**Team Deliverable 3 – CSCE 5430 Software Engineering**

1. **Requirements:**

## **User Registration and Login**

Every user of the system will be required to register and maintain an account on the website before he/she can be able to post or buy products on the system.

* + 1. The system shall request the first name, last name, email address and phone number from the user during registration.
    2. The system shall make sure that a user provides first name, last name and email address during registration. Otherwise, the system shall prevent the user from proceeding.
    3. The system shall check to make sure the data provided by the user on the registration form are of the correct type. If not, the system shall display the errors to the user and request for corrections before proceeding.
    4. The system shall display a registration successful message upon successful registration.
    5. The system shall set-up a profile for the user upon successful registration.
    6. The system shall store the timestamp at which the user created the account.
    7. The system shall request a username and password to authenticate a registered user before he/she can access the buying and posting services on the website.

## **Profile maintenance and update**

The system shall maintain a profile for each registered user.

* + 1. The system shall allow the user to update his/her mailing address on his/her profile
    2. The system shall optionally request a user to provide a profile picture
    3. From the profile of the user, the system shall display transaction history of posted items that includes the item name, category, date and time the user posted the item.
    4. The system shall also display the transaction history of items the user has purchased/obtained for free through the website, including the item name, category, seller, date and time the user purchased the item.
* The above are our requirements for this, we have not implemented 3.2.3 and 3.2.4 because we have not implemented the product posting module yet and transaction history requires products to be posted by the user.
* All the other features and requirements specified in this phase where satisfied, including the criticalities that we mentioned in the deliverable 2.

**Updated Plan**: We will include the unimplemented parts in the phase 2 of our release.

**Phase 2:**

The phase 2 of the project will be focused on development of the features 3.2.3, 3.2.4, 3.3, 3.4 and 3.5. In the second phase as per incremental approach we plan on building the features mentioned in phase 2 and subsequently will focus on improving the look/feel of the website.

**Phase 3:**

The phase 3 of the project will be focused on development of the features 3.6 and 3.7. By the end of this phase we plan on developing a full-fledged application that is ready to use by implementing the final features and testing them accordingly.

