

Course : System Analysis and Project Management (MIS 6308.02)

Professor: Dr. Dawn Owens



**UTD Roommate Search Mobile Application**

MILESTONE 4- Design

**GROUP 11**

Mayank Madhav

Prema Siddu

Rutwika Mohanty

Tejasvi Ramadas Sagar

Table of Contents

[User Interface Design: Create an Interface design 2](#_Toc512594668)

[Sequence Diagram with Control Objects 2](#_Toc512594669)

[Software Design 2](#_Toc512594670)

[Testing 2](#_Toc512594671)

[Test case 1 2](#_Toc512594672)

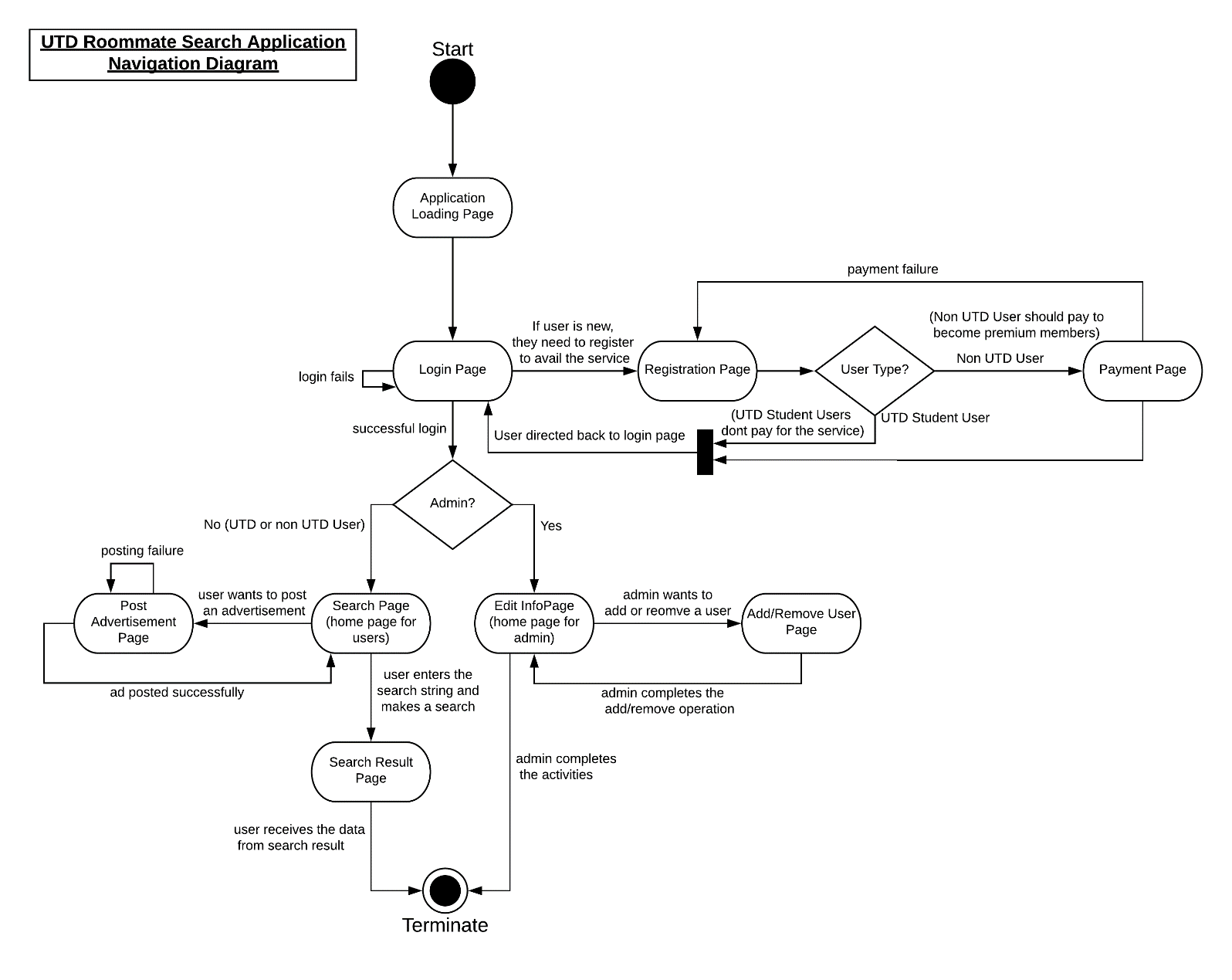
[Test case 2 2](#_Toc512594673)

