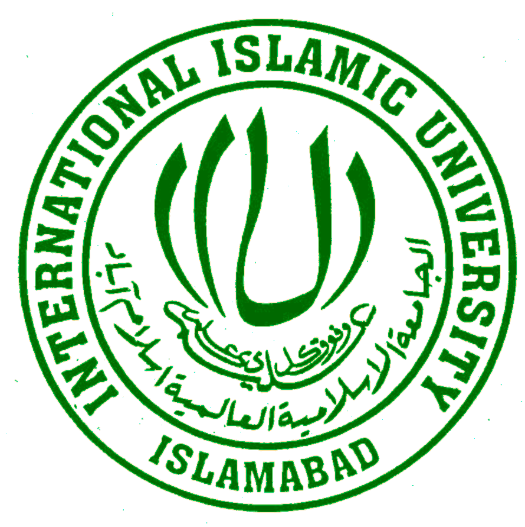
# Hostel Management System



## Project Members

Kaleem Ullah 3329-FBAS/BSSE/F16

Usman Sarwar 3342-FBAS/BSSE/F16

## Supervisor

Mr. Muhammad Nadeem

(Assistant Professor)

**DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING**

**FACULTY OF BASIC AND APPLIED SCIENCES**

# INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD

## (2020)

*Final Approval*

### FINAL APPROVAL

**Dated:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

It is certified that we have read the project report titled “Hostel Management System” submitted by **Mr. Kaleem Ullah (**3329-FBAS/BSSE/F16**)** and **Mr. Usman Sarwar (**3329-FBAS/BSSE/F16**)**. It is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for Bachelor’s Degree in Computer Science.

#### COMMITTEE

**External Examiner:**

Mr. Muhammad Nadeem

Assistant Professor

Department of Computer Sciences & Software Engineering

International Islamic University, Islamabad

**Internal Examiner:**

Mr. Iftekhar Ali Khan

Lecturer

Department of Computer Sciences & Software Engineering

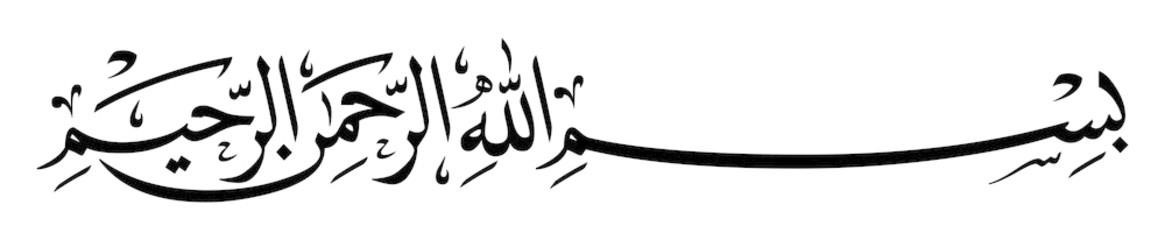
International Islamic University, Islamabad

**Supervisor:**

Mr. Muhammad Nadeem

Assistant Professor

Department of Computer Sciences & Software Engineering International Islamic University, Islamabad



"In the Name of ALLAH, the Most Beneficent, the Most Merciful"

*Dissertation*

### Dissertation

A dissertation submitted to

Department of Computer Science & Software Engineering,

International Islamic University, Islamabad

As partial fulfillment of the requirements

For award of the degree of

BS in Software Engineering.

*Declaration*

### DECLARATION

We hereby declare that we developed this software and this report entirely on the basis of our personal efforts made under the sincere guidance of our project supervisor Mr. Muhammad Nadeem. No portion of this work presented in this report has been submitted in support of our applications for any other degree or qualification of this or any other University or institute of learning. We further declare that this software, hardware and all associated documents, reports are submitted as partial requirements for the degree of BS in Software Engineering.

**Kaleem Ullah**

3329-FBAS/BSSE/F16

**Usman Sarwar**

3342-FBAS/BSSE/F16

*Acknowledgement*

### ACKNOWLEDGEMENT

We might want to express gratitude toward Almighty Allah who blessed us the fortitude and ability to finish this undertaking. Allah made us ready to use every one of our aptitudes, endeavors and information to make this venture. Without His assistance, we would not have done this venture totally. During all the periods of this venture His approval made our ways simple and we achieve this undertaking Alhamdulillah.

We would like to express my special thanks of gratitude to our respected supervisor **Mr. Muhammad Nadeem** who gave us the guidance and support to do this wonderful project on the topic (Write the topic name), which also helped us in doing a lot of Research and we came to know about so many new things. We am really thankful to them.

Secondly we would also like to thank our parents and friends who helped me a lot in finalizing this project within the limited time frame.

*Project in Brief*

|  |  |
| --- | --- |
| **PROJECT** | **T IN BRIEF** |
| Project Title | Faroshgah |
| Version | 1.0 |
| Development Team | Kaleem Ullah  Usman Server |
| Supervised by | Mr. Muhammad Nadeem |
| Date Started | Aug 2020 |
| Date Completed | June 2021 |
| Programming Languages | Java, JavaScript, HTML, CSS |
| Technologies | Express, NodeJS, React Native, MongoDB, Git |
| Tools | Android Studio, VS Code, Postman, Robo 3T, MongoDB Atlas, GitHub |
| Deployment Servers | 1. Heroku, 2. MongoDB Atlas Database- Cloud DBaaS for MongoDB |
| System Used | Core-i7, RAM-6GB, Microsoft Windows 10 |

*Abstract*

### ABSTRACT

Faroshgah, is Mobile Application used to facilitate the students, jobholders, and individuals to interact with Hostels, shopkeepers and job providers via internet.

This app that is being developed as final year project can be helpful in facilitating students to get a job online, in booking hostel online, search hostels by location, book hostel, and third important thing is that this mobile app provides a platform to buy things from shopping stores online.

This project developed using fifth generation languages and technologies. We used languages JavaScript, NodeJS and technologies like React Native.

By using these technologies we were able to develop hybrid mobile app that can run on both android and mac execution environments.

This software is fully secured, scalable, robust and flexible. It provides fully encrypted environment for sensitive data of user like passwords, we were able to develop such secured system by using JSON Web Token.

*Table of Contents*

### Table of Contents

1. INTRODUCTION ........................................................................................................................... 11
   1. PURPOSE .............................................................................................................................. 11
   2. SCOPE ................................................................................................................................... 12
   3. OVERVIEW ........................................................................................................................... 12
   4. POSITIONING ....................................................................................................................... 12
      1. Business Opportunity .......................................................................................................... 12
      2. Problem Statement .............................................................................................................. 13
      3. Product Position Statement ................................................................................................. 14
      4. PRODUCT FEATURES ......................................................................................................... 14
      5. Smart Home .........................................................................
      6. Voice Command ................................................................................................................. 14
      7. Power Monitoring ............................................................................................................... 14
      8. Bill Prediction .................................................................................................................... 14 1.5.5 Auto Mode ......................................................................................................................... 14 1.5.6 Night Mode ........................................................................... **Error! Bookmark not defined.** 1.5.7 Suggestions System ............................................................... **Error! Bookmark not defined.**

1.5.8 Security System ..................................................................... **Error! Bookmark not defined.**

* 1. ADVANTAGES ..................................................................................................................... 15
  2. CONSTRAINTS ..................................................................................................................... 15

1. SYSTEM ANALYSIS ..................................................................................................................... 17
   1. ENTITY RELATION DIAGRAM (ERD ................................................................................ 17
   2. USE CASE MODEL ............................................................................................................... 18
      1. Use Case Descriptions (Brief) ............................................................................................. 20
      2. Use Case Descriptions (Detailed) ........................................................................................ 22
2. SYSTEM DESIGN .......................................................................................................................... 28
   1. SYSTEM SEQUENCE DIAGRAM ........................................................................................ 28
   2. CLASS DIAGRAM ................................................................................................................ 34
3. IMPLEMENTATION ...................................................................................................................... 36
   1. TOOLS AND TECHNIQUES ................................................................................................. 37
   2. ALGORITHMS ...................................................................................................................... 38
      1. Add User ............................................................................................................................ 38 4.2.2 Add Room .......................................................................................................................... 38

4.2.3 Delete User ............................................................................ **Error! Bookmark not defined.** 4.2.4 Delete Room.......................................................................... **Error! Bookmark not defined.**

4.2.5 Control Device ...................................................................... **Error! Bookmark not defined.** 4.2.6 Check Power Usage ............................................................... **Error! Bookmark not defined.** 4.2.7 Set Night Mode ..................................................................... **Error! Bookmark not defined.**

*Table Of Contents*

* + 1. Set Auto Mode ...................................................................... **Error! Bookmark not defined.**
    2. Monitor Activates via Camera ................................................ **Error! Bookmark not defined.**

1. USER MANUAL ................................................................................................................................ 42
   1. Web Application ............................................................................................................................ 42

Dashboard........................................................................................... **Error! Bookmark not defined.**

Admin Panel ....................................................................................... **Error! Bookmark not defined.**

4.3.2 Mobile Application ..................................................................................................................... 56

1. TESTING ........................................................................................................................................ 58
   1. Test Case 01: Add User ........................................................................................................... 58
   2. Test Case 02: Add Room......................................................................................................... 58
   3. Test Case 03: Turn Device On/Off .......................................................................................... 59
   4. Test Case 04: Auto Mode ........................................................................................................ 60
   5. Test Case 05: Night Mode ....................................................................................................... 60
   6. Test Case 06: Set Night Mode ................................................................................................. 61
   7. Test Case 07: Account Setting ................................................................................................. 61
   8. Test Case 08: Update/Delete User or Room ............................................................................. 62
   9. Test Case 09: Camera module ................................................................................................. 62
2. CONCLUSION................................................................................................................................ 64
   1. FUTURE EXTENSION .......................................................................................................... 64

REFERENCES ........................................................................................................................................ 65

## Chapter 01

# INTRODUCTION

## 1. INTRODUCTION

With the advancement of technology, application areas of computer are rising day by day. Every  
sector desires its procurement for fast accurate and automated operations. Therefore different programs are being developed to provide facility using digital technologies.

This project entitled “Faroshgah” a project made by considering the necessities of the modern world. This system aims to provide online facilities using internet. It is a mobile app software which is helpful in the areas of Hostels, shops and jobs. This system provides facility of booking hostel online, shopping online and even someone can apply for job by using this mobile app. This project aims to connect students with hostels via internet, buying things online, and provides facility to get job online.

In terms of security this software is totally secure as it doesn't give access to unauthorized user as the sensitive data like passwords are fully encrypted.

### 1.1 PURPOSE

The main objective behind preparing this project is that to make it possible to book hostel online, get job online, and shop online:

1) To find the Hostel online with ease.  
2) To save time.  
3) To increase efficiency of students.  
4) For fast access of hostels, jobs and shops.  
5) For secure and smooth running of the program.  
6) For error free, effective and easy for database related works.  
7) indicate the hostel availability and the location of the hostels;

### 1.2 SCOPE

Enabling students/jobholders to connect with digital world. Providing facility of using internet in a useful manner form booking things online to buy things online by using fully secured system.

Finding good hostels full of facilitation, finding hostels of your demand on one click easily, in a new city. For finding hostels of your requirements will be easy. Reserving room in the hostel.

The application will help the owners to promote the businesses, as lot of people will get them to know about their services.

* Make hostel, shop and job finding digital
* Making system easy and fast
* Finding reasonable hostels by price, location, charges and facility provided in the place of your choice.
* Make searching shops and jobs online and easy to find.

This system will consist of:

* Owner (who present his service)
* User/Client (who acquire the service)
* A system and Internet Connection
* User Interface , Mobile Application

### 1.3 OVERVIEW

Faroshgah had been development using emerging languages and technologies like NodeJS and React Native, JWT. The main purpose of this program is to enable people to use internet of their ease and facility instead of just entertainment. Due to the fact that NodeJS, React Native are open source programs but also have a very large community, that’s why development of Faroshgah was not too  
difficult and we enjoyed it and learned lots of new technologies covered in developing this project

### 1.4 POSITIONING

#### 1.4.1 Business Opportunity

This application will be more like a Platform for those who want to promote their Hostel business online, for shopkeepers who want to sell online and for job providers who want to get employees online instead of boring advertisement which seeks both money and time. Users can find best and cheap Hostels without spending any money on searching by visiting manually and it reduces time. It also provides an opportunity to get a job online.

#### 1.4.2 Problem Statement

|  |  |
| --- | --- |
| **The problem of** | Difficult to find good hostels for new visitors in a city. Difficult to find jobs and shops within specified area and exact requirements. It is difficult and time taking to go to Streets and cities to look for desired shops, hostels and jobs there are many problems for searching things and there is no platform to upload and manage businesses like this.  By making this online we want to make strong or trustful interaction between user and Providers.  It is difficult for shopkeepers to attract customers while being in a low running area.  It is difficult for students and needy people to look for hostels for living and manage hostels manually. |
| **Affects** | For Hostel owners, it is difficult, time consuming and costly to advertise their business manually.  For students and job holders outside the country it is Difficult to find best hotels, for job seekers there is great opportunity we provide that they can find jobs easily by location. It is very important that you can find shops listing items and go directly to that shop of your interest by location. |
| **the impact of which is** | Hostel owners will enjoy this great opportunity of posting their hostel online and hostel seekers can find online and can book hostel seat online. Hostel seekers can find hostel by location.  Someone can find jobs of specific location and by items he/she want to get or buy from shop. Also job seekers can find jobs and can apply for job in desired city. |
| **a successful solution would be** | Easy access to hostels with best prices and location. Reduce finding time and booking time. Getting desired jobs online by using this platform and buying things online from shops would be really great. |

#### 1.4.3 Product Position Statement

|  |  |
| --- | --- |
| For | Owners & Customers, job seekers, job providers |
| Who | Wants to present and acquire accommodation services as well as want a platform for getting jobs online and buying things online. |
| That | By using this software hostel owners don’t need to advertise their service in a manual way and students, job holders can get hostel by their choice using online system instead of visiting each hostel’s door. |
|  |  |
|  | . |

### 1.5 PRODUCT FEATURES

#### 1.5.1 Login/ Signup

All users must be signed up for getting access to this app. Only logged users can get access to the features of app it provides

#### 1.5.2 Creating Accommodation Service

Owner can provide service to others like to students and jobholders. Service can be like

Hostel searching, booking hostel room online, making payment via credit card.

#### 1.5.2 Acquiring Accommodation Service

On the Client side, the user can acquire rooms booking service online.

#### 1.5.3 Search for the Service

The user can search for the services by specifying the location and can give review about the service.

#### 1.5.4 Online Payment

User can also pay online. Through Online payment feature the user can pay online to the specified account of the owner or can pay directly to the owner.

#### 1.5.5 Buy Things Online

One of the most important feature of our app is that is provides facility of buying things online.

#### 1.5.6 Post Job

Employers can post jobs by using this platform. In this way they can interact with users which are part of that system.

#### 1.5.7 Apply for Job Online

One of the super cool feature for users of that app is that they can get benefit of applying for job of their choice without any hesitation.

#### 1.5.8 Rating & comments

Those who are using our app at the client end, can give review to hostel, shop. In this way we can keep track of the service provided by the owner. .

#### 1.5.9 Getting Notifications

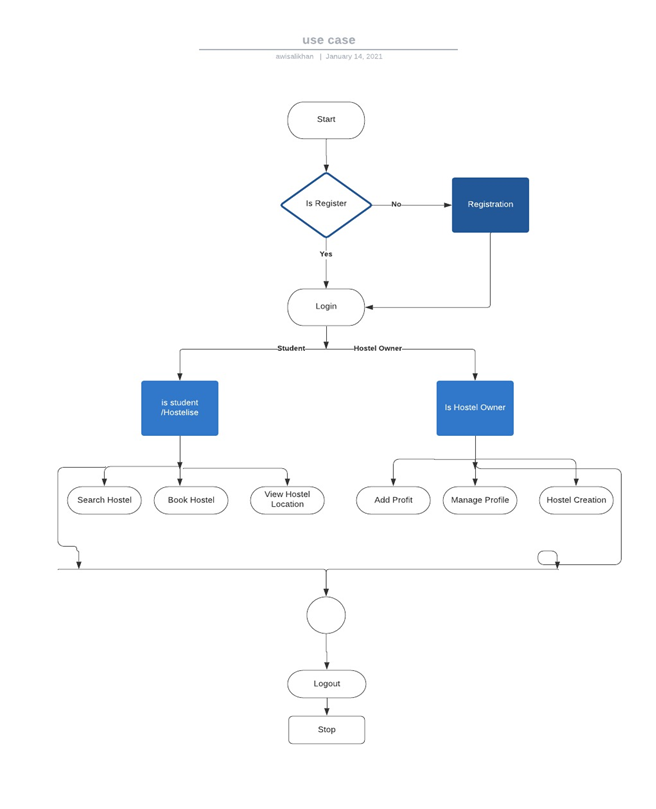
Employers and hostel owners get notifications when someone book hostel.

### 1.6 ADVANTAGES

* Owners will easily add and register their accommodation service by using this app with ease. User interface is quite simple and easy that anyone can understand the system easily.
* Users will get the required service through the App easily and can book it online
* Users will be able to search and check nearby Hotels, shops, and jobs by search. They can compare and can choose the appropriate one for themselves.
* It assures zero fraud service because each registered owner should have to follow the rules and regulations of ours.

### 1.7 CONSTRAINTS

* Both owner and client, should sign up before registering or acquiring any service.
* Only logged users can access the features and services of this application.
* User must signed up if he/she wants to use our service app.



