A PROJECT TITLED

FAROSHGAH

By

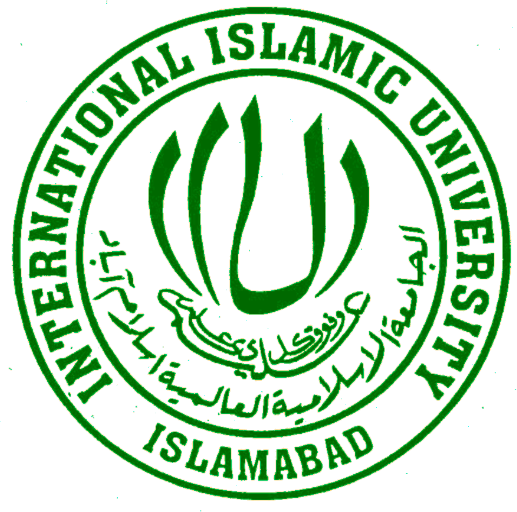
KALEEM ULLAH 3329-FBAS/BSSE/F16 USMAN SERVER 3342-FBAS/BSSE/F16

IN PARTIAL FULLFILLMENT OF

**BS IN SOFTWARE ENGINEERING**

DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

FACULTY OF BASIC AND APPLIED SCIENCES

**INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD (2021)**

## FINAL APPROVAL

**Dated:** It is certified that we have read the project report titled “Faroshgah” 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

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

*Dissertation*

## 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 from any other degree or qualification of this or any other University or institute of learning. We further declare that this software 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

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 special thanks of gratitude to our respected supervisor **Mr. Muhammad Nadeem** who gave us the guidance and support in completing this wonderful project Faroshgah, which also helped us in doing a lot of Research and we came to know about so many new things. We are 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.

|  |  |
| --- | --- |
|  | **BROJECT 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

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.

The aim of this project is to develop software for students who come here in large cities to get higher education but they feel it difficult to find good hostels online. This application aims to help students in finding hostels, getting jobs and shopping via internet using mobile application. This reflects the forecast availability of hostels via internet for students and also provides opportunity to get jobs online.

*Table of Contents*

Table of Contents

[FINAL APPROVAL 1](#_Toc75623375)

[COMMITTEE 1](#_Toc75623376)

[Dissertation 3](#_Toc75623377)

[DECLARATION 3](#_Toc75623378)

[ACKNOWLEDGEMENT 4](#_Toc75623379)

[ABSTRACT 6](#_Toc75623380)

[CHAPTER 1 INTRODUCTION 8](#_Toc75623381)

[1. INTRODUCTION 9](#_Toc75623382)

[PURPOSE 9](#_Toc75623383)

[SCOPE 9](#_Toc75623384)

[OVERVIEW 10](#_Toc75623385)

[POSITIONING 10](#_Toc75623386)

[PRODUCT FEATURES 12](#_Toc75623387)

[ADVANTAGES 12](#_Toc75623388)

[CONSTRAINTS 12](#_Toc75623389)

[CH#2 SYSTEM ANALYSIS 14](#_Toc75623390)

[2. SYSTEM ANALYSIS 14](#_Toc75623391)

[Use Case Diagram 15](#_Toc75623392)

[Table 2.10 Use Case Delete Jobs by Provider 33](#_Toc75623393)

[CHAPTER 3 SYSTEM DESIGN 34](#_Toc75623394)

[System Sequence Diagram 34](#_Toc75623395)

[CLASS DIAGRAM 40](#_Toc75623396)

[3.4 Deployment Diagram 42](#_Toc75623397)

[CHAPTER 4 IMPLEMENTATION 44](#_Toc75623398)

[Functional Features of the Model: 44](#_Toc75623399)

[TOOLS AND TECHNIQUES 44](#_Toc75623400)

[CHAPTER 5 65](#_Toc75623401)

[5. System Testing 65](#_Toc75623402)

[Black Box Testing 65](#_Toc75623403)

[White Box Testing 65](#_Toc75623404)

[Test Cases 65](#_Toc75623405)

[CHAPTER 6 CONCLUSION 74](#_Toc75623406)

[5. CONCLUSION 74](#_Toc75623407)

[FUTURE EXTENSION 75](#_Toc75623408)

[REFERENCES 75](#_Toc75623409)

# CHAPTER 01

# INTRODUCTION

## 

## 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 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.

### 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;

### SCOPE

Enabling students/jobholders to connect with the digital world. Providing the facility of using the internet in a useful manner from finding hostels online to book hostel online and from finding shops online to buying things from the shops via internet. The job module facilitates the students and individuals to get a job online.

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 and booking of rooms in the hostel.

### 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 lot of new technologies covered in developing this project.

### POSITIONING

#### **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.

#### **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. |
| **Successful solution would be** | Easy access to hostels with best prices and location. Reduce finding time and booking time. Get your desired job and shop and get it. |

#### **Product Position Statement**

|  |  |
| --- | --- |
| For | Owners & Customers |
| Who | Wants to present and acquire accommodation services |
| 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. |
|  |  |
|  | . |

### PRODUCT FEATURES

##### All in One

There are three different services like hostels, shops, and jobs in one platform. Users can benefit from these three different services by using our mobile app.

Owner can add the service to other users like students and job holders. Service can be like Hotel searching, reserving seats and providing jobs and shops online.

##### Add up accommodation service

Owner can add the service to other users like students and job holders. Service can be like Hotel searching, reserving seats and providing jobs and shops online.

##### Acquire accommodation service

On the Client side, the user can acquire searching, reserving seat and finding jobs and shops.

##### Search for the Service

The user can search for the services by different keywords like by address, city, name etc.

##### 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 par directly to the owner.

##### Notifications

Owners get notification when hostel seat booked by someone. Similarly when someone purchases product from shop, shop owner gets notification.

##### Reviews

Users can give review to the hostels and shops of their choice. In this way users can guess quality of the service provided.

### ADVANTAGES

* + - Users will easily add and register their accommodation service to App, regarding the user can find it useful after registering.
    - Users will get the required service through the App easily and can book it online, regarding the category of requirement like hostel, seats and can book a seat for them.
    - Users will be able to search and check nearby Hotels, Homes/flats, Marriage halls, hostels, compare them, and choose the appropriate one for self.
    - User doesn’t need to spend money on advertisements and client doesn’t need to spend money to find accommodation service.

### CONSTRAINTS

Users both owner and client, should sign up before registering or acquiring any service. It is role based application. There are three types of roles, admin, publisher and user. Users with role publisher and admin can create hostel, shop or post a job. Users having role user can book room, apply for job, searching, purchase product and can give a review. Admin have fully fledged authority to do anything.

# CHAPTER 02

# SYSTEM ANALYSIS

## SYSTEM ANALYSIS

To identify goals and purposes of the system we use system analysis process. System analysis helps 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 systems and want to develop new system. It helps us to identify and to understand the requirements and domain of the new system.

Following are three different techniques to do system analysis for our current system that is being developed

1. Data Flow Diagrams
2. Use Case Model
3. Use Case Description
4. Use Case Diagrams

### DATA FLOW DIAGRAMS

### A way of representing the flow of data through system is data flow diagram. It helps us to find the correct flow of data from inputs to outputs of the system.

#### LOGIN AND REGISTER

### login flow.png

Figure - Login And Registration Flow Diagram

### CREATE HOSTEL

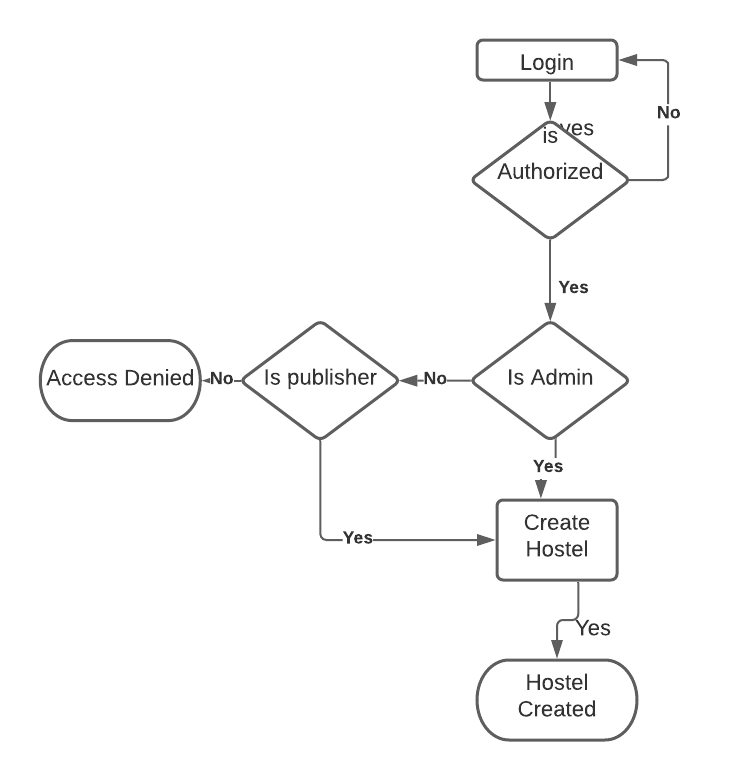


Figure - Create Hostel Flow Diagram

### BOOK SEAT

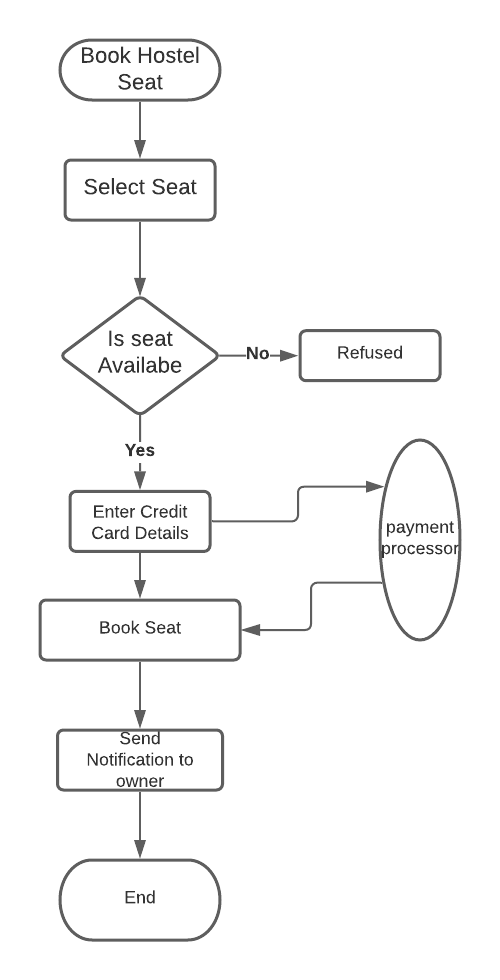


Figure - Book Seat Flow Diagram

### 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)

##### Hostel Use Case Diagram



Figure - Hostel Use Case Diagram

Shops Use Case Diagram

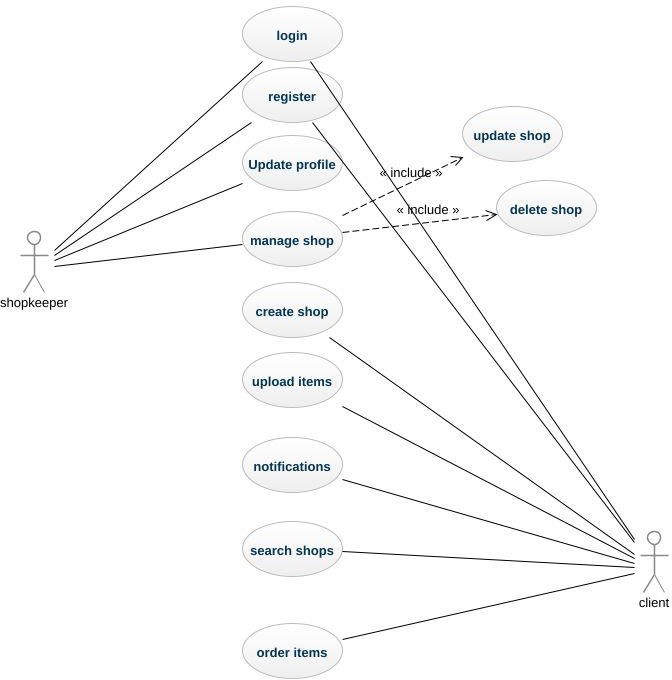


Figure - Shop Use Case Diagram

#### Jobs Use case diagram

# 

Figure - Jobs Use Case Diagram

### Use Case Description

In software and systems engineering, a use case represents actions or event steps typically defining the interactions between a role and a system to achieve a goal. The actor can be a human or other external system.