[Test case 3 2](#_Toc512594674)

[Architecture diagram 2](#_Toc512594675)

[Minutes of Meeting 2](#_Toc512594676)

# User Interface Design: Create an Interface design

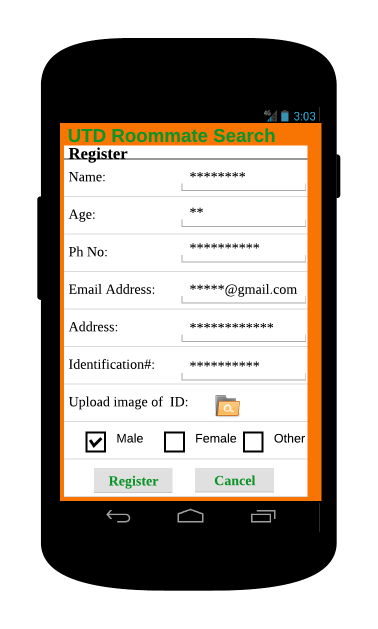


Here are few of the screen mockups, showing the User Interface of our application:

1. Application Loading page



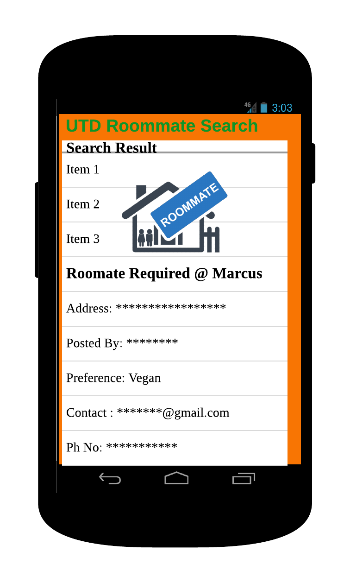
1. Registration Page



1. Search Page



1. Search Result Page



# Sequence Diagram with Control Objects

# Software Design

1. **Search Room/Roommate: - This is method for searching room availability.**

|  |  |  |
| --- | --- | --- |
| **Method Name:** SearchRoom () | **Class Name:** User | |
| **Description of responsibilities:**    Implementing the necessary behavior when the User once logged in to the UTD Search Application navigates to the search engine. The user keys in the keywords for the search and clicks on ’Search’ button. | | |
| **Arguments Received:**  Search | | **Data Type:**  String |
| **Return Value:**  SearchResult | | **Data Type:**  list |
| **Message and Example:**  SearchEngine(Search): Boolean  SearchResult = SearchEngine(2BHK) | | |
| **Algorithm Specification:**  --Checks if Search is present in database  If Search is not null  Retrieve the Search from databases  Display Search Results from the query  Return SearchResult  Else  Return 0  End If | | |

1. **Payment: - This is method for online payment by Non-UTD Users**

|  |  |  |
| --- | --- | --- |
| **Method Name:** Payment () | **Class Name:** Non-UTD User | |
| **Description of responsibilities:**    Implementing the necessary behavior when a non-UTD user registers to the application, he/she needs to make a payment to access the application as a premium account holder. A pop-up for premium account displayed on the main page with price details. | | |
| **Arguments Received:**  UserID | | **Data Type:**  Integer |
| **Return Value:**  MakePayment | | **Data Type:**  Boolean |
| **Message and Example:**  Non-UTDUser(UserID): Boolean  MakePayment = Non-UTDUser(631) | | |
| **Algorithm Specification:**  --Checks if UserID is present in database  If UserID is not null  Retrieve the User ID from databases  --Check for valid User ID  If valid User ID  --Check Whether User is UTD User/Non-UTD User  If Non-UTD User  Make the Payment  Enter Payment Details  If payment is successful  Return 1  Else  Return 0  End If  Else  Return -1  End If  Else  Return -2  Else  Return -3  End If | | |

