Daffodil International University

Department of Software Engineering, FSIT

SWE-431 Project / Thesis

Project Documentation

Smart Hostel finder

Supervised by:

Fatama Binta Rafiq

Lecturer

Department of Software Engineering

Daffodil International University

Submitted by:

Kamruzzaman

161-35-1559

Department of Software Engineering

Daffodil International University

Approval

This Project titled **“Smart Hostel Finder”,** submitted by **Kamruzzaman, ID-161-35-1559** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan
Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman

Md Fahad Bin Zamal
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Md. Shohel Arman
Lecturer
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Prof Dr. Mohammad Abul Kashem
Professor
Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering & Technology, Gazipur

External Examiner

Acknowledgement

First of all, thanks to ALLAH, ALHAMDULILLAH because I am very much blessed as I have successfully reached towards the final semester. From the very beginning of my university life, I have learned a lot about software engineering as well as core computer science related knowledge from my course teachers. Moreover, they teach us ethics, morality and politeness.

It is a fortunate opportunity for me as a student of the Department of Software Engineering, one of the exalted academic cancers of the Science and Information Technology Faculty of the Daffodil international University, to express my deep feelings of gratitude to the department and to my honorable teachers and also to the department staff.

Besides, I am so much thankful to my parents that, I was always supported by them. They always prioritize my opinion and inspired me as well.

I am also so many grateful to my supervisor Fatama Binta Rafiq, Senior Lecturer, Department of Software Engineering, FSIT. Daffodil International University, Dhaka, for her excellent guidance, inspiration, encouragement and also for allowing me to work with this project. It was not possible for me to complete my thesis paper successfully without her help.

Specially, I am very thankful to my friend, elder and younger brothers of our university who always appreciate me to my studies and my daily works for learning new ones.

DECLARATION

I hereby declare that I have taken this thesis under the supervision of Fatama Binta Rafiq, Senior Lecturer, Department of Software Engineering, Daffodil International University. I also declare that neither this project nor any part of this has been submitted elsewhere for award of any degree.

Kamruzzaman

Kamruzzaman

ID: 161-35-1559

Batch: 19th

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Certified by:

Fatama
12.12

Fatama Binta Rafiq

Senior Lecturer

Department of Software Engineering

Faculty of Science & Information Technology

Daffodil International University

Abstract

I have started to develop this project from April. From the very starting of development of this project, it demands hard working, patients, persistency to meet the requirements of stakeholders. After that I have proposed the design. And then start to work.

Database plays a vital role for any application software. And so this is why, I have designed the database diagram having tables with proper relationship. After that, I have prepared the user interface and take their approval to continue to the next part. It is to be said that, the interface of my application is very simple and easy to understand. After completing that, I have started to write the core functionality of the project.

Developing project is not end of all tasks actually. There are some other important tasks to perform. And that is testing. It is also known as quality assurance also. Almost at every software company there are a quality assurance team. Their main responsibility is to find the loop holes or vulnerability of software. If there any bug remains before handover to the stakeholders, there is a change to ruin the whole project. So testing plan is very important. And after developing the project, I have tried to assure the quality of this project.

Identification of the drawbacks of the existing system leads to the development of computerized hostel finding system that will be compatible to the existing system with the system which is more users friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing hostel finding system.less human error, Strength and strain of manual labour can be reduced, High security, Data redundancy can be avoided to some extent, Data consistency, Easy to handle, Easy data updating, Easy record keeping, Backup data can be easily generated.

List of Content

List of Figures Page No

Chapter 1

Introduction

1.1 Project Overview

The system of finding hostel is very complex in Bangladesh. On which students are not able find proper hostel. There are many problems such as, value of hostel, location, direction, Quality etc.

Nowadays many student come Dhaka from outside. For this reason they don’t know which is the best hostels for them. My system helped them find the proper hostel and gives more ability to access information.

Management & integration all aspects of the business key functions including: hostel entry, name, location, rent, food item. Determination to achieve lower prices, better quality and better service is also difficult when people used to use manual ways.

Providing right information to the right people at the right time anywhere in the world, enabling peoples to improve quality, enhance review system capabilities and promote communication between hostel manager.

For all of reason, we are going to make a web based on PC and android phones. By using this web application, employees of hostel and those who want to live in hostel can be benefited as they would not face any hassles to make their work, find specific hostel, give review for specific information. Only normal user who have user id and password he can give review on hostel and read review. Most importantly, hostel authority and client can be cost benefited, find each hostel, manage internal function, commercial report and quality. And the process would be done within a few while.

1.2 Project Purpose

The main purpose of this project named “**Smart Hostel Finding**” is to make an automation system which might be helpful for millions of users from different place of country by giving them a few information. Therefore, we are going to develop such a project.

1.2.1 Background

Suddenly I notice that, people from our country come Dhaka from different places is facing a lot of problem where they live. Most of the time they select wrong place for live and there are a few hostel remaining to escape those problem. Thousands of people are being suffering. So considering that, I have brainstormed and thinking that my project is going to solve their problems.

1.2.2 Benefits & Beneficiaries

The module "Smart Hostel Finding" applications would be beneficial for some point of view. Now, I am mentioning those

Below:

- My system helps Client and hostel manager to make registration easily
- It helps authenticated access to the system.
- It would be helpful for find hostel.
- It is helpful for find best value for money hostel.
- It is also helpful for determining required hostel and location, review, rent, food, packing.
- It helps client and manager to find hostel within a moment.
- This application makes authentication system for hostel finding system.
- Our app helps hostel company to increase their business.
- This app helps client and hostel manager to work easily.
- The system also going to reduce unwanted problems
- Only registered person can get access to the system.

I have also mentioned some benefits as well as beneficiaries. So, I think this application is very much helpful for users

1.2.3 Goals

The main scope of this project is to develop an application based on PC and android device. As most of hostels are giving low quality service, so I implement my system firstly for those persons who wants or live hostel.

This document is also aimed at specifying requirements of software to be developed but it can also be applied to assist in the selection relation between the different stakeholders. The standard can be used to create software requirements specifications directly or can be used as a model for defining the system requirements.

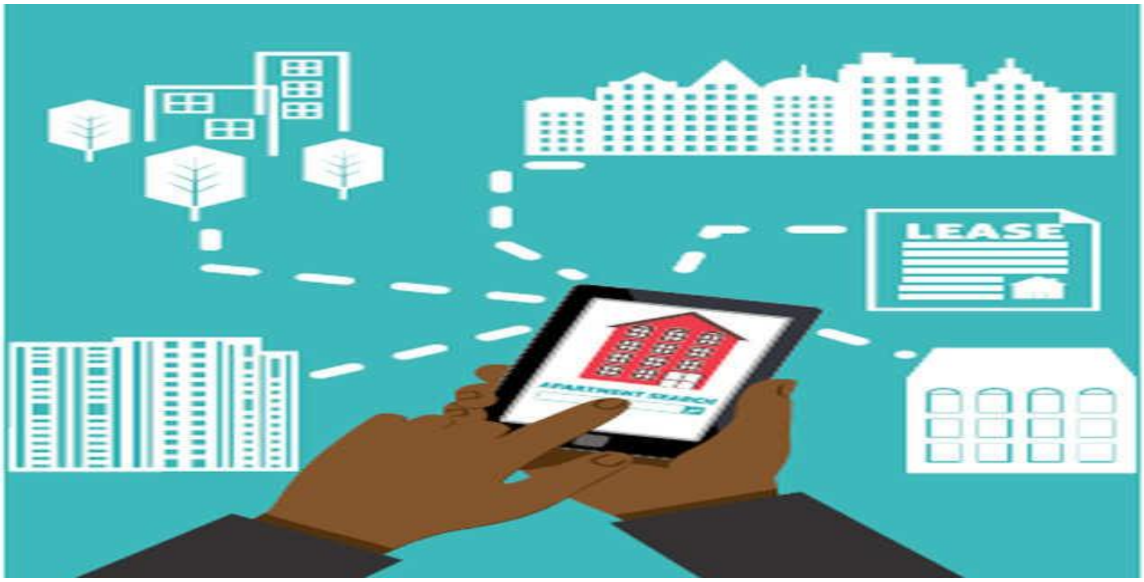
1.3 Stakeholders

There are two types of stakeholders in my total "Smart Hostel Finder" system. Such as:

Manager/Hostel	Hostel manager can register with their phone number, hostel name, location, food menu. They can update their hostel information.
Client/Normal User	User can registered my system, they find hostel with specific location, give review and read review of other user. They can contact with hostel manager and see the hostel information.