#### **Use Case Registration**

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

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-01** |
| **Name** | Registration |
| **Primary Actors** | Hostel Owner, student |
| **Goals** | Enable user to access the system. |
| **Pre-Conditions** | User must be on the registration url. |
| **Post-Conditions** | User registered 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 as user) |
| **Alternates** | The user credentials are not valid. OR  The user already registered in the system, so try to login |

Table - Use Case Registration

#### **Use Case 02: Login**

Detail of use case login which tells a login scenario

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-02** |
| **Name** | Login |
| **Primary Actors** | Hostel owner, student/client |
| **Goals** | Enable user to get access to the system. |
| **Pre-Conditions** | User must have account. |
| **Post-Conditions** | User logged in and redirected to home screen. |
| **Main Success Scenarios** | User Provides username and password. Enters the login button. System redirects user to home page after successful attempt. |
| **Alternates** | Enter correct credentials and try again. |

Table - Use case Login

#### **Use Case 03: Add hostel**

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

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-03** |
| **Name** | Add Hostel |
| **Primary Actors** | Hostel owner/admin |
| **Pre-Conditions** | User must have account with role admin or publisher |
| **Post-Conditions** | Hostel added successfully. |
| **Main Success Scenarios** | Publisher fills the form and submits it.  System Redirects to Home page after successful attempt. |
| **Alternates** | Publisher already have hostel with given name. |

Table - Use Case Add Hostel

#### **Use Case 04: Search Hostel**

Detail of use case Search hostel where user can search hostel by name, address and city.

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-04** |
| **Name** | Search Hostel |
| **Primary Actors** | User |
| **Pre-Conditions** | User must have account and logged in |
| **Post-Conditions** | Hostel found successfully. |
| **Main Success Scenarios** | User inputs the text value in search bar and presses enter. System returns list of matched hostels. |
| **Alternates** | Hostel did not exist with the given searched input. Try different keyword for to search. |

Table - Use Case Search Hostel

#### **Use Case 05: Update Hostel Record:**

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

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-05** |
| **Name** | Update Hostel |
| **Primary Actors** | Publisher/admin |
| **Pre-Conditions** | User must have account with role admin or publisher |
| **Post-Conditions** | Hostel updated successfully. |
| **Main Success Scenarios** | User comes to his profile. Find the hostel menu. Go to the hostel wanted to update. Update the information and click update button. System responses back that hostel record updated successfully. |
| **Alternates** | Provide all the necessary information to update the hostel record |

Table - Use Case Update Hostel

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

Detail of use case deletes hostel record

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-06** |
| **Name** | Delete Hostel |
| **Primary Actors** | Publisher/admin |
| **Pre-Conditions** | User must have account with role admin or publisher must be owner of that hostel |
| **Post-Conditions** | Hostel Deleted successfully. |
| **Main Success Scenarios** | User comes to his profile. Find the hostel menu. Go to the hostel wanted to delete. Delete the hostel and click delete button. System responses back that hostel record deleted successfully. |

Table - Use Case Delete Hostel

#### **Use Case 07: 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-07** |
| **Name** | Book hostel room/seat |
| **Primary Actors** | Student/client |
| **Goals** | To book hostel seat online |
| **Pre-Conditions** | Login, add profile, should have credit card |
| **Post-Conditions** | Booking of hostel is done with notification to owner |
| **Scenarios** | User selects hostel seat. Provides credit card information. Hostel seat booked successfully |
| **Alternates** | Provide authenticated information and try again |

Table - Use Case Book Hostel

#### **Use Case 08: Give Review to Hostel**

User gives a review to the hostel he wants

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-08** |
| **Name** | Give review to hostel |
| **Primary Actors** | Student/client |
| **Goals** | To give a review to hostel |
| **Pre-Conditions** | User must be login |
| **Post-Conditions** | Review given successfully |
| **Scenarios** | User selects hostel. Writes a comment. Submit the comment. System response review given successfully |

Table - Use Case Review to Hostel

### Module 02: Shops

#### **Use Case 09: Create Shop**

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-09** |
| **Name** | Create Shop |
| **Primary Actors** | Owner/admin |
| **Goals** | To create shop |
| **Pre-Conditions** | User should have account and must be logged in |
| **Post-Conditions** | Shop created successfully |
| **Scenarios** | User fills the shop form. Provides necessary information. Submits the form. System shows response of Shop created successfully |

Table - Use Case Create Shop

#### **Use Case 10: Update Product Info for Shop**

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-10** |
| **Name** | Update items for shop |
| **Primary Actors** | Owner/admin |
| **Goals** | Update Item Successfully |
| **Pre-Conditions** | User must be owner of product or an admin. |
| **Post-Conditions** | Item updated successfully |
| **Scenarios** | User clicks the edit button. Server response back with data to edit. User edits the data and submits. Server responses data saved successfully |

Table - Update Product

#### **Use Case 11: Delete Product for Shop**

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-11** |
| **Name** | Delete items for shop |
| **Primary Actors** | Owner/admin |
| **Goals** | Delete Item Successfully |
| **Pre-Conditions** | User should have account and must be logged in |
| **Post-Conditions** | Item must be deleted. |
| **Scenarios** | User clicks delete button. System deletes the shop. System response back to user. |

Table - Delete Product

#### **Use Case 12: Search Shop and Product**

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-12** |
| **Name** | Search shops |
| **Primary Actors** | Buyer / End User |
| **Goals** | Show the list of products according to search text |
| **Pre-Conditions** | User should have account and must be logged in |
| **Post-Conditions** | Show searched items successfully |
| **Scenarios** | User writes text in the search bar. System responses back with matched shops with products. |

Table - Use Case Search Shop

#### **Use Case 13: Order Product**

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-13** |
| **Name** | Order Product |
| **Primary Actors** | Buyer / End User |
| **Goals** | User Purchase products and Order Created Successfully |
| **Pre-Conditions** | User is logged in and entered the correct payment card number and should have amount to pay |
| **Post-Conditions** | Order Item Successfully |
| **Scenarios** | User selects product. Enters credit card information. Payment gateway responses back. User clicks purchase button. System responses back with order created. |

Table - Order Product

#### **Use Case 14: Review and Rate the Shop**

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-14** |
| **Name** | Review and Rate the Shop |
| **Primary Actors** | Buyer / End user |
| **Goals** | User Rate and Reviewed the Shop successfully |
| **Pre-Conditions** | User must be logged in and not already given a review to that shop |
| **Post-Conditions** | User Rate and Reviewed the Shop successfully |
| **Scenarios** | User selects the shop to give a review. Fills the review form with title and description and rating. User clicks submit button. System responses back with successful response of given review |

Table - Order Product

### USE CASES FOR JOBS

#### **USE CASE 15: Post Job**

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

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-15** |
| **Scopes** | Post a job |
| **Name** | Post job for job seekers |
| **Primary Actors** | Provider |
| **Goals** | To show jobs online for job seekers. |
| **Pre-Conditions** | Logged in and should have role owner |
| **Post-Conditions** | Job is posted and available to view online |
| **Scenarios** | User clicks job menu. Press Create Job button. Enters the job details. Presses the submit button. System responses with job posted successfully |

Table 15- Use Case Post Job

#### **Use Case 16: Search Jobs**

Detail of use case where it shows searching of jobs by user

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-16** |
| **Scopes** | To gets recent jobs |
| **Name** | Search Jobs |
| **Primary Actors** | User |
| **Goals** | To get posted jobs |
| **Pre-Conditions** | User must be logged in |
| **Post-Conditions** | Matched jobs found successfully |
| **Scenarios** | User inputs text in the search bar. On change event fires of. System responses back with matched jobs |

Table 16- Search Jobs

#### **Use Case 17: Apply for Job**

Detail of use case where it how user apply for job

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-17** |
| **Name** | Apply for Job |
| **Primary Actors** | User |
| **Goals** | To apply for desired job |
| **Pre-Conditions** | User must be logged in |
| **Post-Conditions** | User applied for job successfully |
| **Scenarios** | User clicks on interested job. Click applies. User fills the form with required data. Uploads the CV. User Clicks the Submit button. System responses back with message applied for job successfully |

Table 17- Search Jobs

#### **Use Case 18: Update Job**

Detail of use case where it how owner update the job

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-18** |
| **Name** | Update Job |
| **Primary Actors** | owner |
| **Goals** | To update the job |
| **Pre-Conditions** | User must be logged and user must have role provider |
| **Post-Conditions** | User updated the job successfully |
| **Scenarios** | User goes to interested job. Clicks update button. System responses back with data to update. User updates the data. Clicks submit. System responses back with message record updated successfully |

Table 18- Update Job

#### **Use Case 19: Delete Job**

Detail of use case where it how owner delete the job

|  |  |
| --- | --- |
| **Use Case ID no.** | **UC-19** |
| **Name** | Delete Job |
| **Primary Actors** | owner |
| **Goals** | To delete the job |
| **Pre-Conditions** | User must be logged and user must have role provider |
| **Post-Conditions** | User deleted job successfully |
| **Scenarios** | User goes to interested job. Clicks delete button. System responses back with message record deleted successfully |

Table 19- Use Case

delete Job

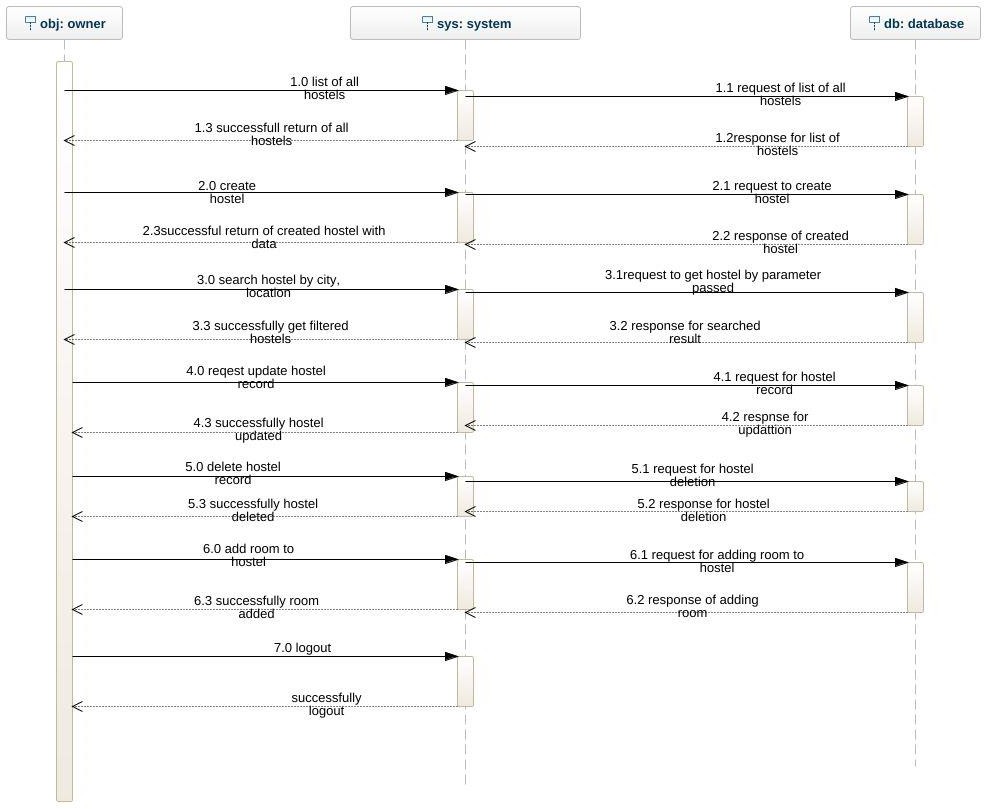
# CHAPTER 3 SYSTEM DESIGN

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.

