#### Hotel Reservation System(Travel BNH)

#### Hamdan Shakeel (BCS151040)

#### Sarmad Hassan (BCS151078)

#### Muhammad Manan (BCS151035)



**Spring-2018**

**Supervised By**

**Dr. Abdul Qadir**

**Department of Computer Science**

**Capital University of Science & Technology, Islamabad**

|  |  |  |
| --- | --- | --- |
|  | **Submission Form for Final-Year**  **PROJECT REPORT** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Version** | | V 1.0 | |  | **NUMBER OF MEMBERS** | 3 |
|  | | | | | | |
| **TITLE** | Hotel Reservation System(Travel BnH) | | | | | |
|  | | | | | | |
| **SUPERVISOR NAME** | | | Dr. Abdul Qadir | | | |

|  |  |  |
| --- | --- | --- |
| **MEMBER NAME** | **REG. NO.** | **EMAIL ADDRESS** |
| Muhammad Manan | BCS151035 | Mananabbasi974@gmail.com |
| Hamdan Shakeel | BCS151040 | Hamdanshakeel20@gmail.com |
| Sarmad Hassan | BCS151078 | Sarmadhassan25@gmail.com |

|  |  |  |
| --- | --- | --- |
| **MEMBERS’ SIGNATURES** |  | |
|  |  | **Supervisor’s Signature** |
|  |  |
|  |  |

Table of Contents

[Chapter 1 4](#_Toc512861178)

[Introduction 4](#_Toc512861179)

[1.1 Project Introduction 4](#_Toc512861180)

[1.2 Existing Examples / Solutions 4](#_Toc512861181)

[1.3 Business Scope 5](#_Toc512861182)

[1.4 Useful Tools and Technologies 5](#_Toc512861183)

[1.5 Project Work Break Down 6](#_Toc512861184)

[Chapter 2 8](#_Toc512861185)

[Requirement Specification and Analysis 8](#_Toc512861186)

[2.1 Functional Requirements 8](#_Toc512861187)

[2.2 Selected Requirement List 11](#_Toc512861188)

[2.3.1 Hotel Manager (HM) Usecase Diagram 13](#_Toc512861189)

[2.3.2 Living Room Manager(LRM) Usecase Diagram 14](#_Toc512861190)

[2.3.3 Customer Use Case 15](#_Toc512861191)

[2.3.4 Service Manager Use Case: 16](#_Toc512861192)

[2.4 Use Case Description 17](#_Toc512861193)

[2.6 System Sequence diagram 44](#_Toc512861194)

[2.6 Domain Model 69](#_Toc512861195)

[Chapter 3 70](#_Toc512861196)

[System Design 70](#_Toc512861197)

[3.1 Multitier Architecture 70](#_Toc512861198)

[3.1.1 Presentation Layer 70](#_Toc512861199)

[3.1.2 Business Logic Layer 71](#_Toc512861200)

[3.1.3 Database Layer 71](#_Toc512861201)

[3.2 Class Diagram 73](#_Toc512861202)

[3.3 Sequence Diagram 74](#_Toc512861203)

[3.3.1 Customer, Hotel room Manager, Service Manager, Living room Manager 74](#_Toc512861204)

[3.4 Entity Relationship Diagram 86](#_Toc512861205)

[3.5 Database Schema 87](#_Toc512861206)

[3.6 User Interface 88](#_Toc512861207)

# Chapter 1

# Introduction

Hotel Reservation System will provide a platform for tourist to find the best hotel in the region and make booking. It will be easy for users to book hotel online and living rooms.

1.1 Project Introduction

Hotel Reservation System is a two sided platform it facilitates the process of booking private living rooms and hotels. People face many problem regarding staying when they visit outstations for enjoyment or business point of view. They search for hotels and ask people to give them rooms because they have no idea regarding the place, so due to this issue they face many problems like time-wastage & they can’t get the rooms according to their living standards. On the other side the owner of the hotel has hired some people who bring the visitors to the hotel and they get their high percentage from the owner and visitors as well.

Keeping this in view, there is a need for an online platform that helps the people to find best hotel or the living room in the region and book online, the owner posts the picture of living rooms to notify the people the accommodation of the rooms and their respective packages.

The Visitor will give their reviews after they checkout the hotel or living room. This feature will give information to other visitors about quality of hotel or living rooms. Owner will get rid of the dealers who get high percentage, and the visitor’s will get the reward point according to the days they live in hotel’s/living rooms like concession in cash & promised breakfast.

# 1.2 Existing Examples / Solutions

There are applications that provide such functionalities:

* Expedia

Expedia provide hotel booking service worldwide, excluding Pakistan. They do not have their own cars and buses for traveling.

* Airbnb

Airbnb provides hotel booking system worldwide, excluding Pakistan.

Hotel Reservation System have transport service at door step to destination.

There is no existing application.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Characteristics** | **Airbnb** | **Expedia** | **Hotel Reservation System(Proposed System)** |
| **1** | Sign Up |  |  |  |
| **2** | Sign In |  |  |  |
| **3** | Post Hotel Rooms |  |  |  |
| **4** | Post Living Rooms |  |  |  |
| **5** | Review |  |  |  |
| **6** | Reward Point |  |  |  |
| **7** | Book Hotel Rooms |  |  |  |
| **8** | Book Living Rooms |  |  |  |
| **9** | Provide Transport |  |  |  |
| **10** | Bill Payment |  |  |  |

# 1.3 Business Scope

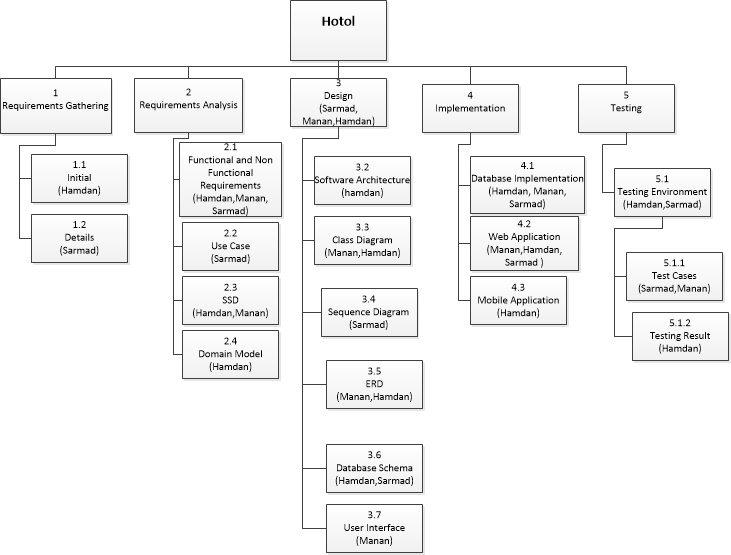
Now a day’s people face many problems when they visit a place where they have not enough knowledge. Due to this issue they face many problems in staying on that place. The Hotel Reservation System (Online Application) help the users to find the hotel rooms/ living rooms and book. Hotel Reservation System will also helpful for the owners of the hotel and the owners of the private living rooms. They can post their rooms in our application this will help the owners of the hotel they can end the booking department of their hotel. The Hotel Reservation System gives reward point to both clients and the hosts. On every successful booking the host and the client gain reward points.

# 1.4 Useful Tools and Technologies

Following are the list of all the possible technologies that will be required during the designing, developmental and testing phases of the project:

* **Programming Language:**
* **C#** will be used at backend for web application.
* **HTML5, CSS3, AngularJS** will be used at frontend for web application.
* **Java** will be used to build android application.
* **Development Environment:**
* **Visual Studio:** It will be used for the development, debugging and simulation of the webserver.
* **Android Studio:** It will be used for the development, debugging and simulation of the android App.
* **Database:**
* **SQL Server:** It will be used to store data, as it is compatible with Microsoft products.
* **SQLite:** It will be used to store data, as it is compatible with android applications.

# 1.5 Project Work Break Down



# Chapter 2

# Requirement Specification and Analysis

Requirements analysis is a process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. In Chapter 2 we will enlist the functional and non-functional requirements and model functional requirements in the form of use case model.

# 2.1 Functional Requirements

A functional requirement defines a function of a system or its component. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

**Actors**

* Hotel Manager (HM)
* Living room manager (LRM)
* Customer
* Service Manager(SM)

**Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref-No** | **Functional Requirements** | **Status** | **Quality Attributes** |
| 1 | HM / LRM should be able to register. | Pending | 1. Usability |
| 2 | HM/LRM post hotel room pictures and information. | Pending | 1. Security w.r.t authentication, authorization 2. Usability 3. Performance |
| 3 | HM/ LRM view the information of customer who book the room. | pending | 1. Security w.r.t authentication, authorization 2. Usability |
| 4 | HM / LRM insert, update, delete and view the catalog of the rooms. | pending | 1. Usability 2. Security w.r.t authentication, authorization. 3. Performance |
| 5 | HM / LRM view the status of the rooms.  (whether occupied or not) | pending | 1. Security w.r.t authorization, authentication.   2. Usability. |
| 6 | HM / LRM give information about facilities (parking). | pending | 1. Usability |
| 7 | HM / LRM approve the booking requests of customers. | pending | 1. Security w.r.t authorization, authentication.  2. Usability. |
| 8 | Customer should be able to view information about Hotel rooms / private living rooms (Facilities & Status). | pending | 1. Usability 2. Security w.r.t authentication, authorization. 3. Performance |
| 9 | Customer should be able to book hotel rooms/private living rooms. | pending | 1. Usability 2. Security w.r.t authentication, authorization. 3. Performance |
| 10 | Customer should be able to rate the hotel room/living room. | pending | 1. Security w.r.t authorization, authentication. 2. Usability. |
| 11 | Customer should be able to pay bill. | pending | 1. Security w.r.t authentication, authorization 2. Usability |
| 12 | Customer should be able to view location of hotels/living rooms. | pending | 1. Usability |
|  |  |  |  |
| 14 | Service Manager should be able to insert/update/delete HM/LRM. | pending | 1. Security w.r.t authentication, authorization. |
| 15 | Service Manager should be able to provide transport catalog. | pending | 1. Security w.r.t authentication, authorization. |
| 16 | Service Manager should be able to view all information of hotels/living rooms. | pending | 1. Usability. 2. Security w.r.t authentication, authorization. |
| 17 | Service Manager should be able to view all information of customers. | pending | 1. Usability. 2. Security w.r.t authentication, authorization. |
| 18 | Service Manager approve the requests of HMs/LRM. | pending | 1. Usability. 2. Security w.r.t authentication, authorization. |
| 19 | User’s view the ranking of the hotels/private living rooms. | pending | 1. Usability. 2. Security w.r.t authentication, authorization. |

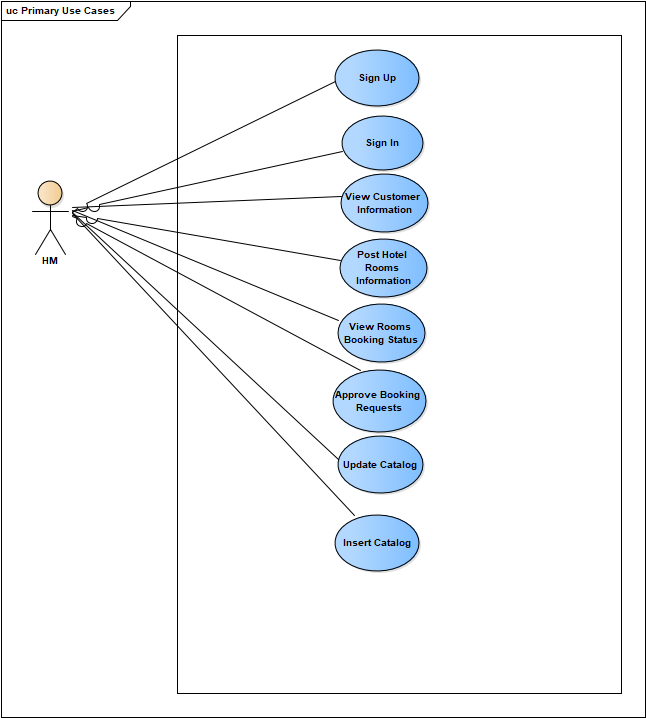
# 2.2 Selected Requirement List

**Features implement in First Iteration**

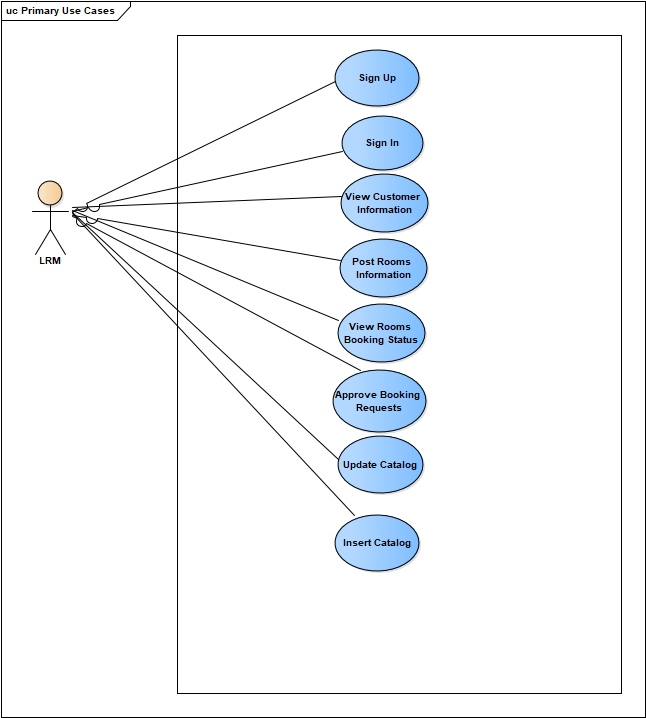
|  |  |
| --- | --- |
| **Ref#** | **Features** |
| 1 | User Management System |
| 2 | Manager Post hotel rooms/living rooms |
| 3 | Customer view the hotel rooms |
| 4 | Service Manager view the information of hotels and customers |

**2.3 USE CASE MODELING**

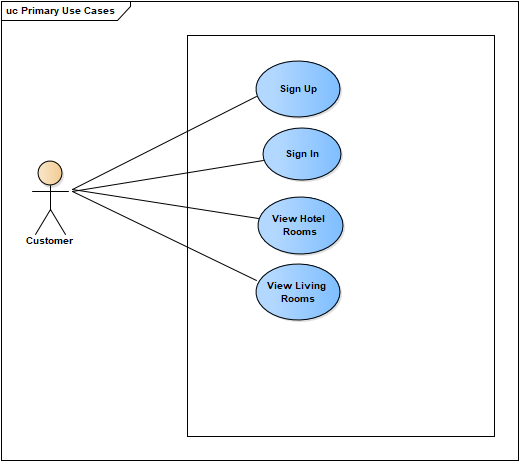
# 2.3.1 Hotel Manager (HM) Usecase Diagram



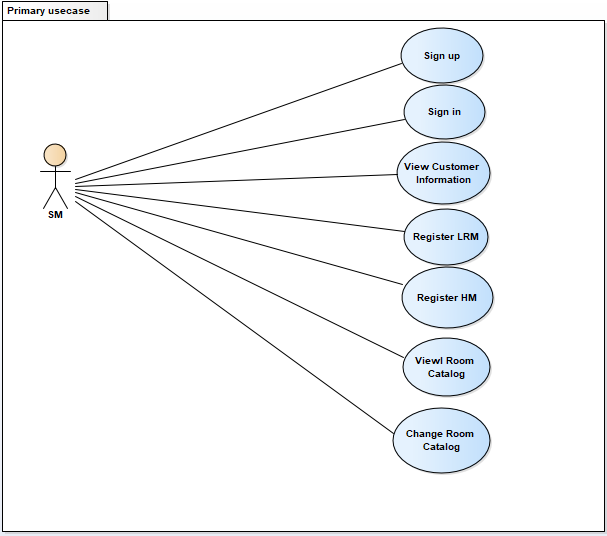
# 2.3.2 Living Room Manager(LRM) Usecase Diagram

