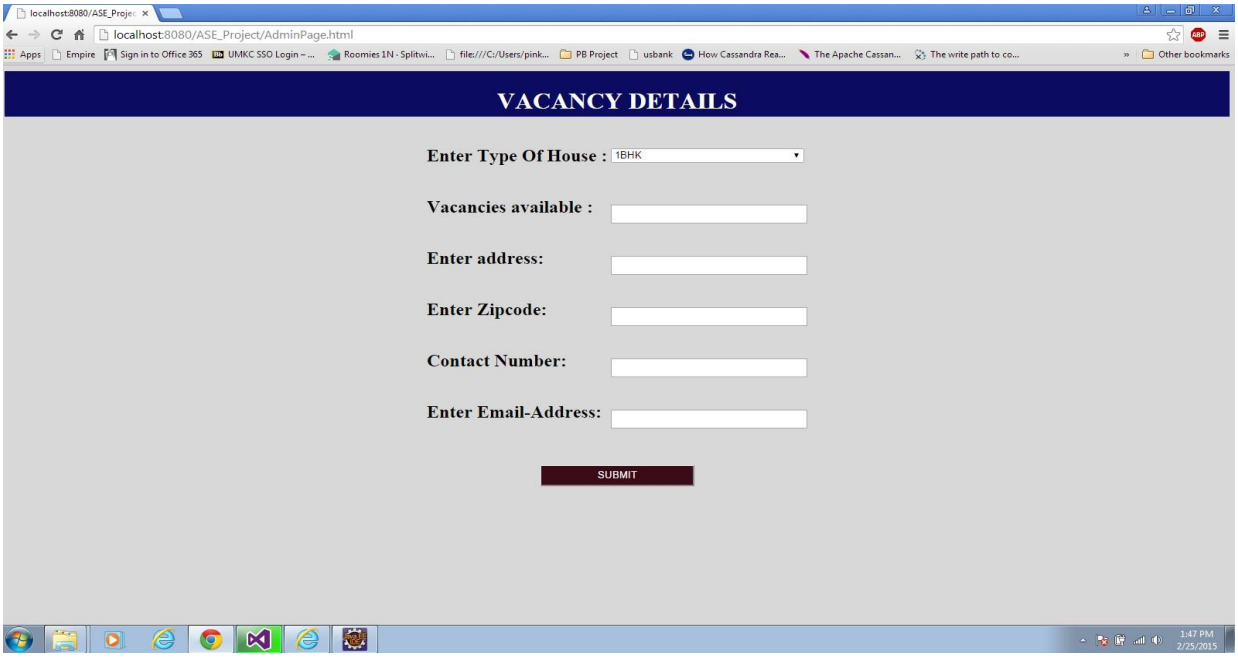
The following page allows the users access the home page of our website.