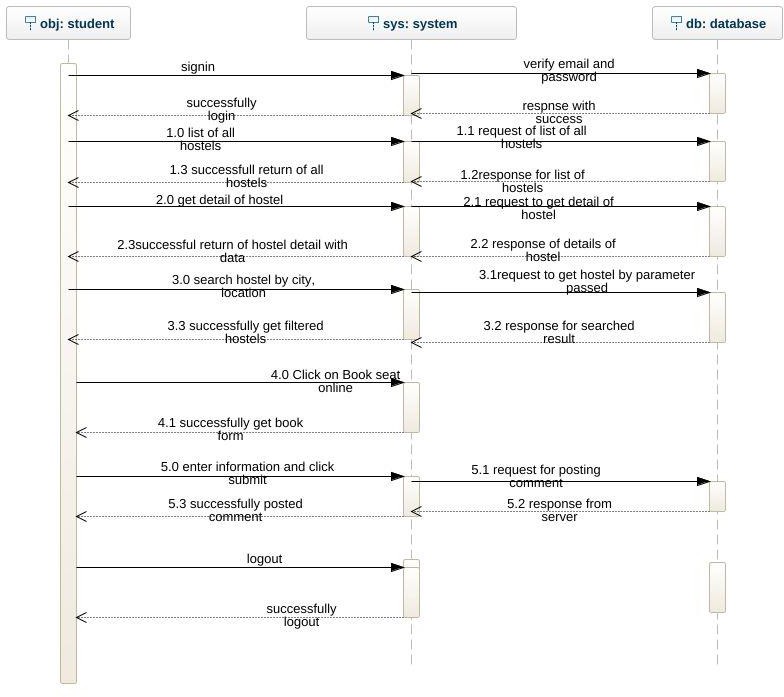
### System Sequence Diagram

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

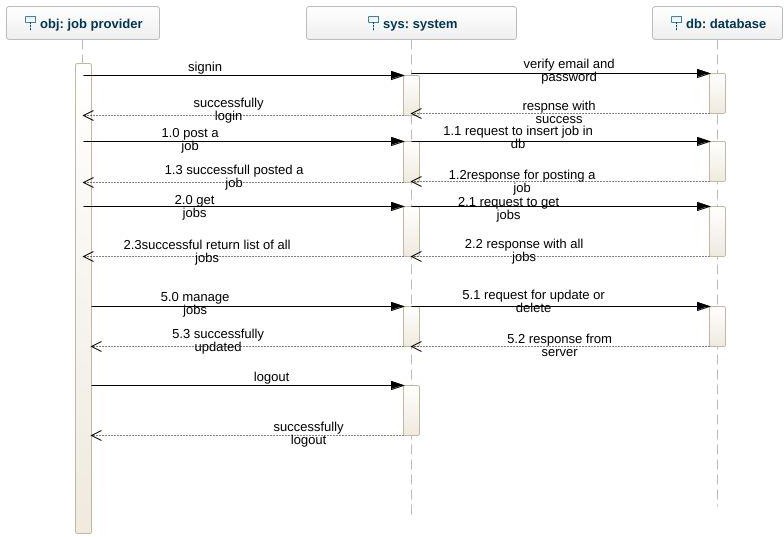
##### Sequence Diagram for owners



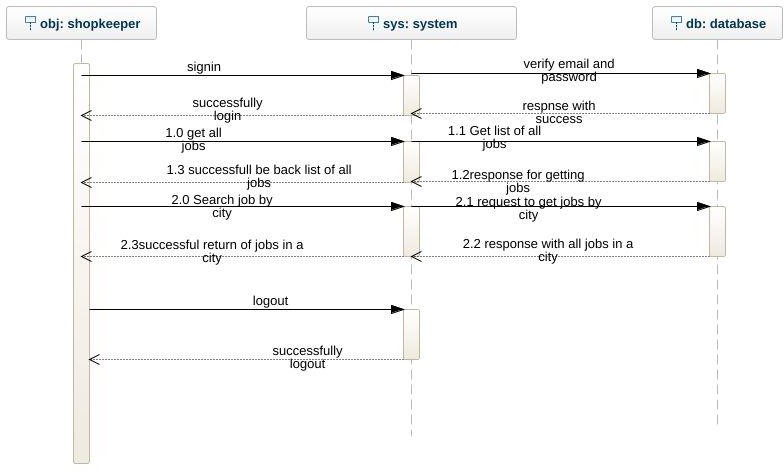
* + 1. **Sequence Diagram for students**



##### 3.23 Sequence Diagram for Job Provider

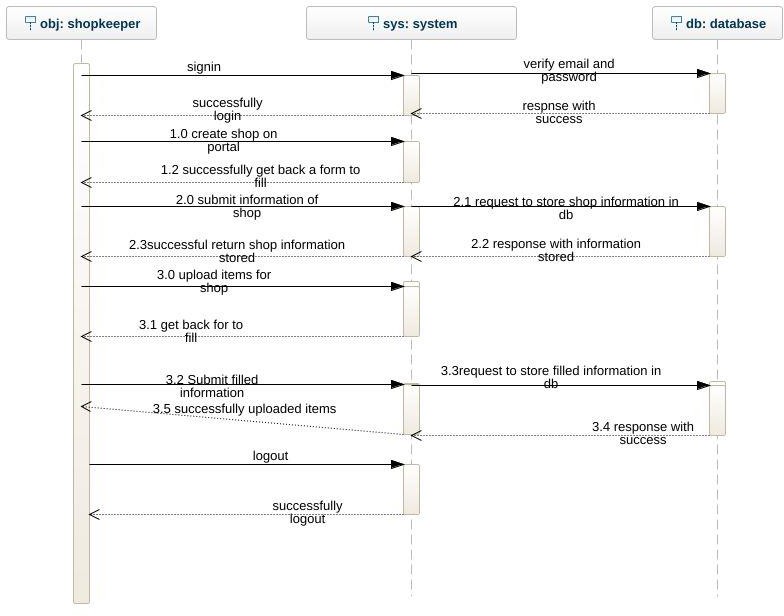


**3.2.4 Sequence Diagram for Job Seeker**

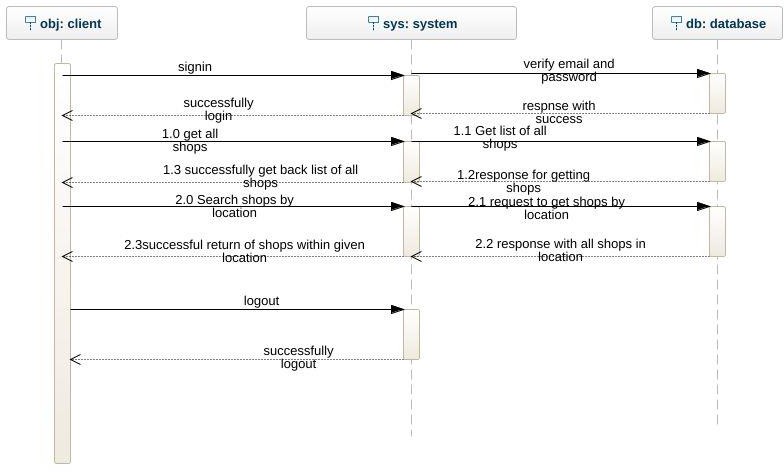


##### 3.2.4 Sequence Diagram for Shopkeeper

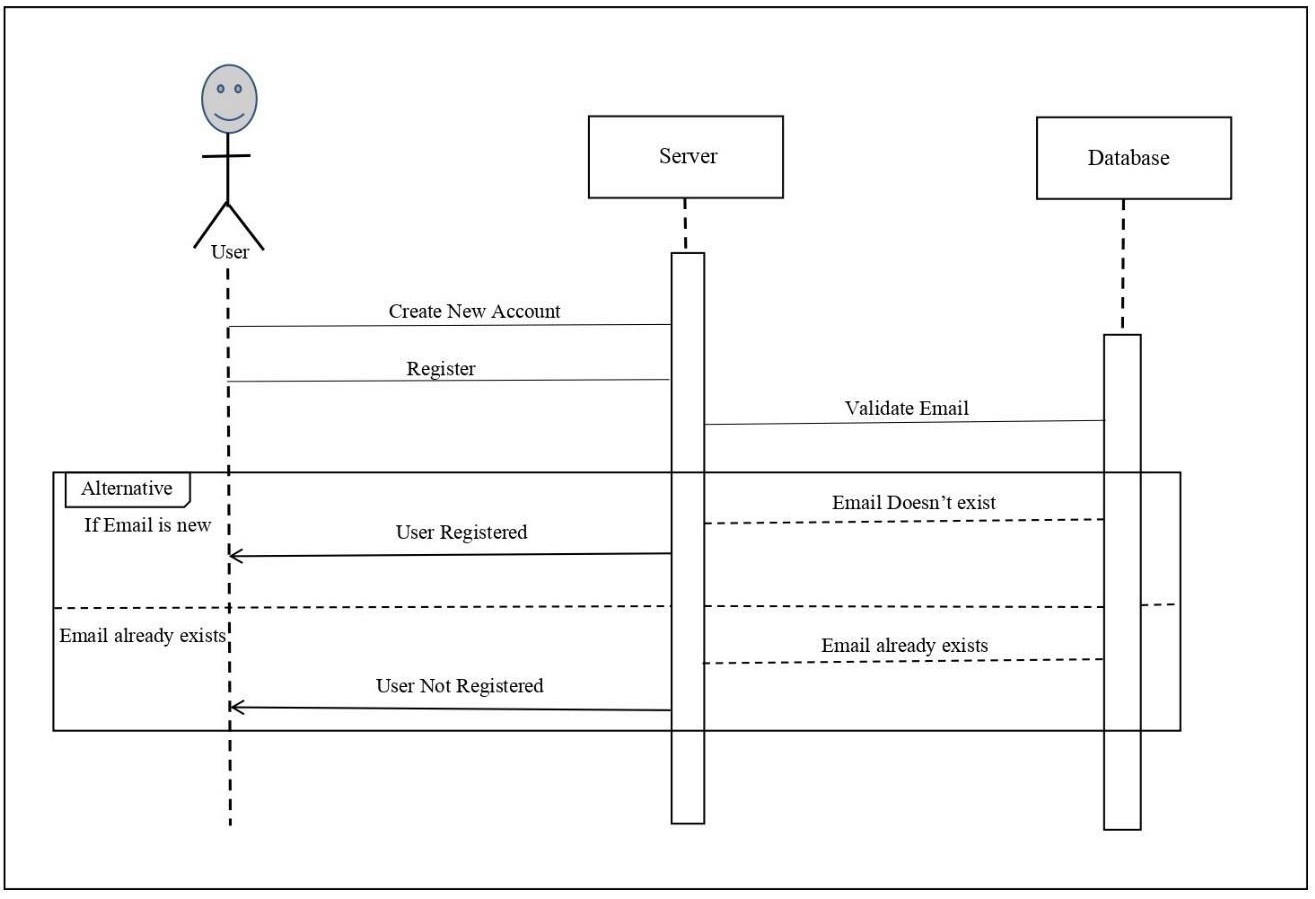
* + 1. **Sequence Diagram for Shopkeeper**



* + 1. **Sequence Diagram for client who wants to buy something from shop**



* + 1. **Register New User**



*System Design*

32

*System Design*

*System Design*

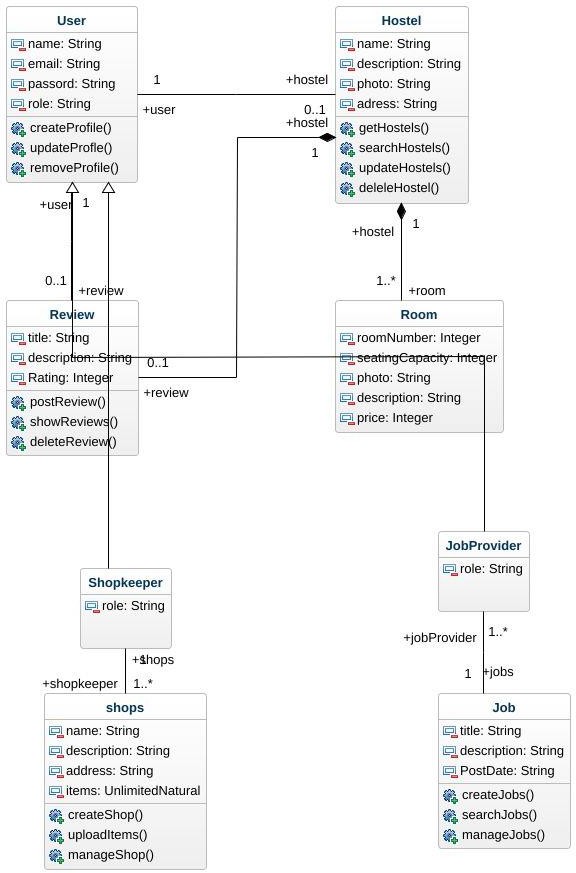
### CLASS DIAGRAM

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.

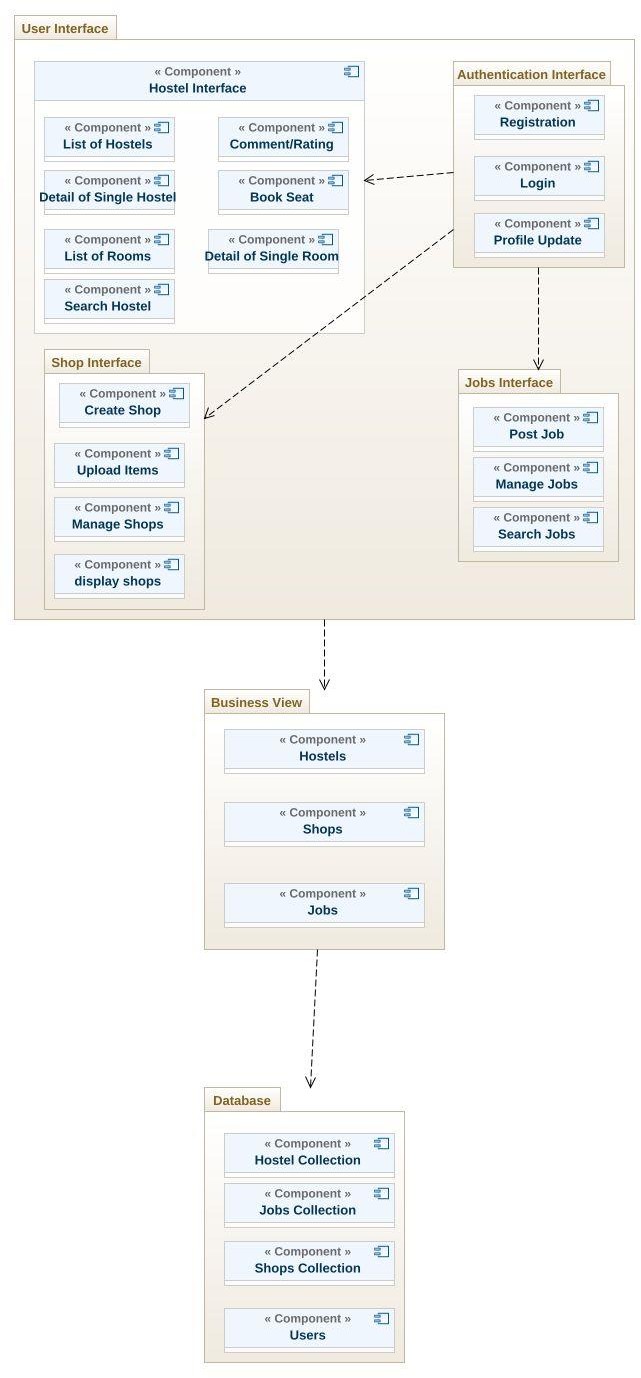
Considered to be the type of the structural diagrams as it represents the whole structure of the specific systems. Also has the similarity with the flow chart. Class diagram has three main parts represented by the rectangular box. Three parts represent the following things:

* + 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.