3. **Update Information: - This is method for Updating information in the enquiry form provided by Users.**

|  |  |  |
| --- | --- | --- |
| **Method Name:** UpdateInformation () | | **Class Name:**  UTDStudent/Non-UTD User |
| **Description of responsibilities:**    Implementing the necessary behavior by updating the information provided by the Users in the enquiry form. | | |
| **Arguments Received:**  ID (UTD UserID or Non-UTD UserID) | **Data Type:**  Integer | |
| **Return Value:**  UpdateInformation | **Data Type:**  Boolean | |
| **Message and Example:**  UpdateInformation(ID): Boolean  UpdateResult = UpdateInformation (75252) | | |
| **Algorithm Specification:**  --Checks if ID is present in database  If ID is not null  Retrieve the ID from databases  --Check for valid ID  If valid ID  Retrieve Enquiry Form from database  Update Enquiry Form  If valid Enquiry Form  Update Enquiry Form in the Database  Return 1  Else  Return 0  End If  Else  Return -1  End If  Else  Return -2  End If | | |

# Testing

# Test Cases for UTD Roommate Search

# Functional Test cases

# **1.1 Test Case Title:**

**Verify login ID and password for the existing user.**

|  |  |
| --- | --- |
| **Purpose:** | To verify whether the user (UTD student) can login into the application successfully. |
| **Prereq:** | The user (UTD student) has already created an account for the application. |
| **Test Data:** | Email ID = {valid: xyz123@utdallas.edu, invalid: @utdallas.edu, %$#@.com, empty}  Password = {valid: abc123& (minimum 7 letters, alphanumeric and one special character), invalid: 123, abc, $%,1sdf, empty} |
| **Steps:** | |  |  | | --- | --- | | **Test Input** | **Expected Output** | | Visit Home page. |  | | Click on Login link on the top. | Login page should be displayed. | | Enter invalid Email ID. | Error message should be displayed as ‘Invalid Email ID’ | | Enter invalid password. | Error message should be displayed as ‘Password should be minimum 7 characters, alphanumeric with one special character’ | | Click on ‘Login’ button. | User landing page should be displayed. | |
| **Notes and Questions:** |  |
| **Test Results:** | |  |  |  | | --- | --- | --- | | **Date** | **Pass/Fail** | **Remarks** | |  |  |  | |  |  |  | |  |  |  | |