**UML Diagrams:**

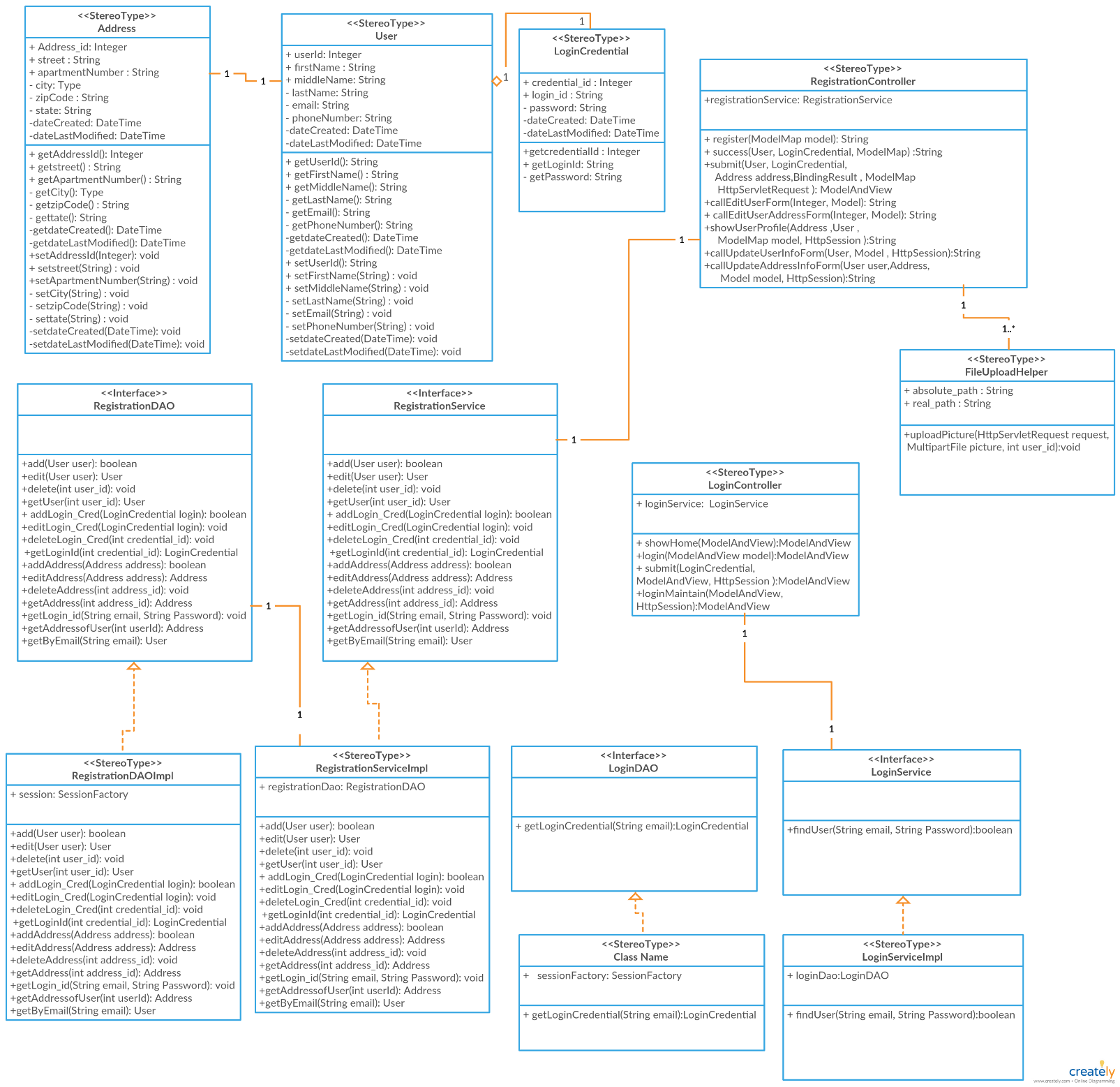
1. **Class Diagram**

Figure 1: Class Diagram Phase 1 (Diagram