1.4 Proposed System Model

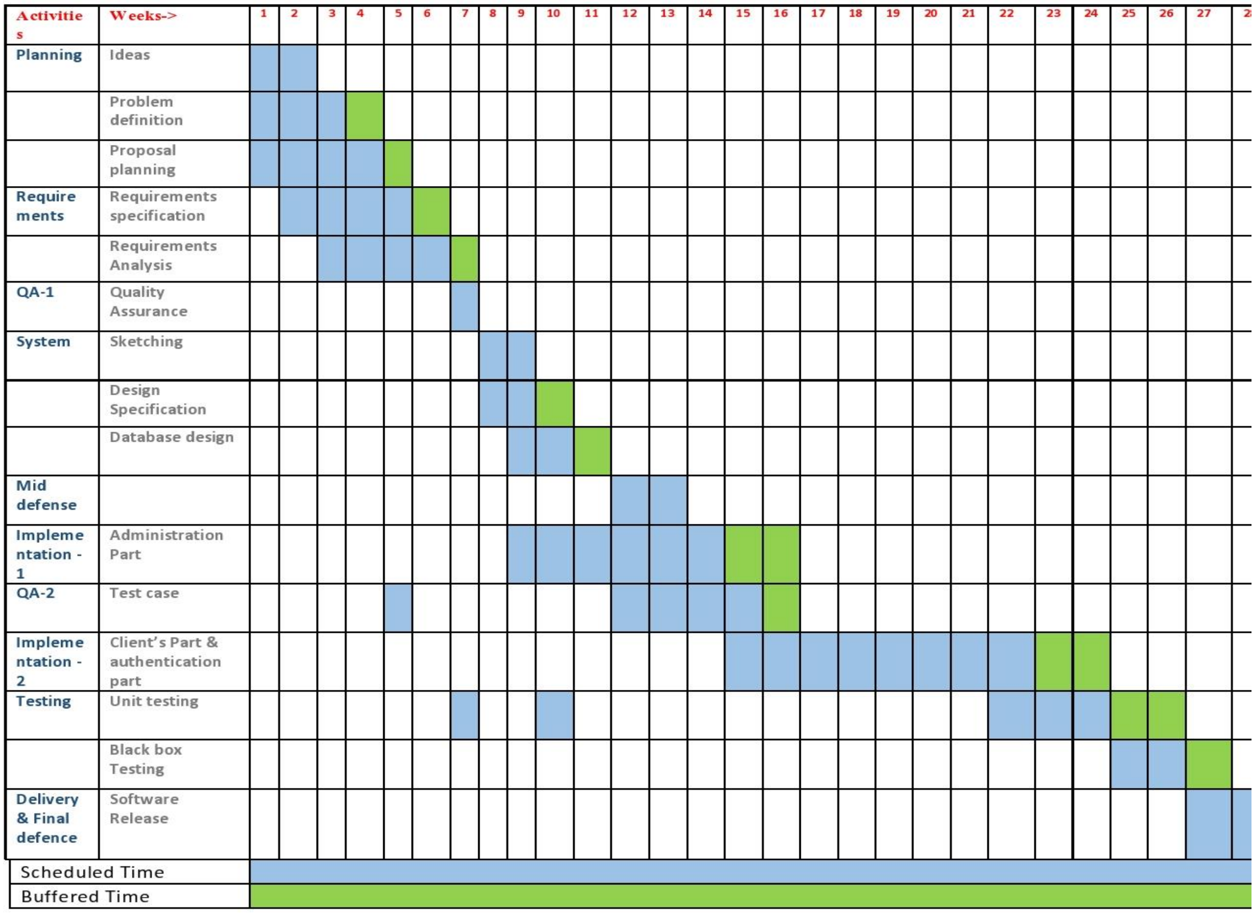
Before going to develop a system it is very important to have a system model. I have find system model. This model proposed system in brief.



1.5 Project Schedule

We need to prepare a scheduling plan to complete the project on time. It also refer to make communication with what task need to get done within timeframe.

1.5.1 Gantt Chart



Activities	Weeks->	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	2
Planning	Ideas	🟦	🟦																										
	Problem definition	🟦	🟦	🟦	🟩																								
	Proposal planning	🟦	🟦	🟦	🟦	🟩																							
Requirements	Requirements specification		🟦	🟦	🟦	🟦	🟩																						
	Requirements Analysis			🟦	🟦	🟦	🟦	🟩																					
QA-1	Quality Assurance							🟦																					
System	Sketching								🟦	🟦																			
	Design Specification								🟦	🟦	🟩																		
	Database design									🟦	🟦	🟩																	
Mid defense													🟦	🟦															
Implementation - 1	Administration Part									🟦	🟦	🟦	🟦	🟦	🟦	🟩	🟩												
QA-2	Test case					🟦							🟦	🟦	🟦	🟦	🟩												
Implementation - 2	Client's Part & authentication part															🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟦	🟩	🟩				
Testing	Unit testing							🟦			🟦												🟦	🟦	🟦	🟩	🟩		
	Black box Testing																									🟦	🟦	🟩	
Delivery & Final defence	Software Release																											🟦	🟦

Scheduled Time	🟦
Buffered Time	🟩

1.5.2 Release Plan or Milestone

Activities	Duration in week	Total week
Brainstorming	Week no. (1,2)	2
Problem identification	Week no. (1,2,3,4)	4
Requirement specification	Week no. (2,3,4,5)	4
Requirement analysis	Week no. (3,4,5,6)	4
Sketching	Week no. (8,9)	2
Design specification	Week no. (8,9)	2
Database design	Week no. (9,10)	2
Update Hostel info and add room information	Week no. (9,10,11,12,13,14)	6
Quality assurance(1,2)	Week no. (7,12,13,14,15)	5
Test case	Week no. (12,13,14,15)	4
Impose case & demerits	Week no. (15,16,17,18,19,20,21,22)	8
Unit testing	Week no. (22,23,24)	3
Black-box testing	Week no. (25,26)	2
Software release	Week no. (27,28)	2

Chapter 2

Software Requirement Specification

2. Requirement Specification

The complete requirement specification based on the elicitation process is described in this section.

2.1 Functional Requirements:

Functional requirements refer to the functions which are mandatory to the system. Functional requirements must be able to perform on the software system. Every system must have some functional requirements. Now, I am going to mention functional requirements associating with my project.

2.1.1 Create Account for Login

Requirements 1	**Create Account.**
Description	After Click on sign in button if user don't have an account then there is two option for registration, one is for client and other one is for hostel manager.
Stakeholders	Client and manager.

2.1.2 Manage Account for Client and Manager

Requirements 2	**Manage Account.**
Description	After Complete sign in. Both user client and manager can successfully login and log out from the system.
Stakeholders	Client and manager.

2.1.3 Add hostel for client

Requirements 3	**Add hostel info.**
Description	Hostel manager can add hostel info and client can viewhostel information.
Stakeholders	Manager and client.

2.1.4 View Hostel info for client

Requirements 4	**View hostel info.**
Description	After Completing registration and login user(Client) can view the information of hostel.
Stakeholders	Client.

2.1.5 Manage hostel information

Requirements 5	**Manage Hostel info.**
Description	Hostel manager can inset update and delete his hostel information.
Stakeholders	Manager.

2.1.6 Manager add room information

Requirements 6	**Add room info.**
Description	After Complete sign in his own account manager can add room information..
Stakeholders	Manager.

2.1.7 View room information.

Requirements 7	**View room info.**
Description	After completing all sign in issue user can view room information..
Stakeholders	Client.

2.1.8 Manage room information by manager.

Requirements 8	**Manage room info.**
Description	After inserting room information he can update his information like rent, room range, quality etc.
Stakeholders	Manager.

2.1.9 Food information added by manager.

Requirements 9	**Add food info.**
Description	Manager can add his food menu of his hostel in my system. He also can update and delete his food information.
Stakeholders	Manager.