# Chapter 02

# System Analysis

## 2. SYSTEM ANALYSIS

To identify goals and purposes of the system we use system analysis process. System analysis help to distinguish system’s objectives and purposes and gives a clear picture about the system requirements. In this way we can achieve reliable and trustworthy software.  
By using system analysis, we can understand the problem and its context. The system analysis can be very useful when we study the  
existing system and other is to understand the requirements and domain of the new system.  
Subsequent to investigating the prerequisites of the undertaking to be played out, the following  
stage is to examine the issue and comprehend its unique circumstance. The main movement in the  
stage is examining the current framework. What's more, other is to comprehend the necessities and  
space of the new framework. Similarly imperative, yet the main movement fills in as a principle  
of giving the useful particulars and afterward effective structure of the proposed framework.  
Understanding the properties and necessities of another framework is increasingly troublesome  
and requires innovative reasoning and comprehension of existing running framework is likewise  
troublesome, ill-advised comprehension of present framework can lead redirection from  
arrangement.  
The model that is fundamentally being pursued is the Water fall Model, which expresses  
that the stages are sorted out in a straight request. As a matter of first importance the attainability  
consider is finished. Analysis model includes:-

1. Use case diagram
2. Use case diagram with description
3. System sequence diagram

#### 2.1 Use Case Diagram

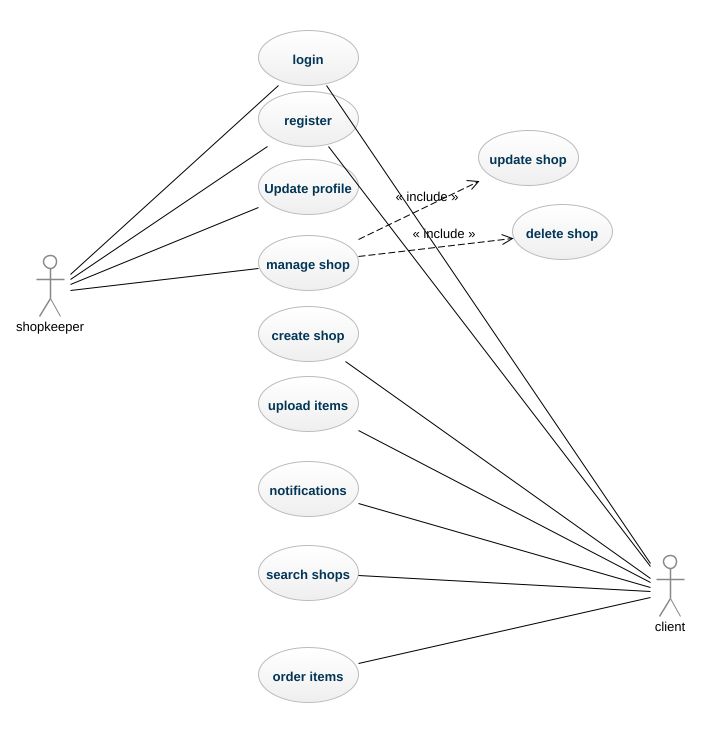
A use case diagram is used to represent the set of actions (use cases) that system can perform  
with in collaboration with external users of the system (actors). By using use case diagram we can figure out what a user can do with system by interacting with it. Use case diagram help us to find out the real actions of users that can be performed.

Figure 2.1 shows the relationship  
between the user and different use cases in which the user is involved.

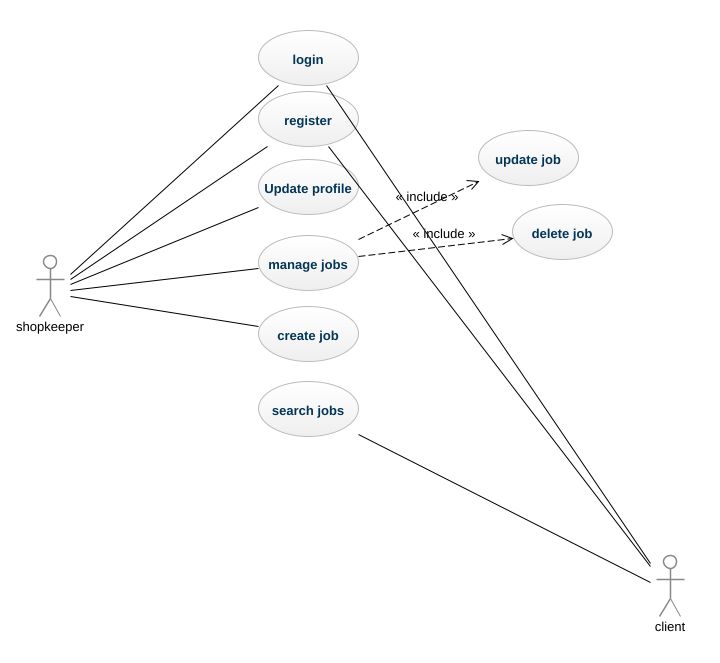
**2.1.1 Hostel Use Case Diagram**



**Shops Use Case Diagram**



**Jobs Use case diagram**



##### 2.2 Use Case Description in Detail Expended Format:

1. In software and systems engineering, a use case shows list of actions or event steps that is interactions between a role and a system to achieve a goal. The actor can be a human or other external system.
2. **2.2.1 Use Case Registration**

Details of use case registration which tells how user can register to the application

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-01** |
| **Scopes** | Login screen for hostel owner and Client/student to access the system |
| **Name** | Registration |
| **Primary Actors** | Hostel owner, student |
| **Goals** | To enable user to run the system. |
| **Pre-Conditions** | User has opened the app and on the registration screen. |
| **Post-Conditions** | Account Created successfully. |
| **Main Success Scenarios** | User requests for registration.  Opt message sent for verification.  User verifies the OTP and Redirected to login after successful registration.  (User can register himself as Hostel owner or simply as user) |
| **Alternates** | The user credentials are not valid.  The user already registered in the system. |

**2.2.2 Use Case Login:**

Detail of use case login which tells a login scenario

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-02** |
| **Scopes** | Meet-up Platform for hostels and students |
| **Name** | Login |
| **Primary Actors** | Hostel owner, student/client |
| **Goals** | To Enable user to run the system. |
| **Pre-Conditions** | User Account Created successfully. |
| **Post-Conditions** | Redirect to home. |
| **Main Success Scenarios** | User Provides username and password.  Redirect to Home page after successful attempt. |
| **Alternates** | Incorrect Username,  Forget password by user. |

* + 1. **Use Case Add hostel:**

Detail of use case post hostel where he can post hostel by providing detail/description

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-04** |
| **Scopes** | Online system for hostels and Client |
| **Name** | Add Hostel |
| **Primary Actors** | Hostel owner |
| **Goals** | To show hostel online to hostel seekers. |
| **Pre-Conditions** | Login |
| **Post-Conditions** | Hostel created |
| **Scenarios** | Creating hostel and publishing online |
| **Alternates** | Register yourself.  Login using your username and password. |

* + 1. **Use Case Search Hostel :**

Detail of use case Search hostel where user can hostel.

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-05** |
| **Scopes** | Meet-up Platform for hostel and students |
| **Name** | Search Hostel |
| **Primary Actors** | Student/user |
| **Goals** | To Enable students/job seekers to search hostels. |
| **Pre-Conditions** | Login, user can search hostel after login |
| **Post-Conditions** | Searching for hostels is done |
| **Scenarios** | User clicks the search bar.  Searches the hostels by city or location/street/zipcode. |
| **Alternates** | Didn’t match your search. |

###### Use Case Update Hostel Record:

1. Detail of use case post job which tells the scenario where user posts a job/requirements

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-05** |
| **Scopes** | Meet-up Platform for hostel owners and clients/students |
| **Name** | Update Hostel Record |
| **Primary Actors** | Hostel owner |
| **Goals** | To update the record of hostel |
| **Pre-Conditions** | Login, only admin or authenticated hostel owner can do that |
| **Post-Conditions** | Updating record of hostel is completed |
| **Scenarios** | Updating record of hostel |
| **Alternates** | Register yourself.  Only authenticated are allowed  Login using your username and password. |

###### 2.2.6 Use Case 06: Delete Hostel Record

Detail of use case delete hostel record

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-04** |
| **Scopes** | Meet-up Platform for hostels and students |
| **Name** | Delete hostel record |
| **Primary Actors** | Hostel owner |
| **Goals** | To delete the record of hostel |
| **Pre-Conditions** | Login, register as a hostel owner |
| **Post-Conditions** | Deletion of hostel is done |
| **Scenarios** | Deleting hostel record from database |
| **Alternates** | Register yourself.  Only authenticated hostel owners can do that  Login using your username and password. |

Table 2. 6 UC deleting hostel record

###### 2.2.7 Search hostel within a specific location

Detail of use case search hostel within given location

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-07** |
| **Scopes** | Meet-up Platform for hostel owners and students/Client |
| **Name** | Search hostel within location |
| **Primary Actors** | Students/client |
| **Goals** | To get hostels located in chosen location |
| **Pre-Conditions** | Navigate to the search page |
| **Post-Conditions** | Searching of hostels is done |
| **Scenarios** | Searching hostel |
| **Alternates** | Try different location |

Table 2. 7 UC Search hostel within location

###### 2.2.8 8: Book Hostel’s Seat

Detail of use case book hostel which tells the scenario where user can book a hostel seat

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-08** |
| **Scopes** | Meet-up Platform for hostels and students |
| **Name** | Book hostel room/seat |
| **Primary Actors** | Student/client |
| **Goals** | To book hostel online |
| **Pre-Conditions** | Login, add profile |
| **Post-Conditions** | Booking of hostel is done |
| **Scenarios** | Booking hostel is done |
| **Alternates** | Register yourself.  Login using your username and password. |

Table 2. 8 UC Book Hostel’s seat

**2.2.9** **USE CASE 09: Review or Rating**

Detail of use case post job which tells the scenario where user posts a job/requirements

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-09** |
| **Scopes** | Meet-up Platform for hostels and students |
| **Name** | Review or Rating |
| **Primary Actors** | Student/ owner |
| **Goals** | To give the rating to the hostel |
| **Pre-Conditions** | Login, add profile |
| **Post-Conditions** | Rating is given to the hostel |
| **Scenarios** | Giving rating to hostel |
| **Alternates** | Register yourself.  Login using your username and password. |

Table 2. 8 UC Review or rate hostel

**Shop Management System**

* **2.2.10 USE CASE-10: Create New Shop**

Detail of use case create new shop which tells the scenario where shopkeeper creates a shop

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-10** |
| **Scopes** | Meet-up Platform for shopkeepers and clients |
| **Name** | Create shop |
| **Primary Actors** | Shopkeeper |
| **Goals** | To create shop on portal |
| **Pre-Conditions** | Login, add profile |
| **Post-Conditions** | Creation of shop is done |
| **Scenarios** | Creation of shop on portal |
| **Alternates** | Register yourself.  Login using your username and password. |

Table 2.10 Use Case Create Shop on Portal

**2.2.11 Upload Items for Shop**

Detail of use case Upload items with its category and detail description

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-11** |
| **Scopes** | Meet-up Platform for shopkeeper and Client |
| **Name** | Upload items for shop |
| **Primary Actors** | shopkeeper |
| **Goals** | To upload items for shop |
| **Pre-Conditions** | Login, only shopkeeper can do that |
| **Post-Conditions** | Uploading items done |
| **Scenarios** | Uploading items for shop |
| **Alternates** | Register yourself.  Login using your username and password. |

Table 2.11 Upload items for shop

2.2.12 Update Items for Shop

Detail of use case update items which tells the scenario where user update items

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-12** |
| **Scopes** | Platform for shopkeepers and Client |
| **Name** | Update items for shop |
| **Primary Actors** | Shopkeeper |
| **Goals** | To update items for shop |
| **Pre-Conditions** | Login, add profile |
| **Post-Conditions** | Updating item is done |
| **Scenarios** | Updating items for shop |
| **Alternates** | Register yourself.  Login using your username and password. |

Table 2.4 Use Case Updating Items for Shop

**2.2.13 Delete Shop Items & detail**

Detail of use case delete shops which tells the scenario where shopkeeper deletes his shop

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-13** |
| **Scopes** | Platform for Shopkeeper and Client |
| **Name** | Delete Shops and its detail |
| **Primary Actors** | shopkeeper |
| **Goals** | To delete the shop from database |
| **Pre-Conditions** | Login, add profile |
| **Post-Conditions** | Deleting shop is done |
| **Scenarios** | Deleting shop and its detail |
| **Alternates** | Register yourself.  Login using your username and password. |

Table 2.4 Use Case Delete Shops

**2.2.14 USE CASE: 14 Search Shop**

Detail of use case post job search which tells the scenario where user can search a shop

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-04** |
| **Scopes** | Platform for shopkeepers and Client |
| **Name** | Search shops |
| **Primary Actors** | client |
| **Goals** | To search the desired shops |
| **Pre-Conditions** | Come to the search page |
| **Post-Conditions** | Searching is done |
| **Scenarios** | Searching shops online |
| **Alternates** | Try to locate by searching alternative keywords. |

Table 2.14 Use Case Search Shop

**2.2.15 Order Item**

Detail of use case order item which tells the scenario where user orders an item

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-15** |
| **Scopes** | Platform for shopkeepers and Client |
| **Name** | Order an item |
| **Primary Actors** | Client |
| **Goals** | To buy an item from the shop |
| **Pre-Conditions** | Login, add profile |
| **Post-Conditions** | Order is done |
| **Scenarios** | Order items from the shop |
| **Alternates** | Register yourself.  Login using your username and password. |

Table 2.15 Use Case Orders Item from the shop

**USE CASES FOR JOBS**

**2.2.16 USE CASE: 16 Create Job**

Detail of use case post job which tells the scenario where user posts a job/requirements

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-16** |
| **Scopes** | Platform for Freelancers and Client |
| **Name** | Create jobs for client |
| **Primary Actors** | Provider |
| **Goals** | To show jobs online for job seekers. |
| **Pre-Conditions** | Login, add profile |
| **Post-Conditions** | Uploading jobs is done |
| **Scenarios** | Uploading jobs for job seekers |
| **Alternates** | Register yourself.  Login using your username and password. |

Table 2.16 Use Case Add Jobs

**2.2.17 Update Jobs detail**

Detail of use case manage jobs by clients where client can manage jobs of itself uploaded for freelancers.

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-17** |
| **Scopes** | Platform for hostels, jobs, shops, owners and Client |
| **Name** | Update job detail |
| **Primary Actors** | Provider |
| **Goals** | Job updating by client those already added |
| **Pre-Conditions** | Registered and logged-in client. |
| **Post-Conditions** | Jobs are managed successfully. |
| **Scenarios** | Client updates the jobs for job seekers |
| **Alternates** | Redirect to home. |

### Table 2.17 Use Case Update Jobs Detail

**2.2.18 Delete the Job**

Detail of use case delete job by provider where provider can delete jobs if no more available

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-18** |
| **Scopes** | Platform for job providers and job seekers |
| **Name** | Delete Job |
| **Primary Actors** | Provider |
| **Goals** | delete jobs by provider those already added |
| **Pre-Conditions** | Registered and logged-in client. |
| **Post-Conditions** | Jobs are deleted successfully. |
| **Scenarios** | Providers defines the jobs for job seekers and delete the job. |
| **Alternates** | If there is no such module, redirect to home screen |

### Table 2.10 Use Case Delete Jobs by Provider

# CHAPTER 3

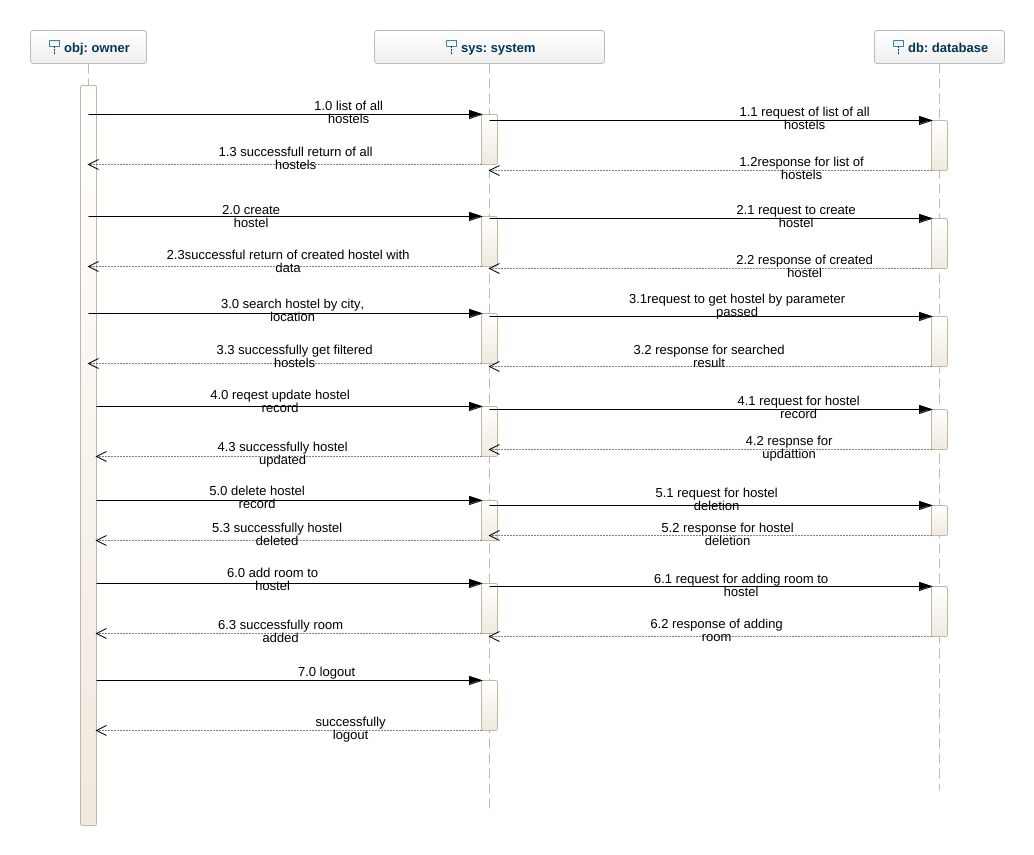
**SYSTEM DESIGN**

System Design is a chapter of our Final Year project documentation in which we have  
explained the design of the system from the implementation perspective.  
We have developed a Class diagram, Sequence diagram and Entity relation diagram. The Class  
diagram describes the classes, their attributes, methods and other object oriented concepts in  
the system. Sequence diagram show a message passing between the objects in a sequential  
manner. Similarly, ERD is an extended version of a domain model and it describes the how  
the tables are created, their fields and the relationships among them.  
The system design helps the developer in the Implementation phase as it gives a basic starting  
to the developer of the system.