created with app on Creately.com)

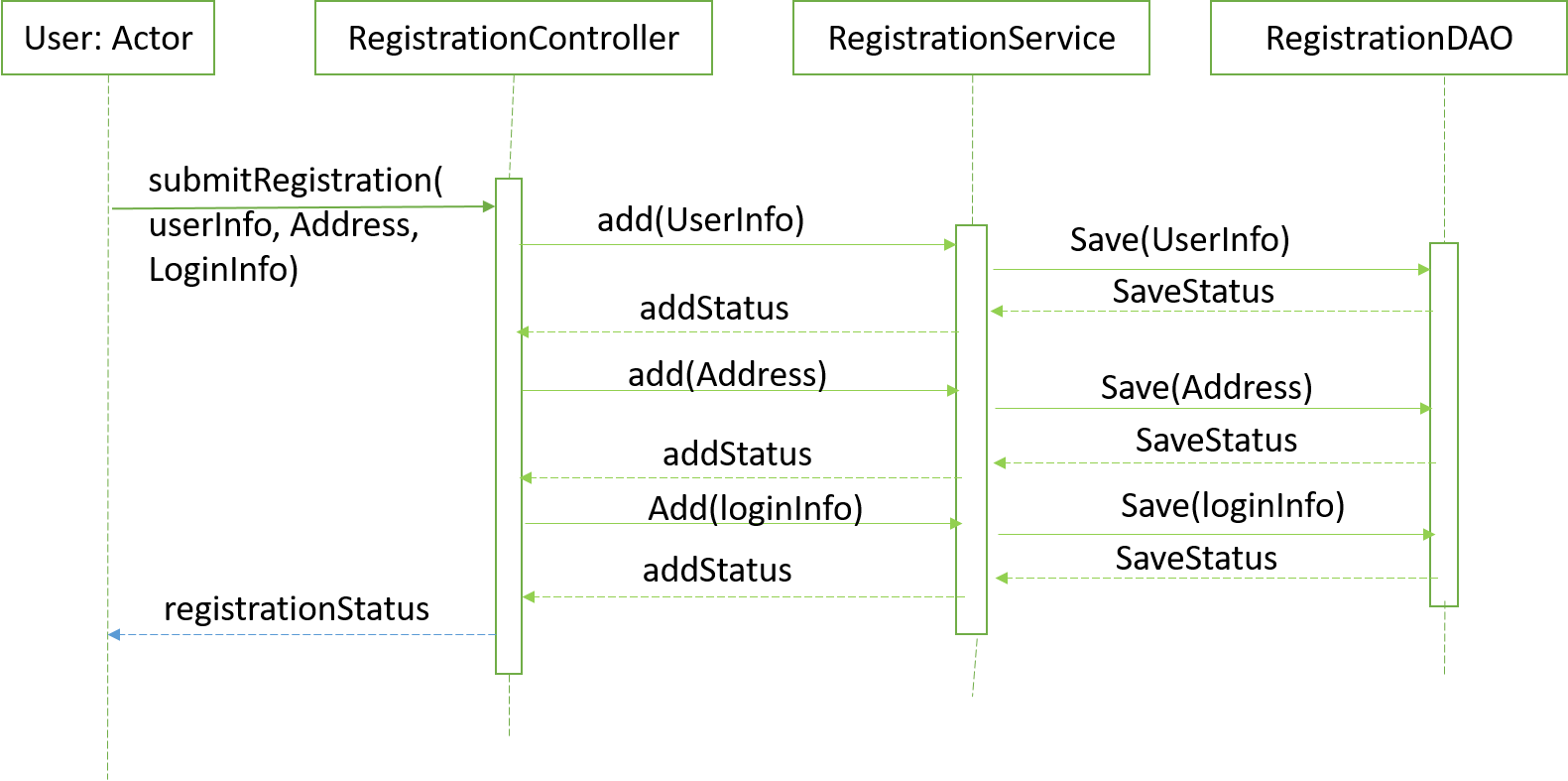
1. **Sequence Diagram**
2. Register Account