2.1.10 View food information for both Manager and client.

Requirements 10	**View food info.**
Description	After adding food information by hostel manager client(user) and manager both can see the food item and food menu.
Stakeholders	Manager and client.

2.1.11 Managing food information by manager.

Requirements 10	**Manage food info.**
Description	After adding food information by hostel manager he can change his hostel food menu. He can update and delete his information.
Stakeholders	Manager.

2.1.12 Review views by client and manager.

Requirements 12	**View review.**
Description	Client and manager can view review. For this user and client have to sign in the system with their registered accounts.
Stakeholders	Client and manager.

2.1.13 Comment submitted by client.

Requirement 13	**Comment.**
Description	Client can share their experience with above hostel by giving their comment or we can say giving rating. Only user(Client) can comment about hostel.
Stakeholders	Client.

2.2 Data Requirements:

For defining data requirements, we need to build the model. For my application maximum data would be loaded from client and hostel employee(Hostel manager). And for that purpose I need to focus on some major points.

Such as:

- Types of entity of the system
- Quantity of Hostel
- Room availability
- Quantity of data
- Hostels service elements.

2.3 Performance Requirements:

It is very important to maintain performance of any software system. To ensure performance, we need to maintain some steps. Now, I will explain some perspective by which we are going to enhance the performance of my project.

2.3.1 Speed & Latency Requirements:

Speed and latency requirements must be ensured while retrieving data from the cloud server.

SLR-1	**Search result must be faster**
Description	When client search for an hostel by giving his location and month, then the result must show within a second.
Stakeholders	Client and manager

2.3.2 Precision & Accuracy Requirements:

Results that is to be shown to the end user is need to be accurate. Because, wrong information might be ruined the whole process.

PAR-1	**Search result must be accurate**
Description	When client or manager search anything the accurate result must show , with inserted information.
Stakeholders	Client and manager.

2.3.3 Capacity Requirements:

The developed system by us must be capable to handle user data, provide accurate information, handling database, manage http request etc.

CR-1	**The system will handle many data**
Description	The system need to handle many data every moment.
Stakeholders	Client and manager

2.4 Dependability Requirements:

The term dependability is measured based on four dimensions. Such as:

- Availability
- Reliability
- Safety
- Security

If we want to say that my application system is dependable then it must fulfill the four dimensions. But there are other tasks. Like there is no way to make mistakes or my system should have the ability to detect and then remove errors. Besides that, it is also very important to limit the damage which might be caused by system failure.

2.4.1 Reliability & Availability Requirements:

Now, I will mention requirement which is related to reliability and availability.

RAR-1	**The system must be available on 24 X 7**
Description	My system must be available all day long, every day in a week • The system must be updated regularly
Stakeholders	Client, Manager.

2.4.2 Robustness or Fault-Tolerance Requirements:

To ensure robustness and fault-tolerance facilities to the end users, it is urgent to ensure 0% crush. Moreover, it must show accurate results.

RFT-1	**The system handles Clients and manager access without system errors**
Description	Not more than 3 Client and manager might hit my application system at a time. All their requests must be handled without any fault.
Stakeholders	Client and manager.

2.4.3 Safety-Critical Requirements:

There are no safety-critical requirements in my project.

2.5 Maintainability & Supportability Requirements:

It is very important to provide after service or support to the end users.

2.5.1 Maintainability Requirements:

Not available for this project.

2.5.2 Supportability Requirements:

Supportability requirements may have related to some extends. Like:

- Testability
- Maintainability
- Configurability
- Serviceability
- Install ability

My application meets all of the above requirements related to supportability.

2.6 Security Requirements:

Making system security as a requirement is very important. System security requirements should be its functional requirement. System security enforces security of an application system.

Functionality related to system security can either be directly tested or observed. Some security related requirements is given below:

- Signing in as client and manager.
- Signing out as client and manager.
- Handling encrypted passwords.

While accessing to the system, each and every user must provide a central authentication mechanism. There is also a process to prevent entering into the system by ensuring hashed password for the unauthenticated users.

2.6.1 Access Requirements:

For accessing to my application system, there remain some authentication and authorization techniques. And every module of my system will provide it. Now I will provide an explanation below.

AR-1	**Application provides security mechanism**
Description	Every module is designed in such a way that it only gives access to the authorized and authenticated users.
Stakeholders	Client and manager.

2.6.2 Integrity Requirements:

Integrity requirements refer to a security system which ensures an expectation of data quality. It also ensures that all data of the system would never be exposed to the malicious modification or accidental destruction. For that reason, I will store my user passwords as encrypted format which is impossible to decrypt. It is also called hashed password.

2.6.3 Privacy Requirements:

It is very important to ensure privacy of the system users. Privacy requirements enhance to protect stakeholder’s privacy. In this way, all data or a partial part of data are going to be disclosed according to system’s privacy policy. To ensure privacy, the central database should be protected by the anonymous. Planners permitted to get access to those data which are being associated by them which can be ensured by the total log in system.

2.7 Usability and Human-Interaction Requirements:

The main target of developing any system is to make the system user friendly and easy to usable for the end users.

2.7.1 Ease of Use Requirements:

My application is easy to use and also easily understandable.

EUR-1	**Application must be usable for the end users**
Description	This app is enough usable to the client and manager by which he can operate this system easily.
Stakeholders	Client and manager.

2.7.2 Personalization and Internationalization Requirements:

There are not any personalization and internationalization requirements in my system. This maiden version of my application is only be operated by a Specific company.

2.7.3 Understand ability and Politeness Requirements:

It is already said that the application which I am going to develop, is understandable enough. The system provides hints to users whether any error occurred or wrong. By reading those errors client and manager can be able to operate the system easily.

2.7.4 Accessibility Requirements: Documentations are mainly two types. One is internal documentation which is generally written by the application engineers. It is prepared to make development life cycle easier for the system engineers or system analysts.

UDR-1	**The system engineer documentation**
Description	To develop my application named Smart Hostel Finding, firstly I have make a system analysis as well as documentation.
Stakeholders	System analysts or software developers

2.7.6 Training Requirements:

Training requirements involved in after service of any application. It is very necessary to properly train up end users to the system so that they would be capable to operate easily. After launching the full package to the market, firstly I provide training to the different end users like Client, Hostel manager, Registered client, registered manager, Unregister manager and client.

2.8 Look and Feel Requirements:

Look and feel requirement mainly refers how the system will look like and how the user interface of my system will display to the users.

2.8.1 Appearance Requirements:

All users must know which input fields are required and which are not. For that reason, I will use labels for all input fields. Input fields might be text type, radio, checkbox etc.

AR-1	**Labels of mandatory fields must be bold**
Description	The mandatory field’s label must be bold and all input fields must have placeholder to make it easier for the users.
Stakeholders	All users of the system

2.9 Operational and Environmental Requirements:

Operational and environmental requirement refers to the capabilities, performance measurements, process, measurements of effectiveness, measurements of performance, measures of sustainability, measurements of technical performances etc.

2.9.1 Expected Physical Requirements:

There are no expected physical requirements in my system.

2.9.2 Requirements for Interfacing with Adjacent Systems:

There are no requirements for interfacing with adjacent system for my project. Because it is desktop and web based system.

2.9.3 Release Requirements:

There are no specific release requirements in my system. Because it is an live project.

2.10 Legal Requirements:

Legal requirements normally refer to the terms and conditions or privacy policy of any organizations. The terms and condition of our application is that, no third party software or persons are allowed to engage to use my data for their business purpose.

2.10.1 Compliance Requirements:

There are no specific compliance requirements for my system.

2.10.2 Standards Requirements:

As I said above there are no specific standards requirements for my system.

Chapter 3

System Analysis

3.1 Use Case Diagram

I have use case diagram. And there are two actors. Each actor plays different role. And those are already indicated to this use case diagram. This diagram will clarify our system in brief.