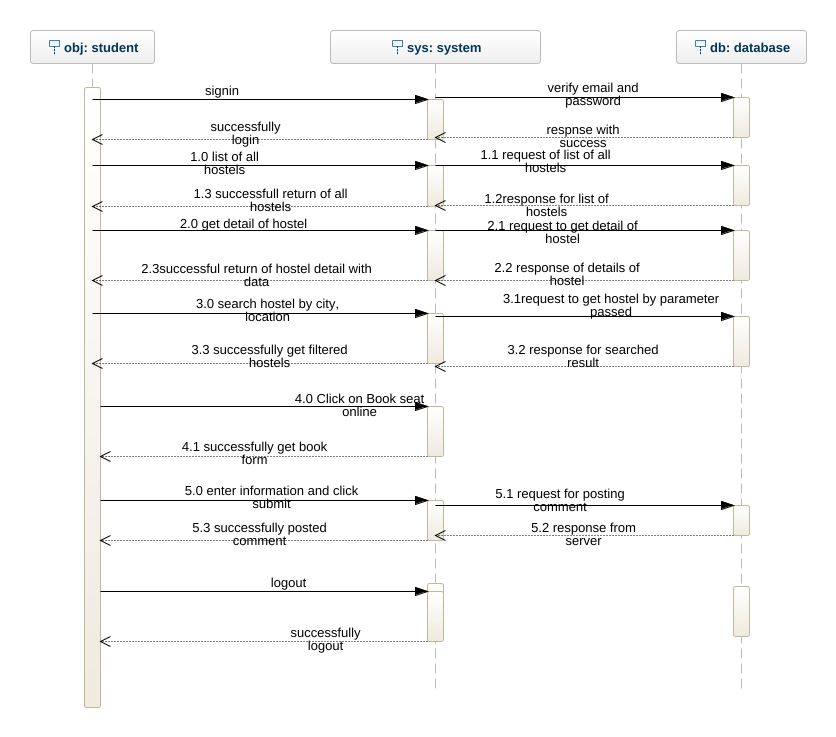
**3.1 System Sequence Diagram**

A sequence diagram of hostel management system represents an object and shows all its points of interaction with other objects in events that are important to it. Figure 3.1 shows the sequence of messages exchanged between objects of hostels and clients

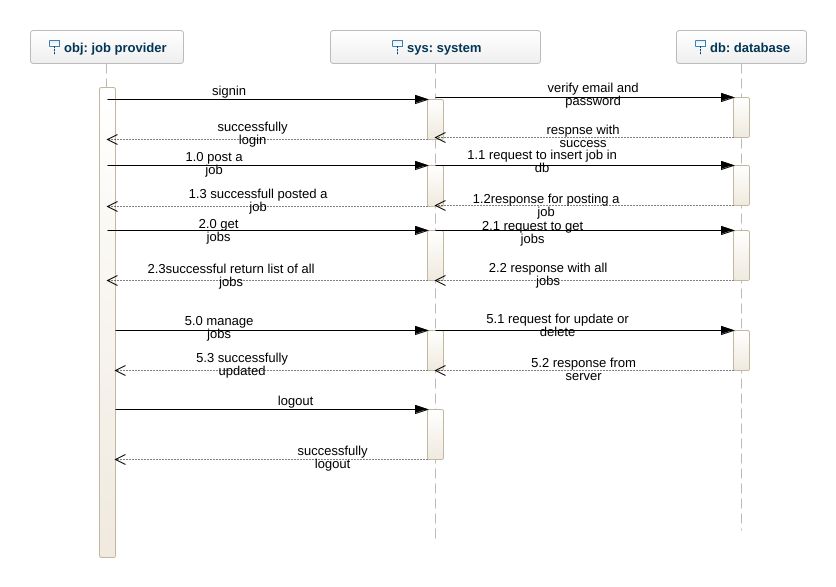
**3.2.1 Sequence Diagram for owners**



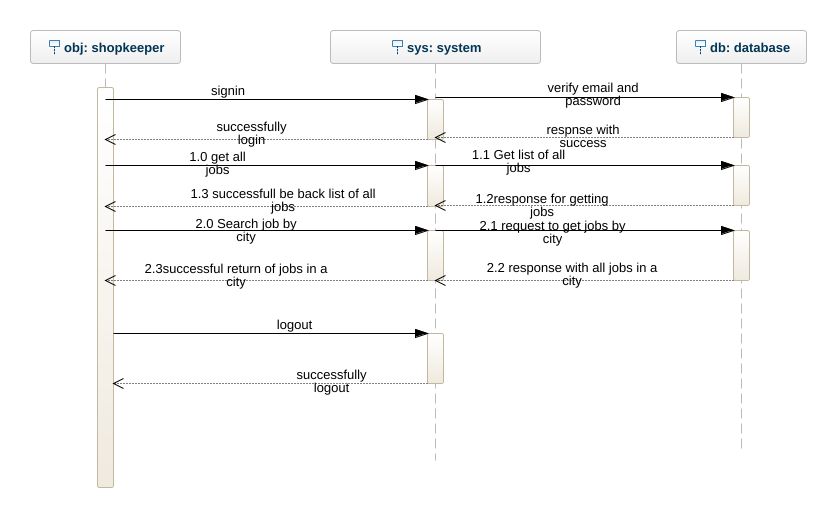
**3.2.2 Sequence Diagram for students**



**3.23 Sequence Diagram for Job Provider**

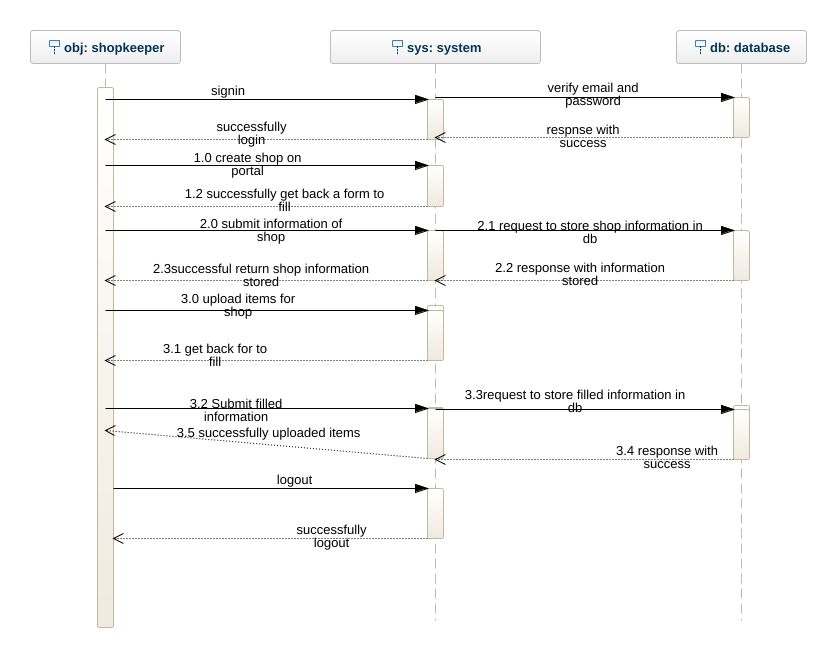


**3.2.4 Sequence Diagram for Job Seeker**

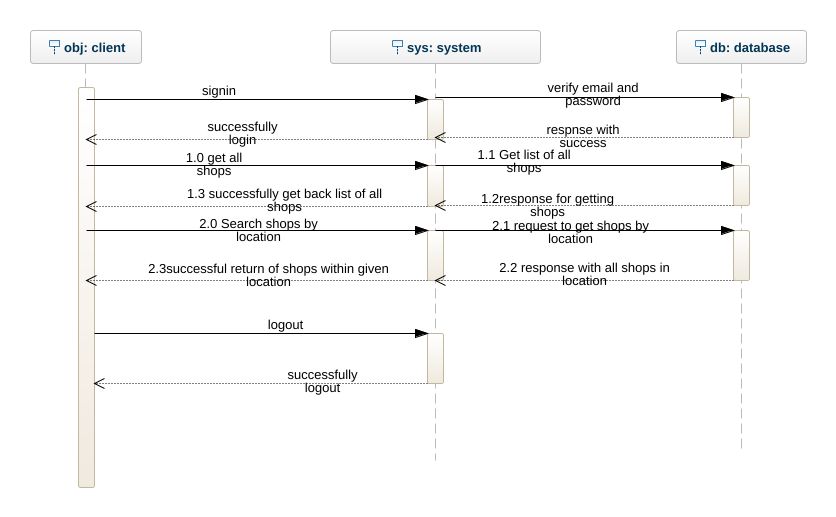
****

**3.2.4 Sequence Diagram for Shopkeeper**

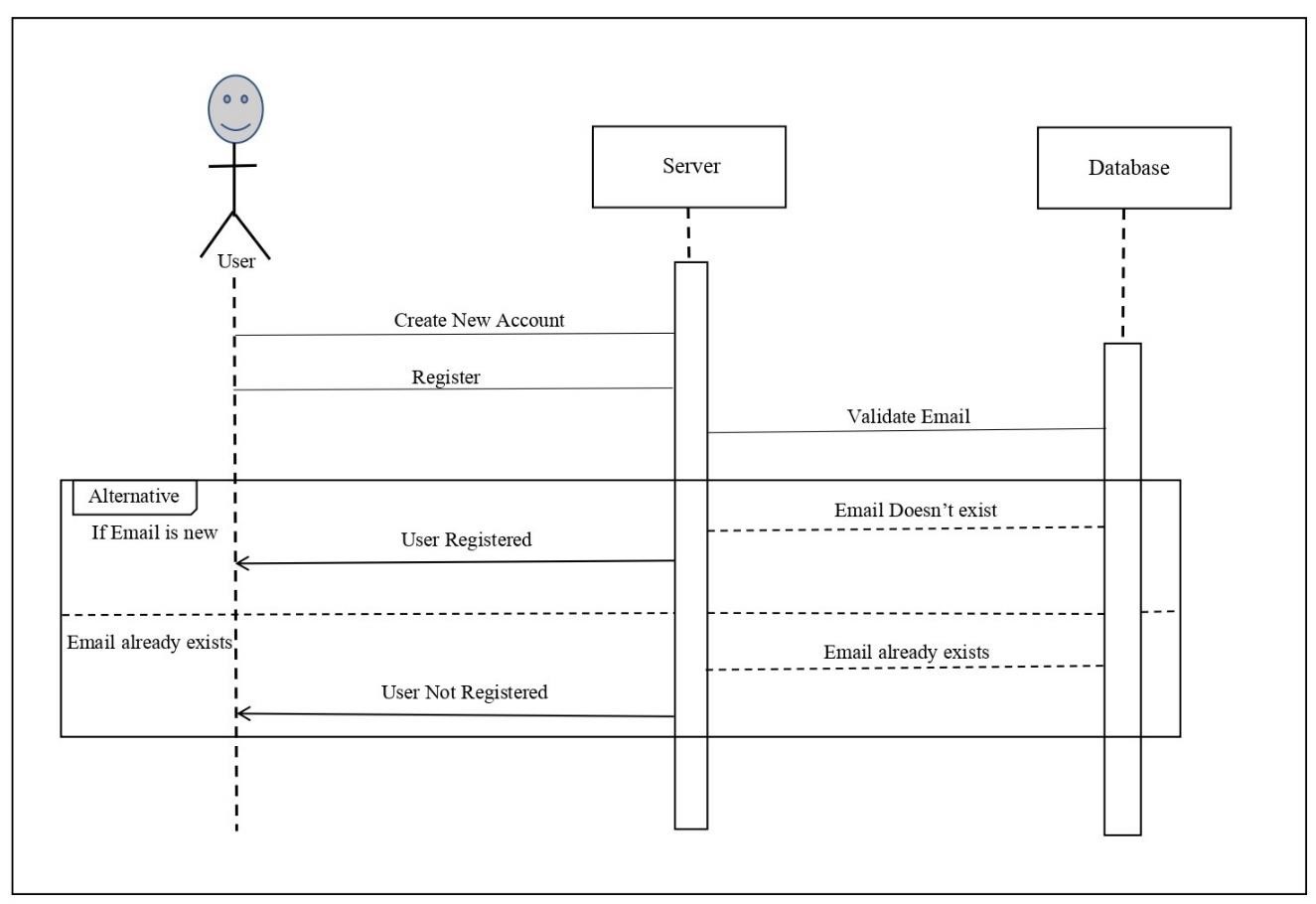
**3.2.4 Sequence Diagram for Shopkeeper**

****

**3.2.5 Sequence Diagram for client who wants to buy something from shop**

****

### 3.2.6 Register New User



*System Design*

32

*System Design*

*System Design*

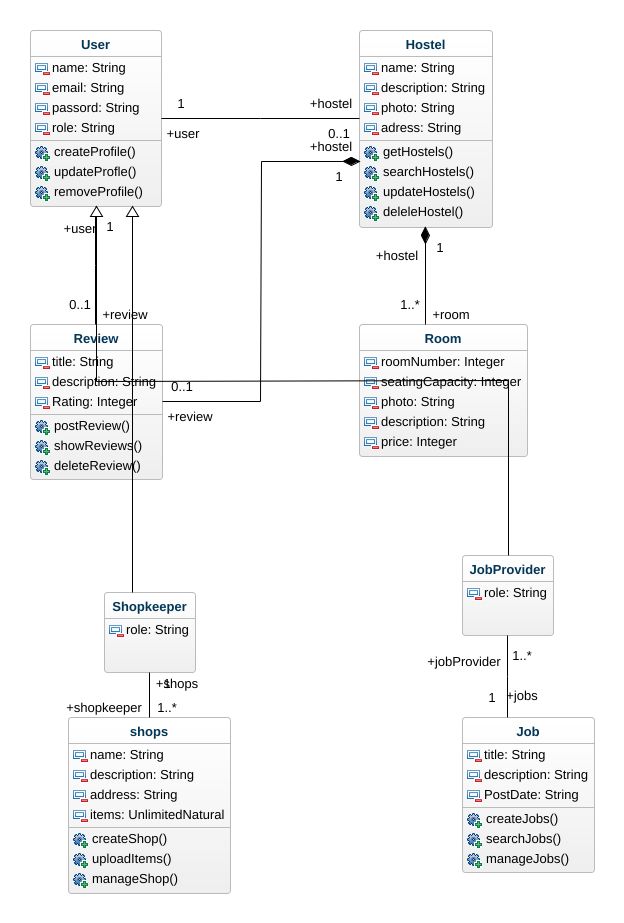
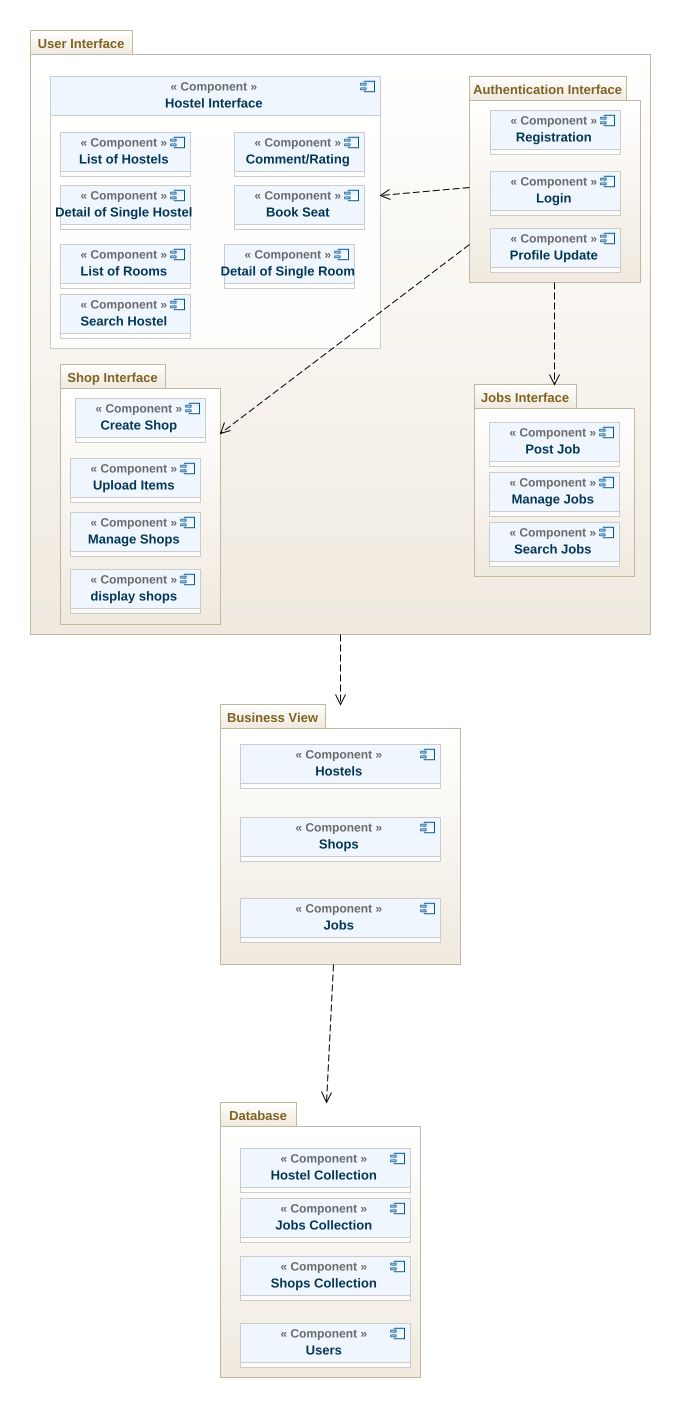
#### 3.2 CLASS DIAGRAM

A class diagram is primarily designed for developers to provide the conceptual model and architecture of the system being developed. Typically, a class diagram consists of more than one class or all the created classes for a system.

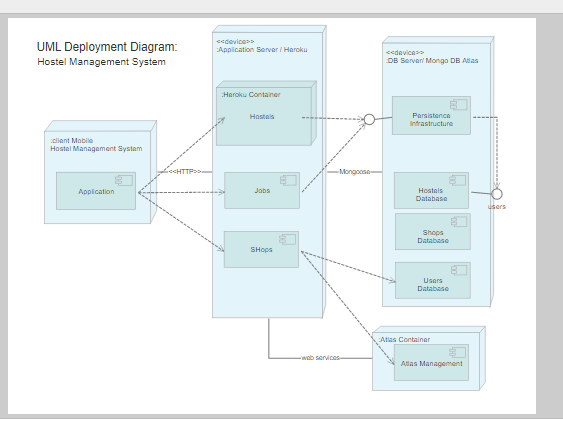
Considered to be the type of the structural diagrams as it represents the whole structure of the specific systems. Also has the similarity with the flow chart. Class diagram has three main parts represented by the rectangular box. Three parts represent the following things:

1. First part: Name of the specific class.
2. Second part: Qualities or Attributes of the specific class.
3. Third part: List of functions a specific class can execute.

Figure 3. 2 Class Diagram

**3.4 Deployment Diagram**

****

# CHAPTER 4

**IMPLEMENTATION**

## 4. Implementation

Implementation is basically another word for coding/software programming. This chapter includes the component diagrams and deployment diagram. The implementation phase explains the design of the internal coding structure of the software.

**4.1 Functional Features of the Model:**

The main features of the system are:

User friendly

Friendly interface

Responsiveness

Reliability

Security

## 3.3 TOOLS AND TECHNIQUES

|  |  |  |  |
| --- | --- | --- | --- |
| **Platform:** API Based Mobile App | |  | |
| **1.** | Web |  | |
| **2.** | Android |  | |
| **3.** | iOS |  | |
| **Tools and Technologies:** | |  | |
| **Front End:** | | **Back End:** | |
| **1.** | React Native | **1.** | NodeJS |
| **2.** | Android Studio | **2.** | MongoDB Cloud |
| **3.** |  | **3.** | Express framework |
| **4.** |  | **4.** | Google Map, Geocoder |
| **5.** |  | **5.** | Ionic framework |
| **6.** |  |  |  |
| **7.** |  |  |  |
| **Supporting Tools and Technologies:** | |  | |
| **1.** | Visual Studio Code |  | |
| **2.** | Robo 3T |  | |
| **3.** | GitHub |  | |
| **4** | Postman |  | |
| **5.** | Android Studio |  | |
| **6.** | Heroku (used for live backend) |  | |
| **7.** |  |  | |

**4.2 Component Diagram**

Figure 4.1 shows the relationship between different components in our application. It depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

## 3.4 ALGORITHMS

### 3.4.1 Add User

Step 1: Start

Step 2: Open the add user form.