# **1.2 Test Case Title:**

**Verify first name, last name, email ID and password for the sign-in page**

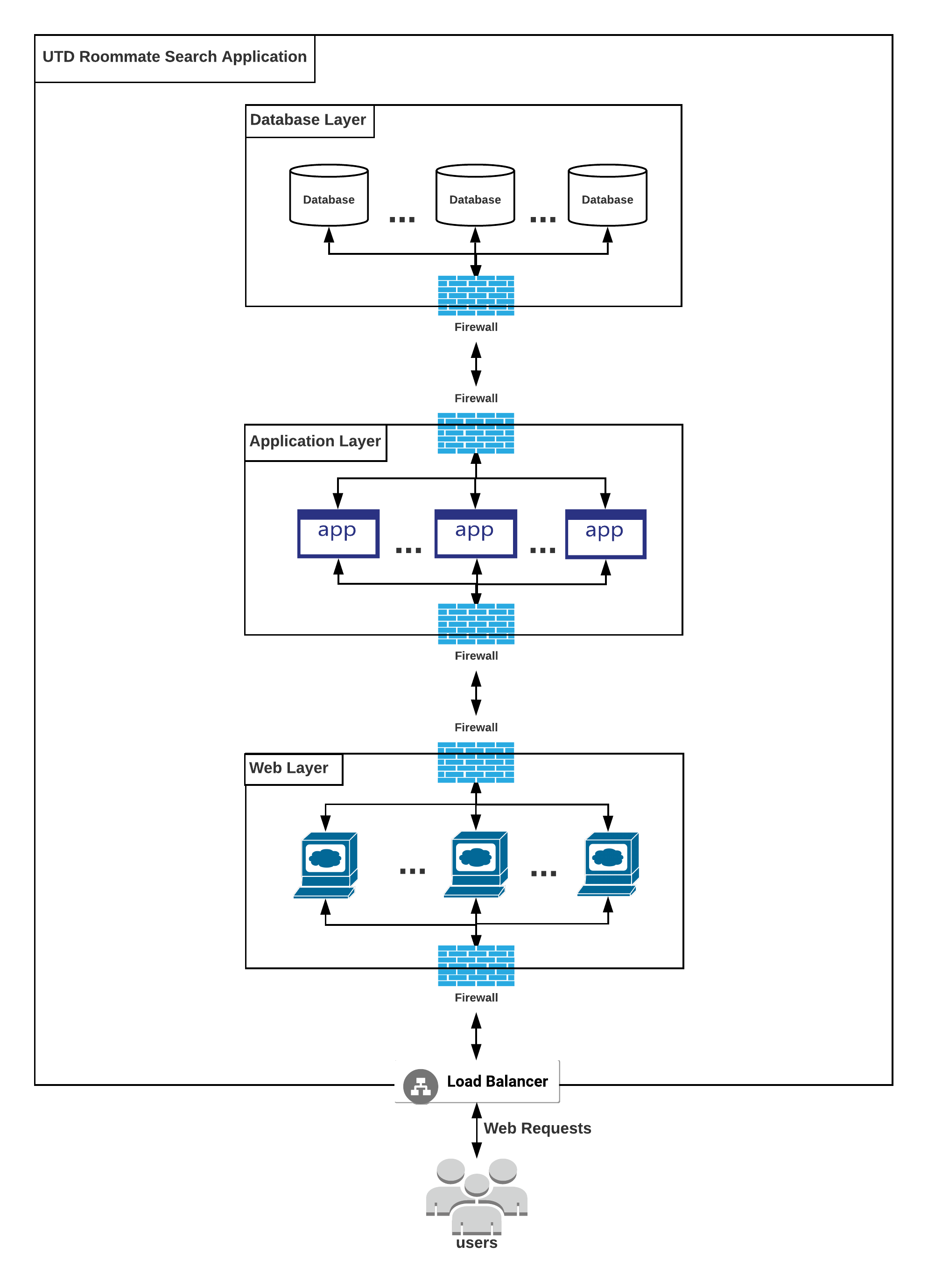
|  |  |
| --- | --- |
| **Purpose:** | To verify whether user (UTD students) can sign-in into the application successfully. |
| **Prereq:** | Users (UTD students) who are new to the application and do not have an account. |
| **Test Data:** | First name = {valid: abcdef, abc1, vb1. (minimum 4 Characters), invalid: abc}  Last name = {valid: abcdef, a123, %^&^& (minimum 4 Characters), invalid: abc}  Email ID = {valid: xyz123@utdallas.edu, invalid: @utdallas.edu, %$#@.com, empty}  Password = {valid: abc123& (minimum 7 characters, alphanumeric with one special character), invalid: 123, abc, $%,1sdf, empty} |
| **Steps:** | |  |  | | --- | --- | | **Test Input** | **Expected Output** | | Visit Home page. |  | | Click on Sign-in link on the top. | Sign-in page should be displayed. | | Enter invalid first name. | Error message should be displayed as ‘Minimum 4 characters’ | | Enter invalid last name. | Error message should be displayed as ‘Minimum 4 characters’ | | Enter invalid Email ID. | Error message should be displayed as ‘Invalid Email ID’ | | Enter invalid password. | Error message should be displayed as ‘Password should be minimum 7 characters, alphanumeric with one special character’ | | Click on ‘Create an account’ button. | The terms of use pop-up page should be displayed. | | Click agree radio button at page bottom | The tick should be displayed. | | Click submit button. | New user landing page should be displayed with welcome message. | |
| **Notes and Questions:** |  |
| **Test Results:** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **vDate** | **Time** | **Unit/Line** | **Pass/Fail** | **Remarks** | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