****

# 2.3.3 Customer Use Case

****

# 2.3.4 Service Manager Use Case:



2.4 Use Case Description

**Hotel Manager (HM) Description Table:**

**Table 1: HM Sign Up**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | Uc1 | | | |
| **Use Case Name:** | Signup | | | |
| **Created By:** | Sarmad Hassan | **Last Updated By:** | | Muhammad Manan |
|  |  |  | |  |
| **Date Created:** | *31/03/2018* | **Last Revision Date:** | | *04/04/2018* |
| **Actors:** | Hotel Manager (HM) | | | |
| **Description:** | The HM (he/she) can sign up first time by providing a Username, Email, Password, Type. | | | |
| **Trigger:** | Register / Signup | | | |
| **Pre-conditions:** | HM have System (PC) in running condition and application in running state. | | | |
| **Post conditions:** | HM Signup / registered successfully. | | | |
| **Normal Flow:** | **HM action** | | **System action** | |
|  | 1. HM Selects HM Signup form option. | | 1. System displays form containing (Username, Email, Password and type) fields. | |
| 1. HM fills all fields and clicks Signup button**.** | | 1. System Send Confirmation email to user. | |
|  |  | | 1. System receives confirmation message(if user clicks on confirmation email ). | |
|  |  | | 1. System will confirm Succesfully Signup activity by sending message to the user. | |
| **Alternative Flows:** | If HM can’t fill all fields of Signup form then system not register HM**.** | | | |

**Table 2: HM Sign In**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | Uc2 | | | |
| **Use Case Name:** | Sign-in | | | |
| **Created By:** | Sarmad Hassan | **Last Updated By:** | | Muhammad Manan |
|  | Muhammad Manan |  | | Sarmad Hassan |
|  |  |  | | Hamdan Shakeel |
| **Date Created:** | *31/03/2018* | **Last Revision Date:** | | *04/04/2018* |
| **Actors:** | Hotel Manager (HM) | | | |
| **Description:** | HM (he/she) can sign-in if HM is registered user. | | | |
| **Trigger:** | Sign in | | | |
| **Preconditions:** | HM system in running state. HM should be registered user. | | | |
| **Post conditions:** | HM will be able to view application catalogs containing features. | | | |
| **Normal Flow:** | **HM action** | | **HM action** | |
|  | 1. HM selects sign-in form. | | 1. System displays form containing (Email, Password) fields. | |
| 1. HM fills all fields and clicks Sign-in button. | | 1. System checks HM is registered user by verifying its email and password. | |
|  |  | | 1. System display application catalog containing features to user. | |
| **Alternative Flows:** | 1. HM fills Sign-in form incompletely and system can’t verify HM. 2. HM fills the field with wrong enteries.   In above cases system can’t provide application features to HM | | | |

**Table 3: HM Post Hotel Room Information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc3* | | | |
| ***Use Case Name:*** | *Post Hotel Room Information* | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Hamdan Shakeel  Muhammad Manan |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *4/4/2018* |
| ***Actors:*** | Hotel Manager (HM) | | | |
| ***Description:*** | HM can post the description and pictures of hotel rooms. | | | |
| ***Trigger:*** | Post Hotel room information. | | | |
| ***Pre-conditions:*** | HM should be signed in. HM should be in insert catalog. HM have hotel rooms pictures and description. | | | |
| ***Post-conditions:*** | HM post pictures and description of rooms successfully. | | | |
| ***Normal Flow:*** | **HM** | | **System** | |
|  | 1. HM selects Insert catalog option. | | 1. System provides form containing (post room pictures, post description, mattress quantity, no of beds, pillow quantity, post room rent-amount) fields. | |
| 1. HM fills the form and clicks submit option. | | 1. System validate fields and send confirmation message to user. | |
|  | |  | |
| ***Alternative Flows:*** | HM Selects wrong catalog. | | | |

**Table 4: HM-View Customers Information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc4* | | | |
| ***Use Case Name:*** | *View Customer Information* | | | |
| ***Created By:*** | Sarmad Hassan  Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan  Hamdan Shakeel |
| ***Date Created:*** | *31/3/2018* | ***Last Revision Date:*** | | *4/4/2018* |
| ***Actors:*** | Hotel Manager (HM) | | | |
| ***Description:*** | HM will select view Customer catalog to view customer’s information and details. | | | |
| ***Trigger:*** | View customer Information. | | | |
| ***Pre-conditions:*** | HM should be signed in. HM Should be in view Customer catalog. | | | |
| ***Post-conditions:*** | HM will be able to view all the details and information of the customer. | | | |
| ***Normal Flow:*** | **HM** | | **System** | |
|  | 1. HM clicks on view  customer catalog option. | | 1. System displays customer view catalog. | |
| 1. HM selects view customer list option. | | 1. System displays form containing (customer-id, booking status, room booked type, customer checked-in date, customer checked-out date ) | |
| 1. HM select customer to view detail. | | 1. System displays customer information in grid view selected by user. | |
| ***Alternative Flows:*** | Database is not responding | | | |

**Table 5: View Room booking status**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc5* | | | |
| ***Use Case Name:*** | *View Room Booking status* | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Hamdan Shakeel |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Hotel Manager (HM) | | | |
| ***Description:*** | HM can view room booking status either room is vacant or booked. | | | |
| ***Trigger:*** | View room status. | | | |
| ***Preconditions:*** | HM must Signed-in. HM must have catalogs options. | | | |
| ***Post conditions:*** | HM will now be able to view the status of the room. | | | |
| ***Normal Flow:*** | **HM** | | **System** | |
|  | 1. HM selects view room status Catalog option. | | 1. The system displays form containing information room status (booked, vacant), mattress quantity, rent paid status, beds quantity. | |
| 1. HM selects room to view its status. | | 1. System displays room status (Booked or Vacant). | |
|  |  | |  | |
| ***Alternative Flows:*** | Database is not responding.  HM cancels the current form. | | | |

**Table 6: Approve Room Booking Request**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | ***Uc6*** | | | |
| ***Use Case Name:*** | ***Approve Room Booking Request*** | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Hotel Manager (HM) | | | |
| ***Description:*** | HM will approve the room booking request of the customer by checking status of room, and either customer have paid the bill. | | | |
| ***Trigger:*** | Approve room booking request. | | | |
| ***Preconditions:*** | HM should be signed in. HM will be in room booking requests list option. Room which customer want to book should be vacant. Customer Should pay the bill. | | | |
| ***Post conditions:*** | HM approved customer successfully. | | | |
| ***Normal Flow:*** | **HM** | | **System** | |
|  | 1. HM selects room Booking requests option. | | 1. System will Display the form containing list of rooms. | |
| 1. HM will select certain room. | | 1. System will display bill invoice form containing information’s of customer (room id, customer info, bill transaction, booking date). | |
|  | 1. HM will check which customer have paid the bill and select that customer and click confirm button. | | 1. System will send confirmation message to the customer and HM. | |
| ***Alternative Flows:*** | If customer invoice status is not confirmed system will not send conformation message.  Database is not responding. | | | |

**Table 7: Insert Hotel Room Catalog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | **Uc7** | | | |
| ***Use Case Name:*** | *Insert Hotel Room Catalog* | | | |
| ***Created By:*** | *Sarmad Hassan*  *Muhammad Manan* | ***Last Updated By:*** | | *Sarmad Hassan*  *Muhammad Manan* |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Hotel Manager (HM) | | | |
| ***Description:*** | HM will insert hotel room pictures in the catalog. HM will insert information and description of hotel rooms its facilities and features. | | | |
| ***Trigger:*** | Insert hotel room catalog. | | | |
| ***Preconditions:*** | HM must be Signed in. HM must be in Insert hotel catalog. | | | |
| ***Post conditions:*** | Hotel room catalog Inserted succesfully. | | | |
| ***Normal Flow:*** | **HM** | | **System** | |
|  | 1. HM will click on the insert hotel room catalog option. | | 1. System will display the Insert catalog interface. | |
| 1. HM selects insert option and insert that hotel room’s pictures and description. | | 1. System display the catalog containing (pictures and description) of the hotel. | |
| ***Alternative Flows:*** | HM will select insert hotel room catalog and without inserting clicks insert option. | | | |
|  | Database is not responding. | | | |

**Table 8: Update Hotel Room Catalog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | **Uc8** | | | |
| ***Use Case Name:*** | *Update Hotel Room Catalog* | | | |
| ***Created By:*** | *Sarmad Hassan*  *Muhammad Manan* | ***Last Updated By:*** | | *Sarmad Hassan*  *Muhammad Manan* |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Hotel Manager (HM) | | | |
| ***Description:*** | HM will insert hotel room pictures in the catalog. HM will insert information and description of hotel rooms its facilities and features. | | | |
| ***Trigger:*** | Update hotel room catalog. | | | |
| ***Preconditions:*** | HM must be Signed in. HM must be in update hotel catalog. | | | |
| ***Post conditions:*** | Hotel room catalog updated succesfully. | | | |
| ***Normal Flow:*** | **HM** | | **System** | |
|  | 1. HM will click on the update hotel room catalog option. | | 1. System will display the update catalog interface. | |
| 1. HM selects update option and update that hotel room’s pictures and description. | | 1. System display the catalog containing (pictures and description) of the hotel. | |
| ***Alternative Flows:*** | HM will select update hotel room catalog and without inserting clicks update option. | | | |
|  | Database is not responding. | | | |

**Living Room Manager (LRM) Description Tables:**

**Table 9: LRM­-Signup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc9* | | | |
| ***Use Case Name:*** | *Signup* | | | |
| ***Created By:*** | *Sarmad Hassan* | ***Last Updated By:*** | | *Muhammad Manan* |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Living Room Manager (LRM) | | | |
| ***Description:*** | The LRM (he/she) can sign up first time by providing a Username, Email, Password, Type. | | | |
| ***Trigger:*** | Register / Signup | | | |
| ***Preconditions:*** | LRM have System (PC) in running condition and application in running state. | | | |
| ***Post conditions:*** | LRM Signup / registered successfully. | | | |
| ***Normal Flow:*** | ***LRM*** | | ***System*** | |
|  | 1. LRM Selects HM Signup form option. | | 1. System displays form containing (Username, Email, Password and type) fields. | |
| 1. LRM fills all fields and clicks Signup button**.** | | 1. System Send Confirmation email to user. | |
|  |  | | 1. System receives confirmation message (if user clicks on confirmation email ). | |
|  |  | | 1. System will confirm Succesfully Signup activity by sending message to the user. | |
| ***Alternative Flows:*** | If LRM can’t fill all fields of Signup form then system not register LRM**.** | | | |

***Table 10: LRM Sign In***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | Uc10 | | | |
| **Use Case Name:** | Sign-in | | | |
| **Created By:** | Sarmad Hassan | **Last Updated By:** | | Muhammad Manan |
|  | Muhammad Manan |  | | Sarmad Hassan |
|  |  |  | | Hamdan Shakeel |
| **Date Created:** | *31/03/2018* | **Last Revision Date:** | | *04/04/2018* |
| **Actors:** | Living room Manager(LRM). | | | |
| **Description:** | HM (he/she) can sign-in if LRM is registered user. | | | |
| **Trigger:** | Sign in | | | |
| **Preconditions:** | LRM system in running state. LRM should be registered user. | | | |
| **Post conditions:** | HM will be able to view application catalogs containing features. | | | |
| **Normal Flow:** | **LRM action** | | **System action** | |
|  | 1. LRM selects sign-in form. | | 1. System displays form containing (Email, Password) fields. | |
| 1. LRM fills all fields and clicks Sign-in button. | | 1. System checks LRM is registered user by verifying its email and password. | |
|  |  | | 1. System display application catalog containing features to user. | |
| **Alternative Flows:** | LRM fills Sign-in form incompletely and system can’t verify LRM.  LRM fills the field with wrong enteries.  In above cases system can’t provide application features to LRM. | | | |

**Table 11: LRM-post Living room information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc11* | | | |
| ***Use Case Name:*** | *Post Living Room Information* | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan |
| ***Date Created:*** | *31/3/2018* | ***Last Revision Date:*** | | *4/4/2018* |
| ***Actors:*** | Living room Manager (LRM) | | | |
| ***Description:*** | LRM can post the description and pictures of hotel rooms. | | | |
| ***Trigger:*** | Post Living room information. | | | |
| ***Pre-conditions:*** | LRM should be signed in. LRM should be in insert catalog. LRM have hotel rooms pictures and description. | | | |
| ***Post-conditions:*** | LRM post pictures and description of rooms successfully. | | | |
| ***Normal Flow:*** | **LRM** | | **System** | |
|  | 1. LRM selects Insert catalog option. | | 1. System provides form containing (post room pictures, post description, mattress quantity, no of beds, pillow quantity, post room rent-amount) fields. | |
| 1. LRM fills the fields and clicks submit option. | | 1. System validate fields and send confirmation message to user. | |
|  | |  | |
| ***Alternative Flows:*** | LRM Selects wrong catalog. | | | |

**Table 12: LRM-View Customers Information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc12* | | | |
| ***Use Case Name:*** | *View Customers Information* | | | |
| ***Created By:*** | Sarmad Hassan  Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan  Hamdan Shakeel |
| ***Date Created:*** | *31/3/2018* | ***Last Revision Date:*** | | *4/4/2018* |
| ***Actors:*** | Living room Manager(LRM) | | | |
| ***Description:*** | LRM will select view Customer catalog to view customer’s information and details. | | | |
| ***Trigger:*** | View customer Information. | | | |
| ***Pre-conditions:*** | LRM should be signed in. LRM Should be in view Customer catalog. | | | |
| ***Post-conditions:*** | LRM will be able to view all the details and information of the customer. | | | |
| ***Normal Flow:*** | **LRM** | | **System** | |
|  | LRM clicks on view  customer catalog option. | | System displays customer view catalog. | |
| LRM selects view customer list option. | | System displays form containing (customer-id, booking status, room booked type, customer checked-in date, customer checked-out date ) | |
| LRM selects customer to view detail. | | System displays customer information in grid view selected by user. | |
| ***Alternative Flows:*** | Database is not responding | | | |
|  | HM select wrong catalog | | | |

**Table 13: View Room booking status**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc13* | | | |
| ***Use Case Name:*** | *View Room Booking status* | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Hamdan Shakeel |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Living room Manager (LRM) | | | |
| ***Description:*** | LRM can view room booking status either room is vacant or booked. | | | |
| ***Trigger:*** | View room status. | | | |
| ***Preconditions:*** | LRM must Signed-in. LRM must have catalogs options. | | | |
| ***Post conditions:*** | LRM will now be able to view the status of the room. | | | |
| ***Normal Flow:*** | **LRM** | | **System** | |
|  | 1. LRM selects view room status Catalog option. | | 1. The system displays form containing information room status (booked, vacant, mattress quantity, rent paid status, beds quantity). | |
| 1. LRM selects room to view its status. | | 1. System displays room status (Booked or Vacant). | |
|  |  | |  | |
| ***Alternative Flows:*** | Database is not responding.  LRM cancels the current form. | | | |

**Table 14: Approve Room booking Request**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc14* | | | |
| ***Use Case Name:*** | ***Approve Room Booking Request*** | | | |
| ***Created By:*** | Muhammad Manan  Sarmad Hassan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Living room Manager (LRM) | | | |
| ***Description:*** | LRM will approve the room booking request of the customer by checking status of room, and either customer have paid the bill. | | | |
| ***Trigger:*** | Approve room booking request. | | | |
| ***Preconditions:*** | LRM should be signed in. LRM will be in room booking requests list option. Room which customer want to book should be vacant. Customer Should pay the bill. | | | |
| ***Post conditions:*** | LRM approved customer successfully. | | | |
| ***Normal Flow:*** | **LRM** | | **System** | |
|  | 1. LRM selects room Booking requests option. | | 1. System will Display the form containing list of rooms. | |
| 1. LRM will select certain room. | | 1. System will display bill invoice form containing information’s of customer (room id, customer info, bill transaction, booking date). | |
|  | 1. LRM will check which customer have paid the bill and select that customer and click confirm button. | | 1. System will send confirmation message to the customer and LRM. | |
| ***Alternative Flows:*** | If customer invoice status is not confirmed system will not send conformation message. | | | |