Step 3: Add all the required information of the user

Step 4: Once all the information is added click register.

Step 5: System will save the information and sends the verification email to the user. Step 6: End

### 3.4.2 Add Hotel Step 1: Start

Step 2: Login as Owner

Step 3: Open the Add Hotel form

Step 4: Add all the required information that is to be added.

Step 5: Add the room information

Step 6: Click Save

Step 7: End

### 3.4.3 Add Restaurant

Step 1: Start

Step 2: Login as Owner

Step 3: Open the Add Restaurant form

Step 4: Add all the required information that is to be added.

Step 5: Add the menu information

Step 6: Add the tables information

Step 7: Click on save button Step 8: End

### 3.4.4 Add Flat Step 1: Start

Step 2: Login as Owner

Step 3: Open the Add Flat form

Step 4: Add all the required information that is to be added

Step 5: Click on Save button Step 6: End

### 3.4.5 Add Marriage Hall Step 1: Start

Step 2: Login as Owner

Step 3: Open the Add Flat form

Step 4: Add all the required information that is to be added

Step 5: Add the menu types and information Step 6: End

### 3.4.6 Book Hotel Step 1: Start

Step 2: Login as Client into the system

Step 3: Select the Hotel from the category

Step 4: Search and select Hotel

Step 5: Click on Show Room Information

Step 6: Fill Room Booking form

Step 7: Book Room

Step 8: Pay payment online

Step 9: End

### 3.4.7 Reserve Restaurant

Step 1: Start

Step 2: Login as Client into the system

Step 3: Select the Restaurant in category

Step 4: Search and select Restaurant

Step 5: Check Menu

Step 6: Check Tables

Step 7: Fill Reservation form

Step 8: Make Reservation

Step 9: End

### 3.4.8 Rent Flat Step 1: Start

Step 2: Login as Client into the system

Step 3: Select Flats in category

Step 4: Search and select Flat

Step 5: Select Rent

Step 6: Fill renting form

Step 7: Submit form Step 8: End

### 3.4.9 Book Marriage Hall

Step 1: Start

Step 2: Login as Client into the system

Step 3: Select Marriage Hall in category

Step 4: Search and select Marriage Hall

Step 5: Select Book

Step 6: Fill Booking Form

Step 7: Save

Step 8: End

**CHAPTER 5**

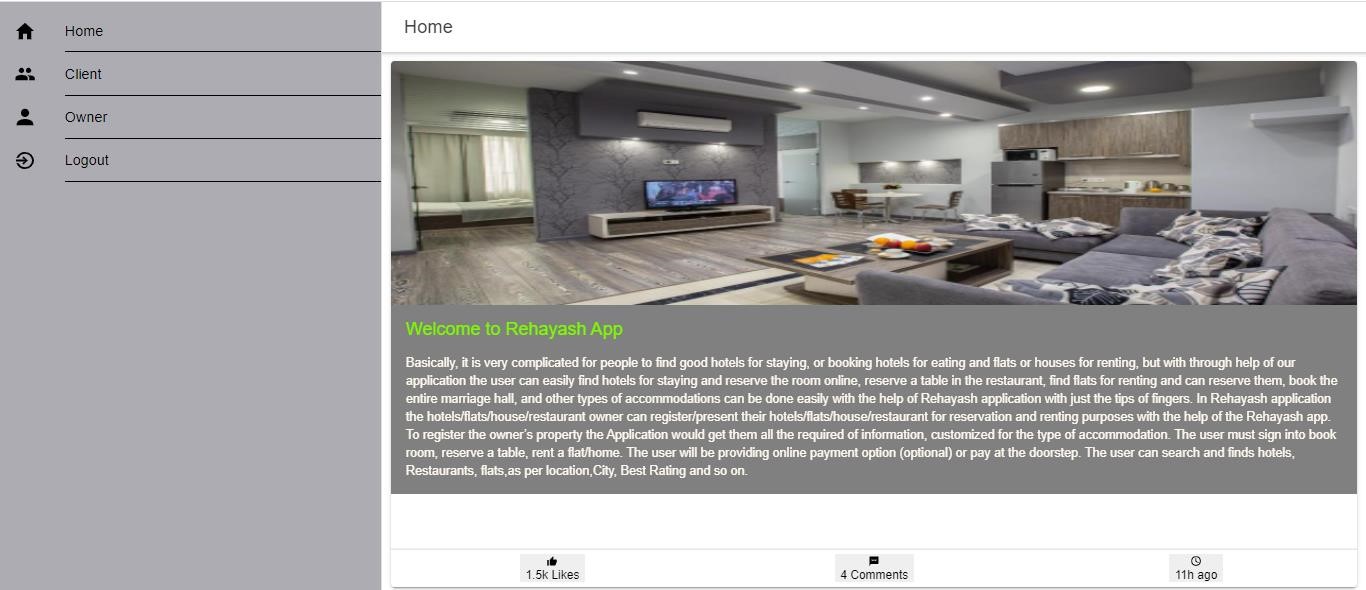
**USER MANUAL**

*User Manual*

## 4. USER MANUAL

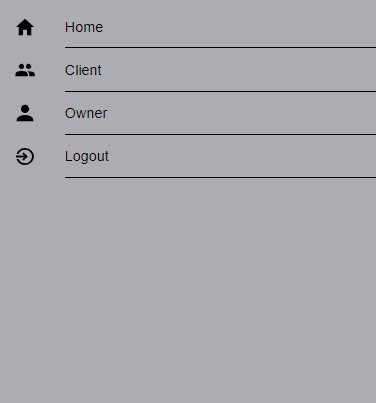
### 4.1 Web Application

This is the homepage of the website. First time when a user open the website this is the first thing that they will see.



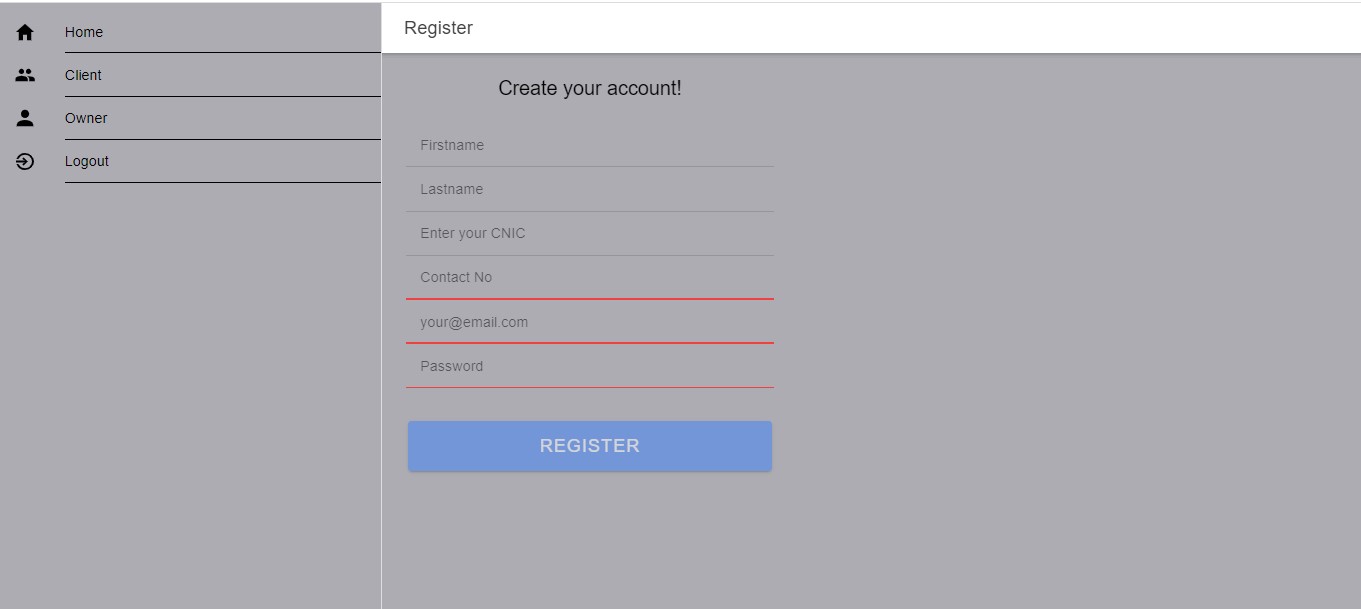
#### Menu

Menu on the website can be accessed on the **Menu** in menu Bar on the left.

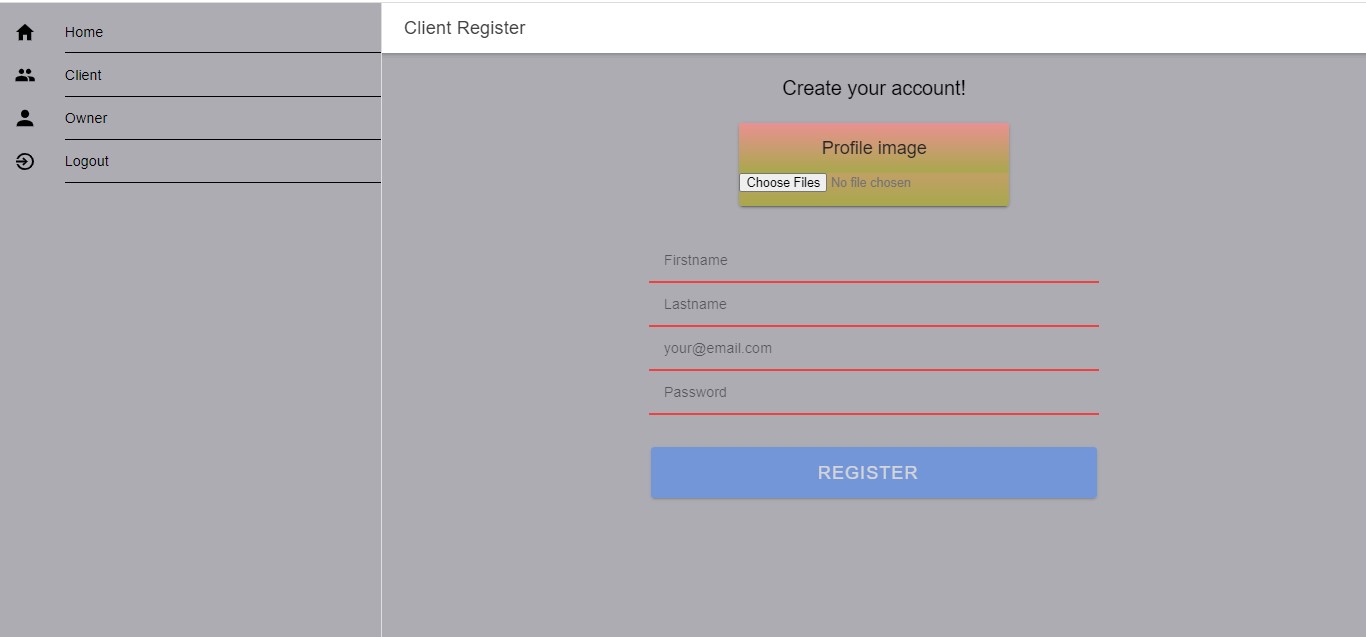


*HomeHaven 42*

Click on the **Client and Owner** button and it will redirect user to a login page if they are not already logged in.

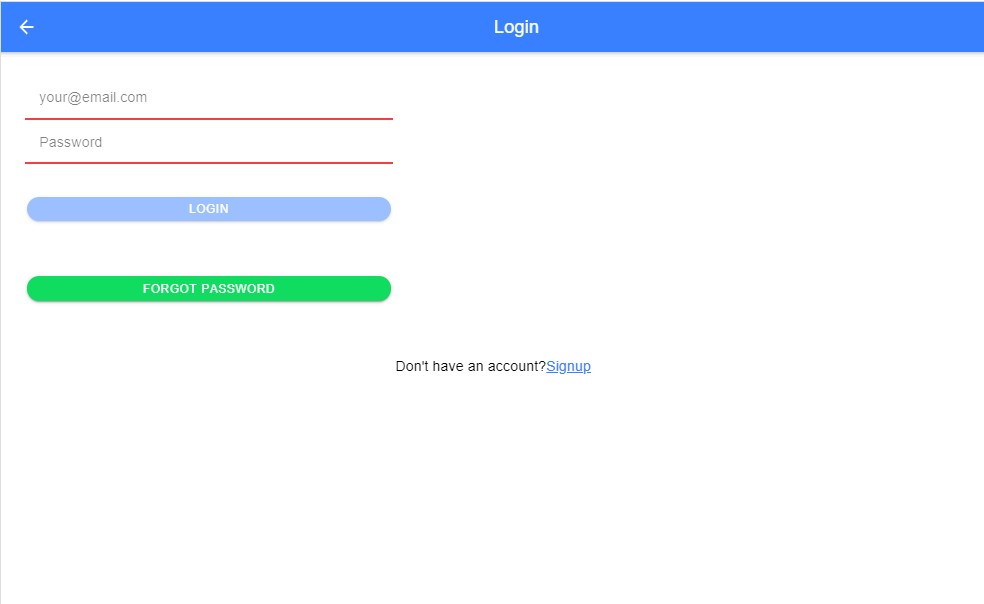
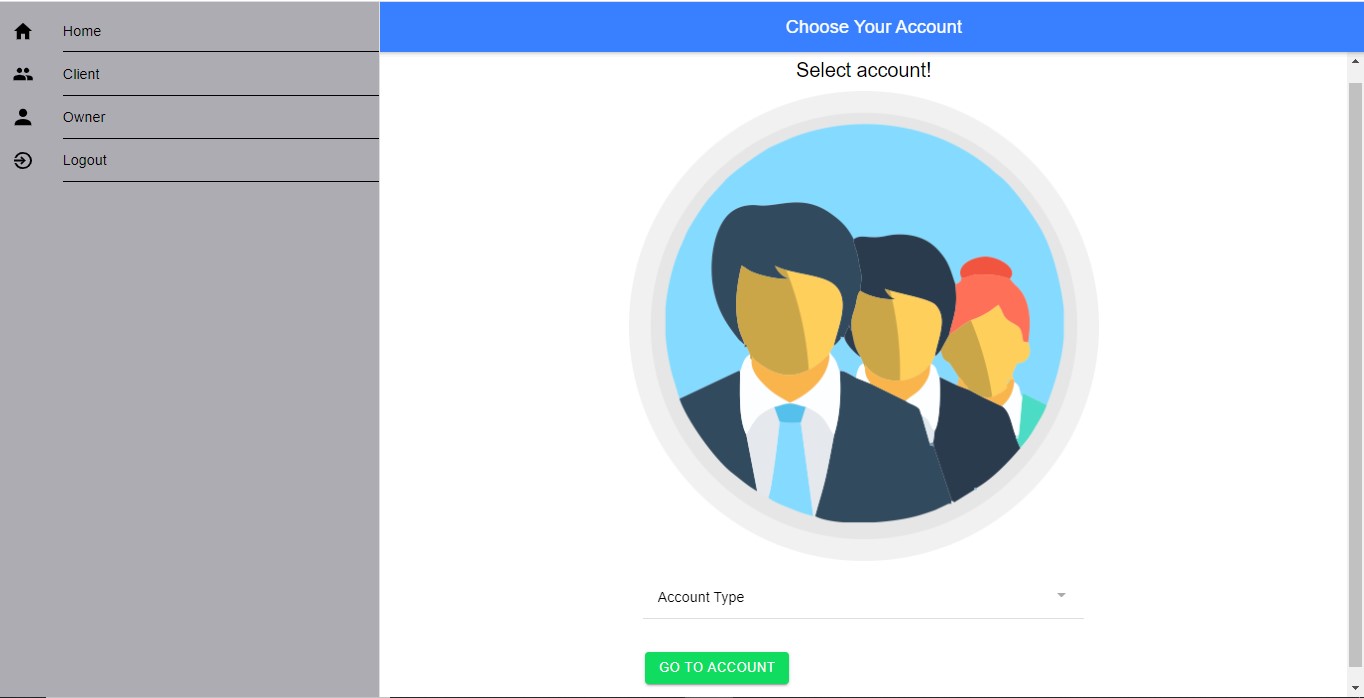