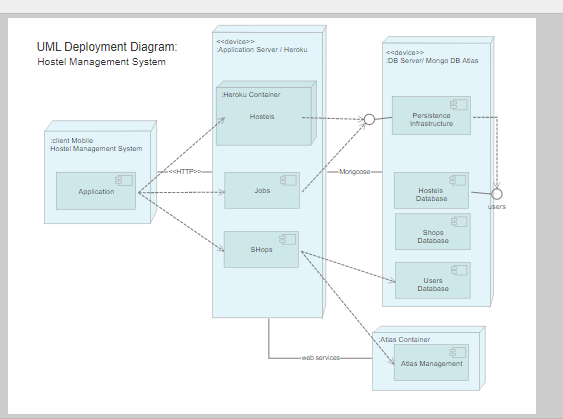
Figure 3. 2 Class Diagram



41



## 3.4 Deployment Diagram



43

# CHAPTER 4 IMPLEMENTATION

* + 1. **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.

### Functional Features of the Model:

The main features of the system are: User friendly

Friendly interface Responsiveness Reliability Security

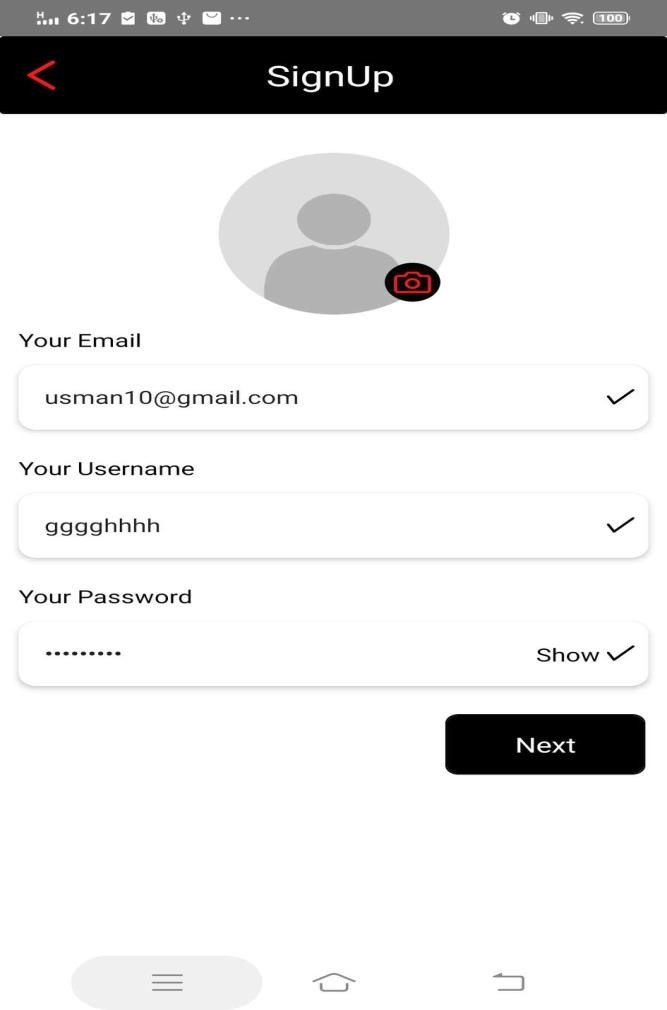
### TOOLS AND TECHNIQUES

|  |  |  |  |
| --- | --- | --- | --- |
| **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.** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **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.** |  | | |

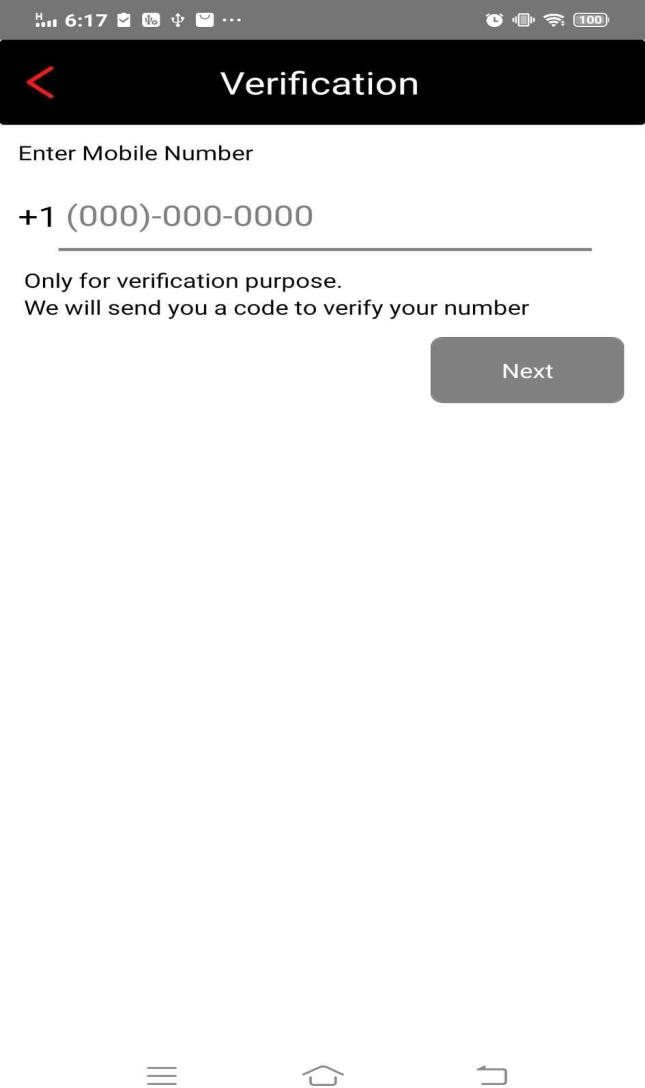
##### User Manual

**4.1 Signup/Login**

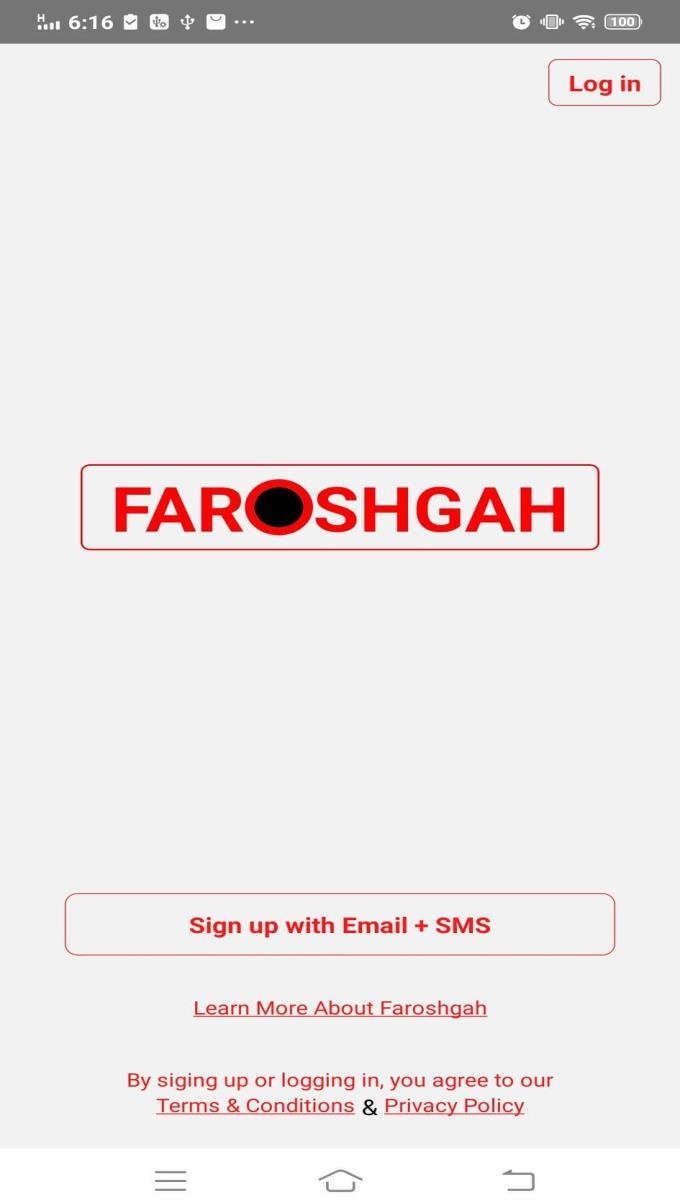


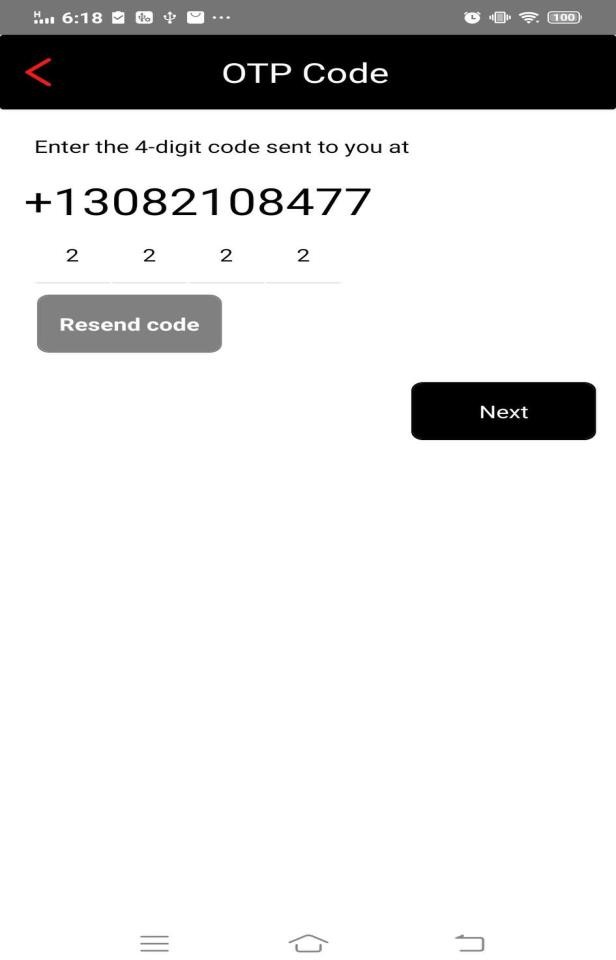
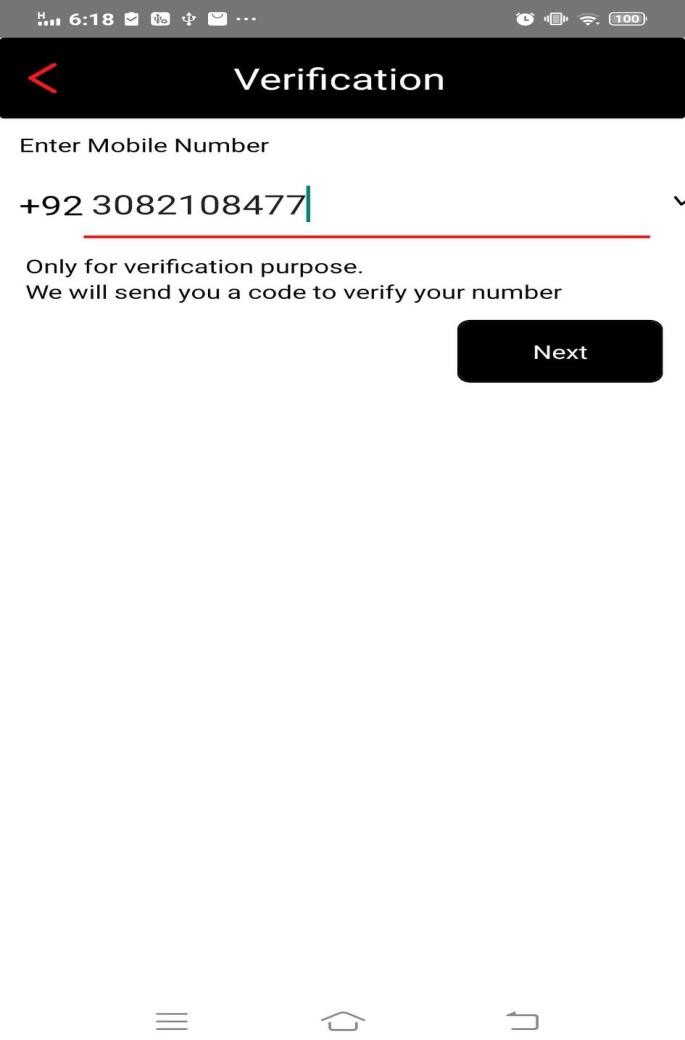
##### Signup: You can signup by using this screen