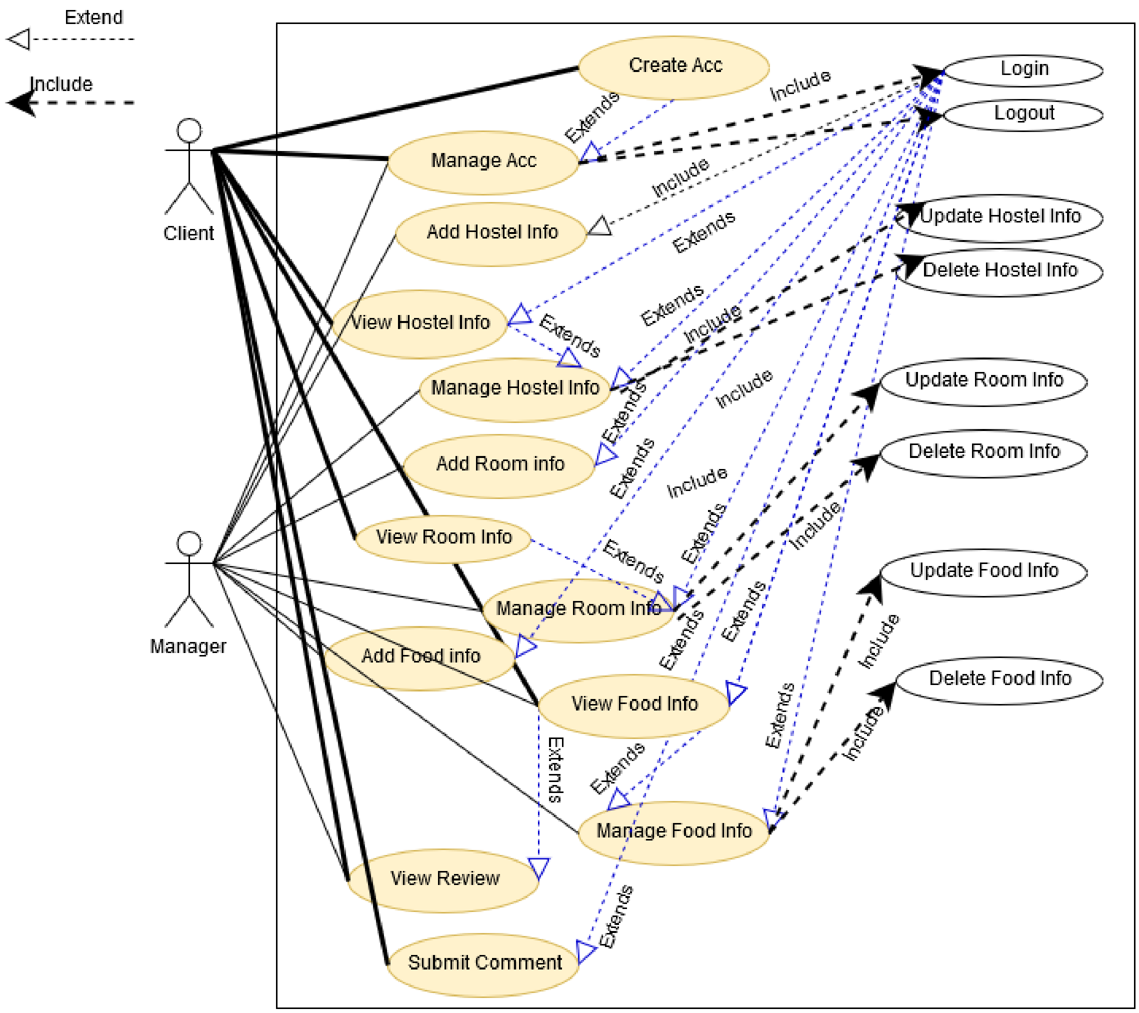


Figure 3.1: Use Case diagram for "Smart Hostel Finder"

3.1.1 Create Account.

<table>
<tr><td><b>Use Case</b></td><td colspan="2"><b>Create Account.</b></td></tr>
<tr><td><b>Goal</b></td><td colspan="2">Client and hostel manager can log in to the system if they don’t have account, they have to choose create account option. There is two options for create account both client and manager.</td></tr>
<tr><td><b>Preconditions</b></td><td colspan="2">• Client and hostel manager must open the system.
• Client and hostel manager must go to the create account option.</td></tr>
<tr><td><b>Success End Condition</b></td><td colspan="2">• Client and hostel manager can open the system successfully.
• Client and hostel manager can create account successfully.</td></tr>
<tr><td><b>Failed End Condition</b></td><td colspan="2">• T&A and planning manager can’t login successfully.
• T&A and planning manager can’t logout successfully.</td></tr>
<tr><td><b>Primary Actors:</b>

<b>Secondary Actors:</b></td><td colspan="2">Client and hostel manager

System</td></tr>
<tr><td><b>Trigger</b></td><td colspan="2">Client and hostel manager must sign in to the system for create account.</td></tr>
<tr><td rowspan="7"><b>Description / Main Success Scenario</b></td><td><b>Step</b></td><td><b>Action</b></td></tr>
<tr><td>1</td><td>Client and hostel manager clicks to the sign in option</td></tr>
<tr><td>2</td><td>Client and hostel manager gets login form.</td></tr>
<tr><td>3</td><td>Client and hostel manager enters right username password and other requirements for create account</td></tr>
<tr><td>4</td><td>Client and hostel manager finally clicks create button.</td></tr>
<tr><td>5</td><td>Client and hostel manager successfully create account.</td></tr>
<tr><td>6</td><td>Client and hostel manager clicks on submit button and successfully create account</td></tr>
<tr><td rowspan="4"><b>Alternative Flows</b></td><td><b>Step</b></td><td><b>Branching Action</b></td></tr>
<tr><td>1</td><td>Client and manager doesn’t sign in system</td></tr>
<tr><td>2</td><td>Client and hostel manager doesn’t get create account form.</td></tr>
<tr><td>3</td><td>Client and hostel manager can’t give right username password</td></tr>
</table>

3.1.2 Manage Account.

Use Case	**Manage Account.**	
Goal	Client and hostel manager can log in to the system if they have account, they have to choose sign in option. There is one options for login account both client and manager.	
Preconditions	• Client and hostel manager must have user name and password. • Client and hostel manager must go to the log in account option.	
Success End Condition	• Client and hostel manager can open log in option in system successfully. • Client and hostel manager can login and log out successfully.	
Failed End Condition	Client and hostel manager can't login successfully. • Client and hostel manager can't logout successfully.	
Primary Actors: **Secondary Actors:**	Client and hostel manager System	
Trigger	Client and hostel manager must have a valid account for log in to the system.	
Description / Main Success Scenario	**Step**	**Action**
	1	Client and hostel manager clicks to the login option
	2	Client and hostel manager gets login form.
	3	Client and hostel manager enters right username password for log in
	4	Client and hostel manager finally clicks sign in button.
	5	Client and hostel manager successfully logged in system.
	6	Client and hostel manager clicks on button and successfully log in and log out from the system
Alternative Flows	**Step**	**Branching Action**
	1	Client and manager doesn't logged in the system
	2	Client and hostel manager cant login with their account
	3	Client and hostel manager can't log out from the system

3.1.3 Add Hostel Info.

Use Case	**Add hostel info.**	
Goal	Hostel manager can log in the system and add his hostel information	
Preconditions	• Client and hostel manager must have an valid account for this • hostel manager must go to the add hostel info option	
Success End Condition	• Hostel manager can add his hostel information in the system successfully.	
Failed End Condition	• Client and hostel manager can't add hostel information successfully.	
Primary Actors: **Secondary Actors:**	Hostel manager System	
Trigger	Hostel manager must have a valid account for add hostel info	
Description / Main Success Scenario	**Step**	**Action**
	1	Client and hostel manager clicks to the login option
	2	Client and hostel manager gets login form.
	3	Client and hostel manager enters right username password for log in
	4	Client and hostel manager finally clicks sign in button.
	5	Client and hostel manager successfully logged in system.
	6	Client and hostel manager clicks on button and successfully log in and log out from the system
Alternative Flows	**Step**	**Branching Action**
	1	Manager doesn't add hostel info in the system
	2	Manager cant login with his account
	3	Hostel manager can't log out from the system