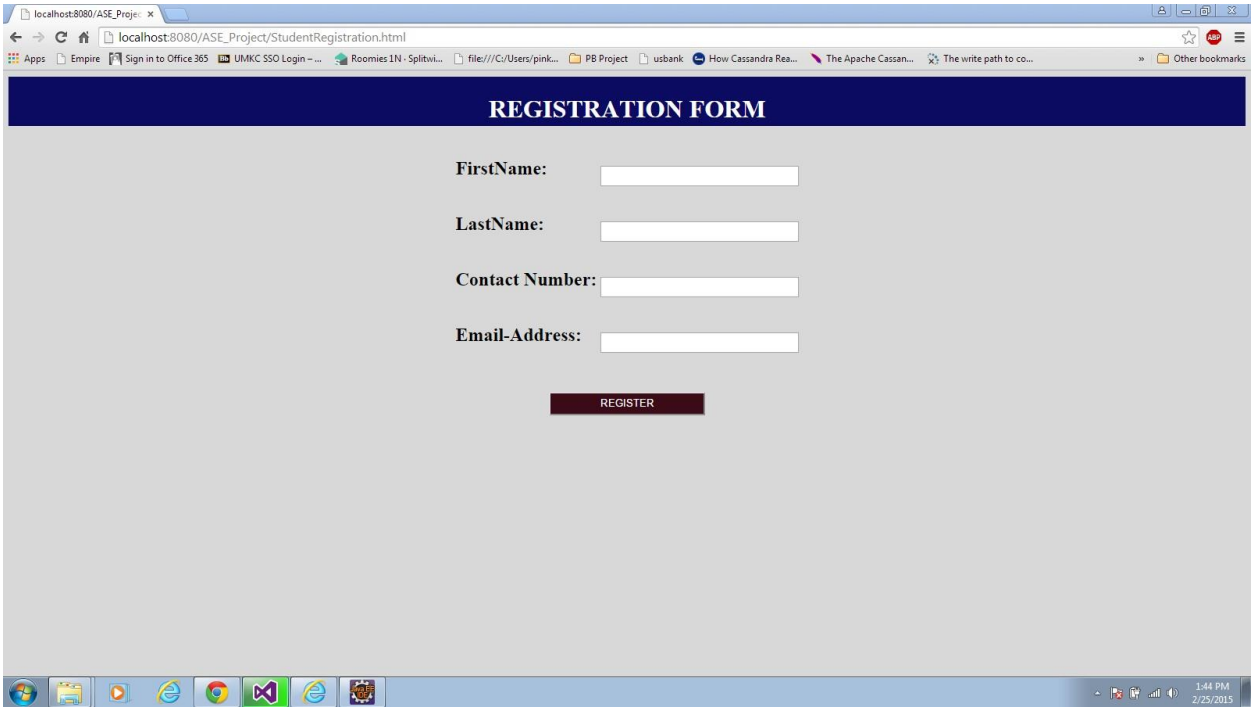
The following log-in page allows administrator to log-in into his account by providing his username and password



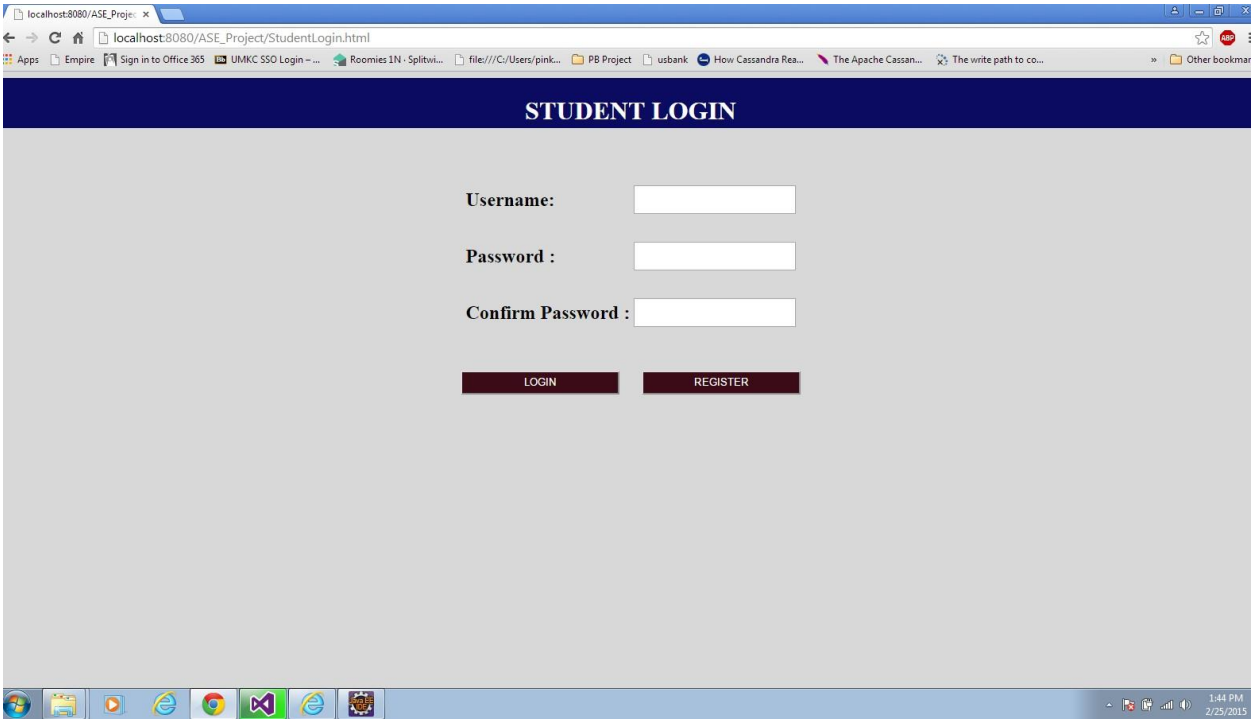
The following page allows administrator to enter the details of the location and other contact information about the available vacancies.