Figure 2: sequence Diagram for Register Account Use case

1. Database Failure (bring online)

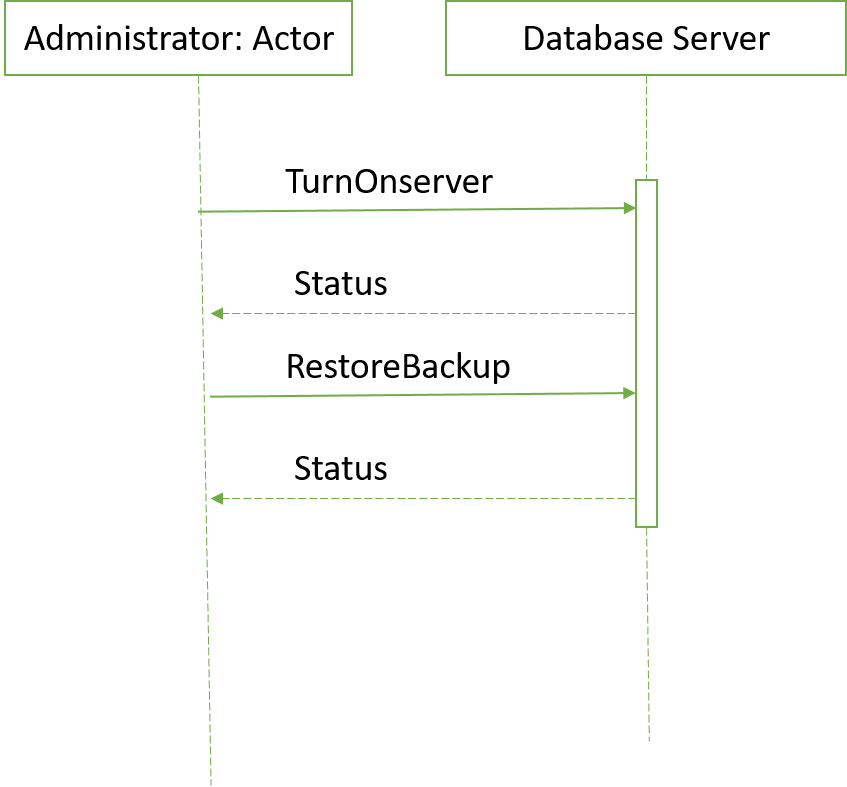


Figure 3: sequence diagram for bringing database server online after failure

1. Use Case Diagram

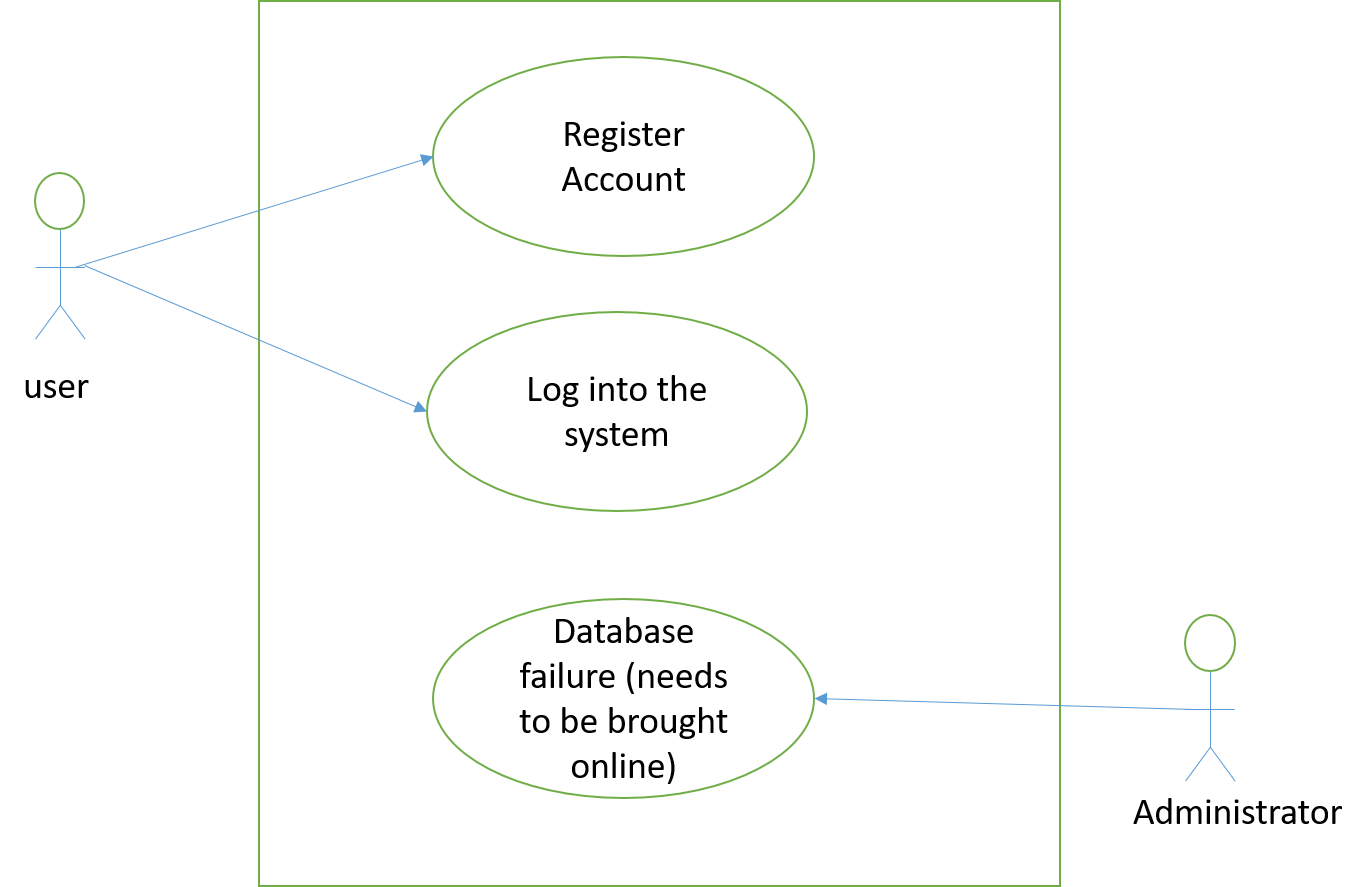


Figure 4: Use Case Diagram

1. **Register Account**

Main Success Scenario

1. User clicks on Sign up button
2. Registration Form is displayed to the user
3. User types his/her information in the respective form fields
4. User clicks on the submit button
5. Submission is processed by the system
6. User is taken to home page with success message

Extension

3a) User does not complete mandatory form fields. Clicking on submit button displays the

Empty fields and form is not submitted

5a) Data processing fails because of validation error is form fields

5b) User stays on registration page and error messages displayed for invalid form fields

1. **Database Failure (bring online) use case**

Main Success Scenario

1. Admin turns on the database server
2. Provides a username and password
3. Restores the database from backup files
4. Database is online and ready

Extension

3a) database backup file corrupted. Recovers data to the most recent stable version

3b) Informs users of the data loss.