**Table 15: Insert Living Room Catalog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | **Uc15** | | | |
| ***Use Case Name:*** | *Insert Living Room Catalog* | | | |
| ***Created By:*** | *Sarmad Hassan*  *Muhammad Manan* | ***Last Updated By:*** | | *Sarmad Hassan*  *Muhammad Manan* |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Living room Manager (LRM) | | | |
| ***Description:*** | LRM will insert living room pictures in the catalog. LRM will insert information and description of hotel rooms its facilities and features. | | | |
| ***Trigger:*** | Insert hotel room catalog. | | | |
| ***Preconditions:*** | LRM must be Signed in. LRM must be in Insert hotel catalog. | | | |
| ***Post conditions:*** | Living room catalog Inserted succesfully. | | | |
| ***Normal Flow:*** | **LRM** | | **System** | |
|  | 1. LRM Selects insert hotel room catalog option. | | 1. System will display the Insert catalog interface. | |
| 1. LRM selects insert option and insert that hotel room’s pictures and description. | | 1. System display the catalog containing (pictures and description) of the hotel. | |
| ***Alternative Flows:*** | LRM will select insert hotel room catalog and without inserting clicks insert option. | | | |

**Table 16: Update Living Room Catalog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc16* | | | |
| ***Use Case Name:*** | *Update Living room Catalog* | | | |
| ***Created By:*** | *Sarmad Hassan*  *Muhammad Manan* | ***Last Updated By:*** | | *Sarmad Hassan*  *Muhammad Manan* |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Living room Manager (LRM) | | | |
| ***Description:*** | LRM will update hotel room pictures in the catalog. LRM will update information and description of hotel rooms its facilities and features. | | | |
| ***Trigger:*** | Update hotel room catalog. | | | |
| ***Preconditions:*** | LRM must be Signed in. LRM must be in update hotel catalog. | | | |
| ***Post conditions:*** | Living room catalog updated succesfully. | | | |
| ***Normal Flow:*** | **LRM** | | **System** | |
|  | 1. LRM selects update hotel room catalog option. | | 1. System will display the update catalog interface. | |
| 1. LRM selects update option and update that hotel room’s pictures and description. | | 1. System display the catalog containing (pictures and description) of the hotel. | |
| ***Alternative Flows:*** | LRM will select update hotel room catalog and without inserting clicks update option. | | | |
|  | Database is not responding. | | | |

**Service Manager (SM) Description Tables:**

**Table 17: SM-Signup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc17* | | | |
| ***Use Case Name:*** | *Signup* | | | |
| ***Created By:*** | *Sarmad Hassan* | ***Last Updated By:*** | | *Muhammad Manan* |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Service Manager (SM) | | | |
| ***Description:*** | The SM (he/she) can sign up first time by providing a Username, Email, Password, Type. | | | |
| ***Trigger:*** | Register / Signup | | | |
| ***Preconditions:*** | SM have System (PC) in running condition and application in running state. | | | |
| ***Post conditions:*** | SM Signup / registered successfully. | | | |
| ***Normal Flow:*** | ***SM*** | | ***System*** | |
|  | 1. SM Selects SM Signup form option. | | 1. System displays form containing (Username, Email, Password and type) fields. | |
| 1. SM fills all fields and clicks Signup button**.** | | 1. System Send Confirmation email to user. | |
|  |  | | 1. System receives confirmation message (if user clicks on confirmation email). | |
|  |  | | 1. System will confirm Succesfully Signup activity by sending message to the user. | |
| ***Alternative Flows:*** | If SM can’t fill all fields of Signup form then system not register SM**.** | | | |

***Table 18: SM Sign In***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | Uc18 | | | |
| **Use Case Name:** | Sign-in | | | |
| **Created By:** | Sarmad Hassan | **Last Updated By:** | | Muhammad Manan |
|  | Muhammad Manan |  | | Sarmad Hassan |
|  |  |  | | Hamdan Shakeel |
| **Date Created:** | *31/03/2018* | **Last Revision Date:** | | *04/04/2018* |
| **Actors:** | Service Manager (SM). | | | |
| **Description:** | SM (he/she) can sign-in if SM is registered user. | | | |
| **Trigger:** | Sign in | | | |
| **Preconditions:** | SM system in running state. SM should be registered user. | | | |
| **Post conditions:** | SM will be able to view application catalogs containing features. | | | |
| **Normal Flow:** | SM **action** | | **System action** | |
|  | 1. SM selects sign-in form. | | 1. System displays form containing (Email, Password) fields. | |
| 1. SM fills all fields and clicks Sign-in button. | | 1. System checks SM is registered user by verifying its email and password. | |
|  |  | | 1. System display application catalog containing features to user. | |
| **Alternative Flows:** | SM fills Sign-in form incompletely and system can’t verify SM.  SM fills the field with wrong enteries.  In above cases system can’t provide application features to SM. | | | |

**Table 19: SM-View Customers Information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc19* | | | |
| ***Use Case Name:*** | *View Customers Information* | | | |
| ***Created By:*** | Sarmad Hassan  Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan  Hamdan Shakeel |
| ***Date Created:*** | *31/3/2018* | ***Last Revision Date:*** | | *4/4/2018* |
| ***Actors:*** | Service Manager(SM) | | | |
| ***Description:*** | SM will select view Customer catalog to view customer’s information and details. | | | |
| ***Trigger:*** | View customer Information. | | | |
| ***Pre-conditions:*** | SM should be signed in. SM Should be in view Customer catalog. | | | |
| ***Post-conditions:*** | SM will be able to view all the details and information of the customer. | | | |
| ***Normal Flow:*** | **SM** | | **System** | |
|  | SM clicks on view  customer catalog option. | | System displays customer view catalog. | |
| SM selects view customer list option. | | System displays form containing (customer-id, booking status, room booked type, customer checked-in date, customer checked-out date ) | |
| SM selects customer to view detail. | | System displays customer information in grid view selected by user. | |
|  |  | |  | |
| ***Alternative Flows:*** | Database is not responding | | | |

***Table 20: SM Register (HM)***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc20* | | | |
| ***Use Case Name:*** | *Register HM* | | | |
| ***Created By:*** | Muhammad Manan  Sarmad Hassan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan  Hamdan Shakeel |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Service Manager (SM) | | | |
| ***Description:*** | HM will request the system to sign up, then SM will receive the request and Register the HM. | | | |
| ***Trigger:*** | Register HM. | | | |
| ***Preconditions:*** | SM must be Signed-in. SM will register HM when HM fill all fields of Sign-up form correctly. | | | |
| ***Post conditions:*** | SM will register HM successfully. | | | |
| ***Normal Flow:*** | **SM** | | **System** | |
|  | 1. SM will select the HM request list option. | | 1. System display the interface of request list of HM. | |
| 1. SM will select the HM from list and check that HM has completely filled the form fields (if form fields correctly and completely filled) , SM will click register option. | | 1. System will register HM Succesfully and send confirmation message to HM(Welcome! You’re registered succesfully). | |
|  |  | |  | |
| ***Alternative Flows:*** | Database is not responding. SM cancels the current form. | | | |

**Table 21: SM-Register (LRM)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc21* | | | |
| ***Use Case Name:*** | *Register LRM* | | | |
| ***Created By:*** | Muhammad Manan  Sarmad Hassan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan  Hamdan Shakeel |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Service Manager (SM) | | | |
| ***Description:*** | LRM will request the system to sign up, then SM will receive the request and Register the LRM. | | | |
| ***Trigger:*** | Register LRM. | | | |
| ***Preconditions:*** | SM must be Signed-in. SM will register LRM when LRM fill all fields of Sign-up form correctly. | | | |
| ***Post conditions:*** | SM will register LRM successfully. | | | |
| ***Normal Flow:*** | **SM** | | **System** | |
|  | 1. SM will select the LRM request list option. | | 1. System display the interface of request list of LRM. | |
| 1. SM will select the LRM from list and check that LRM has completely filled the form fields (if form fields correctly and completely filled) , SM will click register option. | | 1. System will register LRM Succesfully and send confirmation message to HM(Welcome! You’re registered succesfully). | |
| ***Alternative Flows:*** | Database is not responding.  SM cancels the current form. | | | |

**Table 22: View Room Catalog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc22* | | | |
| ***Use Case Name:*** | *View Room Catalog* | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Service Manager (SM) | | | |
| ***Description:*** | Service manager will view living room/Hotel room catalog containing information like description, features and facilities of the room. | | | |
| ***Trigger:*** | View Room Catalog. | | | |
| ***Preconditions:*** | SM must Signed-in. SM must have catalogs options whether to view hotel room or living room’s. | | | |
| ***Post conditions:*** | SM will now be able to view room catalogs. | | | |
| ***Normal Flow:*** | **SM** | | **System** | |
|  | 1. SM selects view room Catalog option. | | 1. The system displays the form to user either user wants to view Living room or Hotel room. | |
| 1. SM selects Type of room to view. | | 1. System displays form to user containing information room status, facilities, features and type. | |
|  |  | |  | |
| ***Alternative Flows:*** | Database is not responding.  SM cancels the current form. | | | |

**Table 23: Change Room Catalog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc23* | | | |
| ***Use Case Name:*** | *Change Room Catalog* | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Muhammad Manan |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Service Manager (SM) | | | |
| ***Description:*** | Service manager will Update living room/Hotel room catalog containing information like description, features and facilities of the room. | | | |
| ***Trigger:*** | Change Room Catalog. | | | |
| ***Preconditions:*** | SM must Signed-in. SM must have catalog options whether to Update hotel room or living room’s. | | | |
| ***Post conditions:*** | SM will now be able to Update room catalogs. | | | |
| ***Normal Flow:*** | **SM** | | **System** | |
|  | 1. SM selects change room Catalog option. | | 1. The system displays the form to user either user wants to Update (Living room or Hotel room). | |
| 1. SM selects Type of room to Update. | | 1. System displays updated form to user containing information room status, facilities, features and type. | |
|  |  | |  | |
| ***Alternative Flows:*** | Database is not responding.  SM cancels the current form. | | | |

**Customer Description Tables:**

**Table 24: Customer Signup**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc24* | | | |
| ***Use Case Name:*** | *Signup* | | | |
| ***Created By:*** | *Hamdan Shakeel* | ***Last Updated By:*** | | *Muhammad Manan* |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Customer | | | |
| ***Description:*** | The user can sign up first time by providing a Username, Email, Password, Type. | | | |
| ***Trigger:*** | Register / Signup | | | |
| ***Preconditions:*** | Customer have System in running condition and application in running state. | | | |
| ***Post conditions:*** | Customer Signup / registered successfully. | | | |
| ***Normal Flow:*** | ***Customer*** | | ***System*** | |
|  | 1. Customer Selects Customer Signup form option. | | 1. System displays form containing (Username, Email, Password and type) fields. | |
| 1. Customer fills all fields and clicks Signup button**.** | | 1. System Send Confirmation email to user. | |
|  |  | | 1. System receives confirmation message (if user clicks confirmation email). | |
|  |  | | 1. System will confirm Succesfully Signup activity by sending message to the user. | |
| ***Alternative Flows:*** | If user(customer) can’t fill all fields of Signup form then system not register user**.** | | | |

***Table 25: Customer Sign In***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | Uc25 | | | |
| **Use Case Name:** | Sign-in | | | |
| **Created By:** | Sarmad Hassan | **Last Updated By:** | | Muhammad Manan |
|  | Muhammad Manan |  | | Sarmad Hassan |
|  |  |  | | Hamdan Shakeel |
| **Date Created:** | *31/03/2018* | **Last Revision Date:** | | *04/04/2018* |
| **Actors:** | Customer. | | | |
| **Description:** | User (customer) can sign-in if customer is registered user. | | | |
| **Trigger:** | Sign in | | | |
| **Preconditions:** | Customer system in running state. Customer should be registered user. | | | |
| **Post conditions:** | Customer will be able to view application catalogs containing features. | | | |
| **Normal Flow:** | **Customer** | | **System action** | |
|  | 1. Customer selects sign-in form. | | 1. System displays form containing (Email, Password) fields. | |
| 1. Customer fills all fields and clicks Sign-in button. | | 1. System checks Customer is registered user by verifying its email and password. | |
|  |  | | 1. System display application catalog containing features . | |
| **Alternative Flows:** | Customer fills Sign-in form incompletely and system can’t verify Customer.  Customer fills the field with wrong enteries.  In above cases system can’t provide application features to Customer. | | | |

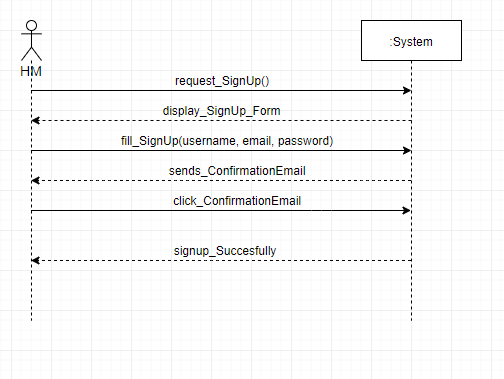
**Table 26: View Hotel Rooms**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc26* | | | |
| ***Use Case Name:*** | *View Hotel Rooms* | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Sarmad Hassan  Hamdan Shakeel |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Customer. | | | |
| ***Description:*** | Customer can view Hotel rooms either room is vacant or booked, room type and facilities. | | | |
| ***Trigger:*** | View Hotel rooms. | | | |
| ***Preconditions:*** | Customer must Signed-in. Customer must has catalog options. | | | |
| ***Post conditions:*** | Customer viewed hotel rooms succesfully. | | | |
| ***Normal Flow:*** | **Customer** | | **System** | |
|  | 1. Customer selects view Hotel rooms Catalog option. | | 1. The system displays form containing list of rooms. | |
| 1. Customer clicks certain room to view hotel room information. | | 1. System displays room information (room booked, vacant, mattress quantity, , room type, beds quantity ,facilities). | |
|  |  | |  | |
| ***Alternative Flows:*** | Database is not responding.  Customer cancels the current form. | | | |

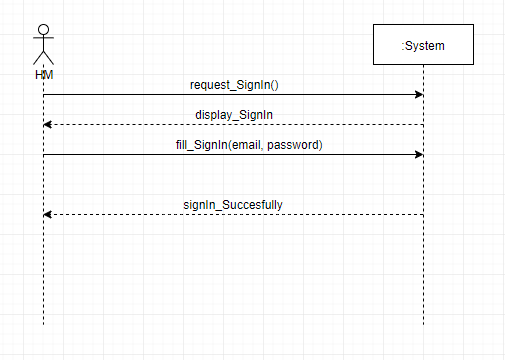
**Table 27: View Living Rooms**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Use Case ID:*** | *Uc27* | | | |
| ***Use Case Name:*** | *View Living Rooms* | | | |
| ***Created By:*** | Muhammad Manan | ***Last Updated By:*** | | Hamdan Shakeel |
| ***Date Created:*** | *31/03/2018* | ***Last Revision Date:*** | | *04/04/2018* |
| ***Actors:*** | Customer. | | | |
| ***Description:*** | Customer can view Living rooms either room is vacant or booked, room type and facilities. | | | |
| ***Trigger:*** | View Living rooms. | | | |
| ***Preconditions:*** | Customer must Signed-in. Customer must has catalog options. | | | |
| ***Post conditions:*** | Customer viewed Living room information succesfully. | | | |
| ***Normal Flow:*** | **Customer** | | **System** | |
|  | 1. Customer selects view Living rooms Catalog option. | | 1. The system displays form containing list of rooms. | |
| 1. Customer clicks certain room to view Living room information. | | 1. System displays room information (room booked, vacant, mattress quantity, , room type, beds quantity ,facilities). | |
|  |  | |  | |
| ***Alternative Flows:*** | Database is not responding.  Customer cancels the current form. | | | |