**if the user enters correct information, he will be loge din**

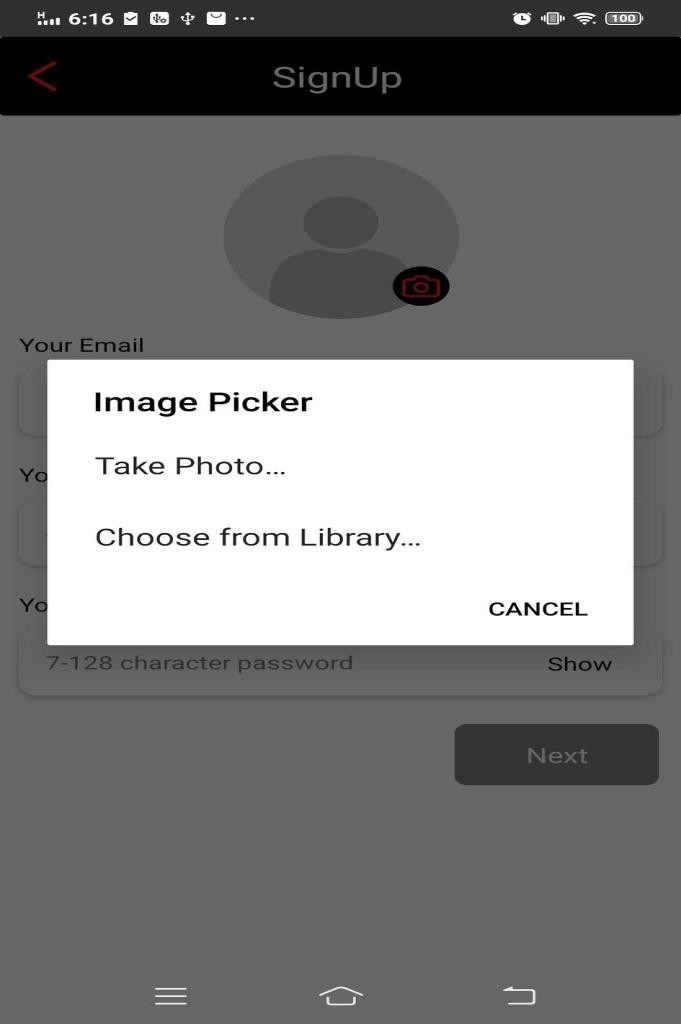


##### At the time signup, user have to get verification code, After this he/she can be able to do next

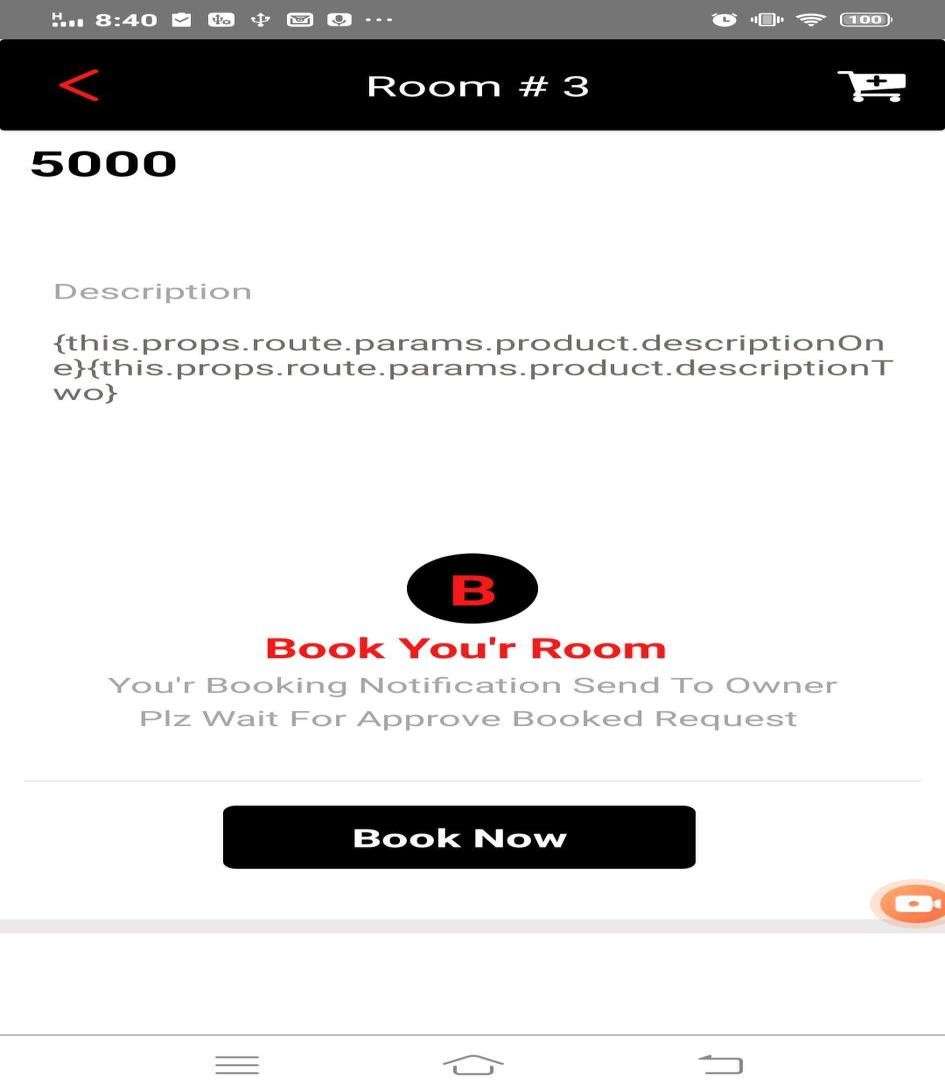


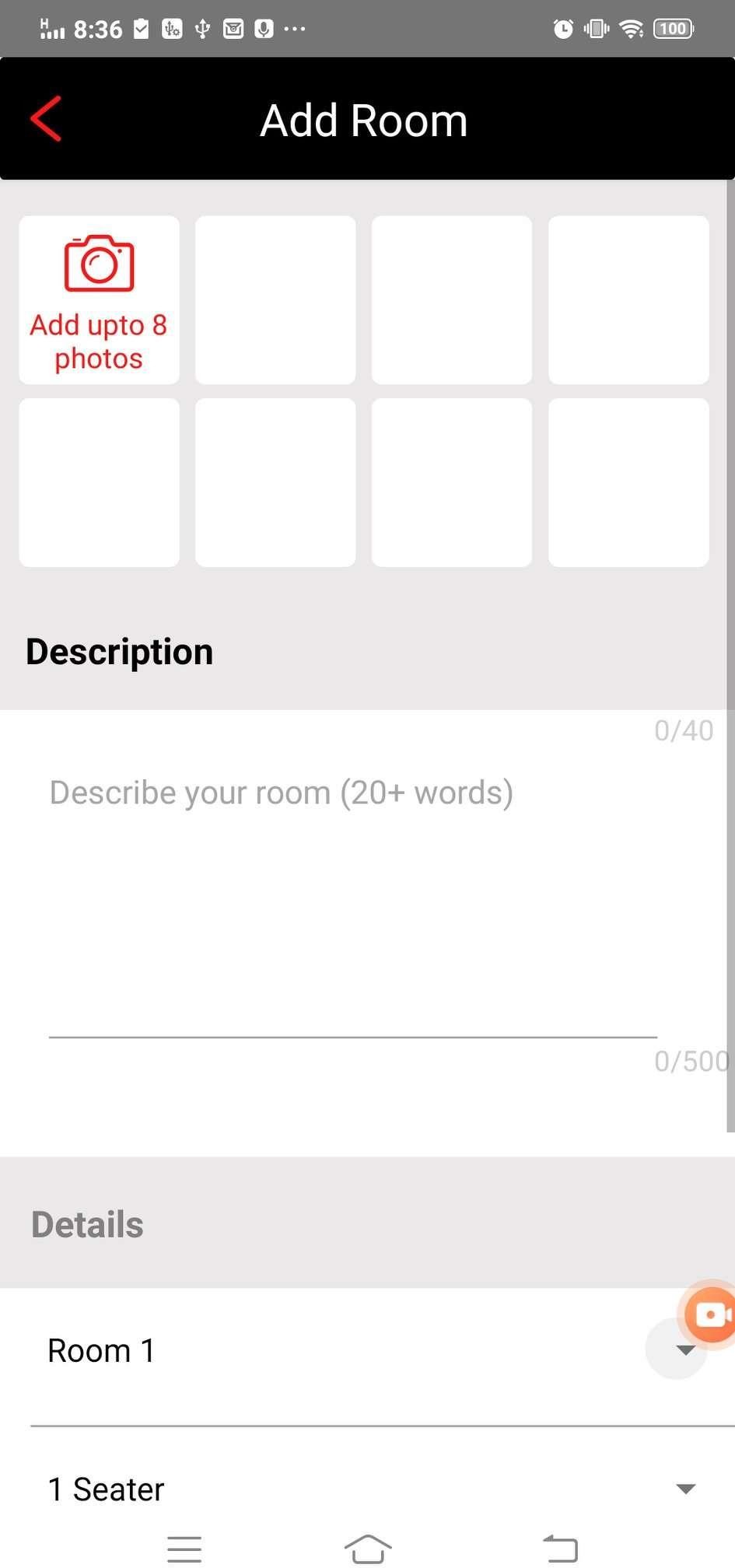
 

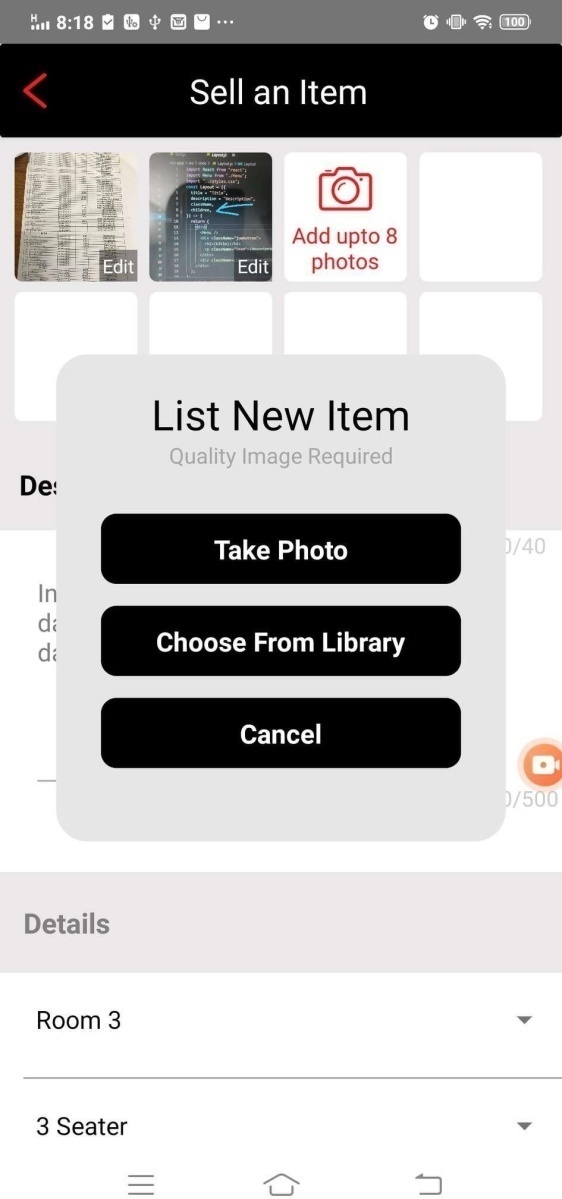
##### After getting verification code on mobile, fill it in the given screen to verify