1. **Log into the system**

Main Success Scenario

1. User clicks login button
2. Login form is displayed
3. User types username and password
4. User clicks submit button
5. User is authenticated and taken to home page.

Extensions

4a) user does not fill in password or username or both. Clicking on submit button displays errors to user.

5a) User authentication fails because of incorrect password and/or username. User stays on login form and is requested to check password and or username.

1. **Test Cases**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Name** | | studentsurplus | | | **Module Name** | | | | | | | Login | | | |
| **Created By** | Haidi Chen | **Created Date** | | 2018/10/28 | **Tested By** | | | | Haidi Chen | | | **Tested Date** | | | 2018/10/29 |
| **Test Scenario 1 – Verify the Functionality of Login Page** | | | | | | | | | | | | | | | |
| **Test Case ID** | **Test Case Description** | **Test Steps** | | **Preconditions** | | **Test Input** | | | | **Expected Output** | | | **Actual Output** | | **Status** |
| 01 | enter valid login\_id & valid password | 1. Enter login\_id 2. Enter password 3. Click on sign in button | | Valid URL,  Test Data | | login\_id =  [nick@gma.com](mailto:nick@gma.com)  password =  888888 | | | | home page | | | home page | | pass |
| 02 | enter valid login\_id & invalid password | 1. Enter login\_id 2. Enter password 3. Click on sign in button | | Valid URL,  Test Data | | login\_id =  [nick@gma.com](mailto:nick@gma.com)  password =  888 | | | | show error message: invalid username or password | | | invalid username or password | | pass |
| 03 | enter invalid login\_id & valid password | 1. Enter login\_id 2. Enter password 3. Click on sign in button | | Valid URL,  Test Data | | login\_id =  [nick@gmail.com](mailto:nick@gmail.com)  password =  888888 | | | | show error message: invalid username or password | | | invalid username or password | | pass |
| 04 | enter invalid login\_id & invalid password | 1. Enter login\_id 2. Enter password 3. Click on sign in button | | Valid URL,  Test Data | | login\_id =  [nick@gmail.com](mailto:nick@gmail.com)  password =  888 | | | | show error message: invalid username or password | | | invalid username or password | | pass |
| 05 | click “Register Here” to registeration page | 1. Click on “register here” link | | Valid URL, | | N/A | | | | registration page | | | registration page | | pass |
| **Created by** | | | **Date** | | | | **Module Name** | | | | **Registration** | | | | |
| **Sri Sravya Tirupachur Comerica** | | | **10/28/2018** | | | | **Tested by** | **Sri Sravya Tirupachur Comerica** | | | **Date** | | | **10/28/2018** | |
| **Test Scenario 2 – Verify the Functionality of Registration Page** | | | | | | | | | | | | | | | |
| **Test Case ID** | **Test Case Description** | **Test Steps** | | **Preconditions** | | **Test Input** | | | | **Expected Output** | | | **Actual Output** | | **Status** |
| 06 | enter all information on the page and agree the terms and conditions | 1. Enter all the fields 2. Check “agree to terms and conditions” 3. Click on “submit” button | | Valid URL,  Test Data | | FirstName=  Hellen  MiddleName=  J  LastName=  Li  email=  [hellen@gma.com](mailto:hellen@gma.com)  phonenumber=  9409993333  street=  123 stocken  apartment No.=  201  city = Denton  state = TX  zipcode = 76201  profile picture = null  password =  hellen123  confirm password =  hellen123 | | | | home page | | | home page | | pass |
| 07 | required information is not provided | 1. Enter part of the fields 2. Check “agree to terms and conditions” 3. Click on “submit” button | | Valid URL  Test Data | | FirstName =  Khai  LastName =  Steven | | | | show warning | | | Show warning | | pass |
| 09 | go to home page from registration | 1. Click on “return to Home page” link | | Valid URL | | N/A | | | | home page | | | profile page | | fail |
| **Test Scenario 3 – Verify the Functionality of Profile Management Page** | | | | | | | | | | | | | | | |
| **Test Case ID** | **Test Case Description** | **Test Steps** | | **Preconditions** | | **Test Input** | | | | **Expected Output** | | | **Actual Output** | | **Status** |
| 10 | change profile information | 1. Click on edit button on the upper right side of page 2. Enter new information 3. Click on submit button | | Valid URL  Test Data | | MiddleName =  K  phoneNumber =  1991998888 | | | | back to profile page with modified information | | | profile page with old information | | fail |
| 11 | change address information | 1. Click on edit button on the right side of Address section 2. Modify information 3. Click on submit button | | Valid URL  Test Data | | Street = 100  Apartment No. =  394 | | | | back to profile page with new information | | | profile page with new information | | pass |