2.6 System Sequence diagram **HM Sign-Up:**

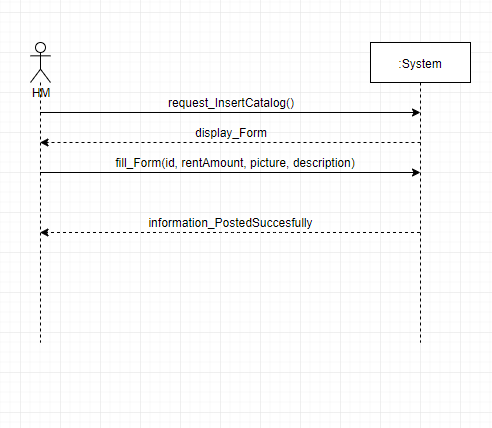
**Figure 2.1:** Signup

**HM Sign-in:**

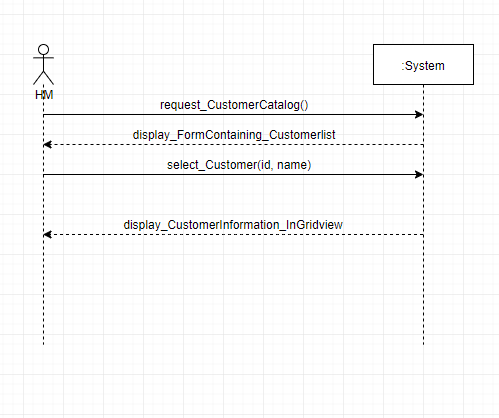
****

**Figure 2.2:** Sign in

**HM Post Hotel room information:**

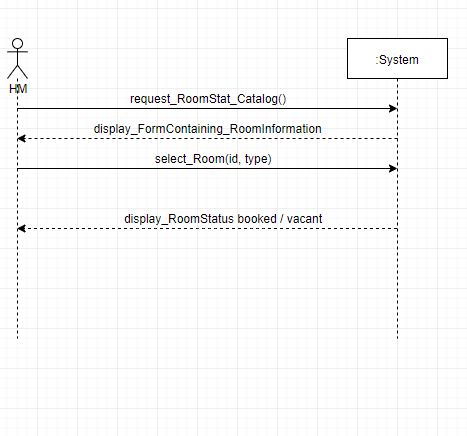
****

**Figure 2.3:** post hotel room information

**HM View Customer information:  
  
**

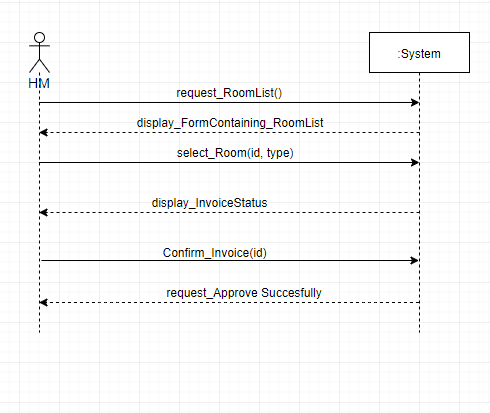
**Figure 2.4:** View Customer information

**HM View Room Booking Status:**

****

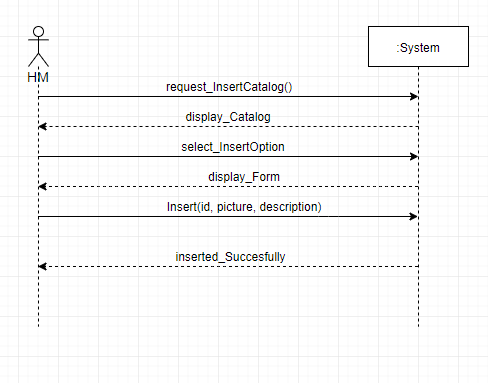
**Figure 2.5:** View room booking status

**HM Approve Room Booking Request :**

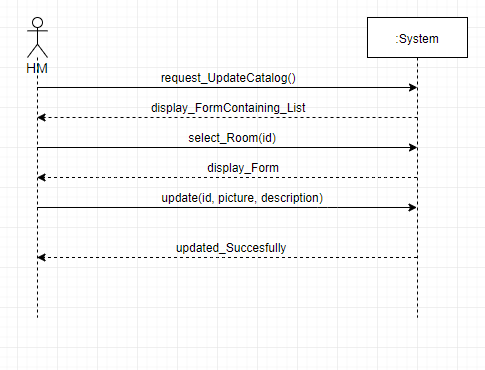
****

**Figure 2.6:** Approve room booking request

**HM Insert Catalog:**

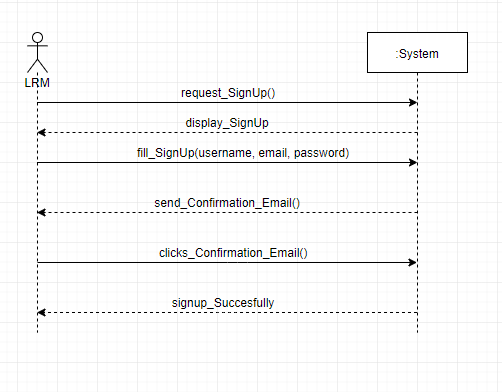
**Figure 2.7:** Insert Catalog

**HM Update Catalog:**



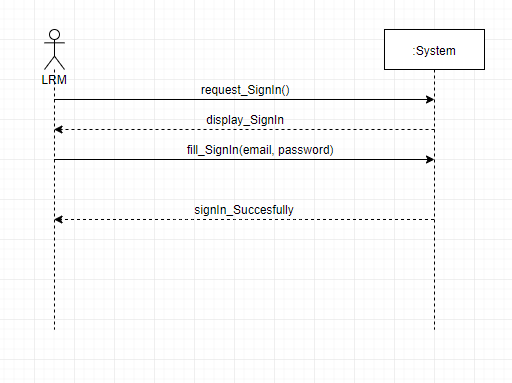
**Figure 2.8:** Update Catalog

**LRM Sign-Up:**

****

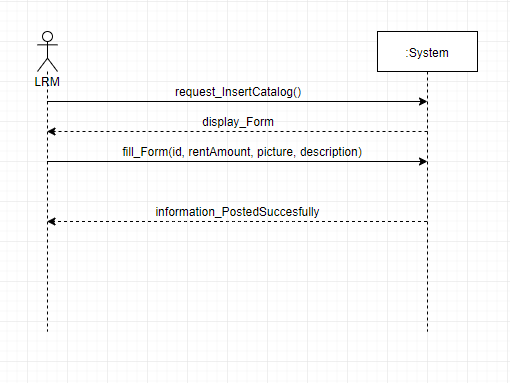
**Figure 2.9:** Signup

**LRM Sign-in:**

****

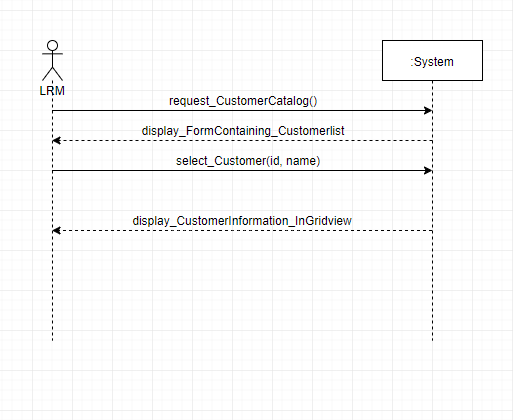
**Figure 2.10:** Sign in

**LRM Post Living room information:**

****

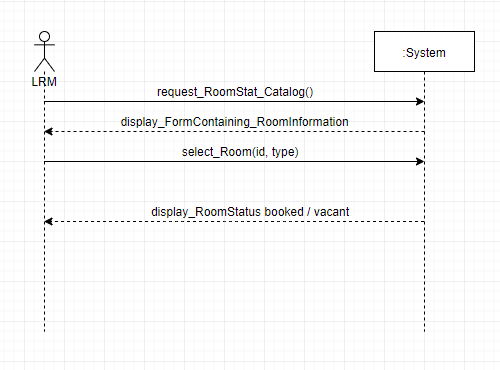
**Figure 2.11:** post living room information

**LRM View Customer information:**

****

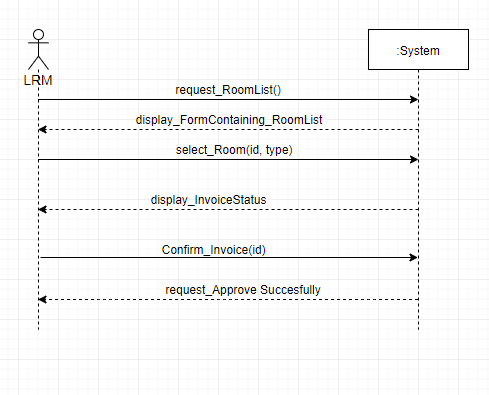
**Figure 2.12:** view customer information

**LRM View Room booking Status:**

****

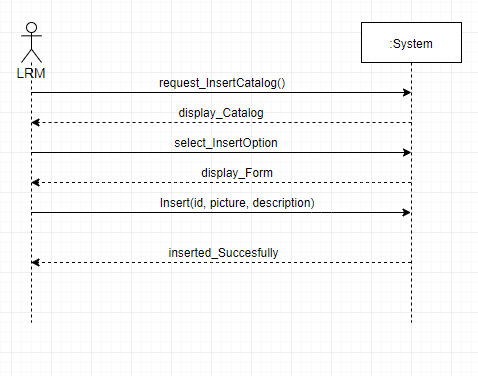
**Figure 2.13:** view room booking status

**LRM Approve Room booking request:**

****

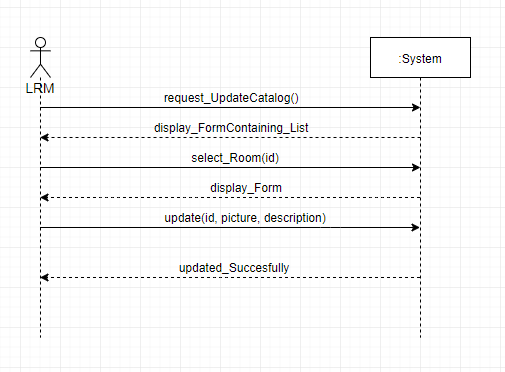
**Figure 2.14:** approve room booking request

**LRM Insert Catalog:**

****

**Figure 2.15:** insert Catalog

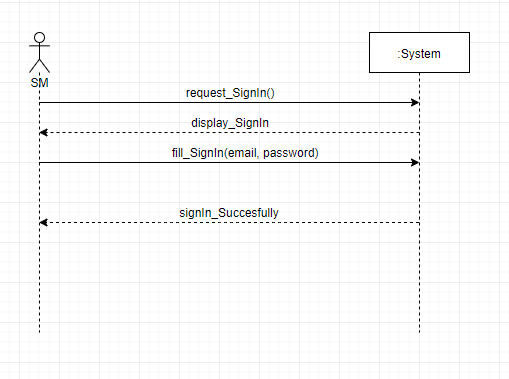
**LRM Update Catalog:**

****

**Figure 2.16:** Update Catalog

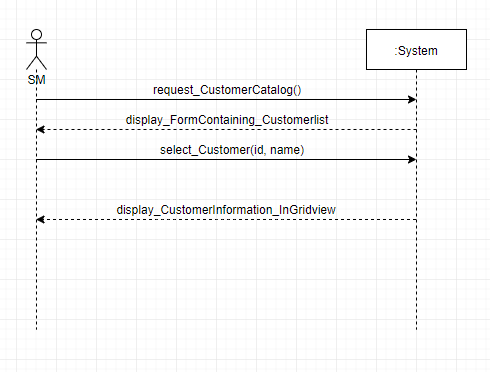
**Figure 2.17:** Sign up

**SM Sign in**

****

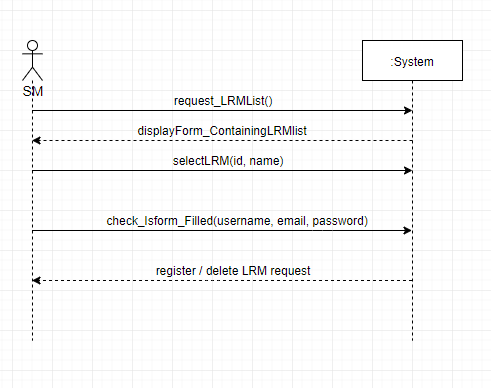
**Figure 2.18:** Sign in

**SM View Customer information:**

****

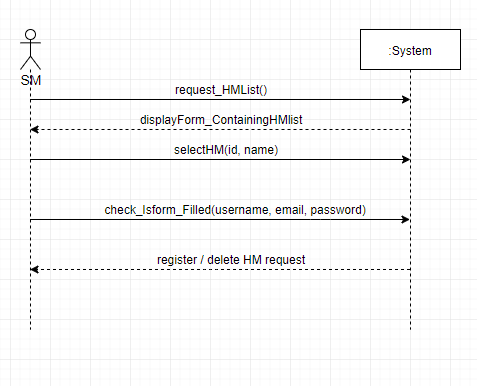
**Figure 2.19:** view customer information