Owner Registration



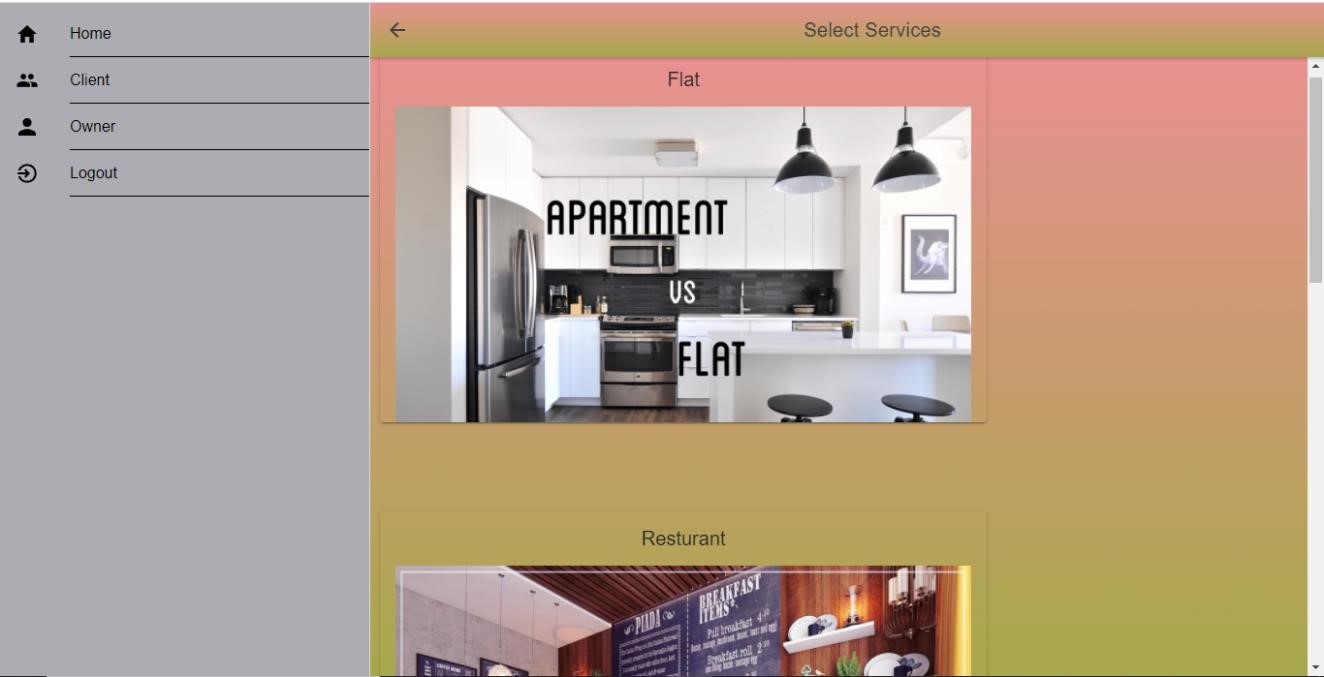
Client Registration

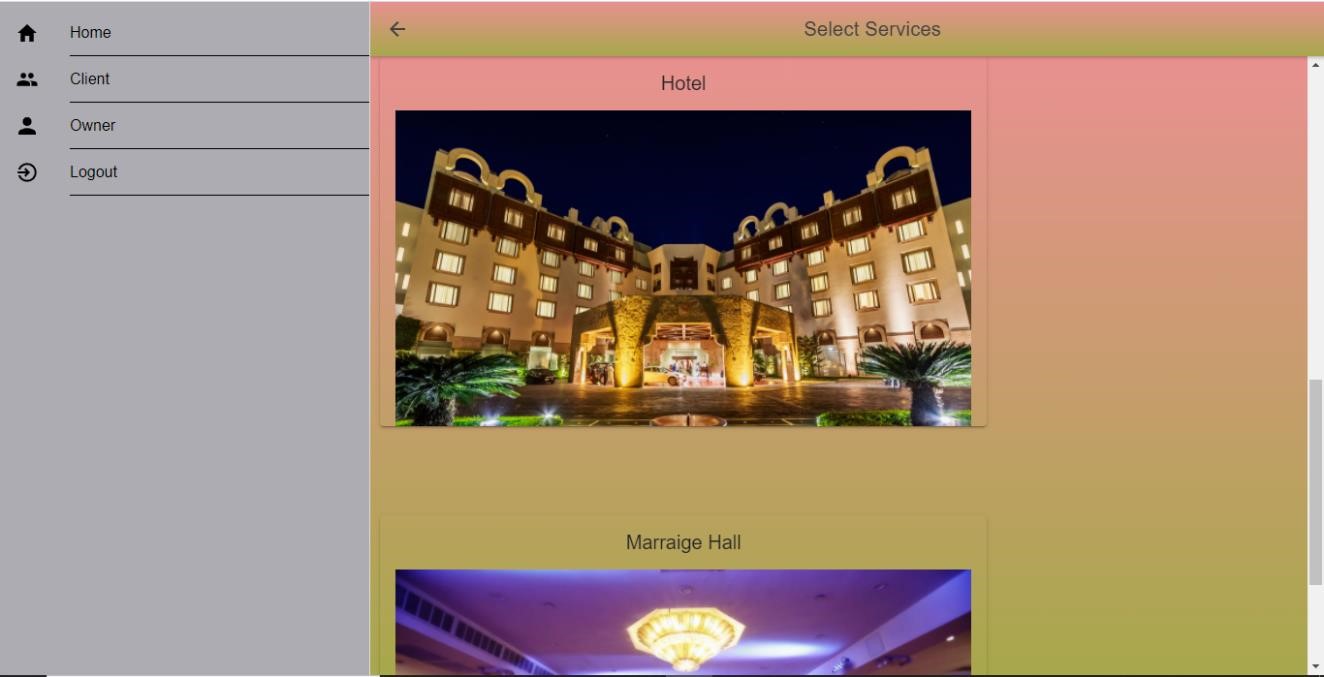
If the user is using the services of **Rehayash** then they can enter their credentials in the **login** page to access their accounts.



Once the user enters correct credentials, they will be redirected to the Service Profile with all of their information.

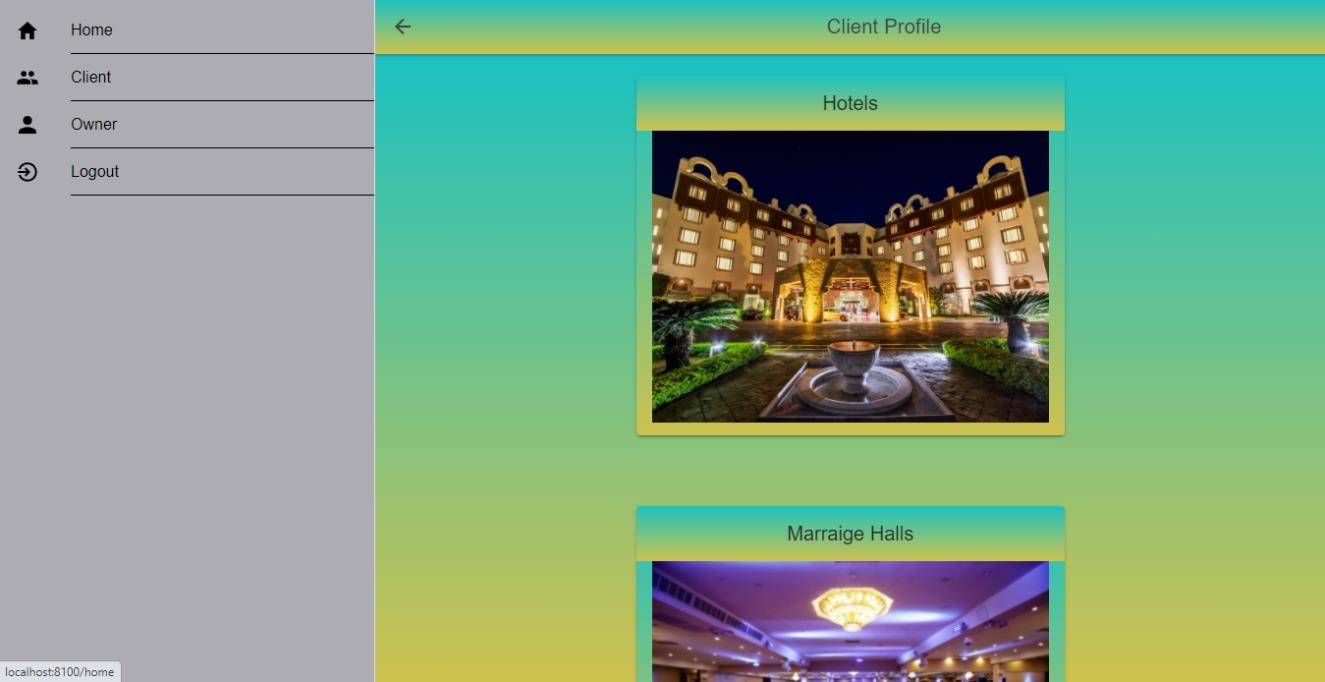
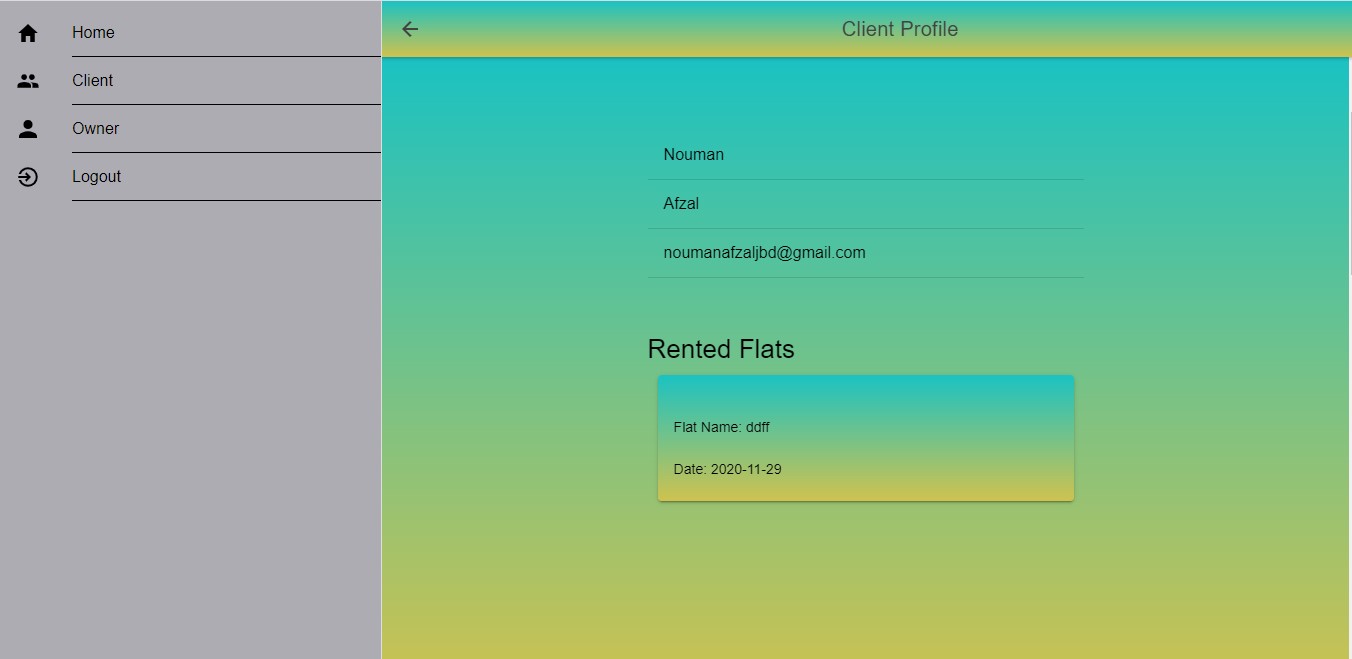
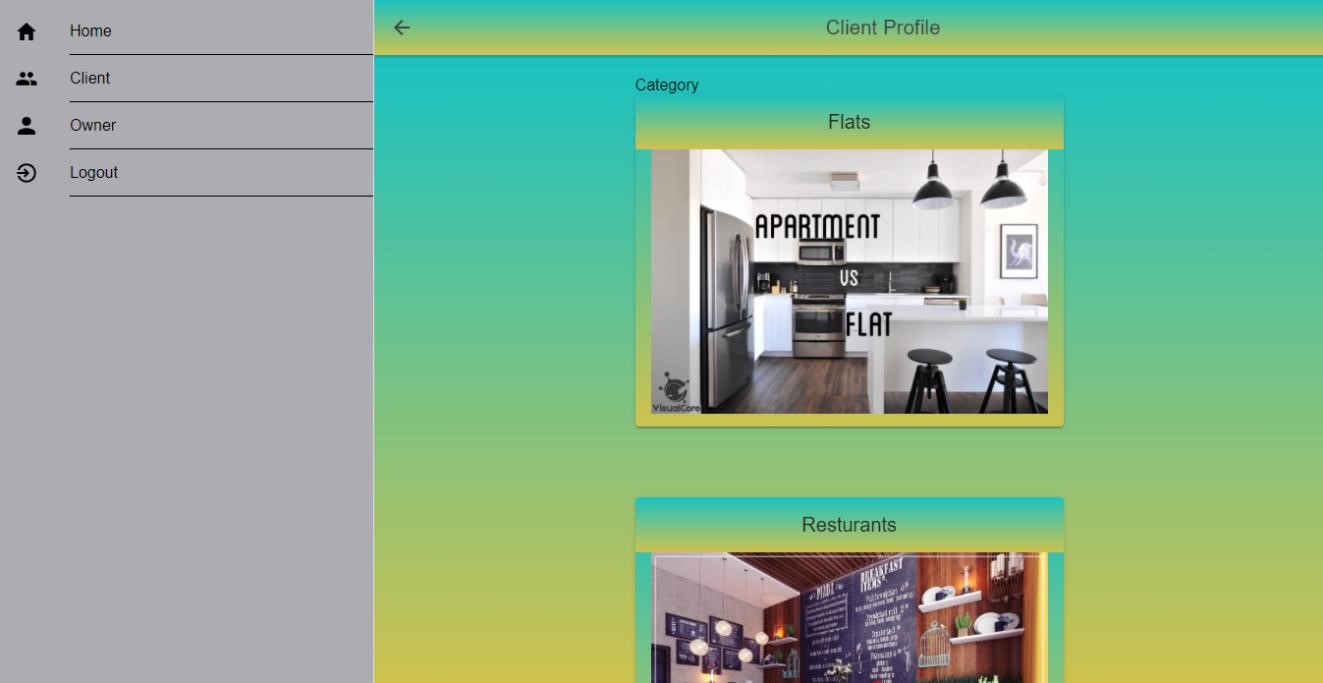
On the owner side all the services can be presented to consumers, are available here on this window. In this window the user can add Hotels, restaurants, flats/homes and marriage halls, just by selecting the required option of service.



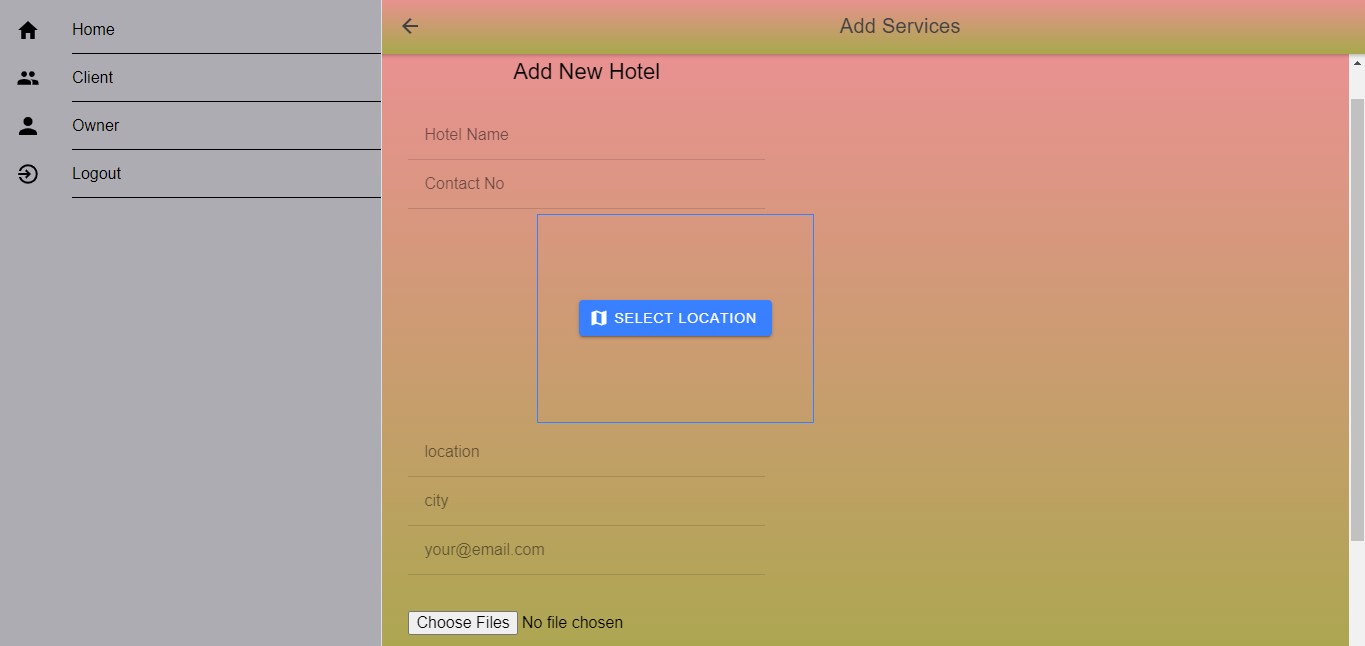


**Owner Profile Side**

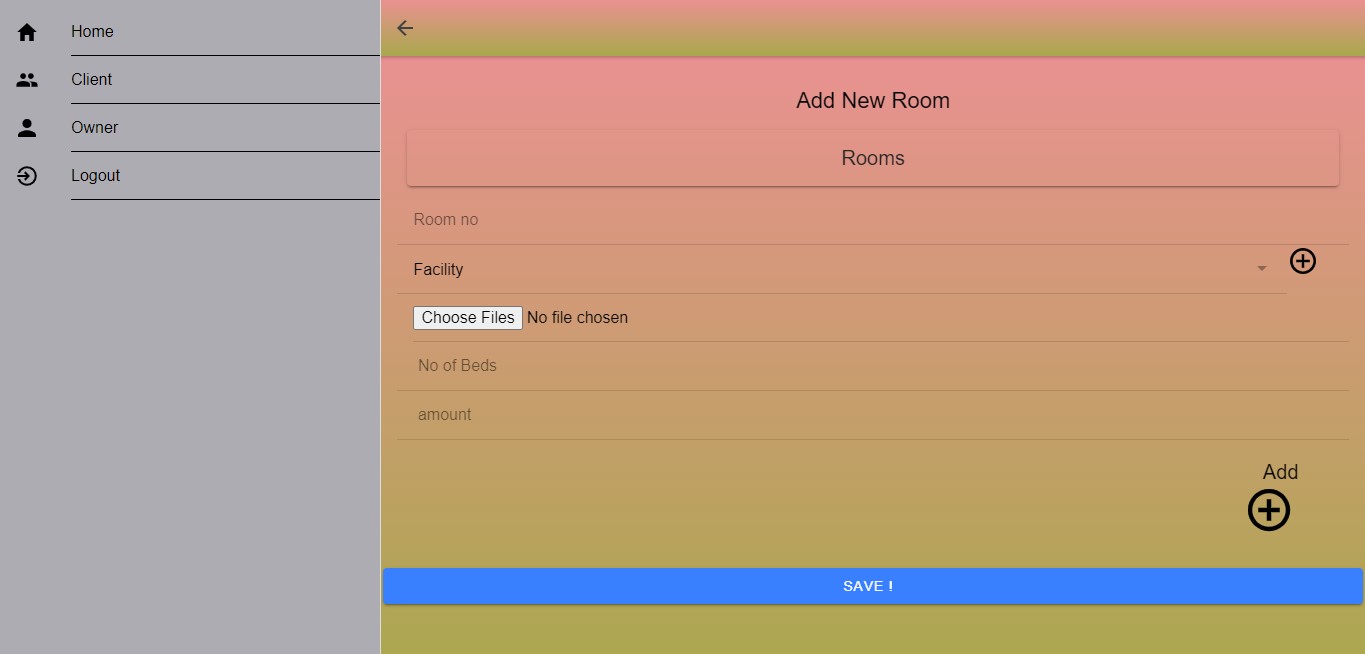
On the Client side of all the options and services a user can acquire, could be seen here. On this window the user can Book Hotel Room, reserve Restaurant, Rent a flat/home, and can also book marriage Hall. The user can also see what services He / She is using right now.