3.1.4 View Hostel Info.

Use Case	**View hostel info.**	
Goal	Client can view the hostel information	
Preconditions	• Client and hostel manager must have an valid account for this • Client must go to the view hostel info option	
Success End Condition	• Client can see the all hostel information from the system successfully.	
Failed End Condition	Client cant view hostel information	
Primary Actors: **Secondary Actors:**	Client System	
Trigger	client must have a valid account and sign in for view hostel info	
Description / Main Success Scenario	**Step**	**Action**
	1	Client have to clicks on the view hostel info option
	2	Client got hostel info form.
	3	Client enters right username password for log in
	4	Client clicks on view hostel info button and successfully see the info
Alternative Flows	**Step**	**Branching Action**
	1	Client doesn’t see hostel info in the system
	2	Client cant login with his account
	3	Client can’t log out from the system

3.1.5 Manage Hostel Info.

Use Case	**Manage hostel info.**	
Goal	Manager can manage his hostel information	
Preconditions	• Hostel manager must have an valid account for this • Manager must go to the manage hostel info option	
Success End Condition	• Manager can add his hostel information successfully and updated his hostel info.	
Failed End Condition	Manager cant manage his hostel information	
Primary Actors: **Secondary Actors:**	Manager System	
Trigger	Manager must have a valid account and sign in for manage hostel info	
Description / Main Success Scenario	**Step**	**Action**
	1	Manager have to clicks on the manage hostel info option
	2	Manager got manage hostel info option
	3	Manager inserts his hostel information
	4	manager clicks on add hostel info button and successfully inserts the info and updated
Alternative Flows	**Step**	**Branching Action**
	1	Manager doesn’t see add hostel info in the system
	2	Manager cant Update his hostel information
	3	Manager cant delete his hostel information

3.1.6 Add Room Info.

Use Case	**Add Room info.**	
Goal	Hostel manager can log in the system and add his room information	
Preconditions	• Hostel manager must have an valid account for this • Hostel manager must go to the add room info option	
Success End Condition	• Hostel manager can add his room information in the system successfully.	
Failed End Condition	• Hostel manager can't add room information successfully.	
Primary Actors: **Secondary Actors:**	Hostel manager System	
Trigger	Hostel manager must have a valid account for add room info	
Description / Main Success Scenario	**Step**	**Action**
	1	Hostel manager clicks to the login option
	2	Hostel manager click the add room info option
	3	Hostel manager clicks on button and successfully add his room information
Alternative Flows	**Step**	**Branching Action**
	1	Manager doesn't add room info in the system
	2	Manager cant login with his account
	3	Hostel manager can't log out from the system

3.1.7 View Room Info

Use Case	**View Room info.**	
Goal	Client can view the room information	
Preconditions	• Client must have an valid account for this • Client must go to the view room info option	
Success End Condition	• Client can see the all rooms information from the system successfully.	
Failed End Condition	Client cant view room information	
Primary Actors: **Secondary Actors:**	Client System	
Trigger	client must have a valid account and sign in for view room info	
Description / Main Success Scenario	**Step**	**Action**
	1	Client have to clicks on the view room info option
	2	Client see the room information
	3	Client enters right username password for log in
	4	Client clicks on view room info button and successfully see the info
Alternative Flows	**Step**	**Branching Action**
	1	Client doesn’t see room info in the system
	2	Client cant login with his account
	3	Client can’t log out from the system

3.1.8 Manage Room Info.

Use Case	**Manage Room info.**	
Goal	Manager can manage his room information	
Preconditions	- Hostel manager must have an valid account for this - Manager must go to the manage room info option	
Success End Condition	- Manager can add his room information successfully and updated his room info.	
Failed End Condition	Manager cant manage his room information	
Primary Actors: **Secondary Actors:**	Manager System	
Trigger	Manager must have a valid account and sign in for manage hostel info	
Description / Main Success Scenario	**Step**	**Action**
	1	Manager have to clicks on the manage room info option
	2	Manager got manage room info option
	3	Manager inserts his hostel information
	4	manager clicks on add room info button and successfully inserts the info and updated
Alternative Flows	**Step**	**Branching Action**
	1	Manager doesn’t see add room info in the system
	2	Manager cant Update his room information
	3	Manager cant delete his room information

3.1.9 Add Food Info.

Use Case	**Add Food info.**	
Goal	Hostel manager can log in the system and add his hostels food information	
Preconditions	• Hostel manager must have an valid account for this • Hostel manager must go to the add food info option	
Success End Condition	• Hostel manager can add his food information in the system successfully.	
Failed End Condition	• Hostel manager can't add food information successfully.	
Primary Actors: **Secondary Actors:**	Hostel manager System	
Trigger	Hostel manager must have a valid account for add food info	
Description / Main Success Scenario	**Step**	**Action**
	1	Hostel manager clicks to the login option
	2	Hostel manager click the add food info option
	3	Hostel manager clicks on button and successfully add his food information
Alternative Flows	**Step**	**Branching Action**
	1	Manager doesn't add food info in the system
	2	Manager cant login with his account
	3	Hostel manager can't log out from the system

3.1.10 View Food Info

Use Case	**View Food info.**	
Goal	Client and manager can view the food information	
Preconditions	• Client and manager must have an valid account for this • Client and manager must go to the view food info option	
Success End Condition	• Client and manager can see the all rooms information from the system successfully.	
Failed End Condition	Client and manager cant view food information	
Primary Actors: **Secondary Actors:**	Client and manager System	
Trigger	Client and manager must have a valid account and sign in for view food info	
Description / Main Success Scenario	**Step**	**Action**
	1	Client and manager have to clicks on the view food info option
	2	Client and manager see the food information
	3	Client and manager enters right username password for log in
	4	Client and manager clicks on view food info button and successfully see the info
Alternative Flows	**Step**	**Branching Action**
	1	Client and manager doesn't see food info in the system
	2	Client and manager cant login with his account
	3	Client and manager can't log out from the system

3.1.11 Manage Food Info.

Use Case	**Manage Food info.**	
Goal	Manager can manage his hostels food information	
Preconditions	• Hostel manager must have an valid account for this • Manager must go to the manage food info option	
Success End Condition	• Manager can add his food information successfully and updated his room info	
Failed End Condition	Manager cant manage his food information	
Primary Actors: **Secondary Actors:**	Manager System	
Trigger	Manager must have a valid account and sign in for manage food info	
Description / Main Success Scenario	**Step**	**Action**
	1	Manager have to clicks on the manage food info option
	2	Manager got manage food info option
	3	Manager inserts his food information
	4	manager clicks on add food info button and successfully inserts the info and updated
Alternative Flows	**Step**	**Branching Action**
	1	Manager doesn’t see add food info in the system
	2	Manager cant Update his food information
	3	Manager cant delete his food information

3.1.12 View Review

<table>
<tr><th>Use Case</th><th colspan="2">View Review</th></tr>
<tr><td><b>Goal</b></td><td colspan="2">Client and hostel can view the review of the other people</td></tr>
<tr><td><b>Preconditions</b></td><td colspan="2">• Client and hostel manager must have an valid account for this
• Client and manager must go to the view review option</td></tr>
<tr><td><b>Success End Condition</b></td><td colspan="2">• Client and manager can see the all review from the system successfully.</td></tr>
<tr><td><b>Failed End Condition</b></td><td colspan="2">Client and manager cant view review information</td></tr>
<tr><td><b>Primary Actors:</b>
<b>Secondary Actors:</b></td><td colspan="2">Client and manager
System</td></tr>
<tr><td><b>Trigger</b></td><td colspan="2">client and manager must have a valid account and sign in for view review</td></tr>
<tr><td rowspan="5"><b>Description / Main Success Scenario</b></td><td><b>Step</b></td><td><b>Action</b></td></tr>
<tr><td>1</td><td>Client and manager have to clicks on the view review option</td></tr>
<tr><td>2</td><td>Client see the review</td></tr>
<tr><td>3</td><td>Client and manager enters right username password for log in</td></tr>
<tr><td>4</td><td>Client clicks on view review button and successfully see the info</td></tr>
<tr><td rowspan="4"><b>Alternative Flows</b></td><td><b>Step</b></td><td><b>Branching Action</b></td></tr>
<tr><td>1</td><td>Client and manager doesn’t see review in the system</td></tr>
<tr><td>2</td><td>Client and manager cant login with his account</td></tr>
<tr><td>3</td><td>Client and manager can’t log out from the system</td></tr>
</table>

3.1.13 Submit Comment

Use Case	Submit Comment	
Goal	Client can give his comment	
Preconditions	• Client must have an valid account for this • Client must go to the submit comment option	
Success End Condition	• Client can comment on the system successfully.	
Failed End Condition	Client cant give comment	
Primary Actors: **Secondary Actors:**	Client System	
Trigger	client must have a valid account and sign in for comment	
Description / Main Success Scenario	**Step**	**Action**
	1	Client have to clicks on the comment option
	2	Only client can comment
	3	Client enters right username password for log in
	4	Client clicks on comment button and successfully submit comment
Alternative Flows	**Step**	**Branching Action**
	1	Client doesn’t see comment option in the system
	2	Client cant login with his account
	3	Client can’t log out from the system

3.2 Activity Diagram

I have prepared some activity diagram according to my use case. These activity diagrams are properly referring the flow of the individual conditions of my project.