**SM Register LRM :**

****

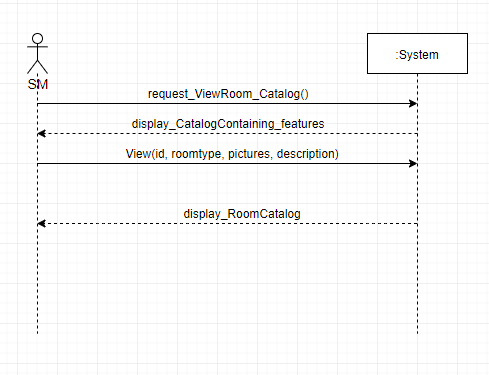
**Figure 2.20:** Register LRM

**SM Register HM:**

****

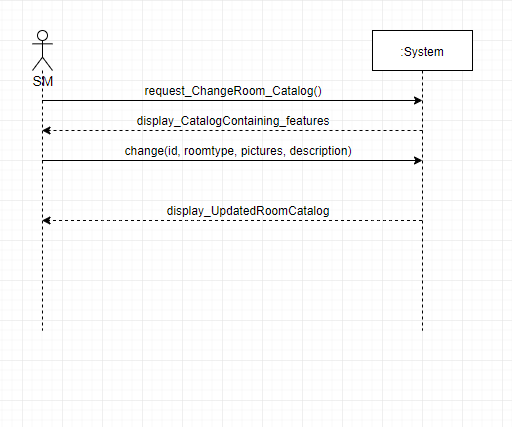
**Figure 2.21:** Register HM

**SM View Room Catalog:**

****

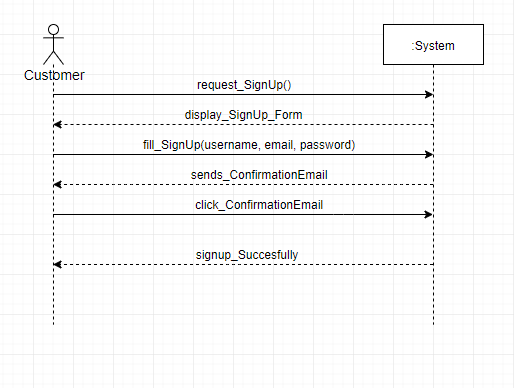
**Figure 2.22:** View room Catalog

**SM Change Room Catalog:**

****

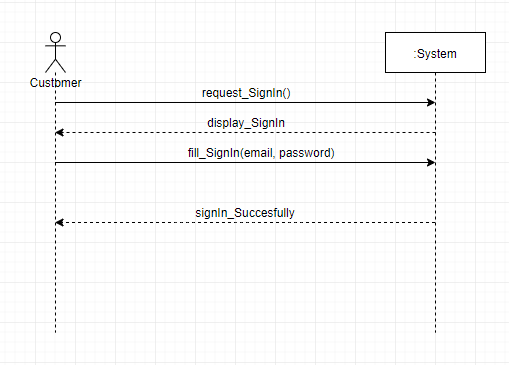
**Figure 2.23:** Change room Catalog

**Customer Sign-up**

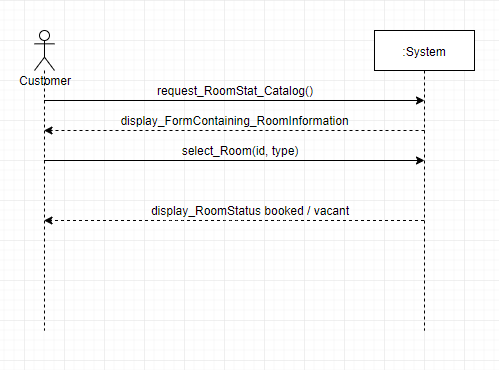
****

**Figure 2.24:** Customer Signup

**Customer Sign-in**

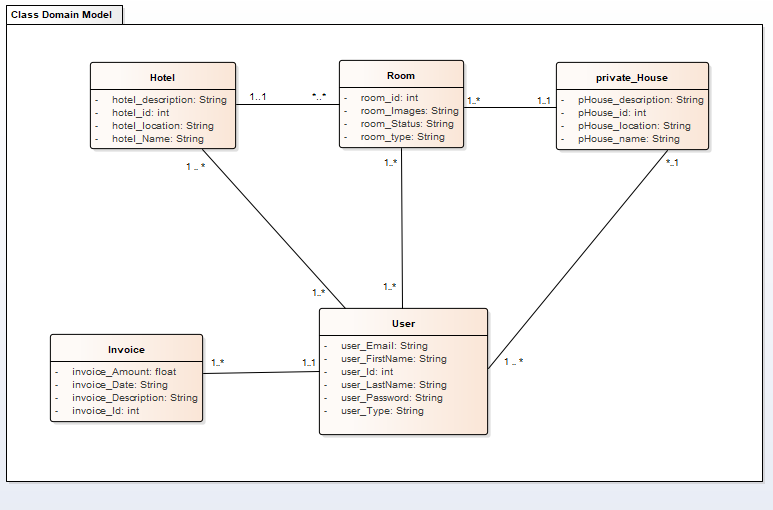
****

**Figure 2.25:** Customer Sign-in

**Customer View Hotel Room  
  
**

**Figure 2.26:** Customer view Hotel rooms

# 2.6 Domain Model

****

# Chapter 3

# System Design

The purpose of this chapter is to provide information that is complementary to the development phase. Without an adequate design, that delivers required function as well as quality attributes, the project will fail. However, communicating architecture to its stakeholders is as important a job as creating it in the first place.

# 3.1 Multitier Architecture

Software architecture is described as the organization or structure of a system, where the system represents a collection of components that accomplish a specific function or set of functions. Below is the architecture diagram of the system:

***Table3.1: Multitier Architecture***

|  |  |
| --- | --- |
| **Layers** | **Description** |
| Presentation Layer | This layer will be used for the interaction with the user through a graphical user interface. |
| Business Logic Layer | This layer contains the business logic. All the constraints and majority of the functions reside under this layer. |
| Database Layer | This layer contains the database of the application being developed. |

# 3.1.1 Presentation Layer

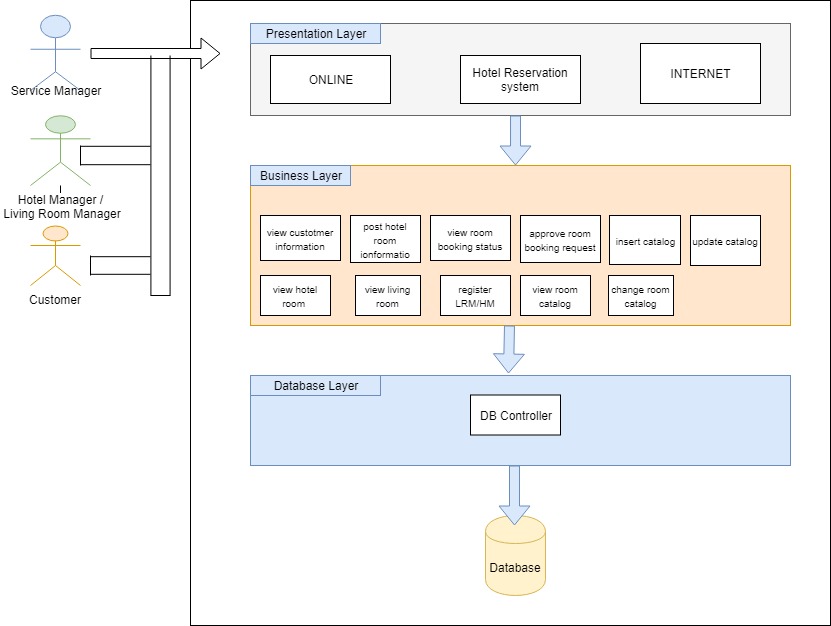
* Occupies the top level and displays information related to services available on a website.
* This tier communicates with other tiers by sending results to the browser and other tiers in the network.

# 3.1.2 Business Logic Layer

Application Layer also called the middle tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.

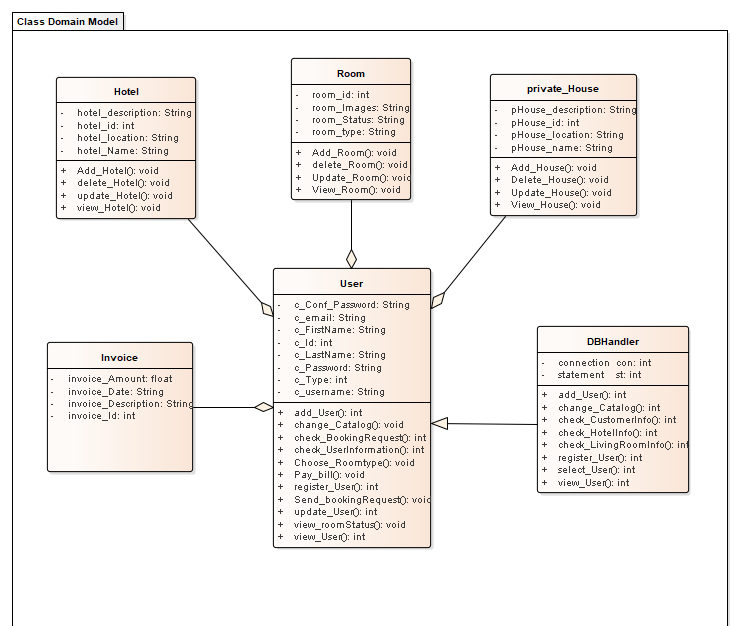
# 3.1.3 Database Layer

Database layer includes database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

****

***Figure 3.1: Architechture Diagram***

# 3.2 Class Diagram

******

# 3.3 Sequence Diagram

Sequence Diagram model the flow of logic within your system in a visual manner enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes.

# 3.3.1 Customer, Hotel room Manager, Service Manager, Living room Manager

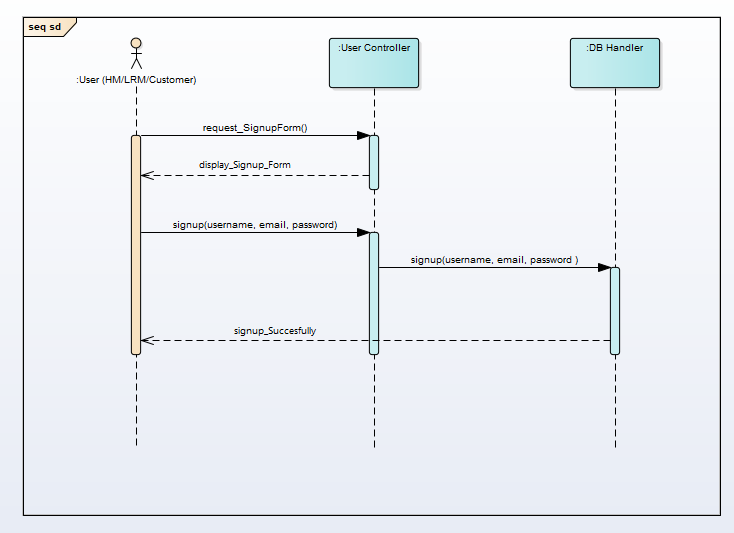
****

Figure 3.4: Customer, Hotel Manager, Living room Manager, Service Manager Sign-up

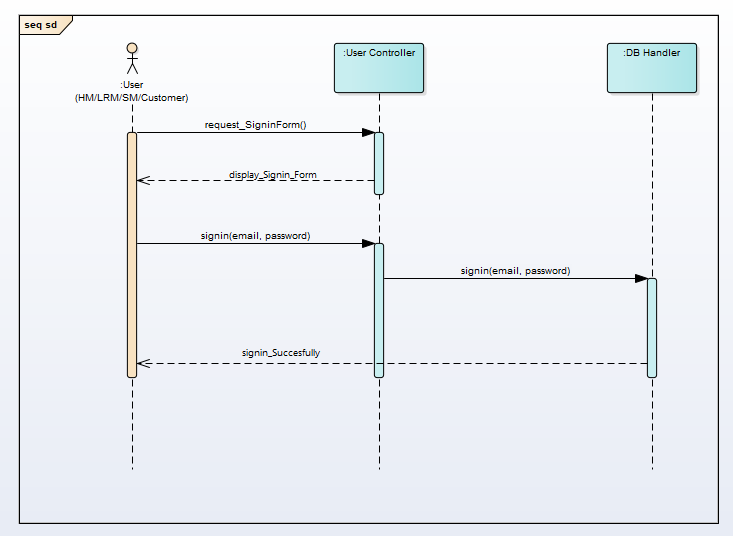


Figure 3.5: Customer, Hotel Manager, Living room Manager, Service Manager Sign-in

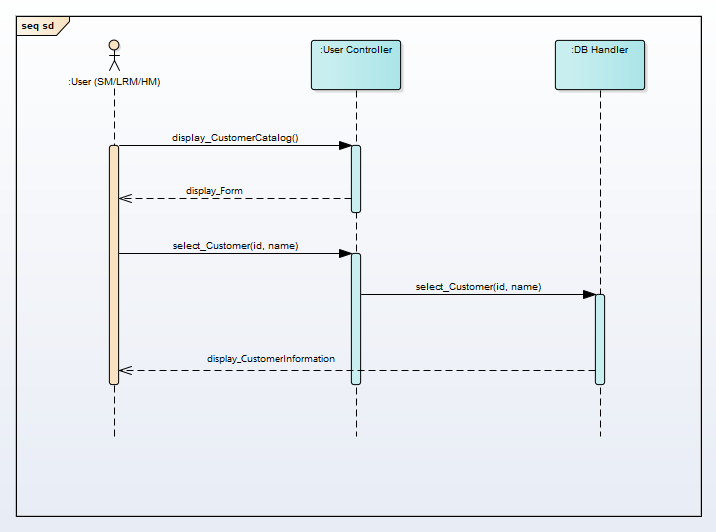
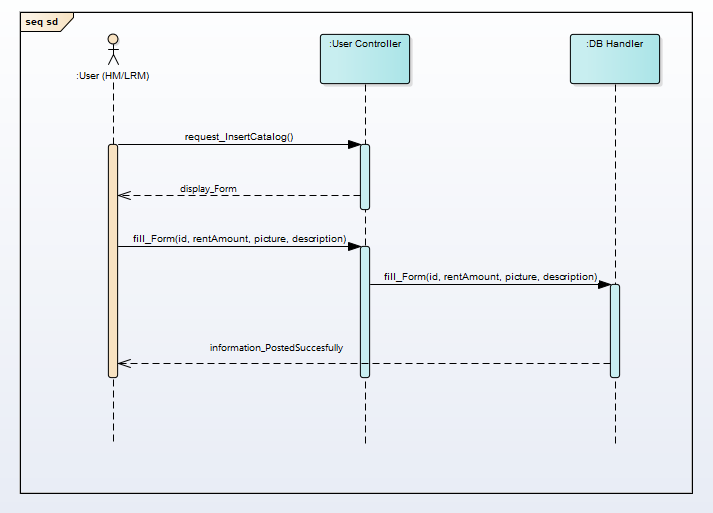
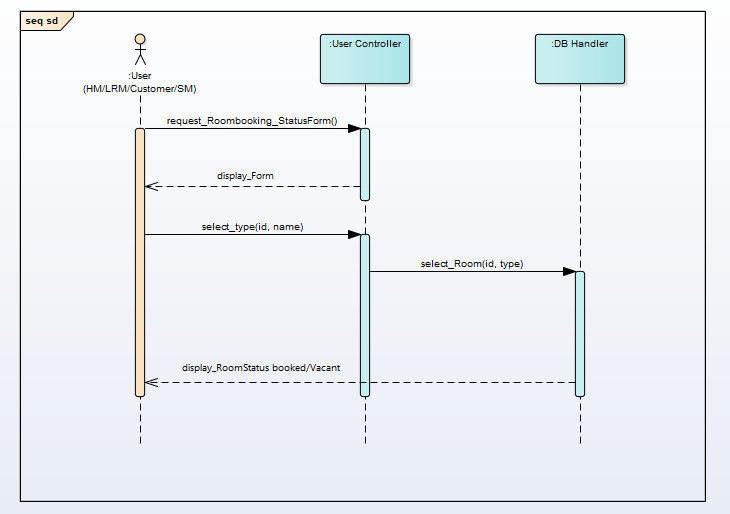


Figure 3.6: Hotel Manager, Living room Manager, Service Manager View Customer Information

  
Figure 3.7: Hotel room Manager, Living room Manager Post (hotel, living) room information

  
  