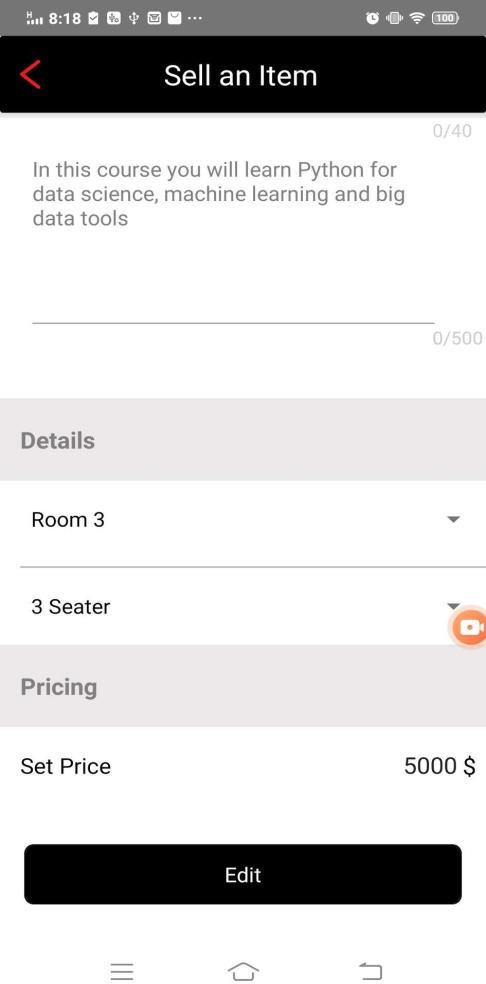


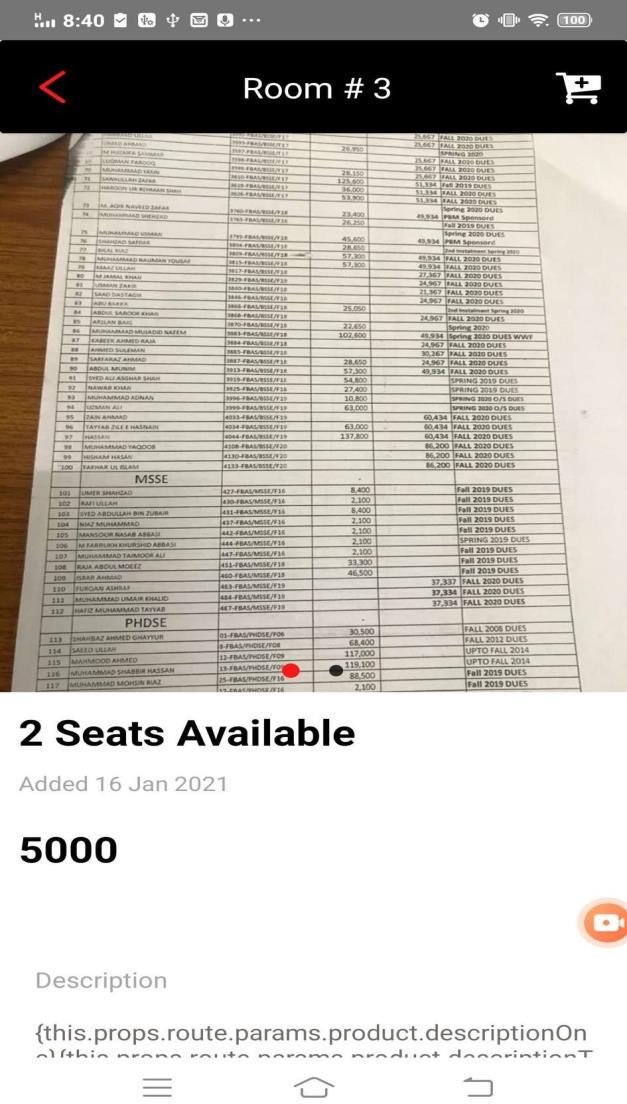
**Pick photo to set it on the profile**



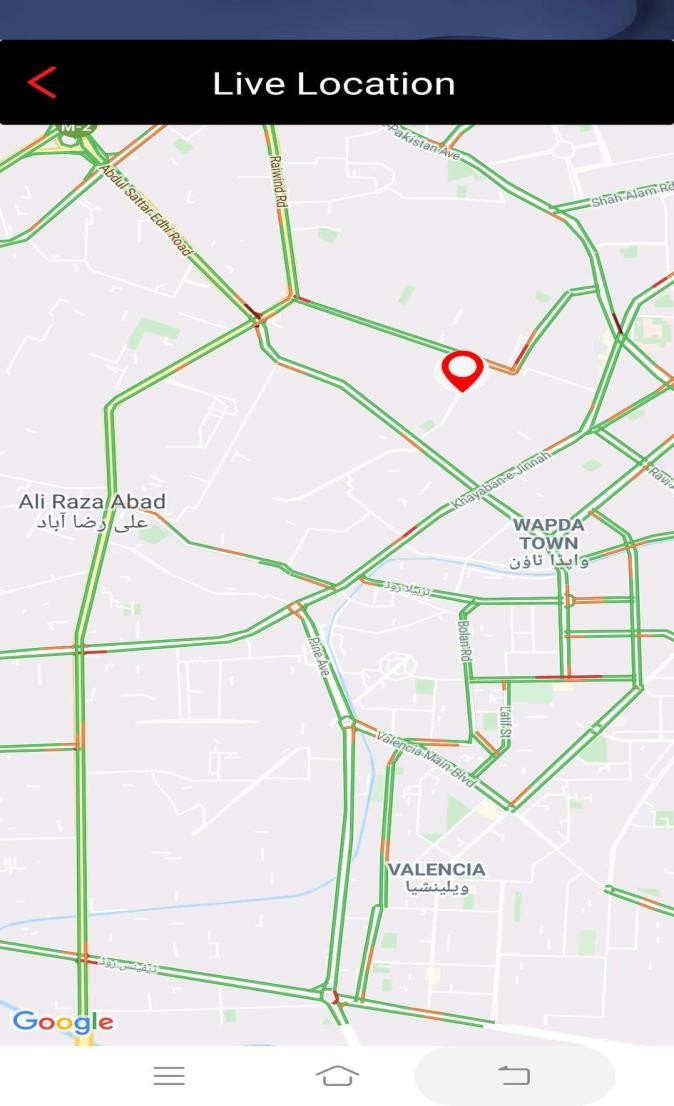




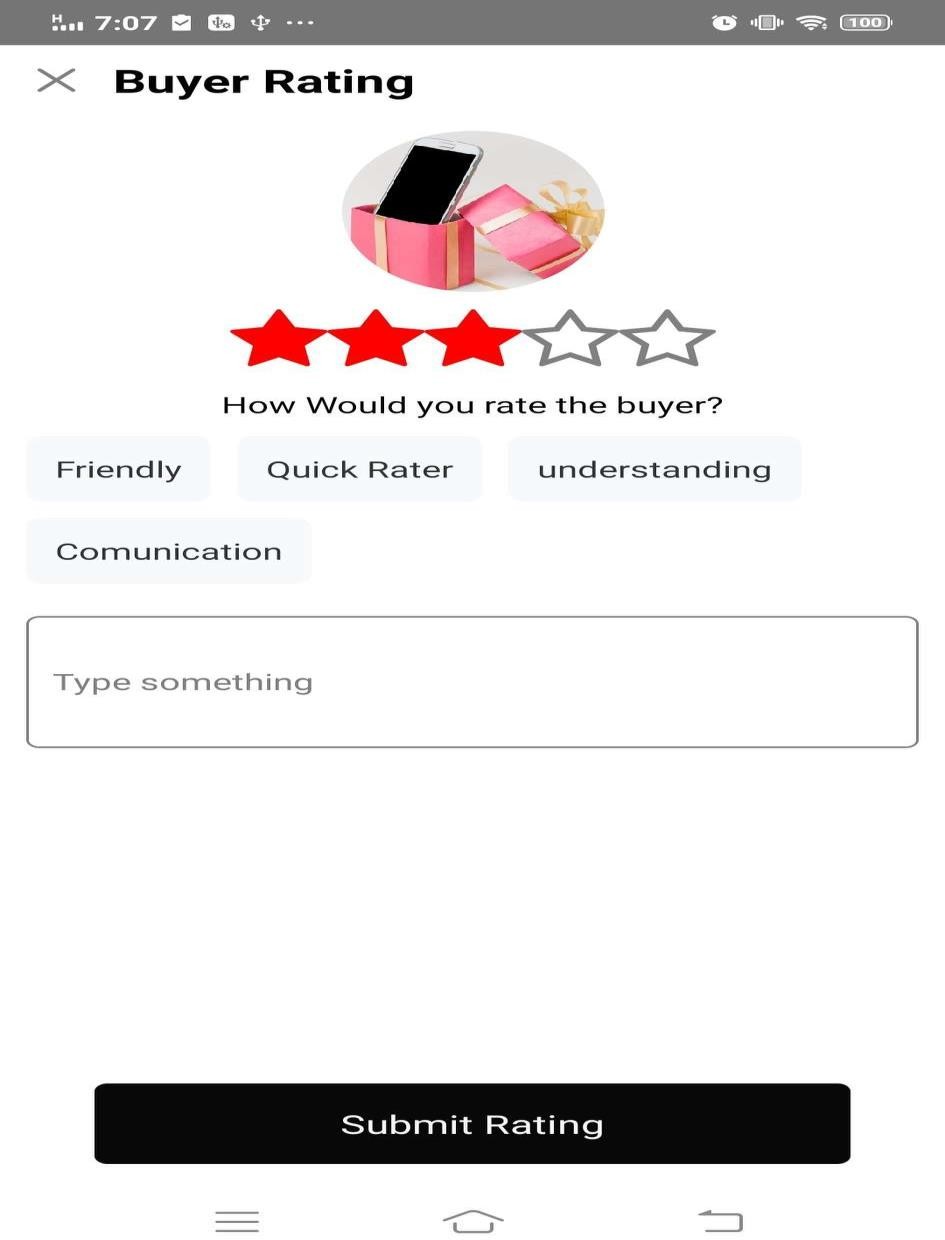




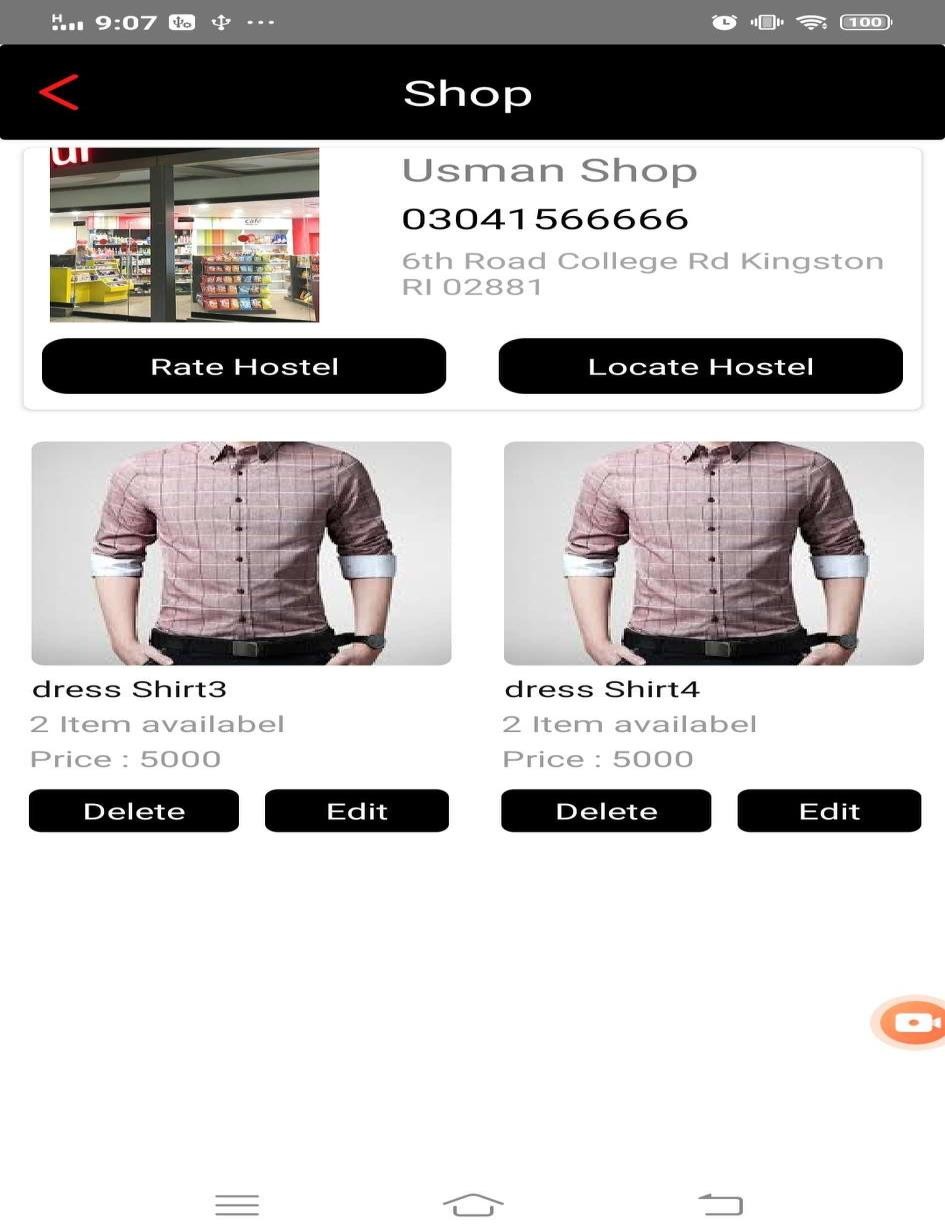
Live location



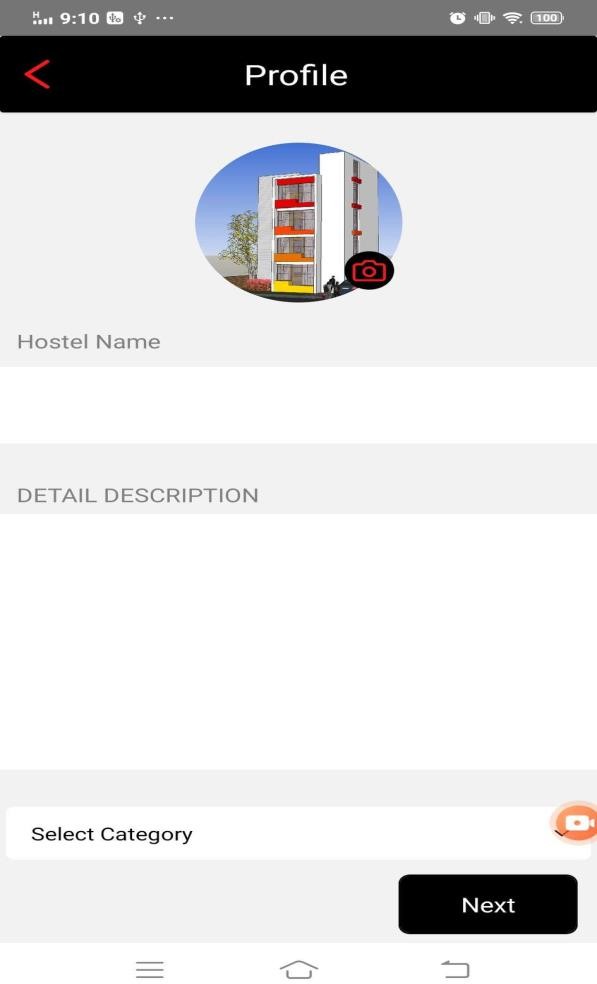
Rating

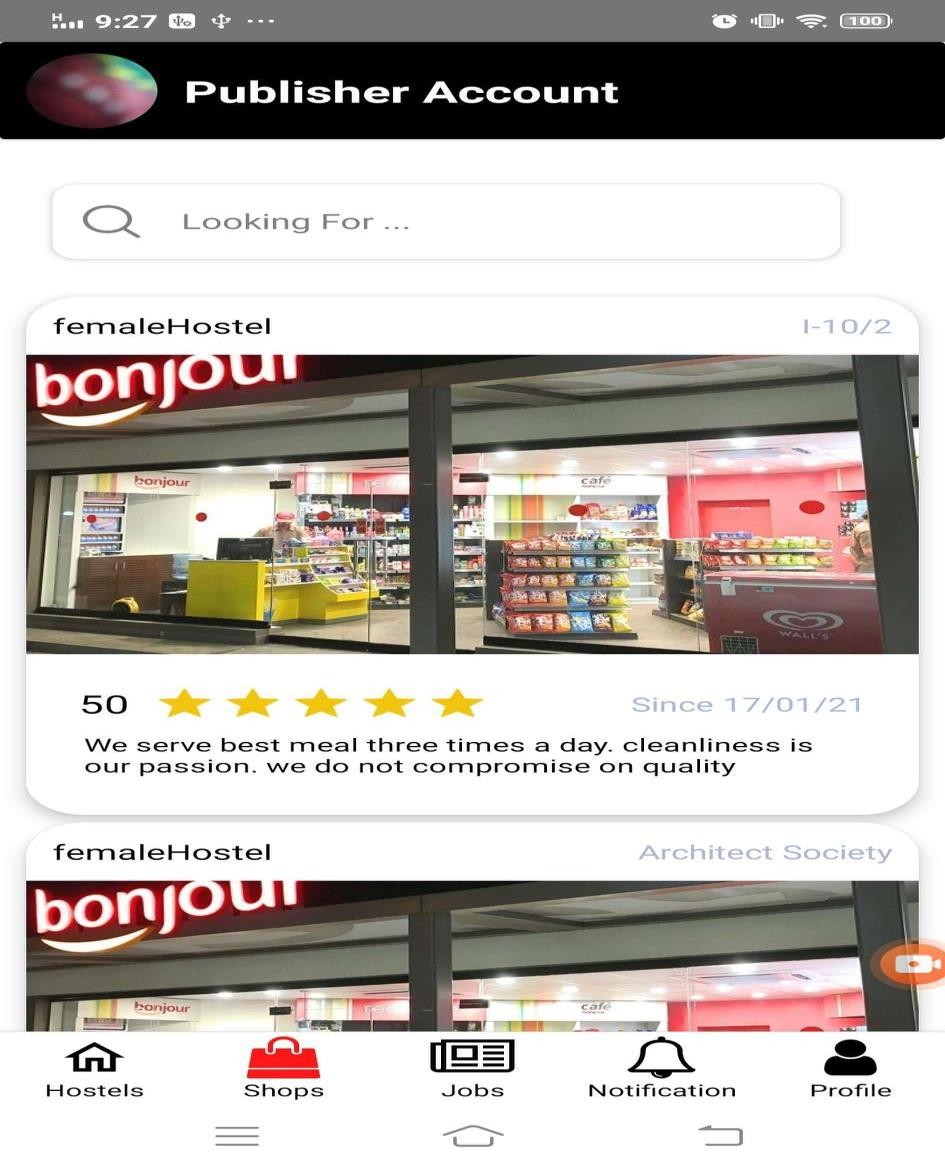


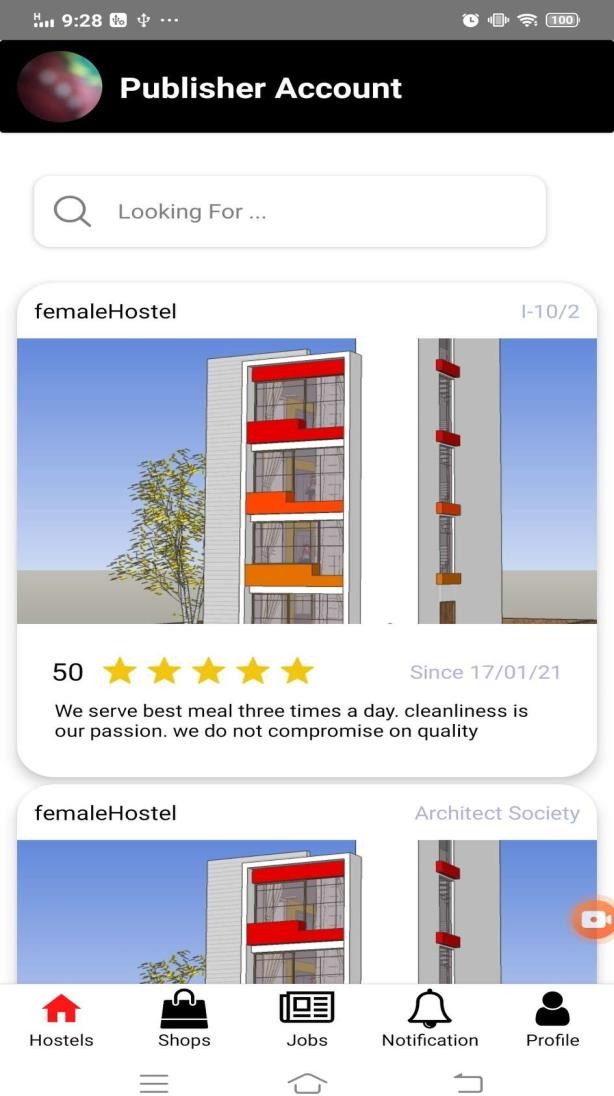
Publisher Account

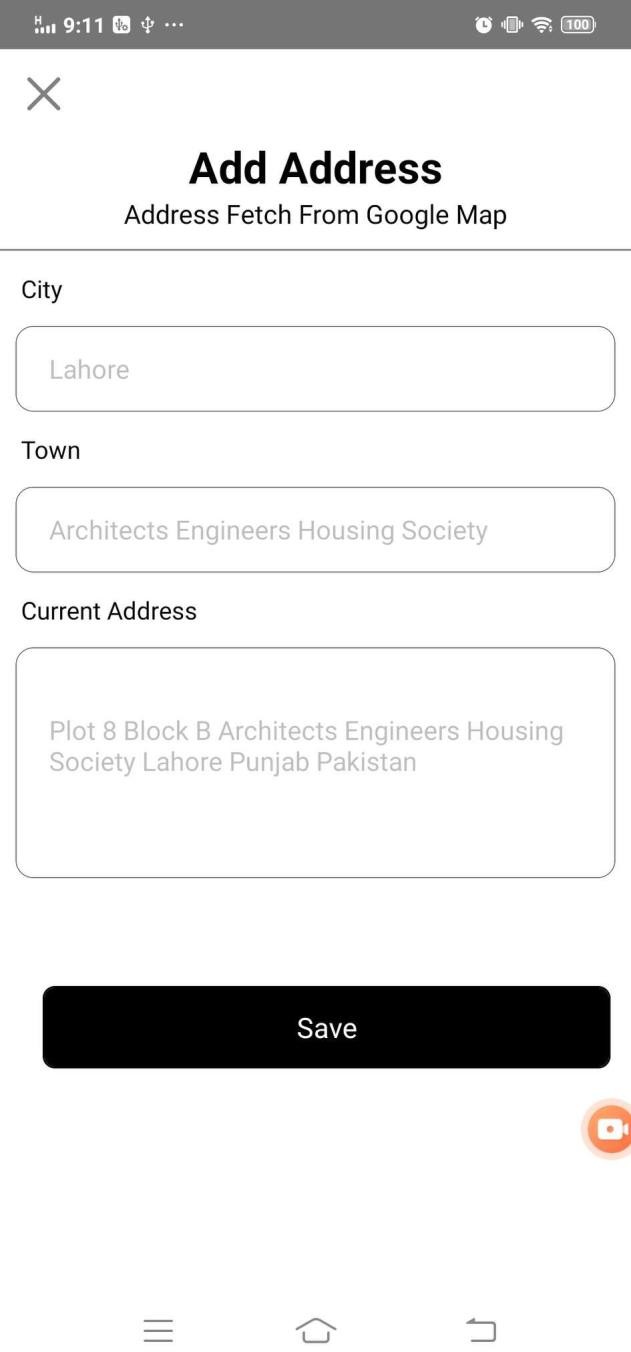
Shop Section