3.2.1 Create account

If user have account then direct going to login, otherwise click on create account option for account.

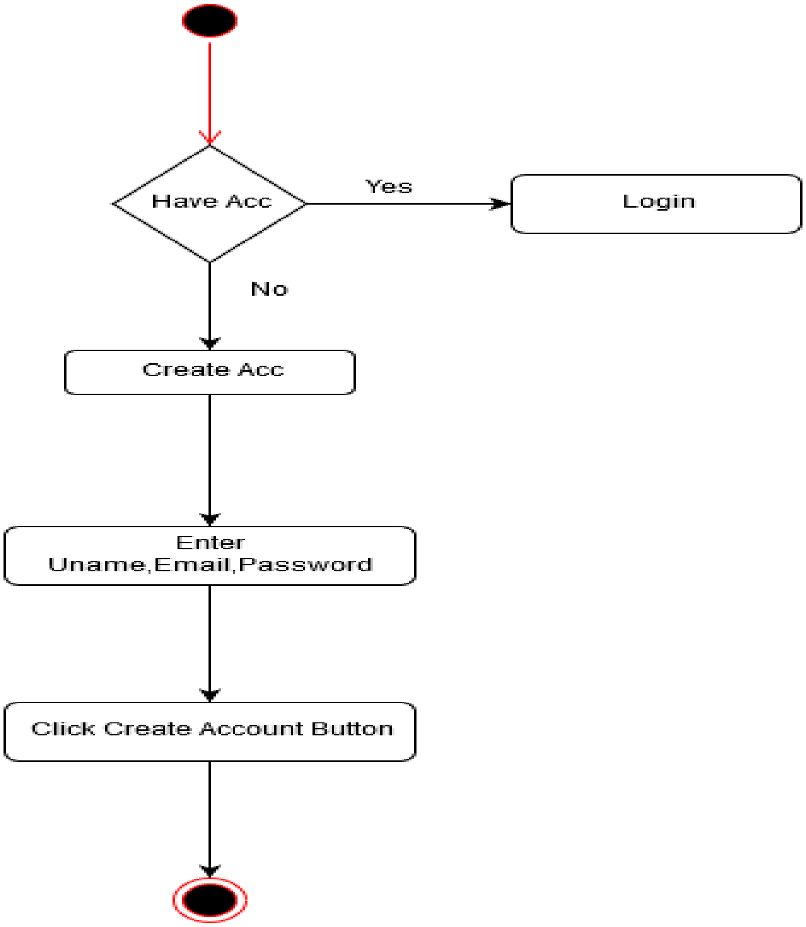


Figure 3.2.1: Activity diagram for Create Account

3.2.2 Manage account

Users already have an account then click on sign in option and click on enter username and password further works.

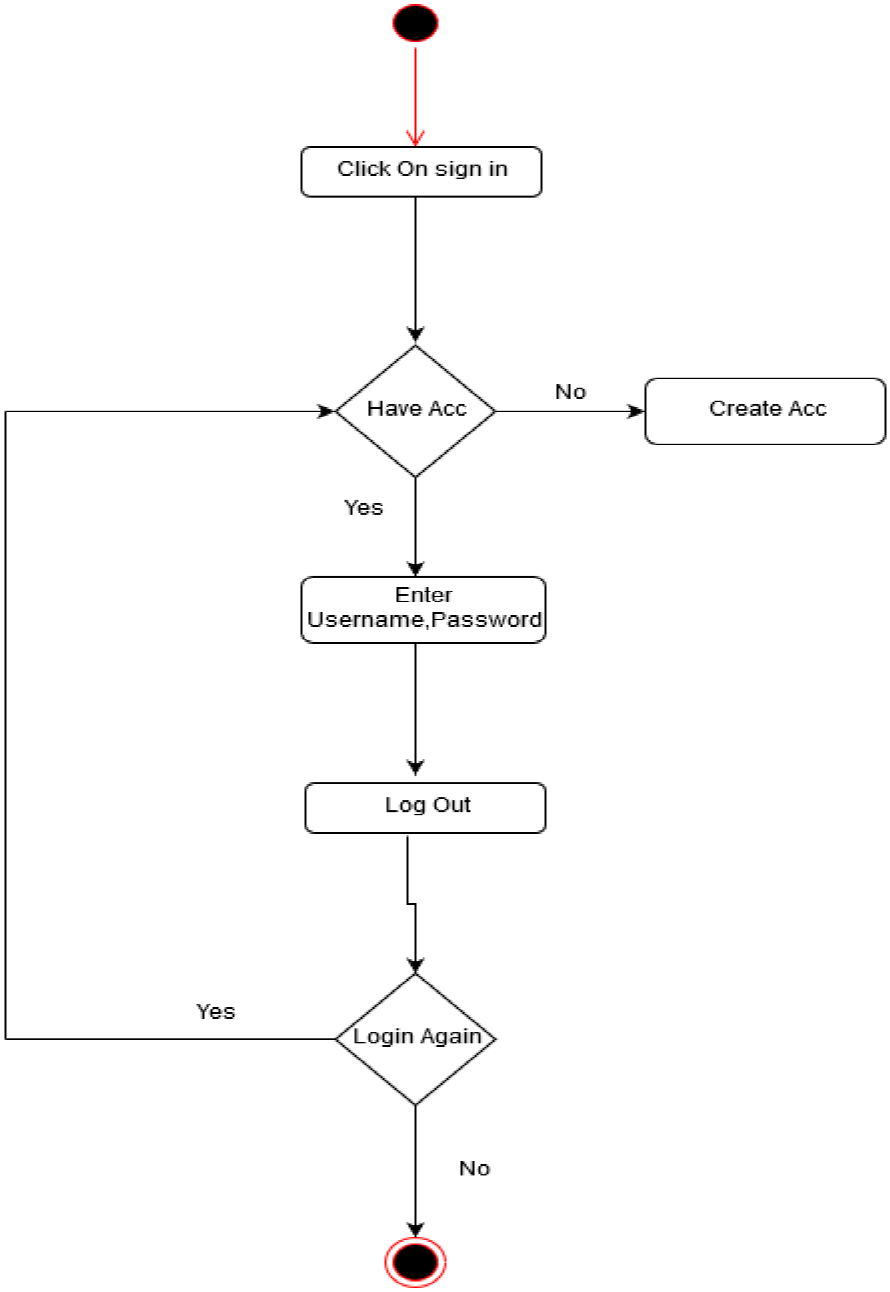


Figure 3.2.2: Activity diagram for Manage account

3.2.3 Add Hostel Info

Manager can add his hostel information b click on add hostel info and others option.