Figure 3.8: HM, LRM, Customer view room booking status

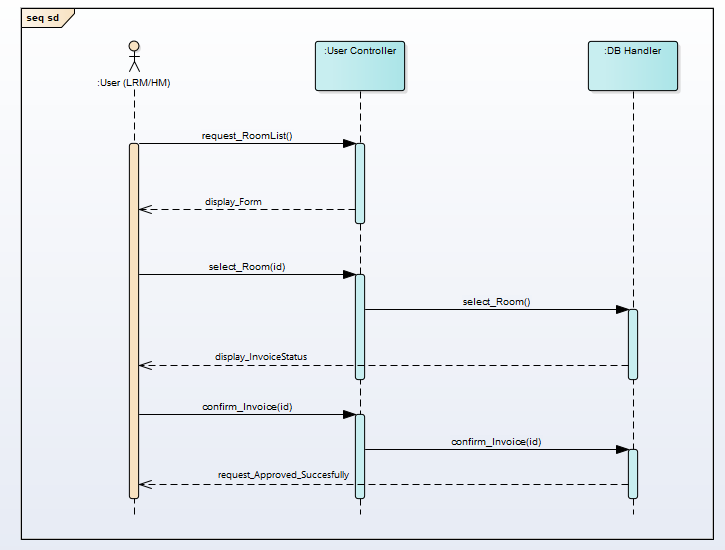


Figure 3.9: HM, LRM Approve room booking request

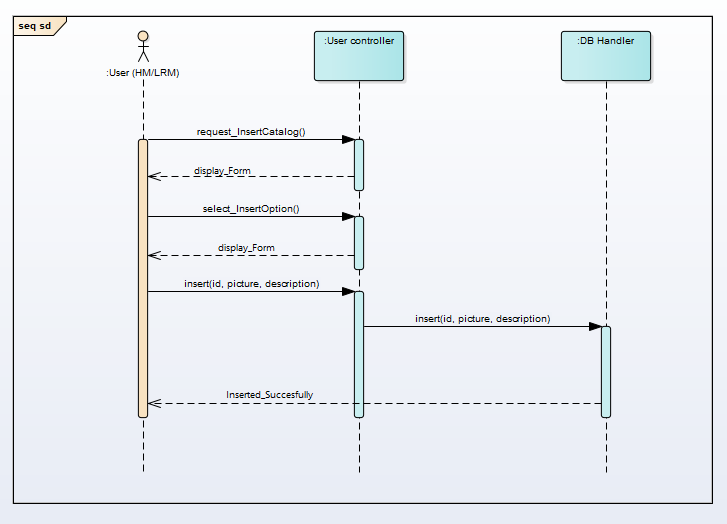


Figure 3.10: HM, LRM Insert Catalog

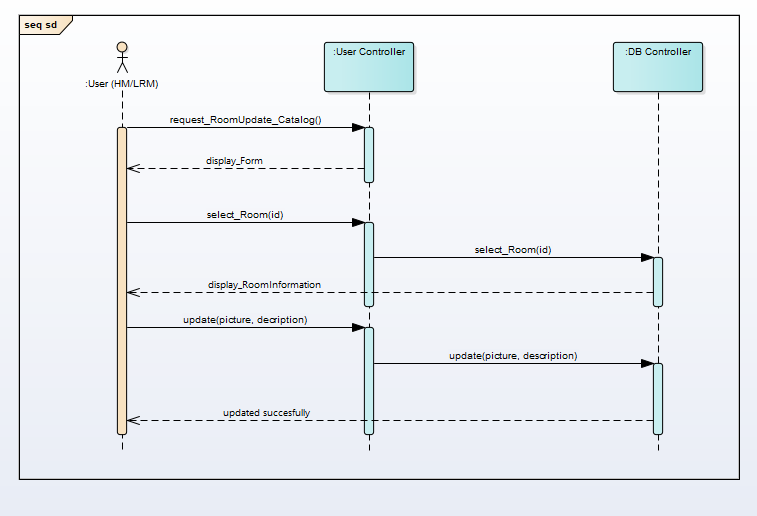


Figure 3.11: HM, LRM update Catalog

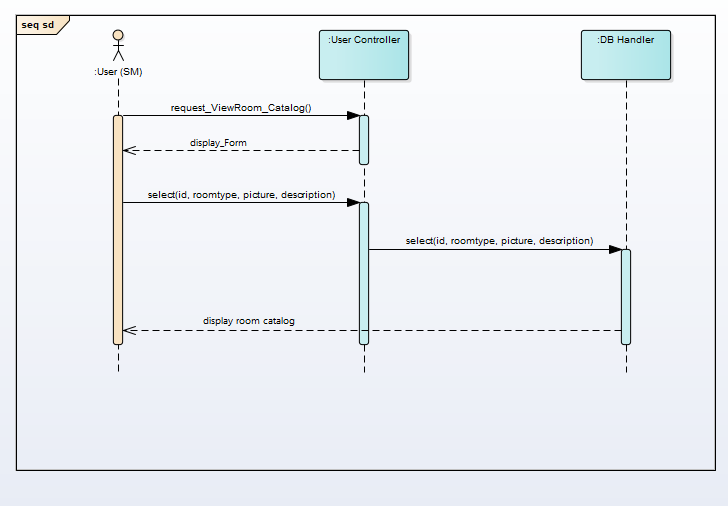


Figure 3.12: SM View room catalog

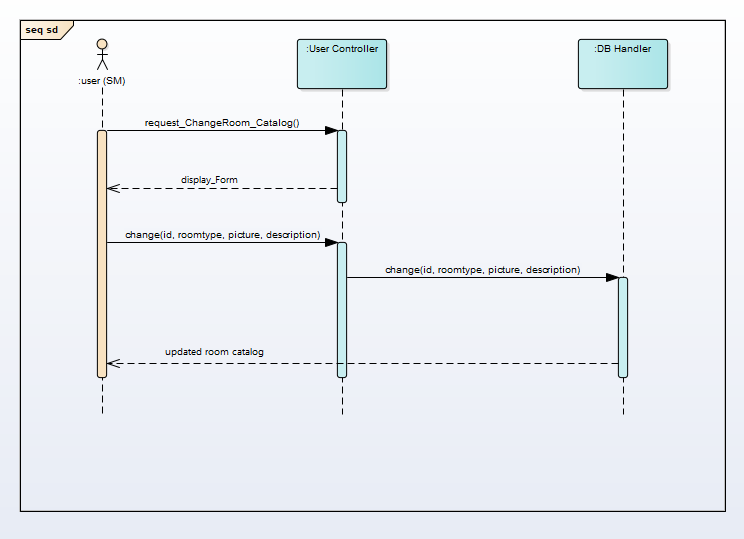
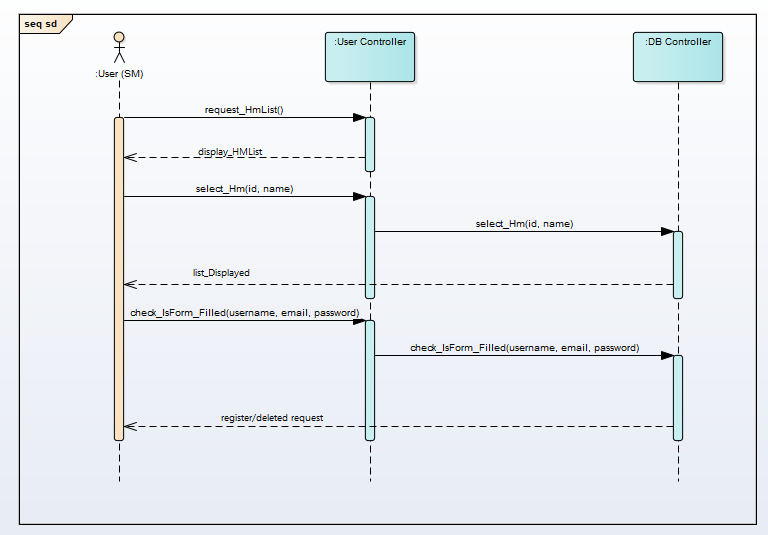


Figure 3.13: SM change room catalog

****  
  
Figure 3.14: SM register HM

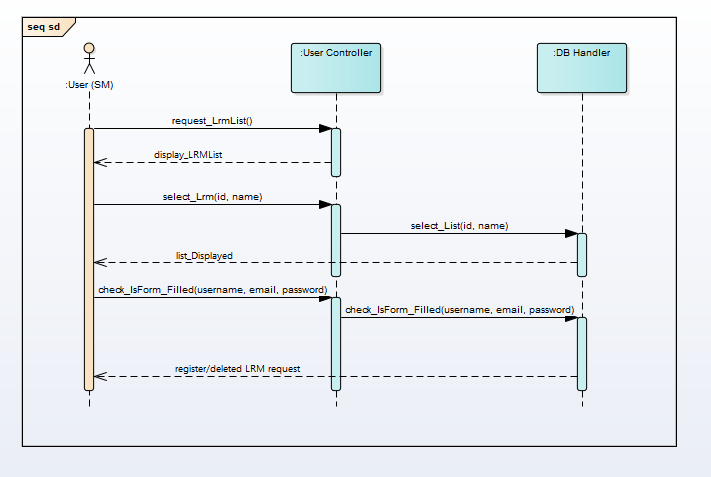
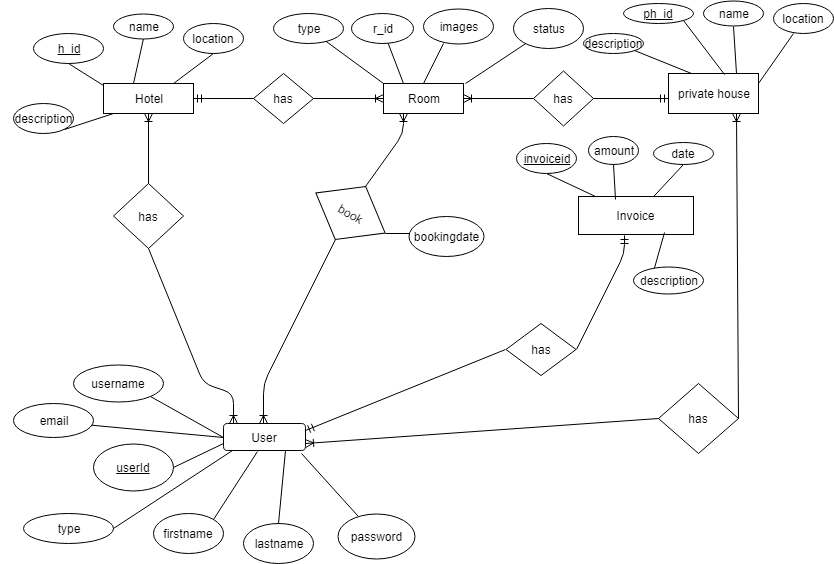
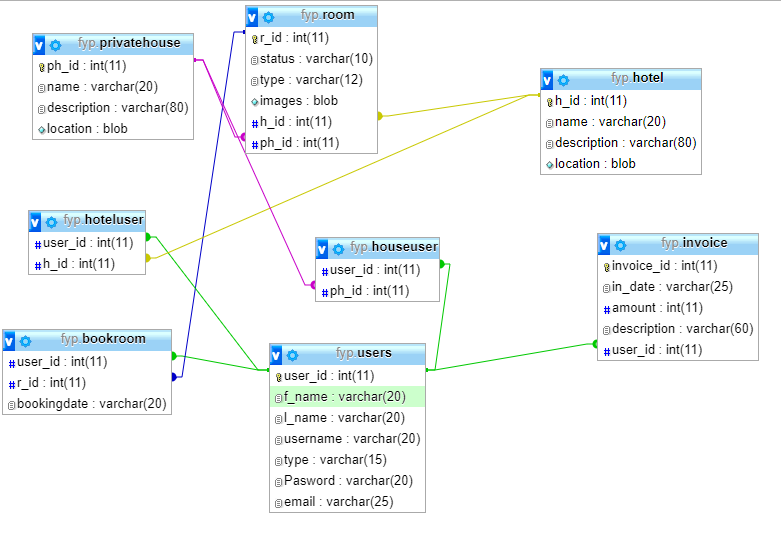