**The path of test case implementation:**

* SourceCode\unt.cse.studentsurplus\src\test\java\unt\cse\studentsurplus\controller\LoginControllerTest.java
* SourceCode\unt.cse.studentsurplus\src\test\java\unt\cse\studentsurplus\controller\RegistrationControllerTest.java
* SourceCode\unt.cse.studentsurplus\src\test\java\unt\cse\studentsurplus\service\**LoginServiceImpTest.java**
* SourceCode\unt.cse.studentsurplus\src\test\java\unt\cse\studentsurplus\service\Registration**ServiceImpTest.java**
* SourceCode\unt.cse.studentsurplus\src\test\java\unt\cse\studentsurplus\dao\ **LoginDAOImplTest.java**
* SourceCode\unt.cse.studentsurplus\src\test\java\unt\cse\studentsurplus\dao\ **RegistrationDAOImplTest.java**

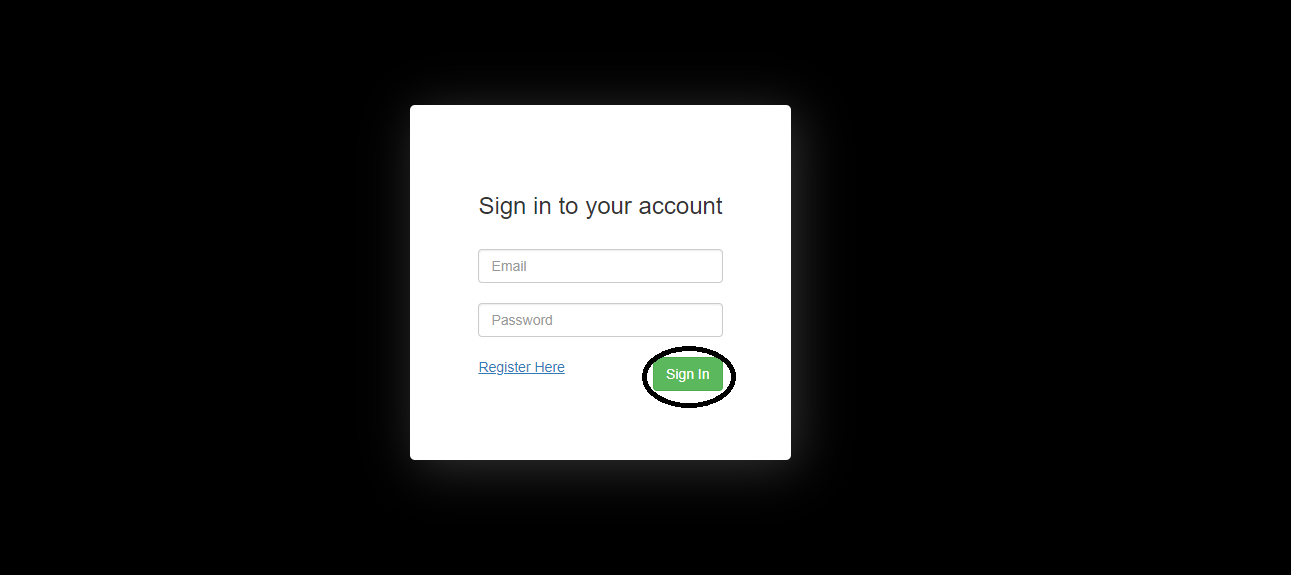
**D. Components:**

|  |  |
| --- | --- |
| **Components** | **Developed By** |
| Registration Module and Profile Update Module | Sri Sravya Tirupachur Comerica |
| Registration Front- End View, Profile View and Profile Update view. | Aboubakar Mountapmbeme |
| Login Module and Session Maintenance Module | Haidi Chen |
| Home Front End view and Login View | Amisha Gadhia |