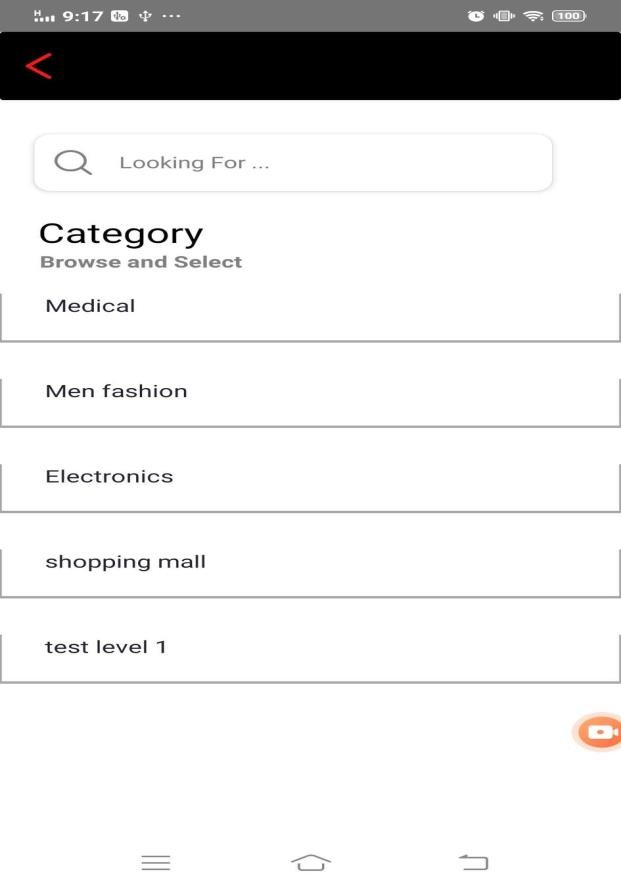
Hostel Section

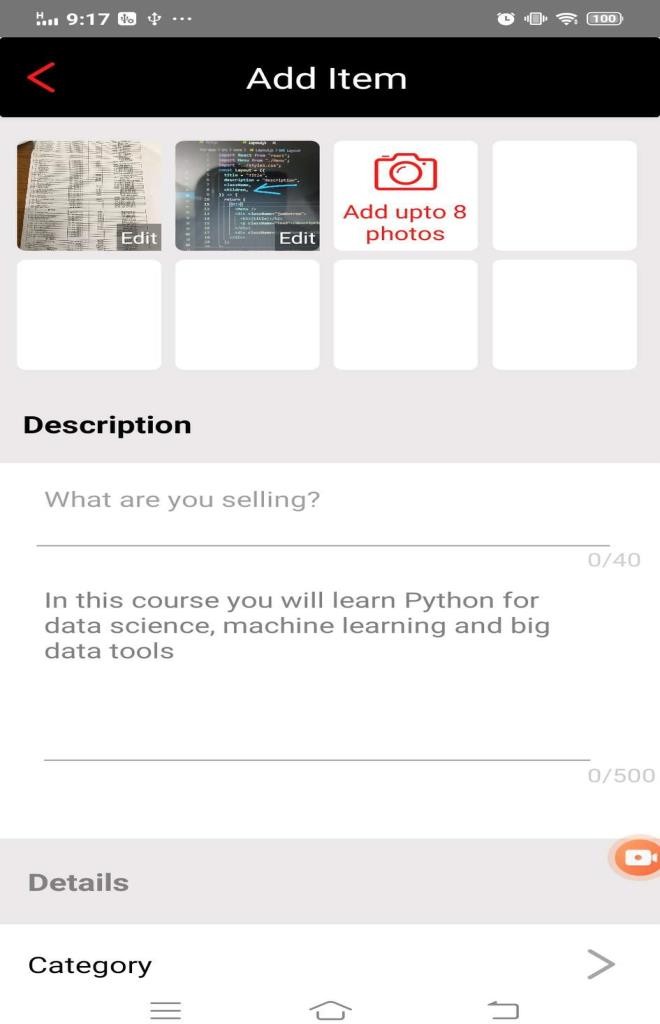


Publisher Account









# CHAPTER 5

**System Testing**

## System Testing

Software testing is a post developmental step. It must be performed to minimize any potential errors or ambiguities. We have tested all the use cases and each of them is passed. The testing of the software gives a contentment to the developer as well as the user and it gives a quality assurance to all the stakeholders concerned.