# **1.3 Test Case Title:**

**Verify payment page when Non- UTD user logs into the application for the first time.**

|  |  |
| --- | --- |
| **Purpose:** | To verify payment page for Non-UTD user. |
| **Prereq:** | Non- UTD user logs into the application for the first time. |
| **Test Data:** | Card type: Visa, Mastercard, American Express, Discover  Card number = {valid: 16-digit number, invalid: incorrect digits, digits less than 16}  Expiration date = {valid: month and year, invalid: year and month older than current month and year}  CVV = {valid:319, invalid: 1, 02, abc, %^^, empty}  Cardholder name: Minimum 20 characters  Country: Select the country name from the drop-down  State: {valid: Texas, Arizona Invalid: TX, AZ, T, TTYYY}  Contact Phone number: {Valid:10digit number, invalid: 9digit number, &^&, agjh,empty}  Email address: {valid: xyz123@utdallas.edu, invalid: @utdallas.edu, %$#@.com, empty} |
| **Steps:** | |  |  | | --- | --- | | **Test Input** | **Expected Output** | | Visit new user landing page. | Pop-up page should be displayed. | | Click on ‘Make payment’ button. | Payment page should be displayed. | | No card type selected | Error message should be displayed as ‘Please select the card type’. | | Enter invalid card number. | Error message should be displayed as ‘Minimum 4 characters’ | | Enter invalid expiration date. | Error message should be displayed as ‘Minimum 4 characters’ | | Enter invalid security code/CVV. | Error message should be displayed as ‘Invalid Email ID’ | | Enter invalid cardholder name | Error message should be displayed as ‘Minimum 20 characters’ | | Select no country from the drop-down | Error message should be displayed as ‘Please select a country’. | | Enter invalid state | Error message should be displayed as ‘Please enter a valid state’. | | Enter invalid contact phone number | Error message should be displayed as ‘Please enter a 10- digit contact number’. | | Enter invalid email address | Error message should be displayed as ‘Invalid Email ID’ | | Click on ‘Submit’ button. | Payment successful page should be displayed with order ID. | |
| **Notes and Questions:** |  |
| **Test Results:** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Date** | **Time** | **Unit/Line** | **Pass/Fail** | **Remarks** | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

# Architecture diagram



# MINUTES OF MEETINGS

## Minutes of Meeting 1

Date: April 23rd, 2018

Location: JSOM

Present: Mayank Madhav

Prema Siddu

Rutwika Mohanty

Tejasvi Ramadas Sagar

Agenda:

* To construct software design by choosing three methods from our UTD roommate search application
* To develop test cases for each of the methods specified for software design.

Action Items:

* Discussed on which three methods to create software design and algorithm for each module.
* Discussed types of testing to use for our UTD roommate search application.

Decisions Reached:

* Brainstormed algorithm steps for each method.
* Created software design for each module and constructed test cases for these modules
* Assigned tasks to each member further progress in the respective topics.

# Minutes of Meeting 2

Date: April 27th, 2018

Location: JSOM

Present: Mayank Madhav

Prema Siddu

Rutwika Mohanty

Tejasvi Ramadas Sagar

Agenda:

* To create a software design and algorithm for each module.
* To discuss test strategy and testing in detail

Decisions Reached:

* The User interface design was developed and design was finalized.
* Testing strategy and approach was determined.
* Test scenario was discussed and flow was identified.