**E. User Manual:**

**Login:**

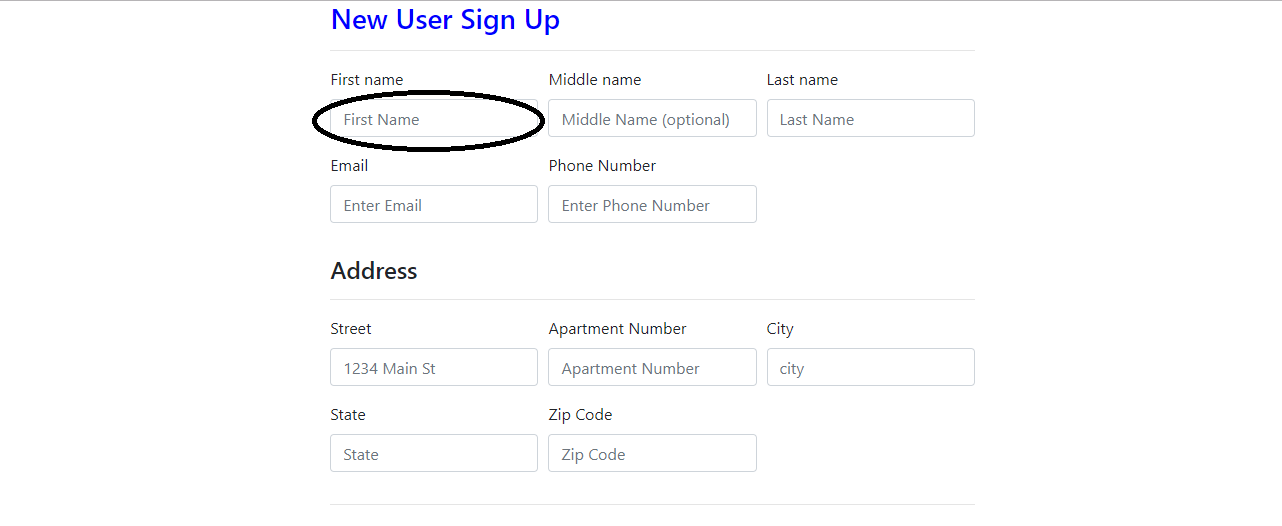
* By clicking on the register here link on the page, the user will be redirected to the registration page.
* If already registered, he can simply login using his username and password.

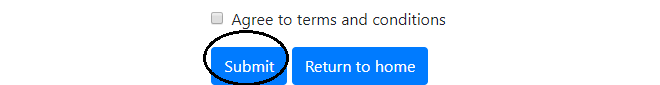
****

Picture (i) Login Page

**Registration:**

* A user can register by filling in the fields that are provided below.
* By clicking on the submit button given below, the user will be successfully registered.