Figure 3.15: SM register LRM

# 3.4 Entity Relationship Diagram

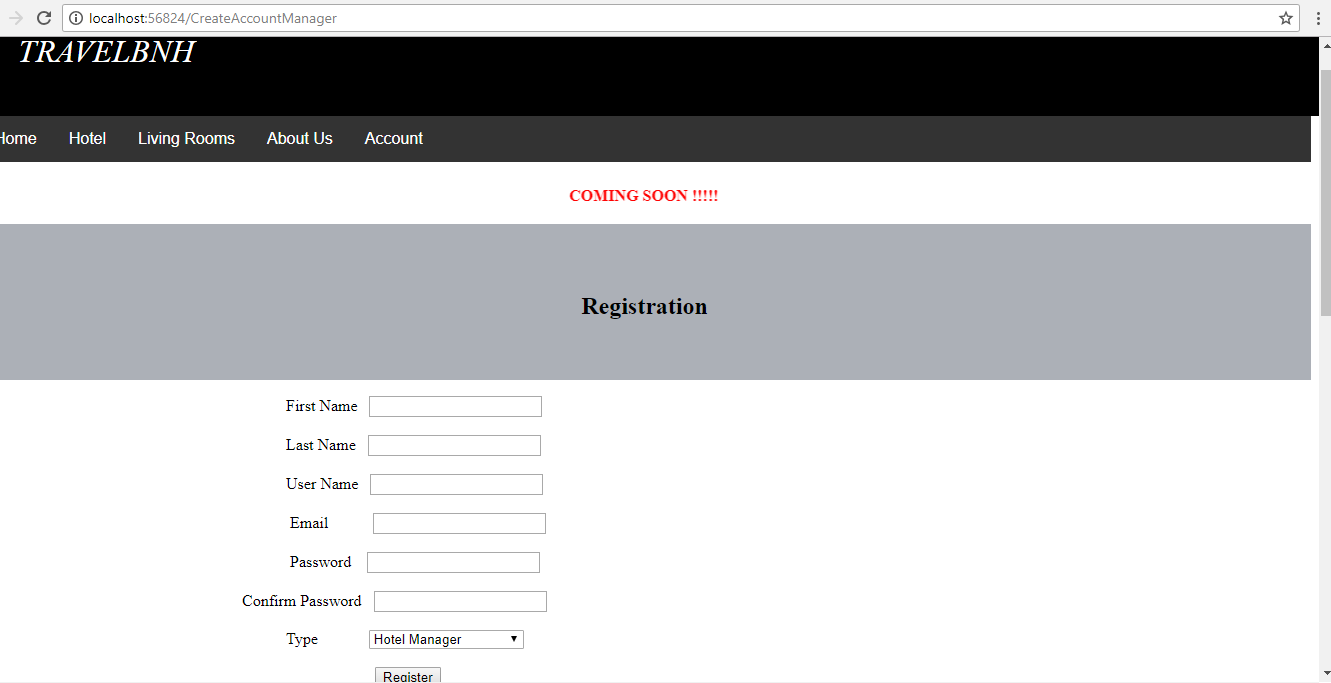


# 3.5 Database Schema

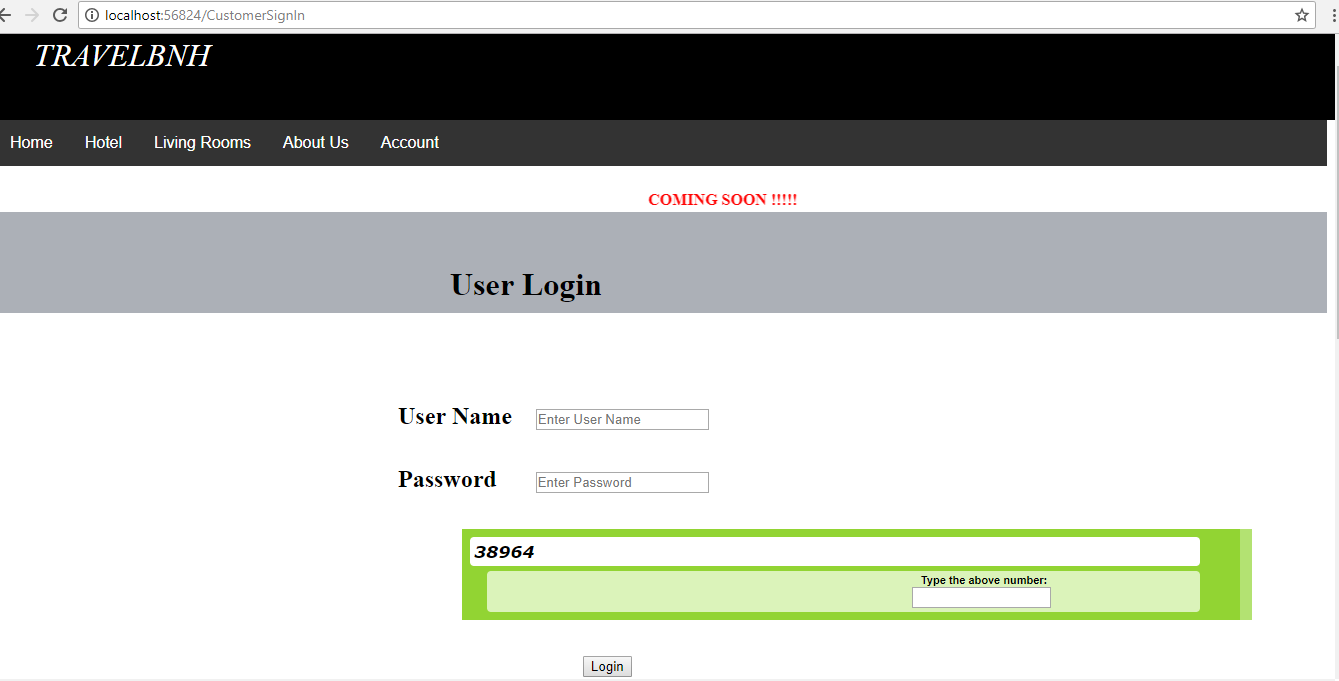


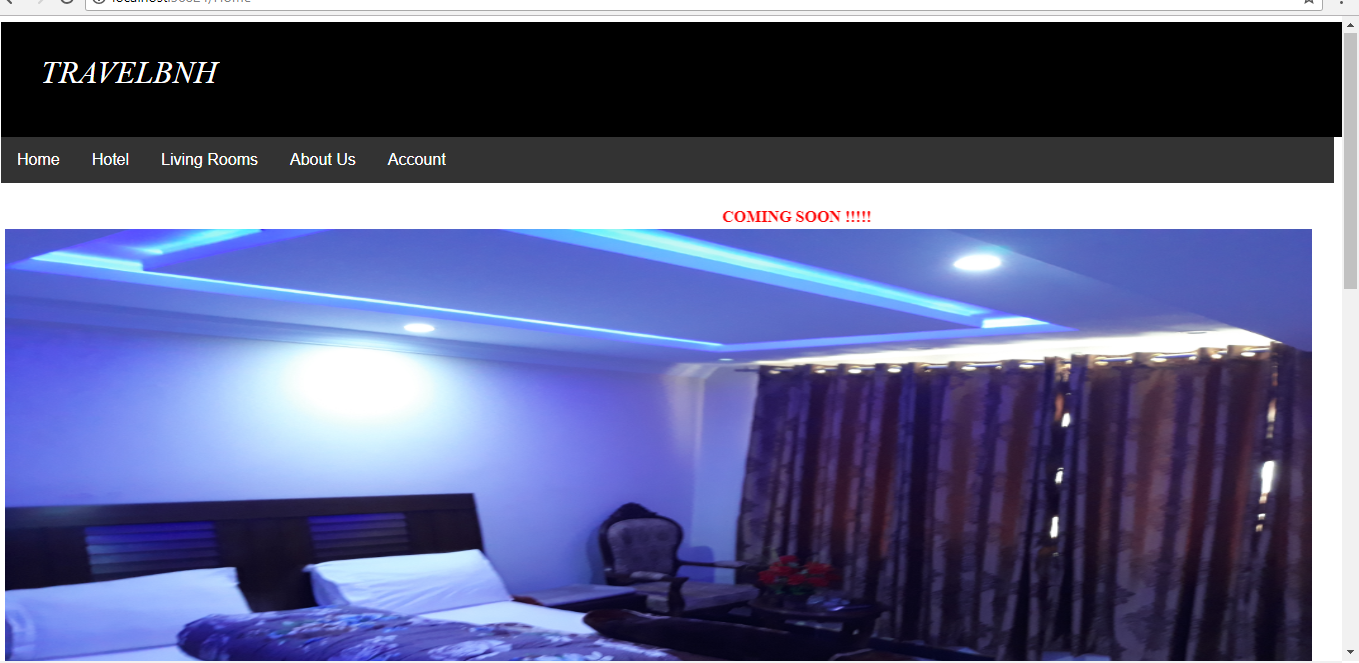
# 3.6 User Interface

***Sign Up:***

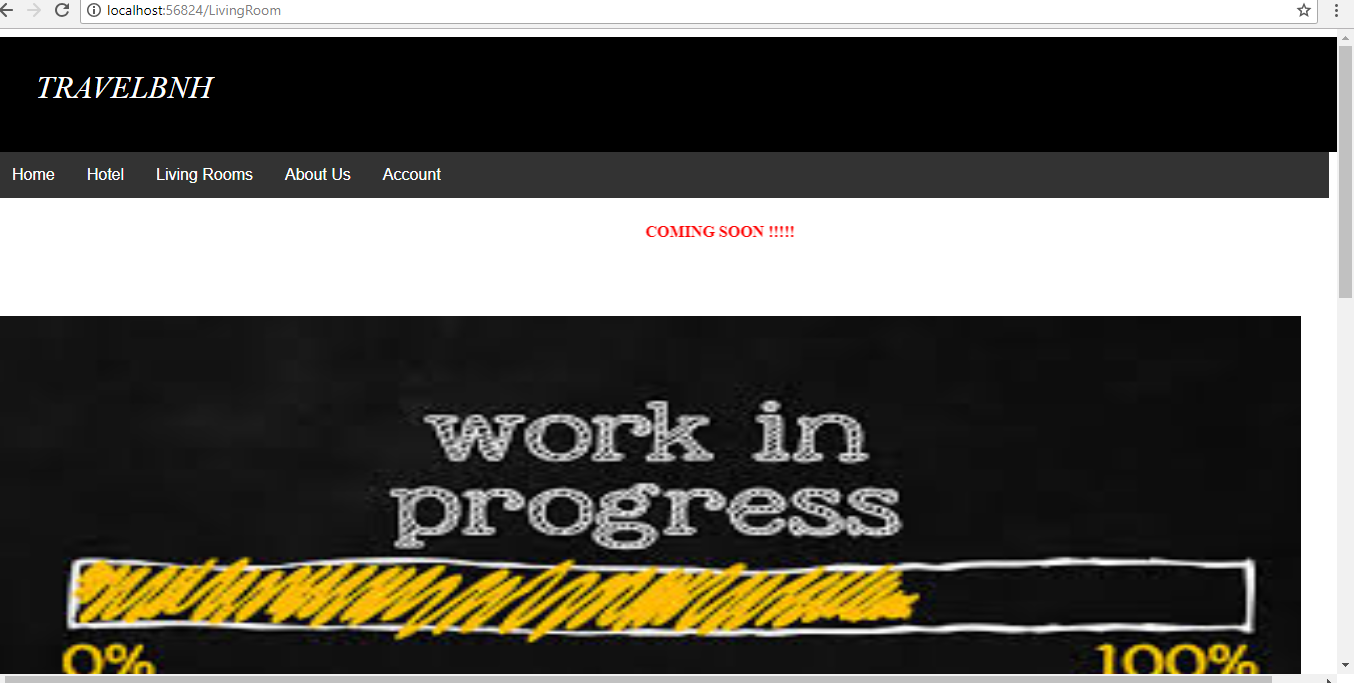
******

***Sign In:***



***Home:*** ******

***Living Room:***

******

**Chapter 4**

**Software Development**

Hotel Reservation System will provide a platform for tourist to find the best hotel in the region

and make booking. It will be easy for users to book hotel online and living rooms. Hotel Reservation System is a two sided platform it facilitates the process of booking private living rooms and hotels.

During implementation we face some complexities but were later resolved.

* Connection of controller with view.
* Fetching location from map.

Keeping this in view, there is a need for an online platform that helps the people to find best hotel or the living room in the region and book online, the owner posts the picture of living rooms to notify the people the accommodation of the rooms and their respective packages.

**4.1 Coding Standards**

* **Camel Case**: For arguments and local variable.
* **Pascal Case**: For class names and methods.

# 4.1.1 Indentation

The indentation pattern should be consistently followed throughout.

**4.1.2 Declaration**

* Instance variables are placed in the sequence: First public instance variables, protected,
* Package level with no access modifier and then private.
* Next the class constructors are declared.
* Html and stylesheet are used.

**4.2 Development environment**

There are different types of languages used in the project and the type of Environment used is given below.

* **Programming Language:**
  + **C#** will be used at backend for web application.
  + **HTML5, CSS3, AngularJS** will be used at frontend for web application.
* **Development Environment:**
* **Visual Studio:** It will be used for the development, debugging and simulation of the webserver.

## Software Description

Main modules of our project are

* User Registration.
* User Login.
* Customer view Hotels.
* Hotel manager Post Hotels.

**Snippet no # 1:**

**User:**

namespace HotelReservationSystem.Models

{

using System;

using System.Collections.Generic;

public partial class User

{

public int UserId { get; set; }

public string Name { get; set; }

public string Email { get; set; }

public string Password { get; set; }

public Nullable<System.DateTime> DateofBirth { get; set; }

public string Gender { get; set; }

public string Address { get; set; }

public string City { get; set; }

public string PhoneNo { get; set; }

public string Type { get; set; }

public bool IsEmailVerified { get; set; }

public System.Guid ActivationCode { get; set; }

}

}

**Description:**

The above code describe how this process will work. These are the data type of the user being used further for user (SM service manager, Hotel Room Manager, living room manager).These are the datatype having setter and getter values.

**Snippet No # 2**

**UserLogin:**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace HotelReservationSystem.Models

{

public class UserLogin

{

[Display (Name="Email ID")]

[Required(AllowEmptyStrings=false,ErrorMessage="Email is Required")]

public string Email { get; set; }

[Required(AllowEmptyStrings = false, ErrorMessage = "Password is Required")]

[DataType(DataType.Password)]

public string Password { get; set; }

[Display(Name ="Remember Me")]

public bool RememberMe { get; set; }

}

}

**Description:**

Users are login into the system in this module. User enters the username and password if the username and password matches with the saved data in database it will navigate to the next page. If the required filed is incorrect it will display Email is required, and password is required.

**Snippet No # 3**

**HotelHouse**

namespace HotelReservationSystem.Models

{

using System;

using System.Collections.Generic;

public partial class HotelHouse

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

public string Address { get; set; }

public string Rating { get; set; }

public Nullable<decimal> Lat { get; set; }

public Nullable<decimal> Long { get; set; }

public string Type { get; set; }

}

}

**Description:**

In this above code Hotel House has datatypes which includes the description, address etc pointing the hotel details.

**Snippet No # 3**

**UserController:**

using HotelReservationSystem.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Mail;

using System.Web;

using System.Web.Mvc;

using System.Web.Security;

namespace HotelReservationSystem.Controllers

{

public class UserController : Controller

{

// GET: User

[HttpGet]

public ActionResult Registration()

{

return View();

}

//Registration post action

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Registration([Bind(Exclude = "IsEmailVerified,ActivationCode")] User user)

{

bool Status = false;

string message = "";

//model validation

if (ModelState.IsValid)

{

#region

var isExist = IsEmailExist(user.Email);

if (isExist)

{

ModelState.AddModelError("EmailExist", "Email already exist");

return View(user);

}

#endregion

#region Generate Activation Code

user.ActivationCode = Guid.NewGuid();

#endregion

#region Password Hashing

user.Password = Crypto.Hash(user.Password);

user.ConfirmPassword = Crypto.Hash(user.ConfirmPassword);

#endregion

user.IsEmailVerified = false;

#region save to Database

using (FypDataBaseEntities1 dc = new FypDataBaseEntities1())

{

dc.Users.Add(user);

dc.SaveChanges();

// send email to user

SendVerificationLinkEmail(user.Email, user.ActivationCode.ToString());

message = "Registration is successfully done,Account Activation link" + "has been send to your email id:"

+ user.Email;

Status = true;

}

#endregion

}

else

{

message = "Invalid Request";

}

//Email already exist

//generate activation code

// Password Hashing

// Save data to database

//send email to user

ViewBag.Message = message;

ViewBag.Status = Status;

return View(user);

}

//verify account

[HttpGet]

public ActionResult VerifyAccount(string id)

{

bool Status = false;

using (FypDataBaseEntities1 dc = new FypDataBaseEntities1())

{

dc.Configuration.ValidateOnSaveEnabled = false; //This line i have added here to avoid

//confirm password does not match issue on save changes

var v = dc.Users.Where(a => a.ActivationCode == new Guid(id)).FirstOrDefault();

if (v != null)

{

v.IsEmailVerified = true;

dc.SaveChanges();

Status = true;

}

else

{

ViewBag.Message = "Invalid Request";

}

}

ViewBag.Status = Status;

return View();

}

//login

[HttpGet]

public ActionResult Login()

{

return View();

}

//login post

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Login(UserLogin login, string ReturnUrl = "")

{

string message = "";

using (FypDataBaseEntities1 dc = new FypDataBaseEntities1())

{

var v = dc.Users.Where(a => a.Email == login.Email).FirstOrDefault();

if (v.Type == "Customer")

{

if (v != null)

{

if (string.Compare(Crypto.Hash(login.Password), v.Password) == 0)

{

int timeout = login.RememberMe ? 525600 : 1; //525600 min=1 year

var ticket = new FormsAuthenticationTicket(login.Email, login.RememberMe, timeout);

string encrypted = FormsAuthentication.Encrypt(ticket);

var cookie = new HttpCookie(FormsAuthentication.FormsCookieName, encrypted);

cookie.Expires = DateTime.Now.AddMinutes(timeout);

cookie.HttpOnly = true;

Response.Cookies.Add(cookie);

if (Url.IsLocalUrl(ReturnUrl))

{

ViewBag.Error = "Invalid email or password";

message = "Invalid Email /Password";

return Redirect(ReturnUrl);

}

else

{

ViewBag.Error = "Invalid email or password";

// message = "Invalid Email /Password";

return RedirectToAction("Index", "Home");

}

}

else

{

ViewBag.Error = "Invalid email or password";

message = "Invalid credential provided";

}

}

else

{

ViewBag.Error = "Invalid email or password";

message = "Invalid credential provided";

}

/\*ViewBag.Message = message;

return View();\*/

}

else if (v.Type == "HotelManager")

{

if (v != null)

{

if (string.Compare(Crypto.Hash(login.Password), v.Password) == 0)

{

int timeout = login.RememberMe ? 525600 : 1; //525600 min=1 year

var ticket = new FormsAuthenticationTicket(login.Email, login.RememberMe, timeout);

string encrypted = FormsAuthentication.Encrypt(ticket);

var cookie = new HttpCookie(FormsAuthentication.FormsCookieName, encrypted);

cookie.Expires = DateTime.Now.AddMinutes(timeout);

cookie.HttpOnly = true;

Response.Cookies.Add(cookie);

if (Url.IsLocalUrl(ReturnUrl))

{

ViewBag.Error = "Invalid email or password";

message = "Invalid Email /Password";

return Redirect(ReturnUrl);

}

else

{

ViewBag.Error = "Invalid email or password";

// message = "Invalid Email /Password";

return RedirectToAction("SM", "SM");

}

}

else

{

ViewBag.Error = "Invalid email or password";

message = "Invalid credential provided";

}

}

else

{

ViewBag.Error = "Invalid email or password";

message = "Invalid credential provided";

}

}

else if (v.Type == "LivingRoomManager")

{

if (v != null)

{

if (string.Compare(Crypto.Hash(login.Password), v.Password) == 0)

{

int timeout = login.RememberMe ? 525600 : 1; //525600 min=1 year

var ticket = new FormsAuthenticationTicket(login.Email, login.RememberMe, timeout);

string encrypted = FormsAuthentication.Encrypt(ticket);

var cookie = new HttpCookie(FormsAuthentication.FormsCookieName, encrypted);

cookie.Expires = DateTime.Now.AddMinutes(timeout);

cookie.HttpOnly = true;

Response.Cookies.Add(cookie);

if (Url.IsLocalUrl(ReturnUrl))

{

message = "Invalid Email /Password";

return Redirect(ReturnUrl);

}

else

{

// message = "Invalid Email /Password";

return RedirectToAction("LRM", "Home");

}

}

else

{

message = "Invalid credential provided";

}

}

else

{

message = "Invalid credential provided";

}

}

}

ViewBag.Message = message;

return View();

}

//captcha

[HttpPost]

public ActionResult FormSubmit()

{

var response = Request["g-recaptcha-response"];

string secretKey = "6LdNu1cUAAAAABpvIQ5tQnIOe04d-7ee9zdhgAmc";

var client = new WebClient();

ViewData["Message"] = "Google recaptcha validation success";

return View("Registration");

}

//logout

[Authorize]

[HttpPost]

public ActionResult Logout()

{

FormsAuthentication.SignOut();

return RedirectToAction("Login", "User");

}

[NonAction]

public bool IsEmailExist(string emailID)

{

using (FypDataBaseEntities1 dc = new FypDataBaseEntities1())

{

var v = dc.Users.Where(a => a.Email == emailID).FirstOrDefault();

return v != null;

}

}

[NonAction]

public void SendVerificationLinkEmail(String EmailID, string activationCode)

{

var verifyUrl = "/User/VerifyAccount/" + activationCode;

var link = Request.Url.AbsoluteUri.Replace(Request.Url.PathAndQuery, verifyUrl);

var fromEmail = new MailAddress("travelbnh@gmail.com", "TravelBnh");

var toEmail = new MailAddress(EmailID);

var fromEmailPassword = "58876785049"; // replace this with actual password

string subject = "Your account is successfully created";

string body = "<br/><br/> We are excited to tell you that your travelbnh account is successfully created, please click on the below link to verify your account" +

"<a href='" + link + "'>" + link + "</a>";

var smtp = new SmtpClient

{

Host = "smtp.gmail.com",

Port = 587,

EnableSsl = true,

DeliveryMethod = SmtpDeliveryMethod.Network,

UseDefaultCredentials = false,

Credentials = new NetworkCredential(fromEmail.Address, fromEmailPassword)

};

using (var message = new MailMessage(fromEmail, toEmail)

{

Subject = subject,

Body = body,

IsBodyHtml = true

})

smtp.Send(message);

} }

}

**Description:**

The above class User controller manage the registration and login of SM, HM, LRM ,saving the data in the database, and verification of account through email, Verification by email the respective user is the major part. If the entries are invalid will redirect to another page respective to the User.