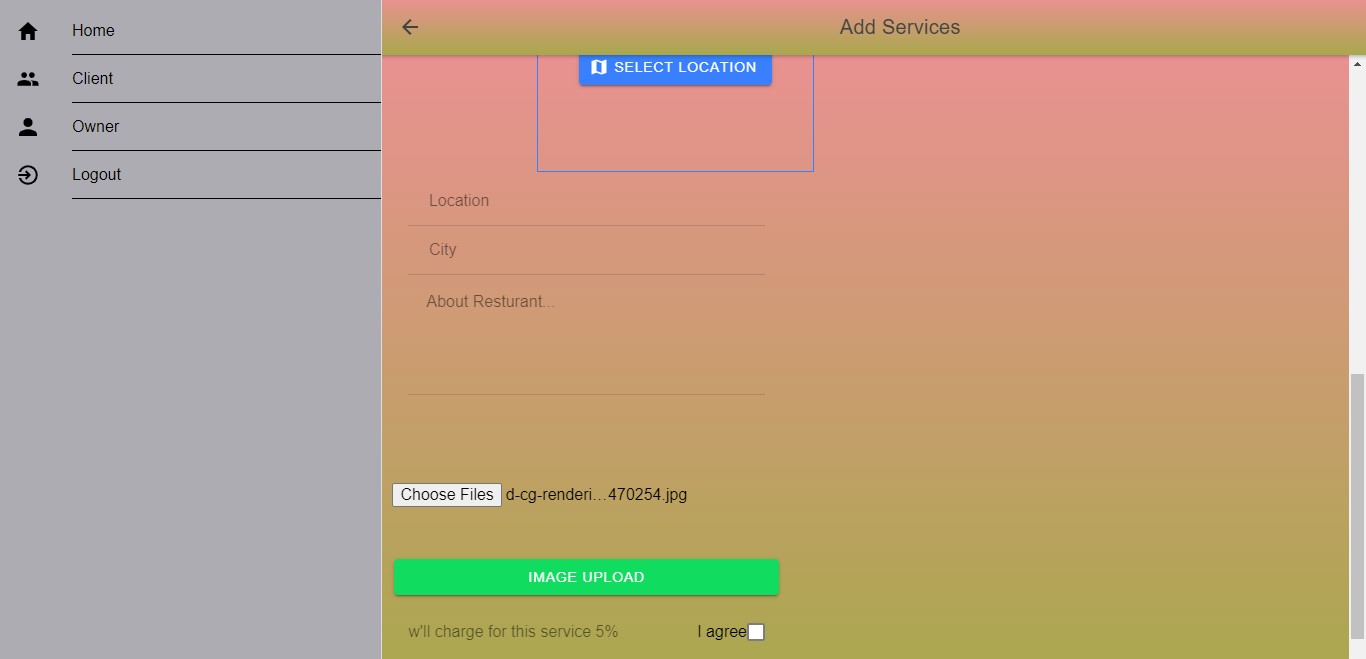
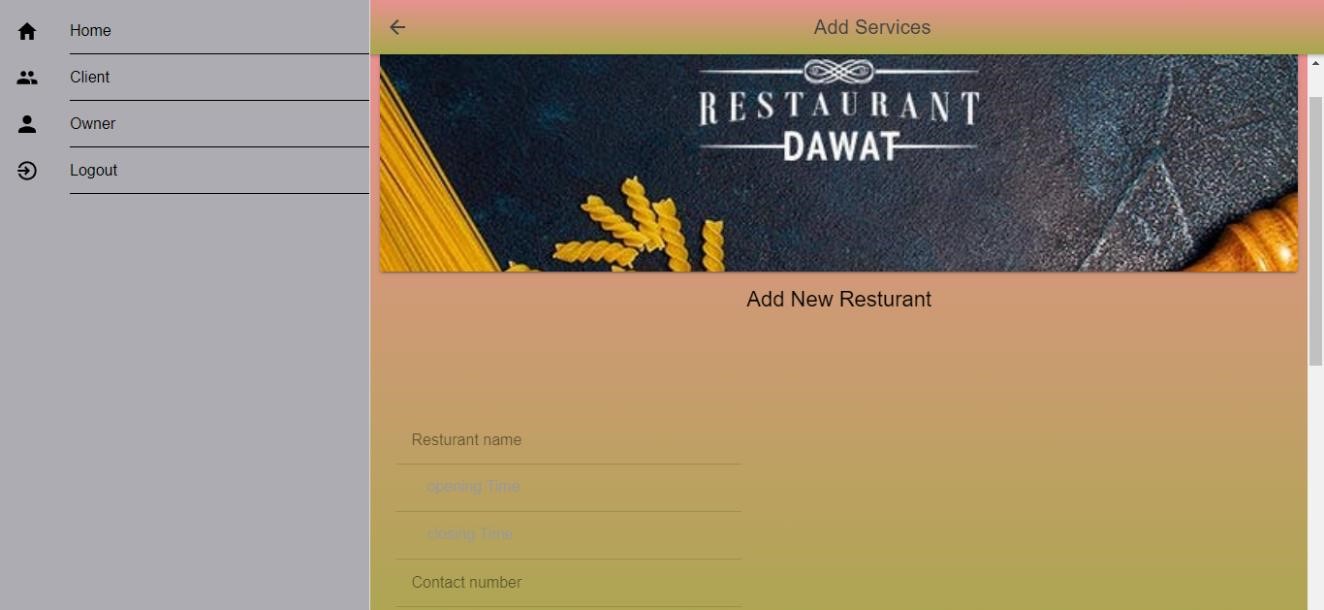
**Client Profile Side** On the owner side by selecting the Hotel option the user can add hotel rooms service to other users for booking. Here the user should provide all the required fields of data.



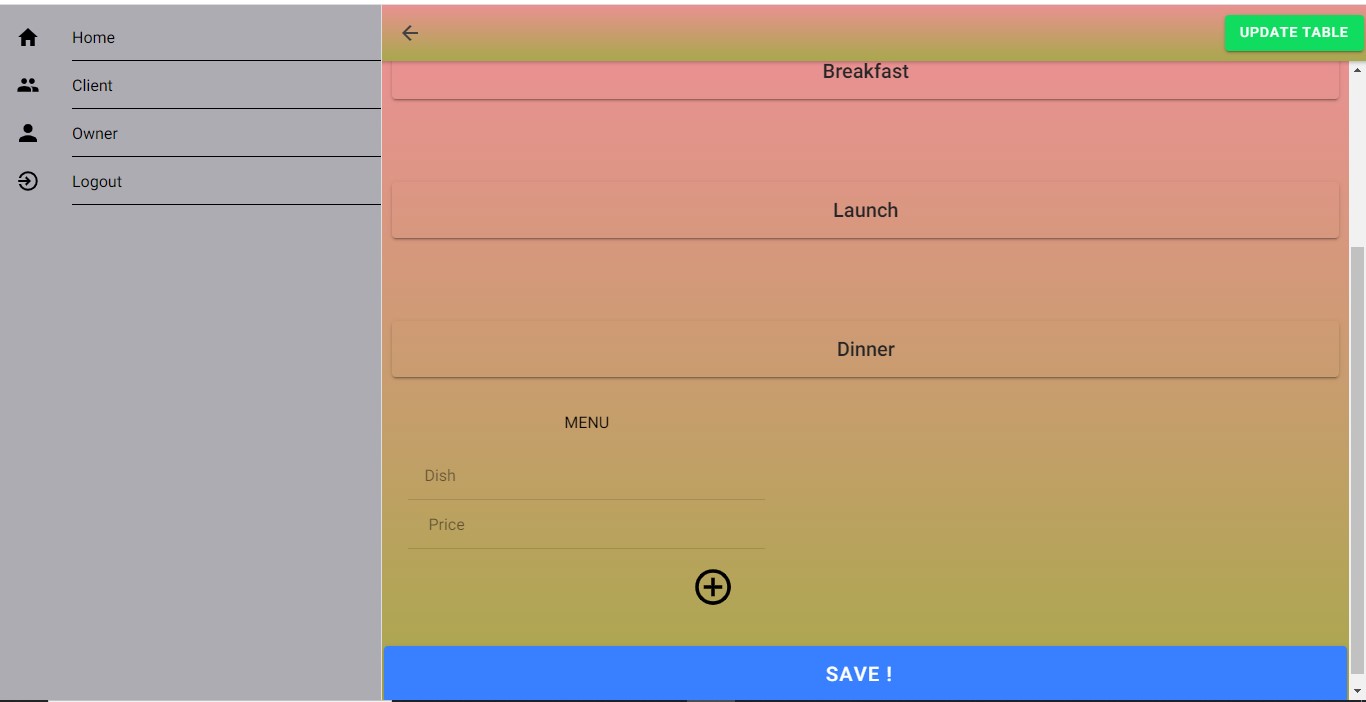
After entering all hotel data, then the user can add types and numbers of rooms to his hotel.



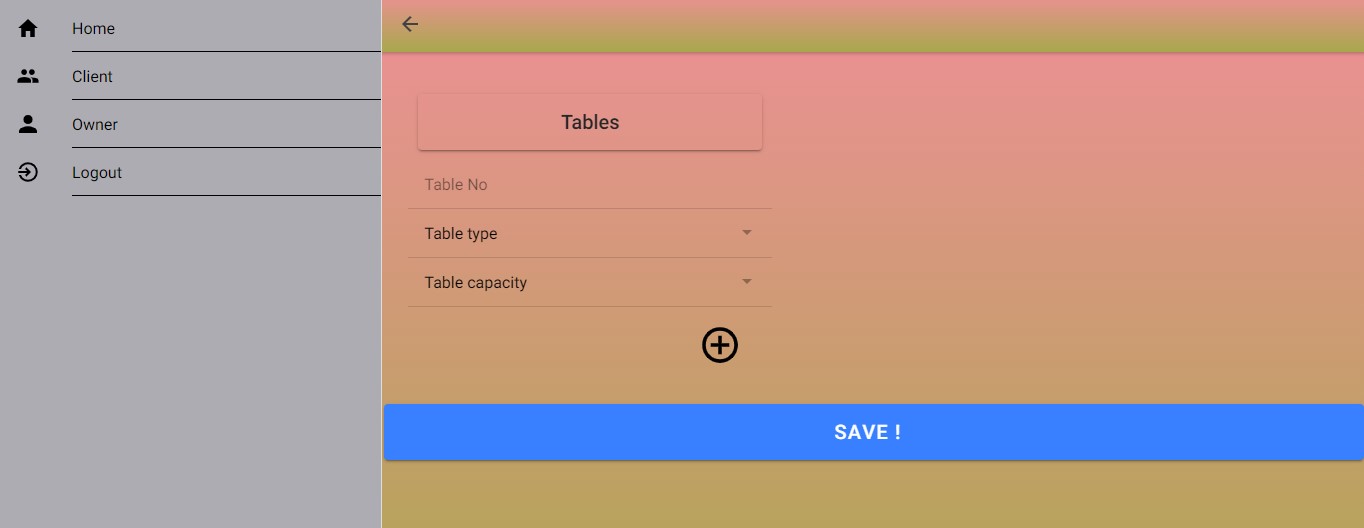
On the owner side by selecting the Restaurant option the user can add restaurant table services to other users for eating. Here the user should provide all the required fields of data for the restaurant.



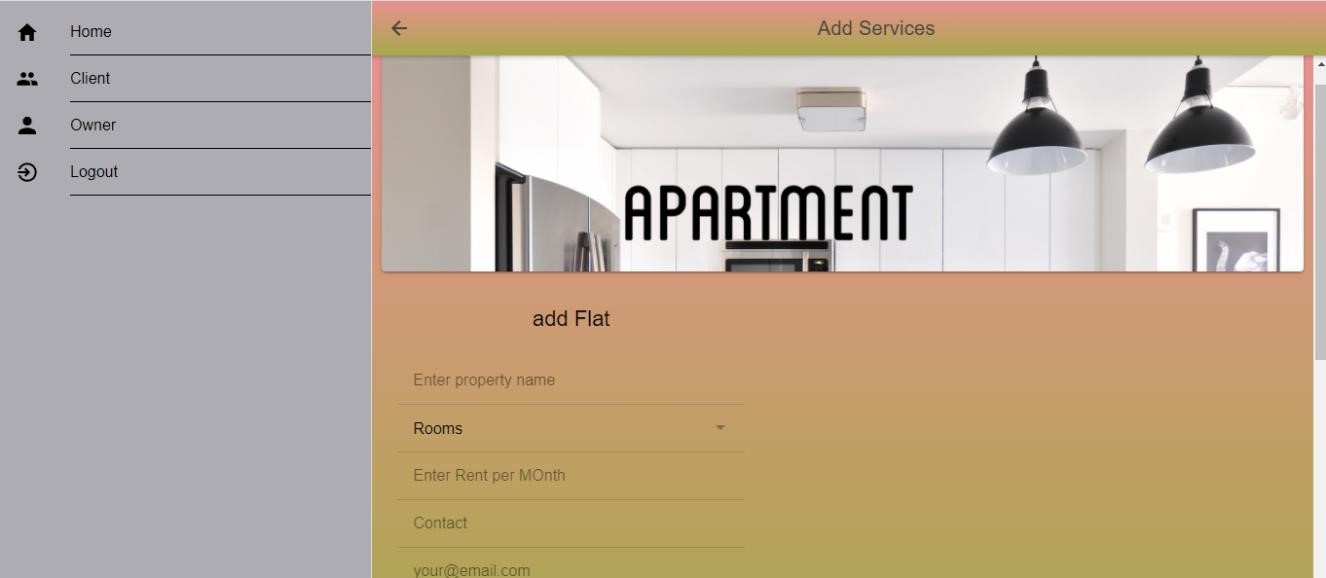
After entering all the data for the restaurant, then the user can add menu of every time and types and numbers of tables.

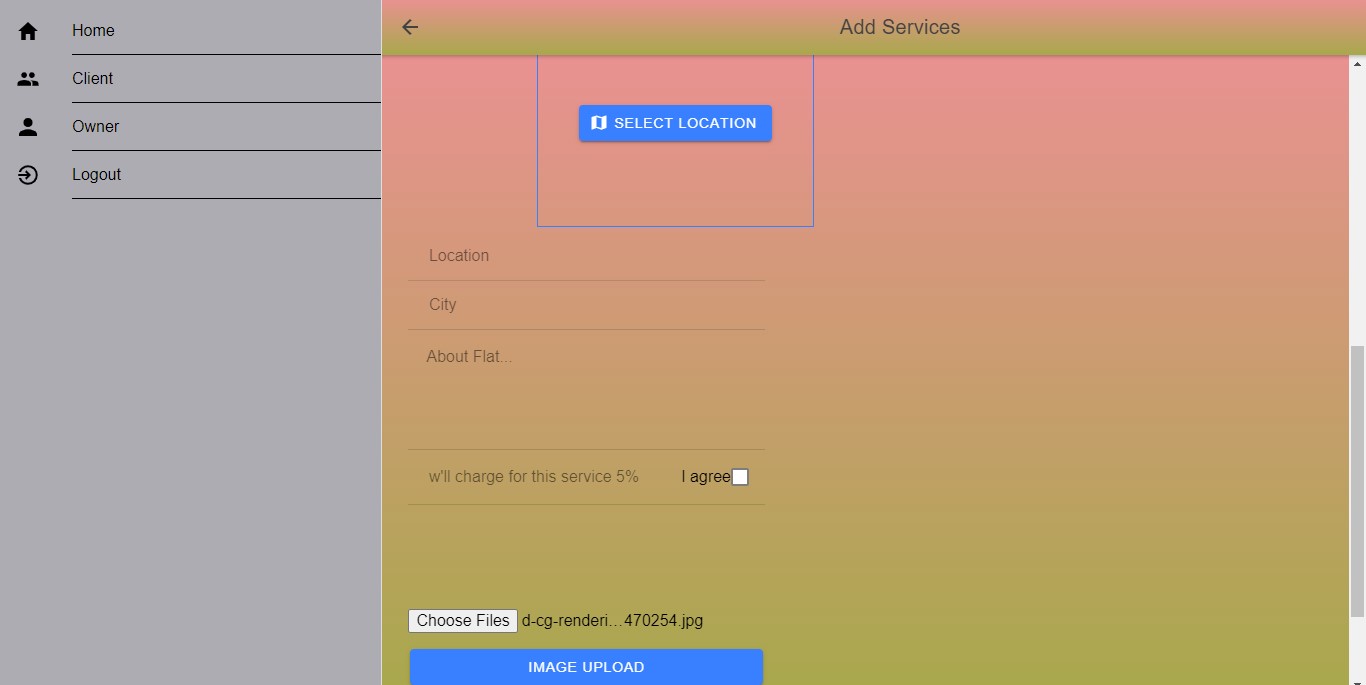


Here the user can register for types and numbers of tables in the restaurant.

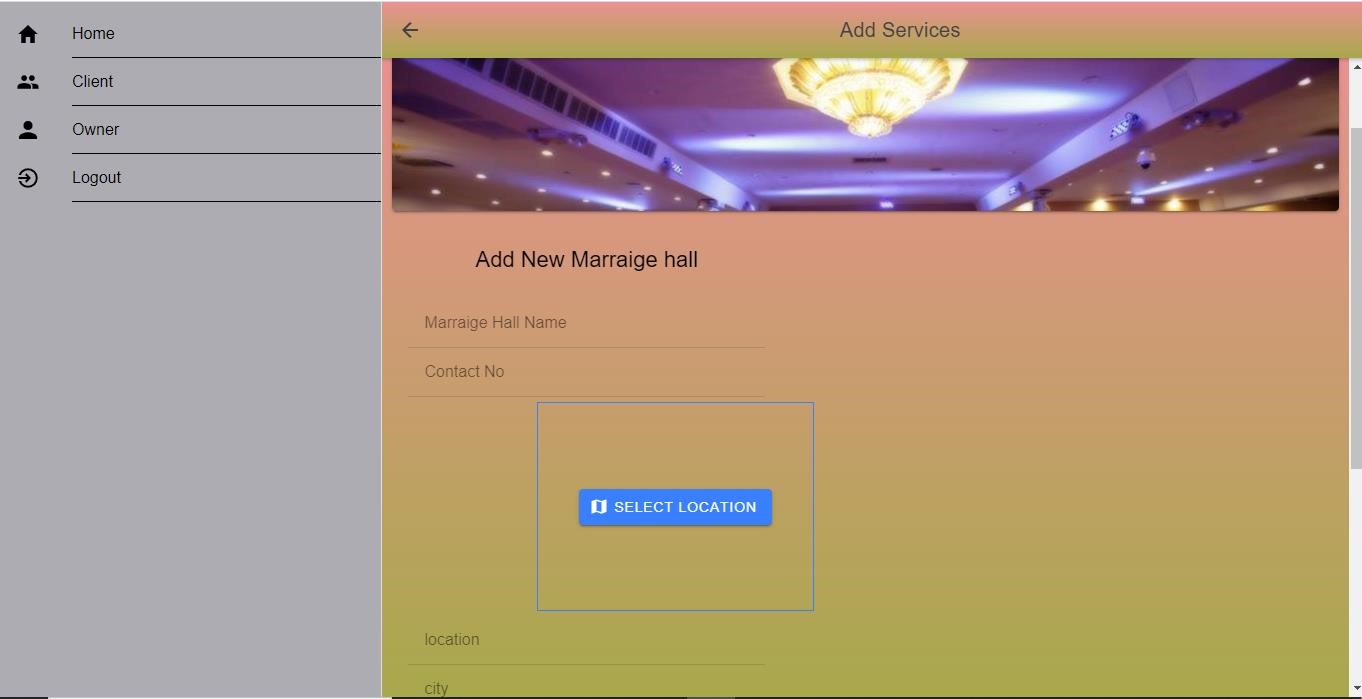


On the owner side the third service that a user can add is by selecting the Flat option, through this user can add flat or home for rent to other users. Here the user should provide all the required fields of data for the flat.