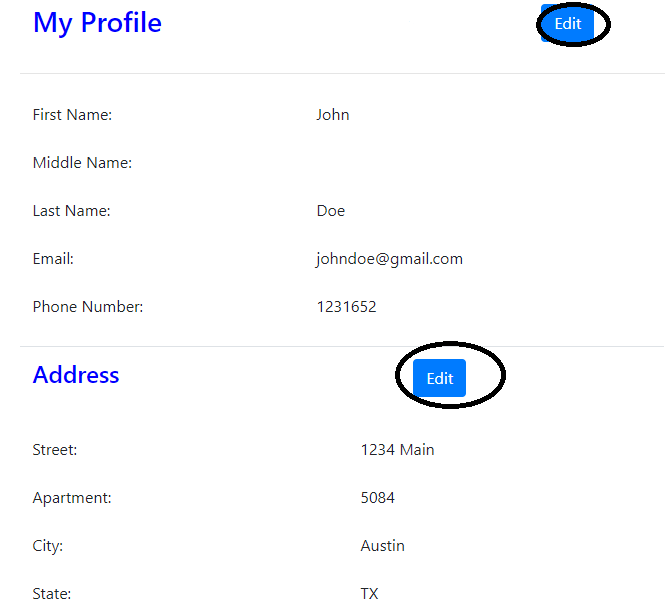
****

****

Picture (ii) Registration Page

**My Profile:**

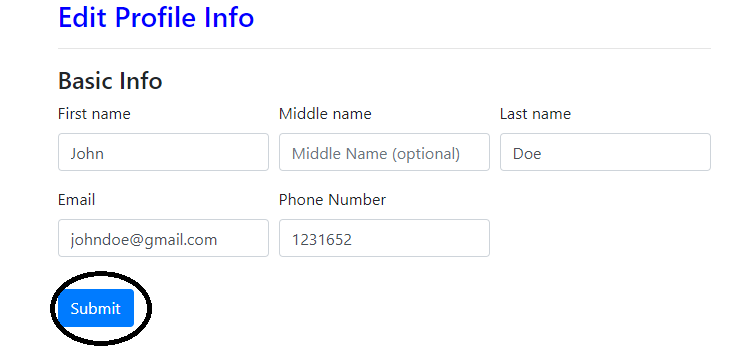
* This user profile page will show the details of the user and they can also change their details by clicking on the edit button

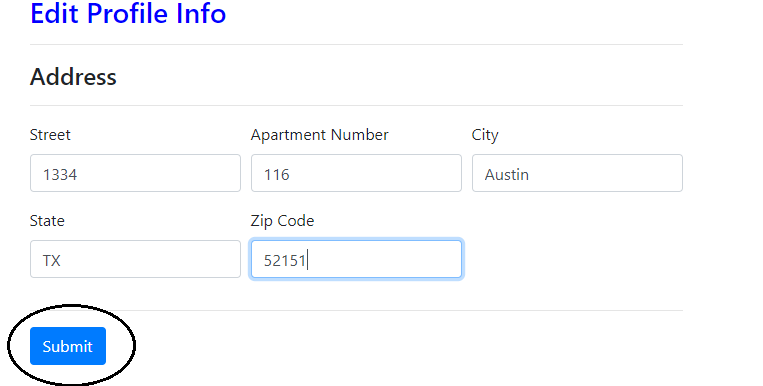
****

Picture (iii) Profile Info Page

**Profile update:**

* After clicking the edit button the user will see fields where he can manually edit his details
* Clicking on the submit button will change the user details and display it to him on the user profile page. [1]





Picture (iv) Profile Update Page

**Home Page:**

* Home will provide an option for user to view his profile by clicking on the user-profile on the navigation bar and also feature to register/login if haven’t done so already.[1]



Picture (v) Home Page

**F. Instruction on how to compile and run:**

* Software Required: Tomcat Server 9, MySql Database and Eclipse IDE.

* Program:

1. Open the project file in Eclipse IDE.
2. Open MySql.
3. Open the studentsurplus.xml file and set the data source values of username and password to your local database’s username and password.
4. unt.cse.studentsurplus-> run as-> Maven Build
5. unt.cse.studentsurplus->run as-> run on server

* Test Cases

1. unt.cse.studentsurplus->src/test/java->RegistrationControllerTest.java
2. RegistrationControllerTest.java-> run as-> JUnit Test

**G. Project Peer Review:**

**Topics:**

1. Data Security
2. Insufficient Requirements Specification
3. Formatting of the SRS document
4. Elaboration on phases and their criticality
5. **Data Security:**

* Discussed about the security of data in the database, fixed it by encrypting the sensitive information of the user before storing it in the database.

1. **Insufficient Requirements Specification:**

* The user constraints explained by us in the SRS document where not elaborate, and hence we elaborated the user constraints in the next version of SRS document.

1. **Formatting of the SRS document**:

* Changed the formatting of the SRS document as per IEEE standards in the next deliverable.

1. **Elaboration on phases and their criticality**:

* Analyzed more and elaborated on how plan on implement our phases and mentioned the criticalities that where involved in their implementation.

**References:**

1. https://www.bridgemaneducation.com/documents/how-to-guide.pdf

Group Name: **DASH**

Project: **Student Surplus**