**Snippet No # 4**

**HomeController:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace HotelReservationSystem.Controllers

{

public class HomeController : Controller

{

// GET: Home

[Authorize]

public ActionResult Index()

{

return View();

}

public ActionResult HM()

{

return View();

}

public ActionResult LRM()

{

return View();

}

public ActionResult SM()

{

return View();

}

}

}

**Description:**

Home Controller generate view for all type of users.

**Snippet no # 5**

**Hotel Controller:**

using HotelReservationSystem.Models;

using System.Web.Mvc;

namespace HotelReservationSystem.Controllers

{

public class HotelController : Controller

{

// GET: Hotel

public ActionResult Hotel()

{

return View();

}

[HttpPost]

public ActionResult AddHotelHouse(HotelHouse ho)

{

using (FypDataBaseEntities1 db = new FypDataBaseEntities1())

{

db.HotelHouses.Add(ho);

db.SaveChanges();

}

return View(ho);

}

}

}

**Description:**

Hotel controller manage information of hotels.

**Snippet no # 6**

**SmController:**

using HotelReservationSystem.Models;

using System;

using System.Collections.Generic;

using System.Data.Entity;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace HotelReservationSystem.Controllers

{

public class SmController : Controller

{

// GET: Sm

public ActionResult Sm()

{

return View();

}

public ActionResult GetData()

{

using (FypDataBaseEntities1 db = new FypDataBaseEntities1())

{

List<User> empList = db.Users.ToList<User>();

return Json(new { data = empList }, JsonRequestBehavior.AllowGet);

}

}

[HttpGet]

public ActionResult AddOrEdit(int id = 0)

{

if (id == 0)

return View(new User());

else

{

using (FypDataBaseEntities1 db = new FypDataBaseEntities1())

{

return View(db.Users.Where(x => x.UserId == id).FirstOrDefault<User>());

}

}

}

[HttpPost]

public ActionResult AddOrEdit(User u)

{

using (FypDataBaseEntities1 db = new FypDataBaseEntities1())

{

if (u.UserId == 0)

{

db.Users.Add(u);

db.SaveChanges();

return Json(new { success = true, message = "Saved Successfully" }, JsonRequestBehavior.AllowGet);

}

else

{

db.Entry(u).State = EntityState.Modified;

db.SaveChanges();

return Json(new { success = true, message = "Updated Successfully" }, JsonRequestBehavior.AllowGet);

}

}

}

}

}

**Description:**

In the above code Sm Controller will control the Service Manager it will manage the User to add them to the database. SM will manage all the user (LRM, HM) .If any modification is required SM will update the respective user and displaying the Update successful message.

# Chapter 5

# Software Testing

Software Testing is the most crucial part of Software Development Process. It is the investigation or evaluation of a software component, improving them, and finding bugs and defects. Testing is usually done by executing a system in such a way that it identifies any gaps, errors, or missing requirements in contrary to the actual requirements.

* 1. **Testing Methodology**

It is essential to have a testing plan in place to ensure that the product delivered is stable, and is delivered on a predictable timeline.

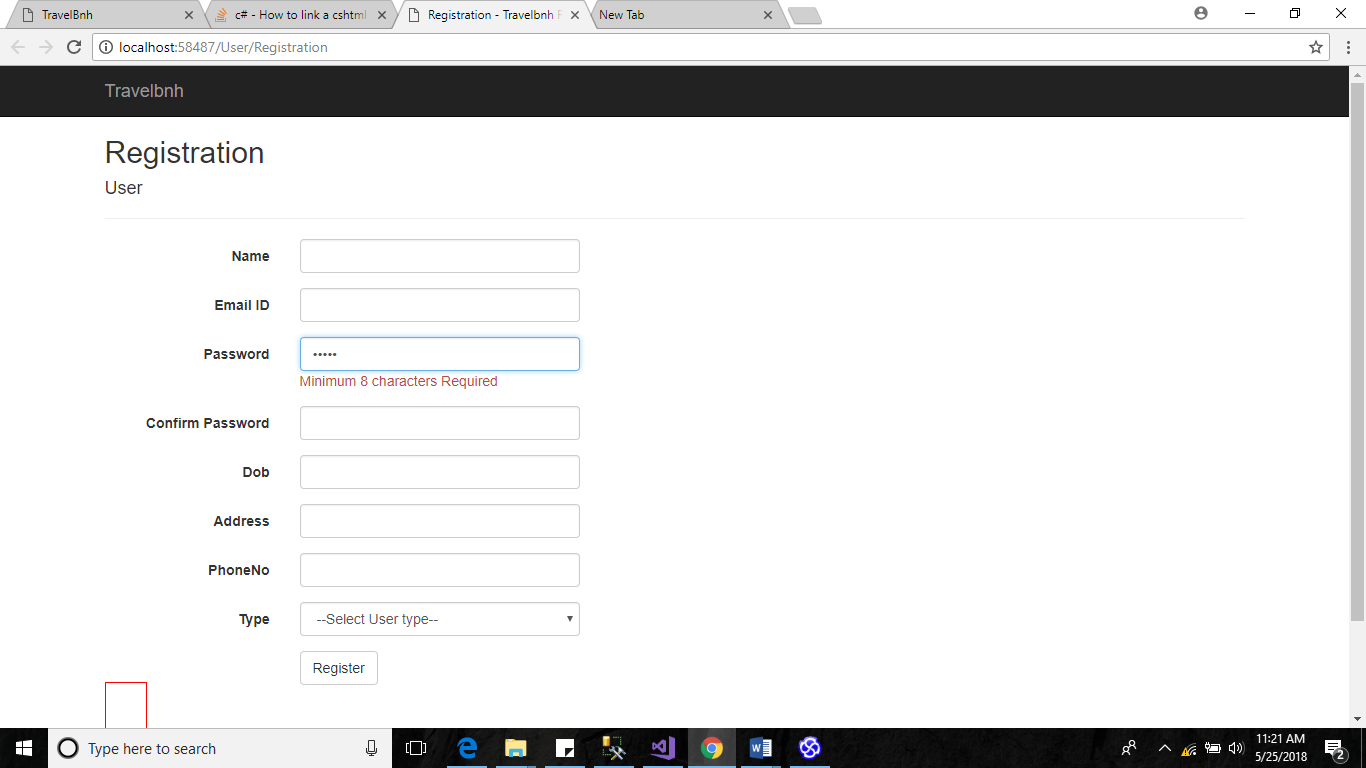
* + Testing is done
  + Project is stable.
  + If a person gives a wrong value’s system should display error message.
  + If a person gives a Right value’s system should display right result.

# Test Case

**Test Case: Registration**

Table : Registration/Password Validation

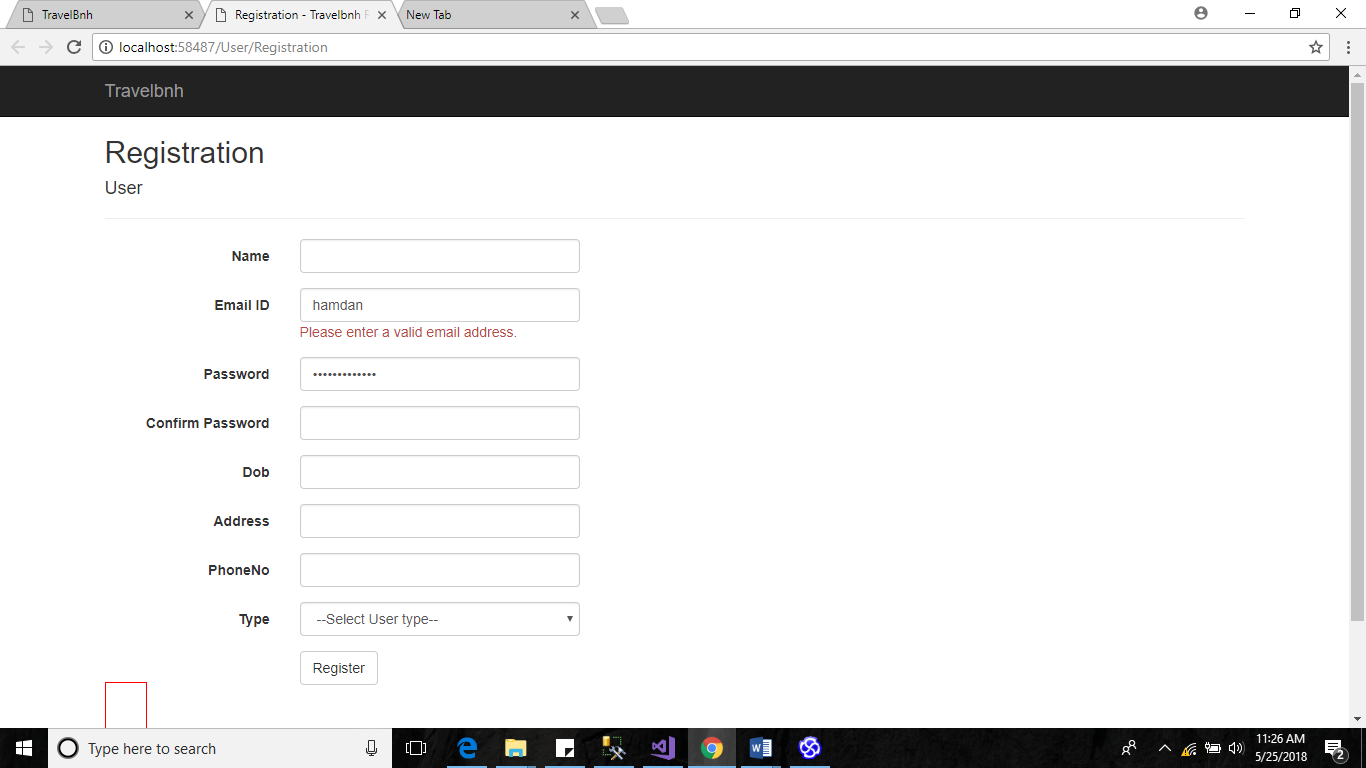
|  |  |
| --- | --- |
| Date: 25 May 2018 |  |
| *System:* Travelbnh |  |
| *Objective:* Registration/Password Validation | *Test ID:*1 |
| *Version:*1 | *Test Type:* Black Box testing |
| *Input:*  Password: 12345 | |
| *Expected Result:* Return Error of minimum 8 characters required | |
| *Actual Result:* Error of minimum 8 characters required | |



**TestCase: Registration**

***Table 2: Registration/Email Validation***

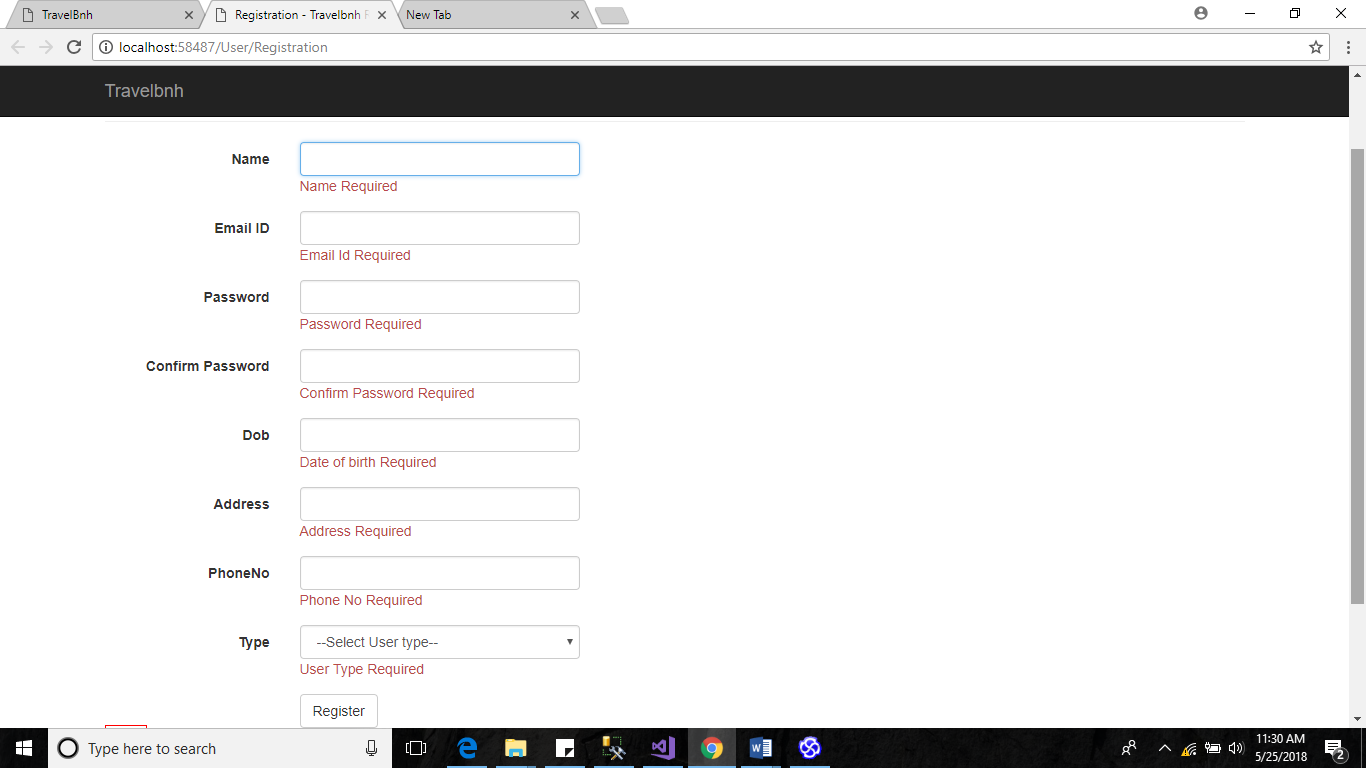
|  |  |
| --- | --- |
| Date: 25 May 2018 |  |
| *System:* TravelBnh |  |
| *Objective:* Registration/ Email Validation | *Test ID:*2 |
| *Version:*1 | *Test Type:* Black Box testing |
| *Input:*  Email Id: hamdan | |
| *Expected Result:*return Error of please enter valid email address | |
| *Actual Result:* Error of invalid email address | |



**TestCase: Registration**

**Table 3: Registration/Required Field Validation**

|  |  |
| --- | --- |
| Date: 25 May 2018 |  |
| *System:* TravelBnh |  |
| *Objective:* Registration/ Required field Validation | *Test ID:*3 |
| *Version:*1 | *Test Type:* Black Box testing |
| *Input: null* | |
| *Expected Result:*return name, password, confirm password, email, date of birth, address, phone no required | |
| *Actual Result:* Error of name, password, confirm password, email, date of birth, address, phone no required | |



**TestCase:**