### Black Box Testing

Black box testing, also known as behavioral testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not these tests can be functional or non-functional though usually useful.

### White Box Testing

White box also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing and Structural Testing. This is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester.

### Test Cases

System-tester tests the system under specific circumstances and conditions also checks whether work or not under given circumstance.

##### Test Case Registration

Table 5.1 shows the test case registration:

|  |  |
| --- | --- |
| **Test Case ID** | TC-01 |
| **Functional Area/Module** | Registration |
| **Purpose** | To enable users to operate/use the system. |
| **Action to Perform** | 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 as user) |
| **Prerequisites** | Hostel owner and student are on the Website. |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Account Created successfully. |
| **Comments:** Test passed successfully | |

Table 5.1 Test Case Registration

##### Test Case Login

Table 5.2 shows the test case login

|  |  |
| --- | --- |
| **Test Case ID** | TC-02 |
| **Functional Area/Module** | Login |
| **Purpose** | To Enable user to run the system |
| **Action to Perform** | User Provides username and password.  Redirect to Home page after successful attempt. |
| **Prerequisites** | User Account Created successfully. |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Redirect to home |
| **Comments:** Test passed successfully | |

Table 5.2 Test Case Login

##### Test Case Add Hostel:

Table 5.3 shows the test case add hostel:

|  |  |
| --- | --- |
| **Test Case ID** | TC-03 |

|  |  |
| --- | --- |
| **Functional Area/Module** | Add hostel |
| **Purpose** | To show hostel online to hostel seekers. |
| **Action to Perform** | Create a hostel  And publish it online |
| **Prerequisites** | Login |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Hostel created |
| **Comments:** Test passed successfully | |

Table 5.3 Test Case Add Hostel

##### Test Case Search Hostel:

Table 5.4 shows the test case search hostel:

|  |  |
| --- | --- |
| **Test Case ID** | TC-04 |
| **Functional Area/Module** | Search Hostel |
| **Purpose** | To Enable students/job seekers to search hostels. |
| **Action to Perform** | User clicks the search bar.  Searches the hostels by city or location/street/zipcode. |
| **Prerequisites** | Login, user can search hostel after login |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Searching for hostels is done |
| **Comments:** Test passed successfully | |

Table 5.4 Test Case Search Hostel

##### Test Case Update Hostel Record:

Table 5.5 shows the test case Update Hostel Record:

|  |  |
| --- | --- |
| **Test Case ID** | TC-05 |
| **Functional Area/Module** | Update Hostel Record |
| **Purpose** | To update the record of hostel |
| **Action to Perform** | Updating record of hostel |
| **Prerequisites** | Login, only admin or authenticated hostel owner can do that |

67

|  |  |
| --- | --- |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Updating record of hostel is completed |
| **Comments:** Test passed successfully | |

Table 5.5 Test Case Update Hostel Record

##### Test Case Delete Hostel Record:

Table 5.6 shows the test case delete hostel record:

|  |  |
| --- | --- |
| **Test Case ID** | TC-06 |
| **Functional Area/Module** | Delete hostel record |
| **Purpose** | To delete the record of hostel |
| **Action to Perform** | Deleting hostel record from db |
| **Prerequisites** | Login, register as a hostel owner |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Deletion of hostel is done |
| **Comments:** Test passed successfully | |

Table 5.6 Test Case Delete Hostel Record

##### Test Case Search hostel within a specific location:

Table 5.7 shows the test case Search hostel within location:

|  |  |
| --- | --- |
| **Test Case ID** | TC-07 |
| **Functional Area/Module** | Search hostel within location |
| **Purpose** | To get hostels located in chosen location |
| **Action to Perform** | Searching hostel |
| **Prerequisites** | Navigate to the search page |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Searching of hostels is done |
| **Comments:** Test passed successfully | |

Table 5.7 Test Case Search Hostel within a Specific Location

##### Test Case Book Hostel’s Seat:

Table 4.8 shows the test case Book hostel room/seat:

|  |  |
| --- | --- |
| **Test Case ID** | TC-08 |
| **Functional Area/Module** | Book hostel room/seat |
| **Purpose** | To book hostel online |
| **Action to Perform** | Booking hostel is done |
| **Prerequisites** | Login, add profile |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Booking of hostel is done |
| **Comments:** Test passed successfully | |

Table 4.8 Test Case Book Hostel Room/Seat

##### Test Case Review or Rating:

Table 5.9 shows the test case Review or Rating:

|  |  |
| --- | --- |
| **Test Case ID** | TC-09 |
| **Functional Area/Module** | Review or Rating |
| **Purpose** | To give the rating to the hostel |
| **Action to Perform** | Giving rating to hostel |
| **Prerequisites** | Login, add profile |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Rating is given to the hostel |
| **Comments:** Test passed successfully | |

Table 5.9 Test Case Review or Rating

##### TEST CASES FOR SHOP MANAGEMENT SYSTEM

* + 1. **Test Case Create New Shop:**

Table 5.10 shows the test case Create New Shop:

|  |  |
| --- | --- |
| **Test Case ID** | TC-10 |
| **Functional Area/Module** | Create New Shop |
| **Purpose** | To create shop on portal |
| **Action to Perform** | Creation of shop on portal |
| **Prerequisites** | Login, add profile |

69

|  |  |
| --- | --- |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Creation of shop is done |
| **Comments:** Test passed successfully | |

Table 5.10 Test Case Create New Shop

* + 1. **Test Case Upload Items for Shop:**

Table 5.11 shows the test case Upload Items for Shop:

|  |  |
| --- | --- |
| **Test Case ID** | TC-11 |
| **Functional Area/Module** | Upload Items for Shop |
| **Purpose** | To upload items for shop |
| **Action to Perform** | Uploading items for shop |
| **Prerequisites** | Login, only shopkeeper can do that |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Uploading items done |
| **Comments:** Test passed successfully | |

Table 5.11 Test Case Upload Items for Shop

##### Test Case Update Items for Shop:

Table 5.12 shows the test case Update Items for Shop:

|  |  |
| --- | --- |
| **Test Case ID** | TC-12 |
| **Functional Area/Module** | Update Items for Shop |
| **Purpose** | To update items for shop |
| **Action to Perform** | Updating items for shop |
| **Prerequisites** | Login, add profile |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Updating item is done |
| **Comments:** Test passed successfully | |

Table 5.12 Test Case Update items for shop

##### Test Case Delete Shop Items & Detail:

Table 5.13 shows the test case Delete Shop Items and detail:

|  |  |
| --- | --- |
| **Test Case ID** | TC-13 |

|  |  |
| --- | --- |
| **Functional Area/Module** | Delete Shops and its detail |
| **Purpose** | To delete the shops from database |
| **Action to Perform** | Deleting shop and its detail |
| **Prerequisites** | Login, add profile |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Deleting shop is done |
| **Comments:** Test passed successfully | |

Table 5.13 Test Case Delete Shops and its detail

##### Test Case Search Shop:

Table 5.14 shows the test case Search Shop:

|  |  |
| --- | --- |
| **Test Case ID** | TC-14 |
| **Functional Area/Module** | Search Shop |
| **Purpose** | To search the desired shops |
| **Action to Perform** | Searching shops online |
| **Prerequisites** | Come to the search page |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Searching is done |
| **Comments:** Test passed successfully | |

Table 5.14 Test Case Search Shop

##### Test Case Order Item:

Table 5.15 shows the test case Order Item:

|  |  |
| --- | --- |
| **Test Case ID** | TC-15 |
| **Functional Area/Module** | Order Item |
| **Purpose** | To buy an item from the shop |
| **Action to Perform** | Order items from the shop |
| **Prerequisites** | Login, add profile |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Order is done |

71

**Comments:**

Test

passed successfully

Table 5.15 Test Case Order Item

##### USE CASES FOR JOBS

* + 1. **Test Case Create Job:**

Table 5.16 shows the test case Create Job:

|  |  |
| --- | --- |
| **Test Case ID** | TC-16 |
| **Functional Area/Module** | Create Job |
| **Purpose** | To show jobs online for job seekers. |
| **Action to Perform** | Uploading jobs for job seekers |
| **Prerequisites** | Login, add profile |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Uploading jobs is done |
| **Comments:** Test passed successfully | |

Table 5.16 Test Case Create Job

##### TEST CASES FOR ADD JOBS

* + 1. **Test Case Update Jobs detail:**

Table 5.17 shows the test case Update Jobs detail:

|  |  |
| --- | --- |
| **Test Case ID** | TC-17 |
| **Functional Area/Module** | Update Jobs detail |
| **Purpose** | Job updating by client those already added |
| **Action to Perform** | Client updates the jobs for job seekers |
| **Prerequisites** | Registered and logged-in client. |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Jobs are managed successfully. |
| **Comments:** Test passed successfully | |

Table 5.17 Test Case Update Jobs detail

##### Test Case Delete the Job:

Table 5.18 shows the test case Delete the Job:

|  |  |
| --- | --- |
| **Test Case ID** | TC-18 |
| **Functional Area/Module** | Delete the Job |
| **Purpose** | delete jobs by provider those already added |
| **Action to Perform** | Providers defines the jobs for job seekers and delete the job. |
| **Prerequisites** | Registered and logged-in client. |
| **Test Case Engineer** | Kaleen Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Jobs are deleted successfully. |
| **Comments:** Test passed successfully | |

Table 5.18 Test Case Delete the Job By Provider

##### Test Case Delete the Job:

Table 5.19 shows the test case Delete the Job:

|  |  |
| --- | --- |
| **Test Case ID** | TC-19 |
| **Functional Area/Module** | Delete the Job |
| **Purpose** | delete jobs by provider those already added |
| **Action to Perform** | Providers defines the jobs for job seekers and delete the job. |
| **Prerequisites** | Registered and logged-in client. |
| **Test Case Engineer** | Kaleem Ullah |
| **Environment** | Android |
| **Expected Result(s)** | Jobs are deleted successfully. |
| **Comments:** Test passed successfully | |

Table 5.19 Test Case Delete the Job

73

*Feroshgah 62*

# CHAPTER 6 CONCLUSION

*References Appendix*

## CONCLUSION

By use of Feroshgah application the user can search and finds hostels, shops, jobs, as per location. This application with elegant design provides ease to students and owners and can make their life by saving lots of time. It provides facility to get hostels, shops and jobs online without wasting your time by visiting hostels or shops on site. You can search to find shops, hostels and jobs. It gives facility to provide the best result based on searched you made. You can manage shops, hostels without any difficulty.

It is fully secured and working application. It is secured from all types of security threads and provide authentication and authorization to the users. Now one can make you fool by

using this platform. It provides big opportunity to those who seeks hostel to get comfortable living in hostels.

### FUTURE EXTENSION

Moving further Feroshgah 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 Feroshgah in near future.

* + - **Add Hotels:** Feroshgah is planning to add Hotels service module where user will be able to add hotels and other users can find it for rent.

*Chapter 06 conclusion*

## REFERENCES

[www.stackoverflow.com](http://www.stackoverflow.com/) [www.wikipedia.com](http://www.wikipedia.com/) [www.tutorialpoints.com](http://www.tutorialpoints.com/) [www.youtube.com](http://www.youtube.com/) [www.w3school.com](http://www.w3school.com/)