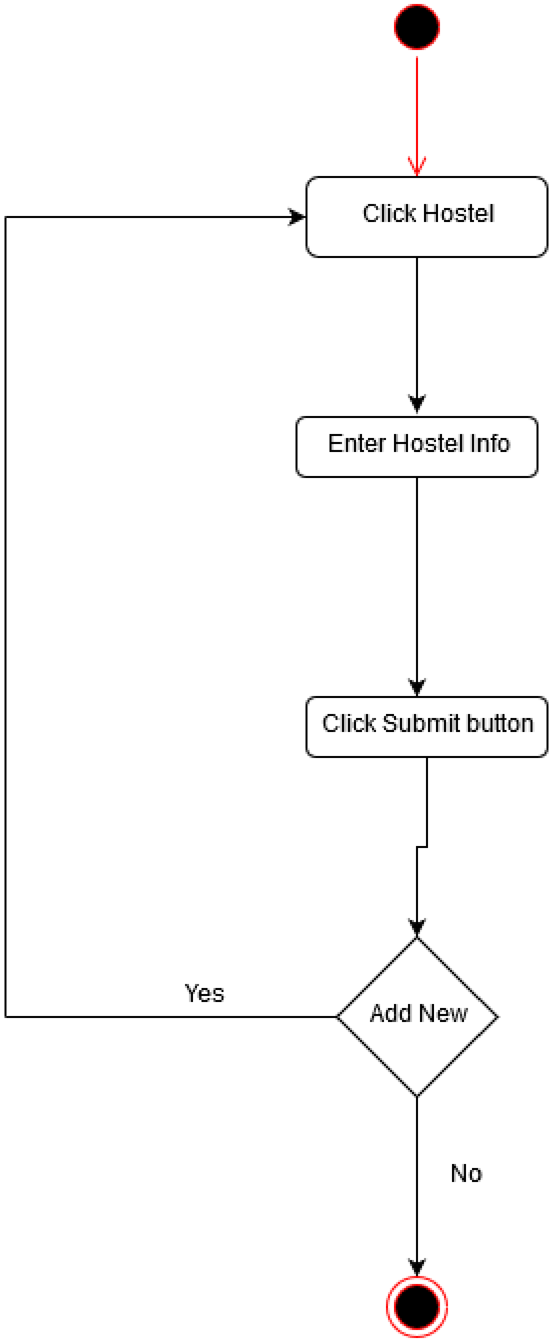


Figure 3.2.3: Activity diagram for Add Hostel Info

3.2.4 View Hostel Info

Users can view hostels information , if they have already registered accounts.

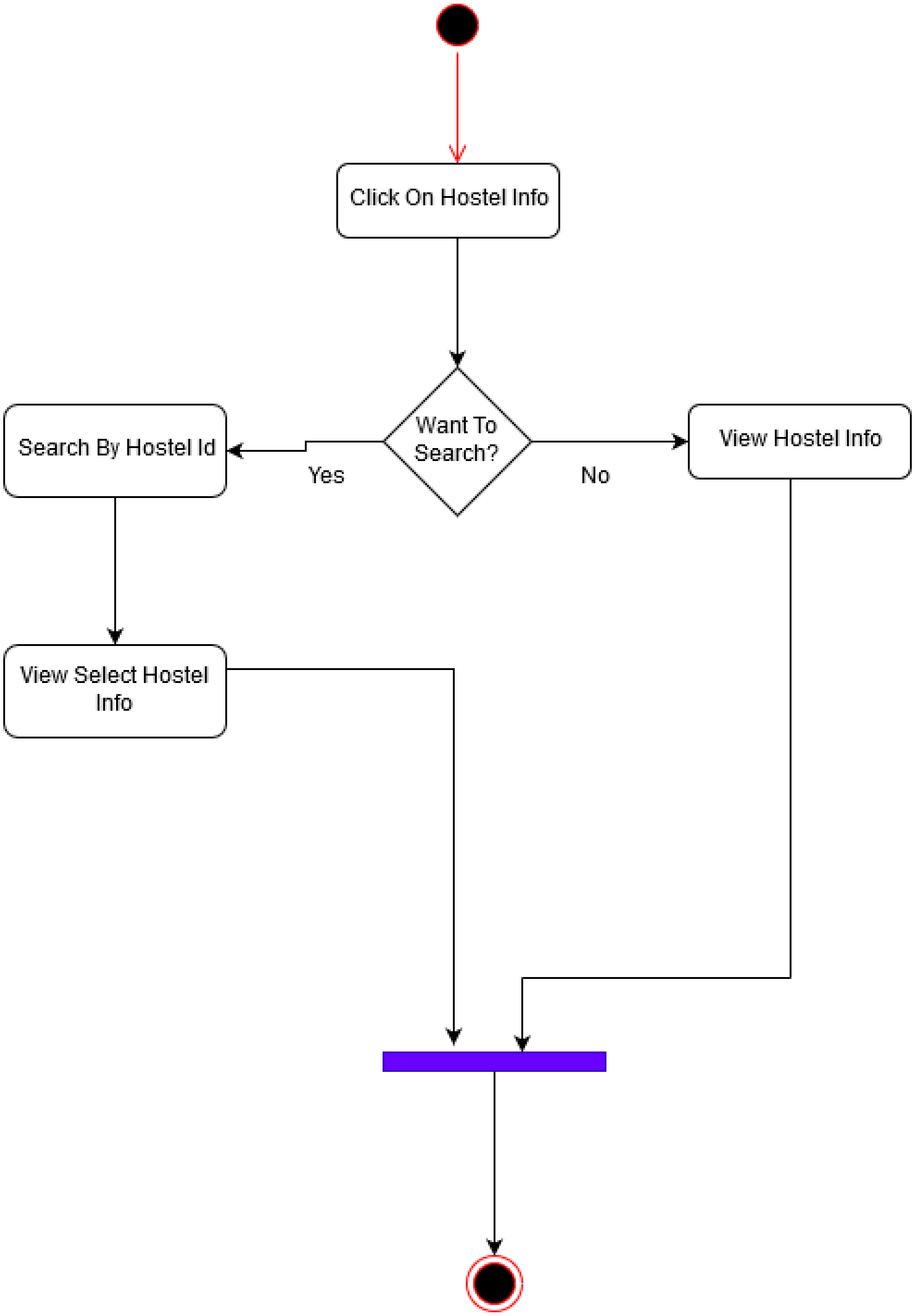


Figure 3.4: Activity diagram for view hostel

3.2.5 Manage Hostel Info

Hostel manager can update his uploaded hostel information to click on hostel.