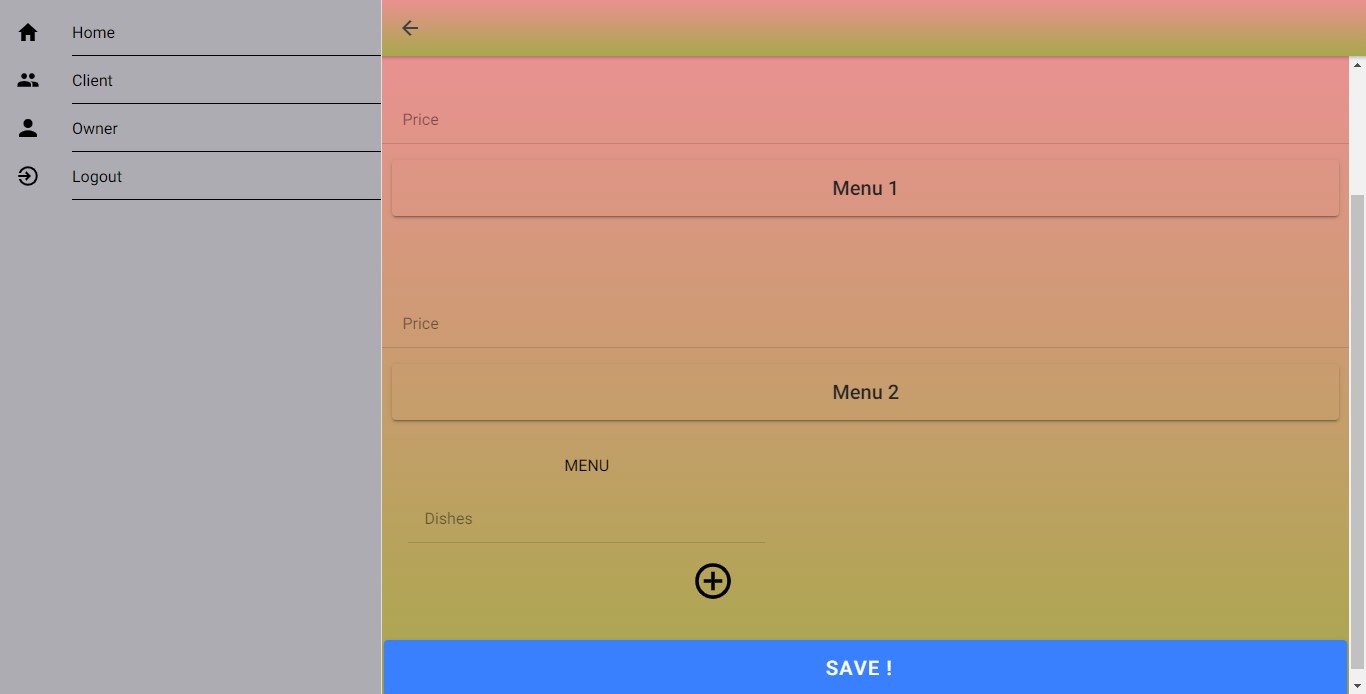




On the owner side the fourth service that a user can add is by selecting the Marriage Hall option, through this user can add Marriage Hall for booking to other users. Here the user should provide all the required fields of data for the Marriage Hall.

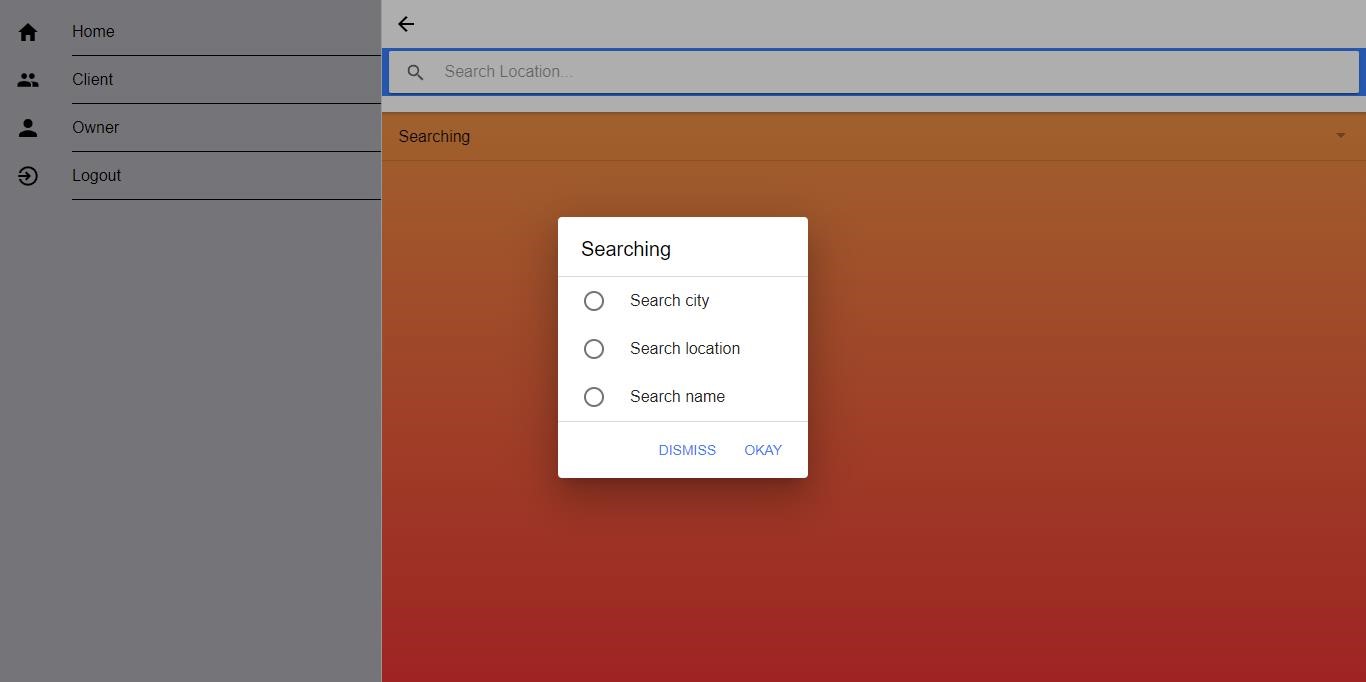


After entering all the data for the marriage hall, then the user can add menu of every type with prices and dishes.

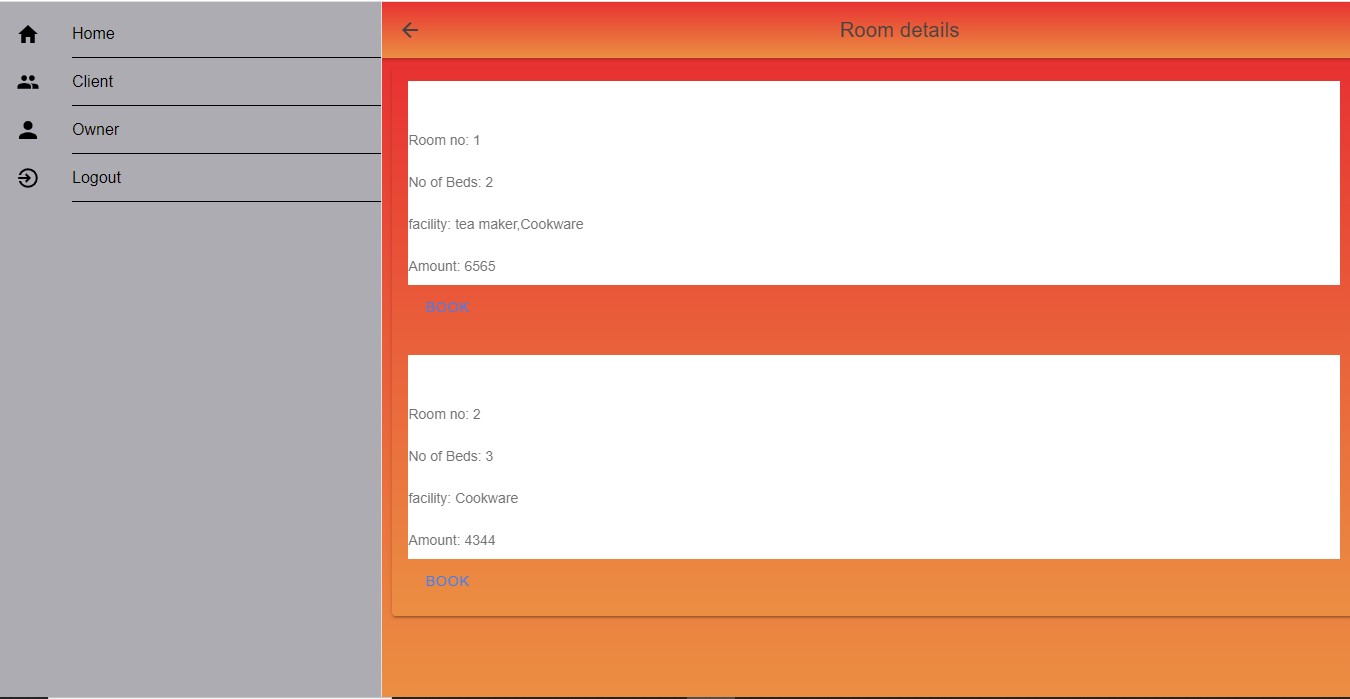


#### Client Side

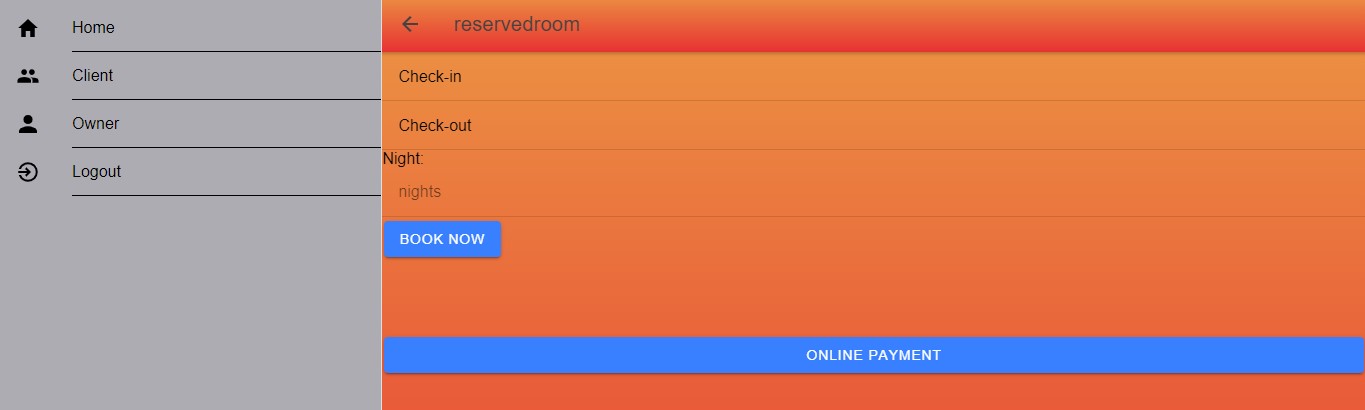
On the Client side the user can acquire the services. To acquire Hotel rooms the user should select the Hotel option from the Client profile. Search for Hotels and select Hotel:



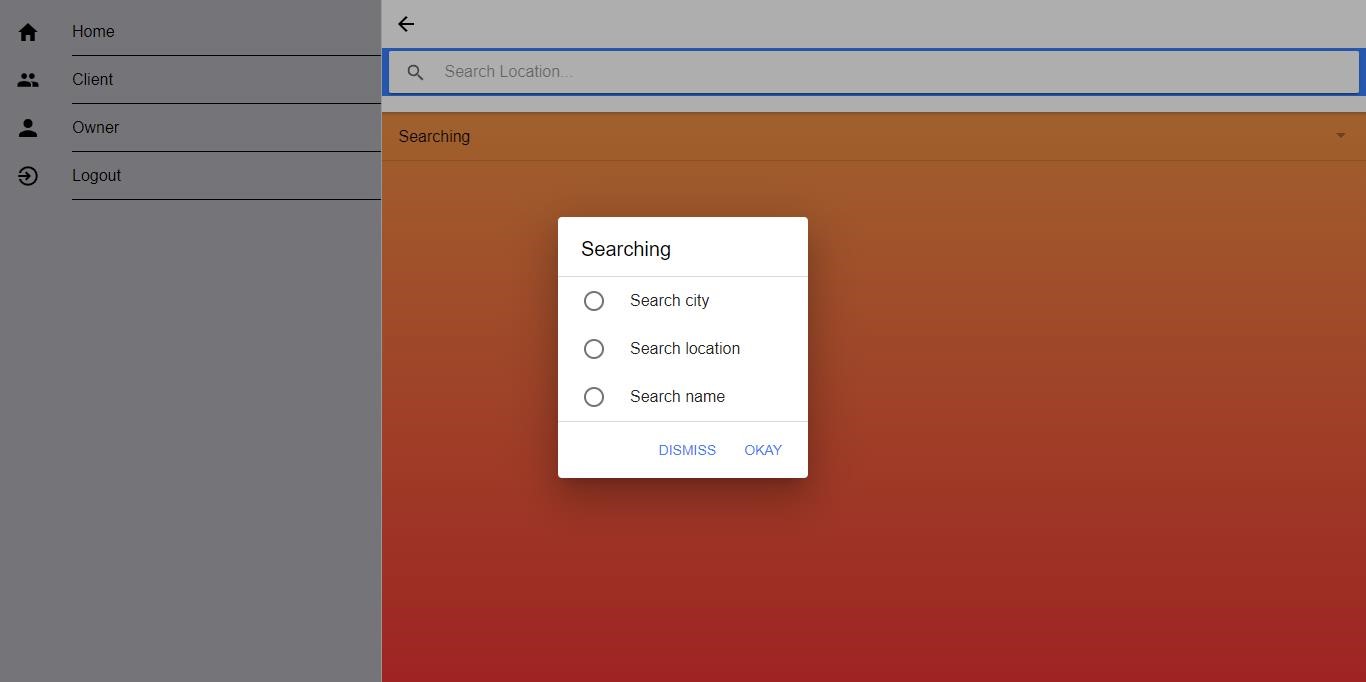
Check out rooms



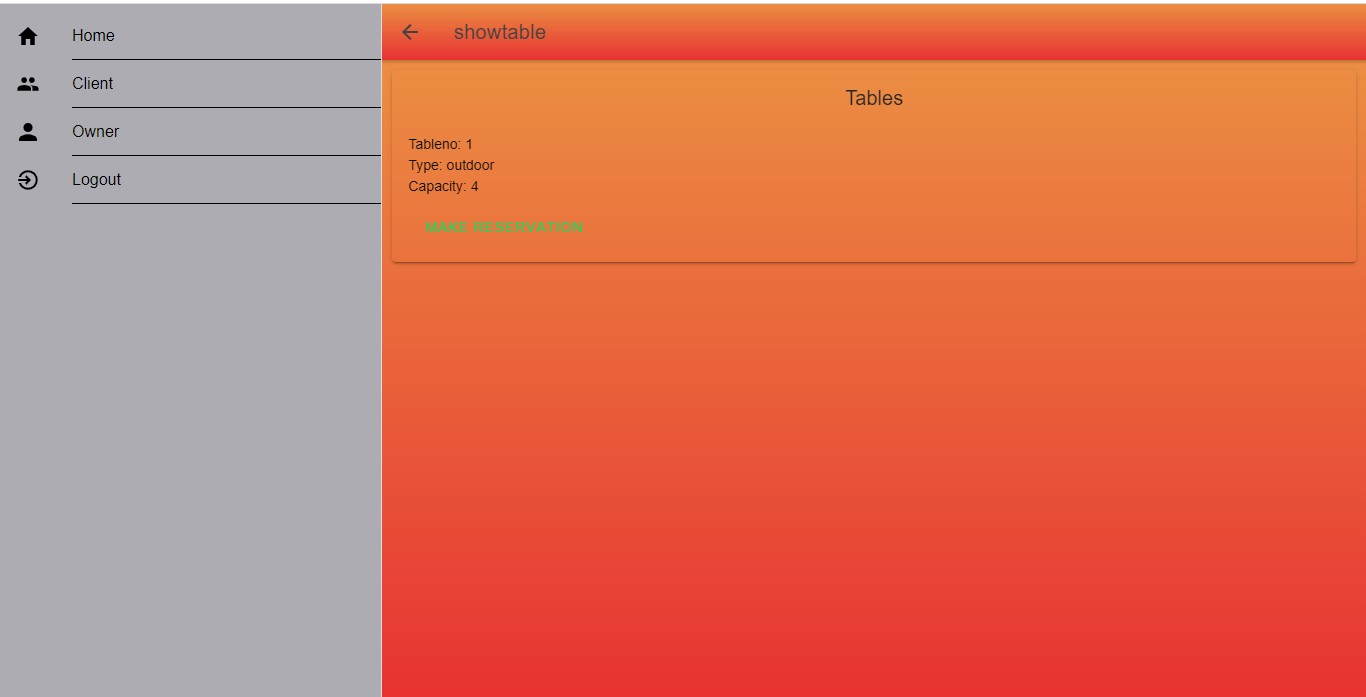
and Fill form for room booking



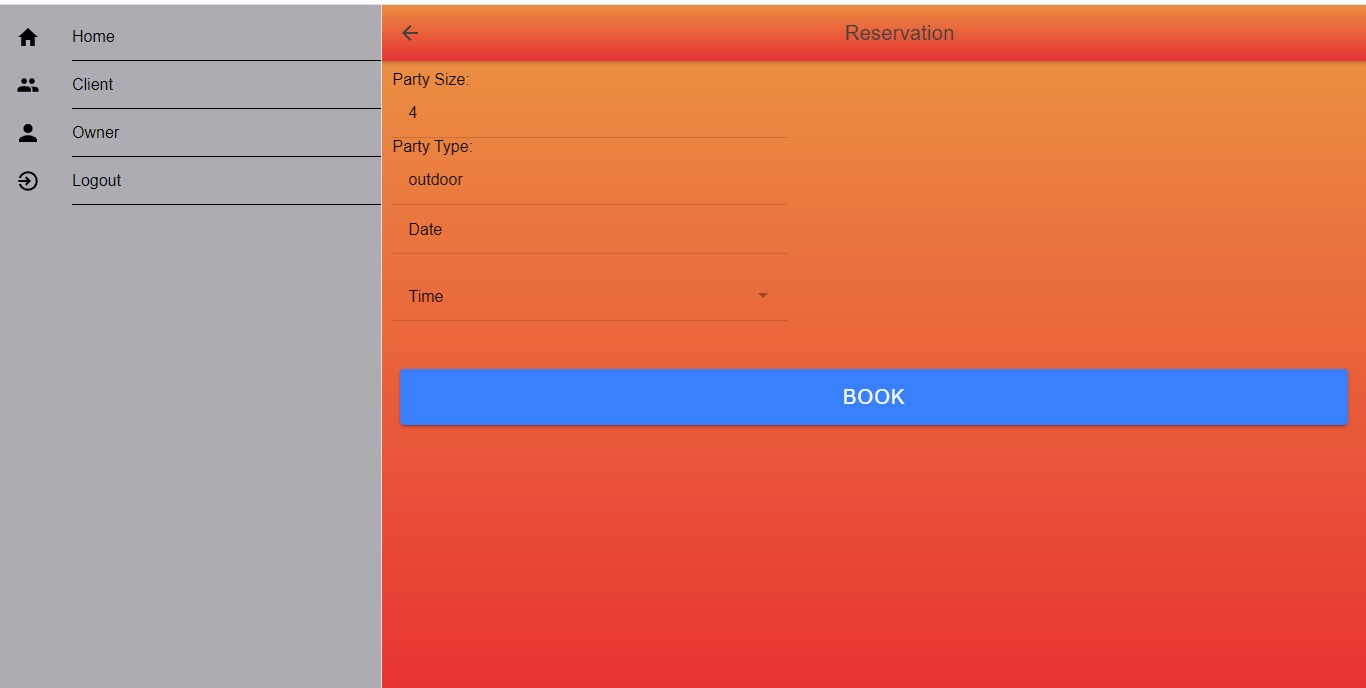
The second thing a user can do is to reserve table in the restaurant. To do so the user should select the Restaurant option from the Client profile. Search for Restaurants and select Restaurant:



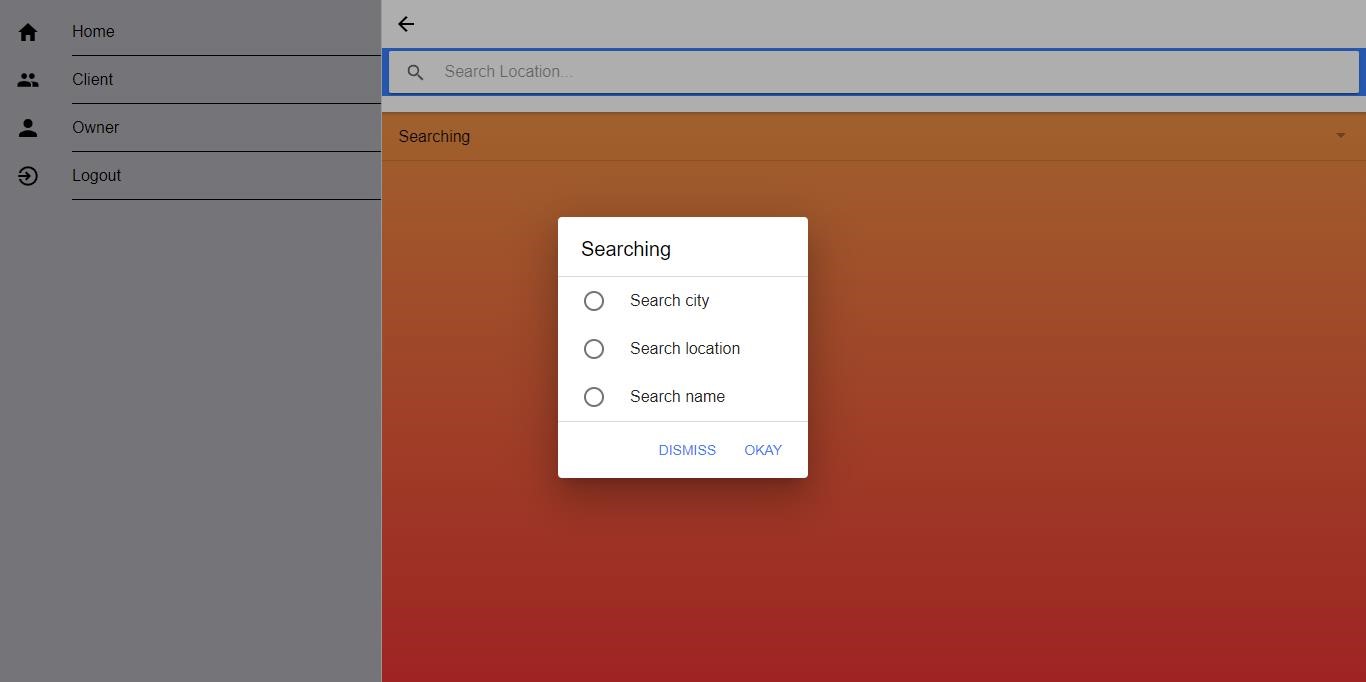
The check out the tables



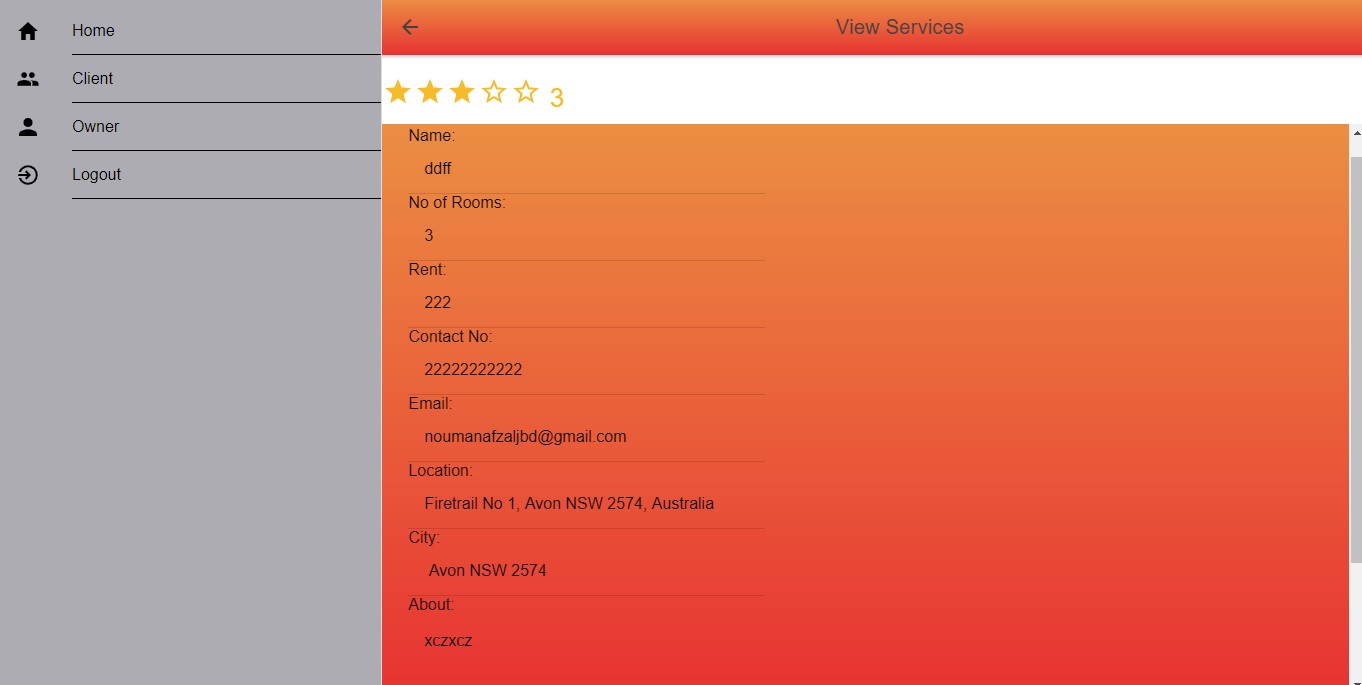
The user can reserve table by filling the form

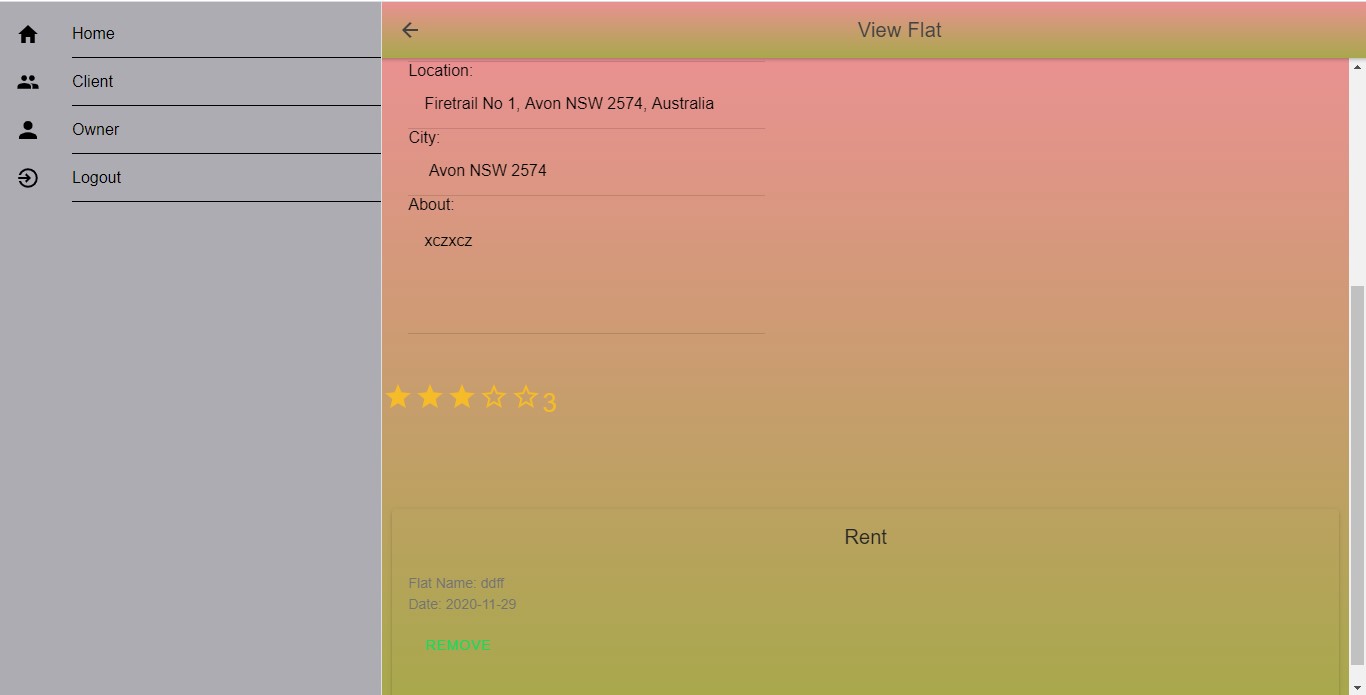


The next thing a user can do is to rent a flat. To do so the user should select the Flat option from the Client profile. Search for Flats and select Flat:



The user can view the profile of the flat:





**4.3.2 Mobile Application**

Chapter 06 Testing

# CHAPTER 6 TESTING

Chapter 06 Testing

## 4. TESTING

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

### 4.1 Test Case 01: Register Users

|  |  |
| --- | --- |
| **Test Case ID** | **TC 01** |
| **Test Item** | Register Users (owner and client) |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if the user can register properly or not. |
| **Action to Perform** | 1. Test the module by leaving fields empty and saving data. 2. Adding inappropriate data in the fields and saving. 3. Test if the email is sent with verification to the user or not. |
| **Pre-Condition** | Application is working |
| **Post-Condition** | System will display a success message and sends Email to the user with verification. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM |
| **Comments:** Module ran successfully. Email was received by the test user | |

Table 6. 1TC: Add User

### 4.2 Test Case 02: Add Hotel

|  |  |
| --- | --- |
| **Test Case ID** | **TC 02** |
| **Test Item** | Add new Hotel Room for Booking |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if module is adding new Hotel and saving data successfully or not. |

*HomeHaven 58 Chapter 06 Testing*

|  |  |
| --- | --- |
| **Action to Perform** | 1. Leave few fields empty and try to save data. 2. Add inappropriate data and try to save it. 3. Test if the email is sent with information to the user or not. |
| **Pre-Condition** | Application is working and the Owner has successfully logged in. |
| **Post-Condition** | Hotel Room data will be saved in DB and Owner will get success message. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM |
| **Comments:** Module ran successfully. Hotel was added to the DB with success message | |

Table 6. 2TC: Add Hotel

### 4.3 Test Case 03: Add Restaurant

|  |  |
| --- | --- |
| **Test Case ID TC 03** | |
| **Test Item** | Add new Hotel Room for Booking |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if module is adding new Restaurant and saving data successfully or not. |
| **Action to Perform** | 1. Leave few fields empty and try to save data. 2. Add inappropriate data and try to save it. 3. Test if the email is sent with information to the user or not. |
| **Pre-Condition** | Owner User is logged in to the System |
| **Post-Condition** | Restaurant data will be saved in DB and Owner will get success message. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM / Android 9.0, 6Gb RAM |
| **Comments:** Module ran successfully. Restaurant was added to the DB with success message. | |

Table 6. 3TC: Add Restaurant

59

*Chapter 06 Testing*

### 4.4 Test Case 04: Add Flat

|  |  |
| --- | --- |
| **Test Case ID** | **TC 04** |
| **Test Item** | Add Flat |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if module is adding new Flat and saving data successfully or not. |
| **Action to Perform** | 1. Leave few fields empty and try to save data. 2. Add inappropriate data and try to save it. 3. Test if the email is sent with information to the user or not. |
| **Pre-Condition** | Owner User is logged in into the system. |
| **Post-Condition** | Flat data will be saved in DB and Owner will get success message. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM / Android 9.0, 6Gb RAM |
| **Comments:** Module run successfully. Restaurant was added to the DB with success message. | |

Table 6.4 TC: Add Flat

### 4.5 Test Case 05: Add Marriage Hall

|  |  |
| --- | --- |
| **Test Case ID** | **TC 05** |
| **Test Item** | Add Marriage Hall |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if module is adding new Marriage Hall and saving data successfully or not. |
| **Action to Perform** | 1. Leave few fields empty and try to save data. 2. Add inappropriate data and try to save it. 3. Test if the email is sent with information to the user or not. |
| **Pre-Condition** | Owner User is logged in into the system. |
| **Post-Condition** | Marriage Hall data will be saved in DB and Owner will get success message. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM / Android 9.0, 6Gb RAM |
| **Comments:** Module run successfully. Marriage Hall was added to the DB with success message. | |

Table 6. 5TC: Add Marriage Hall

60

Chapter 06 Testing

### 4.6 Test Case 06: Book Hotel Room

|  |  |
| --- | --- |
| **Test Case ID** | **TC 06** |
| **Test Item** | Book Hotel Room |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if user can Book Hotel Room properly and saving data successfully or not. |
| **Action to Perform** | 1. Leave few fields empty and try to save data. 2. Add inappropriate data and try to save it. 3. Test if the email is sent with information to the user or not. |
| **Pre-Condition** | Client User is logged in into the system. |
| **Post-Condition** | Hotel room will be booked, and System will display success message. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM / Android 9.0, 6Gb RAM |
| **Comments:** Module ran successfully. Hotel Room was booked with success message. | |

Table 6. 6TC: Book Hotel Room

### 4.7 Test Case 07: Reserve Restaurant

|  |  |
| --- | --- |
| **Test Case ID** | **TC 07** |
| **Test Item** | Reserve Restaurant |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if user can reserve a table in Restaurant properly and saving data successfully or not. |
| **Action to Perform** | 1. Leave few fields empty and try to save data. 2. Add inappropriate data and try to save it. 3. Test if the email is sent with information to the user or not. |
| **Pre-Condition** | Client User is logged in into the system. |
| **Post-Condition** | Restaurant table will be reserved, and System will display success message. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM / Android 9.0, 6Gb RAM |
| **Comments:** Module ran successfully. Restaurant table was reserved with success message. | |

Table 6.7 TC: Reserve Restaurant

*HomeHaven 61*

Chapter 06 Testing

### 4.8 Test Case 08: Rent Flat

|  |  |
| --- | --- |
| **Test Case ID** | **TC 08** |
| **Test Item** | Rent a Flat |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if the User can rent a flat properly and saving data successfully or not. |
| **Action to Perform** | 1. Leave few fields empty and try to save data. 2. Add inappropriate data and try to save it. 3. Test if the email is sent with information to the user or not. |
| **Pre-Condition** | Client User is logged in into the system. |
| **Post-Condition** | Flat will be rented, and System will display success message. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM / Android 9.0, 6Gb RAM |
| **Comments:** Module ran successfully. Flat was rented with success message. | |

Table 6. 8TC: Rent Flat

### 4.9 Test Case 09: Book Marriage Hall

|  |  |
| --- | --- |
| **Test Case ID TC 09** | |
| **Test Item** | Marriage Hall |
| **Test Case Engineer** | Matiullah |
| **Purpose** | To check if the User can book marriage hall properly and saving data successfully or not. |
| **Action to Perform** | 1. Leave few fields empty and try to save data. 2. Add inappropriate data and try to save it. 3. Test if the email is sent with information to the user or not. |
| **Pre-Condition** | Client User is logged in into the system. |
| **Post-Condition** | Marriage Hall will be booked, and system will display success message. |
| **Environment** | Windows 10, 64 bit, 8Gb RAM / Android 9.0, 6Gb RAM |
| **Comments:** Module ran successfully. Marriage Hall was booked with success message. | |

Table 6. 9TC: Book Marriage Hall

*HomeHaven 62*

**CHAPTER 7**

**CONCLUSION**

*References Appendix*

## 5. CONCLUSION

By use of Rehayash application the user can search and finds hotels, Restaurants, flats, Marriage halls as per location, Best Prices, Best Rating and so on Provide easy access to rent every type of flats & houses, hotels room, marriage halls and reserve table in the specified areas based on user requirements. Enable hotels and flat owners to present their services on the ease. Provide Comfortability & Easy Access to Staying services.

Over all the idea is, just to bring ease and simplicity for people to find services like Hotels, Restaurants, flats , homes and marriage halls near them, and to present accommodation services like Hotels, Restaurants, flats , homes and marriage halls easily to other people.

By doing so all (the owner side and client side) both will feel ease for finding and presenting the accommodation services easily. Rehayash will keep improving its features to keep up with the expectations of its clients.

### 5.1 FUTURE EXTENSION

Moving further Rehayash is planning to improve its current features as well as add new features that will meet all the expectations of its clients. Here are some planned improvements that will be part of Rehayash in near future.

• **Add Hostels:** Rehayash is planning to add Hostels service module where user will be able to add hostels and other users can find it for rent.

*Chapter 06 Testing*

## REFERENCES

65