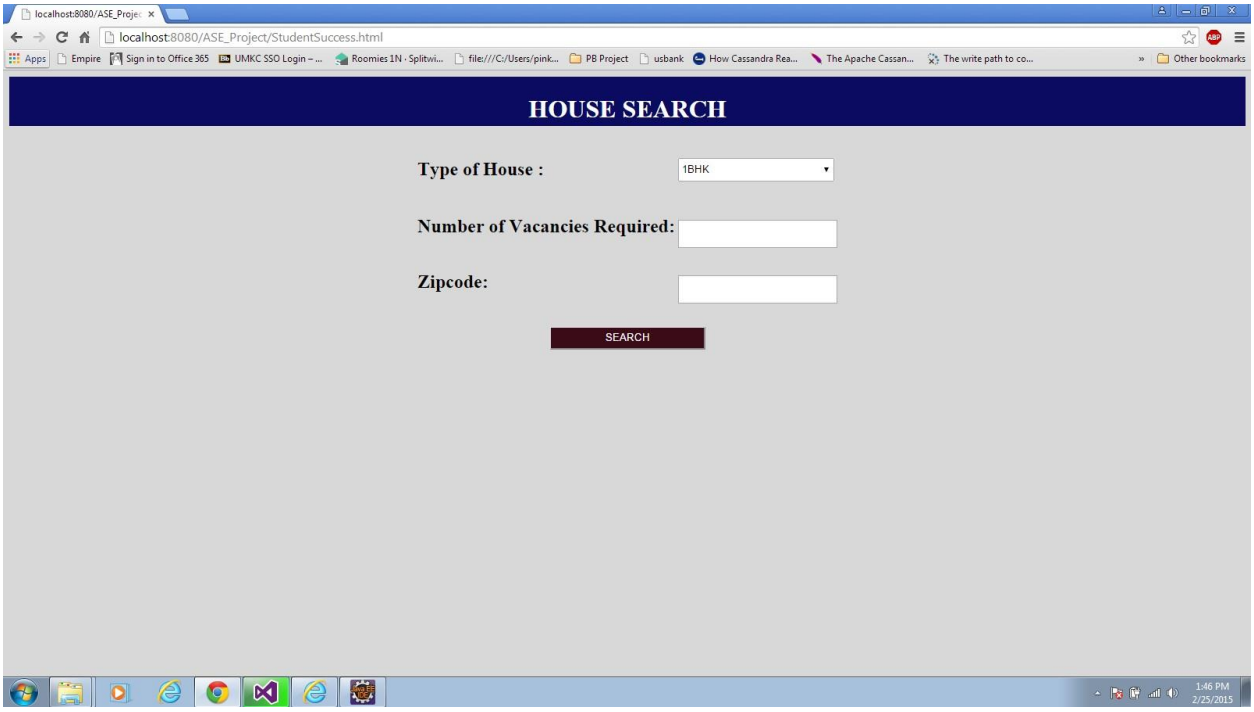
The following registration page allows students to register to the system for further access



The following log-in page allows students to login to the system provided they are already registered to the system



The following page allows students to search for the vacancies at a particular location



For the second increment, work was on front end as well as on back end. Database is created for storage and retrieval of student and administrator information.

PROJECT MANAGEMENT:

- **Implementation status report**
  - **Work completed:**
    - Description: Created database for student login, administrator login.
    - Created Database connection using JDBC Driver.
    - Responsibility:
      Task - Database storage and retrieval (Manusha Reddy)
      Task - Database storage (Sneha Lagandula)
      Task –Html validations (Dheeraj Reddy Bethi)
    - Time Taken: 3 Days
  - **Work to be completed:**
    - **Description:** Mash-up creation using API's