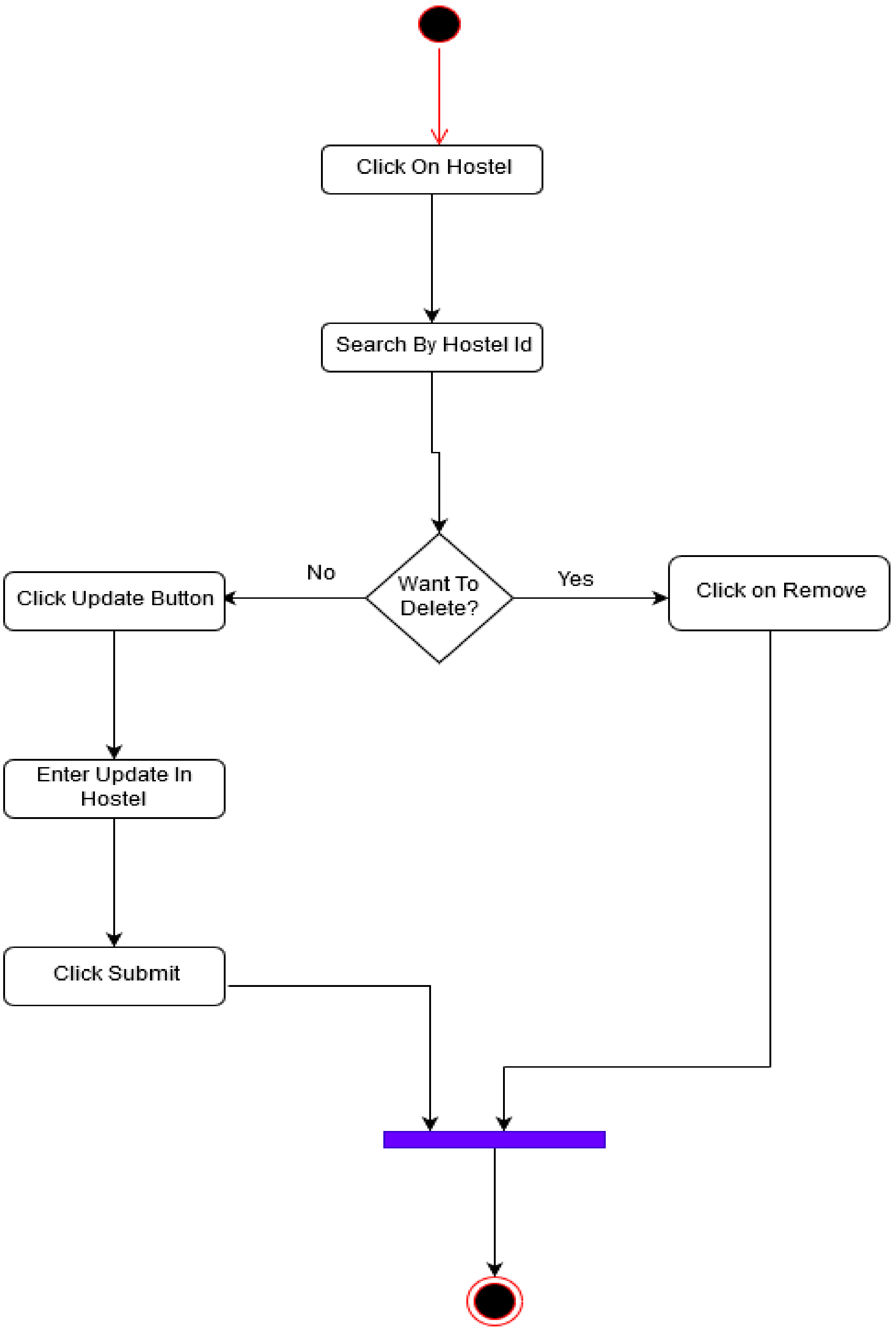


Figure 3.5: Activity diagram for Manage Hostel Info

3.2.6 Add Room Info

Hostel manager can add his hostel room information and pictures.

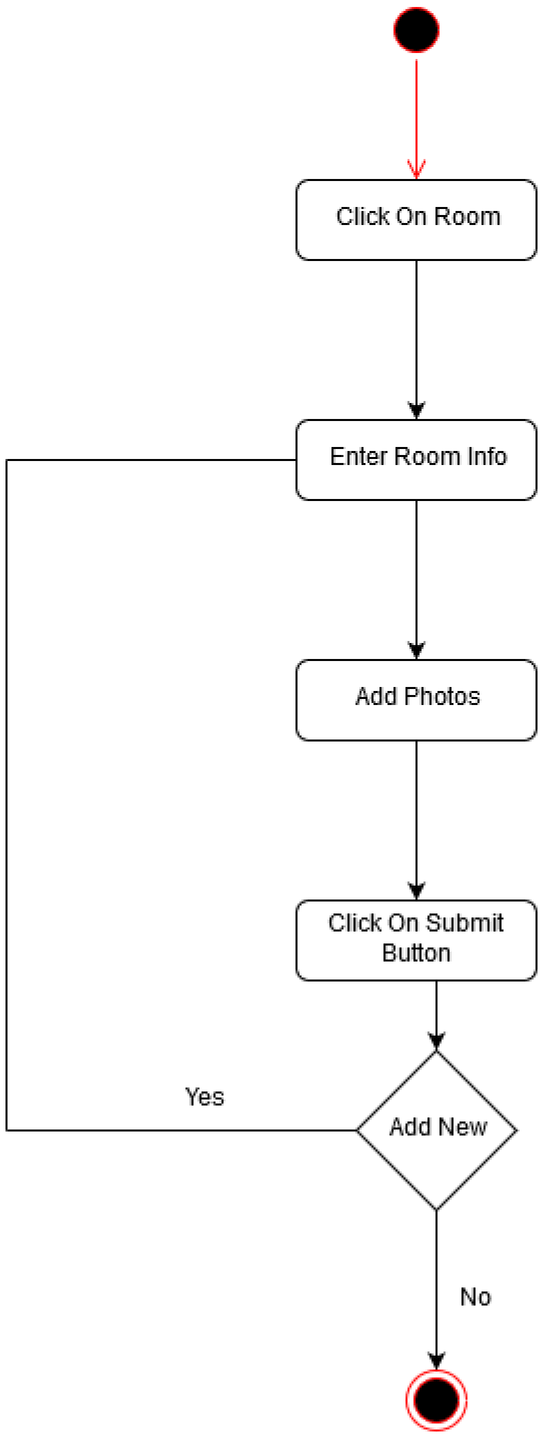


Figure 3.6: Activity diagram for Add Room Info

3.2.7 View Room Info

Users can views room information and photos of room by click on room info.

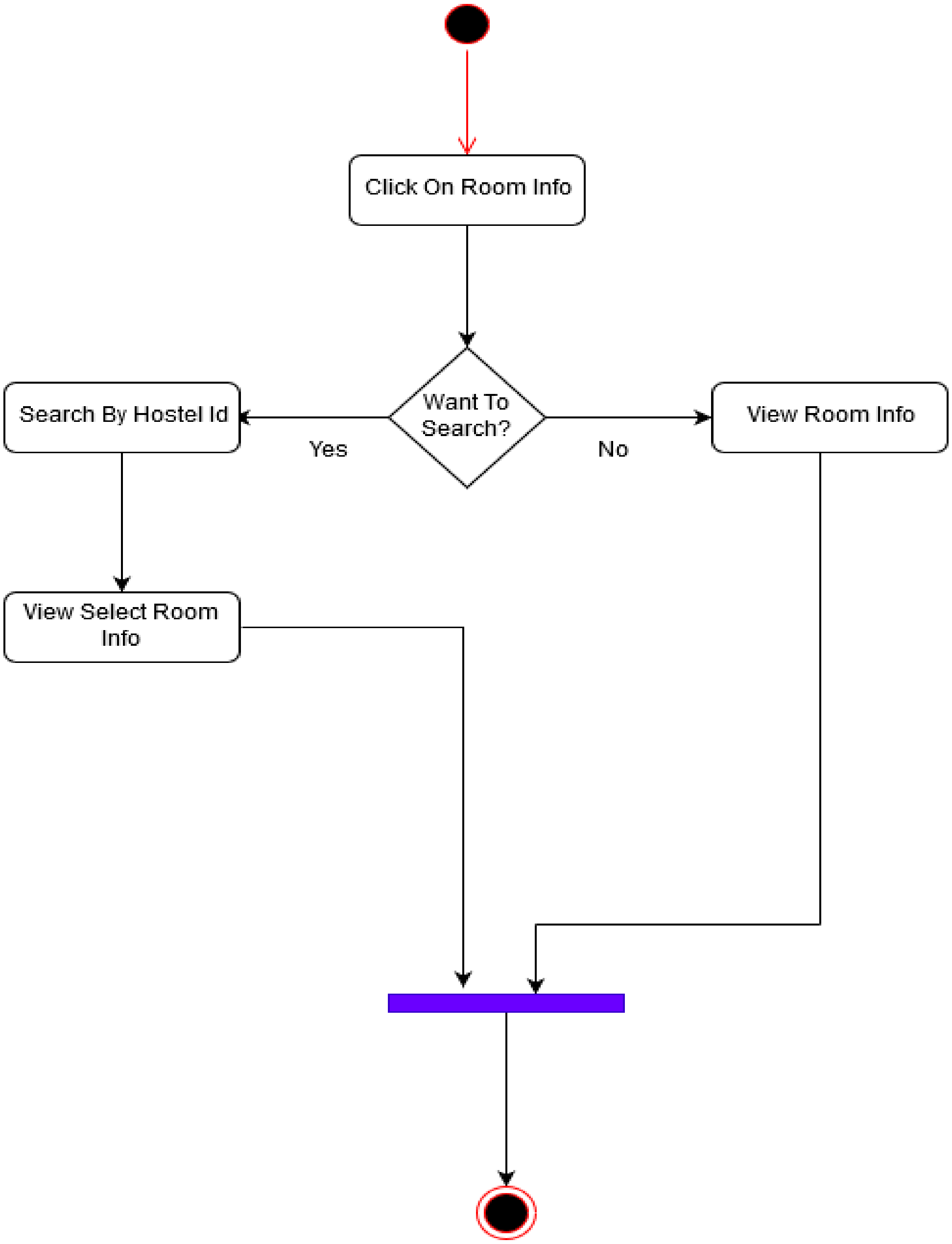


Figure 3.7: Activity diagram for view Room info

3.2.8 Manage Room Info

Hostel Managers can add and update his room information.

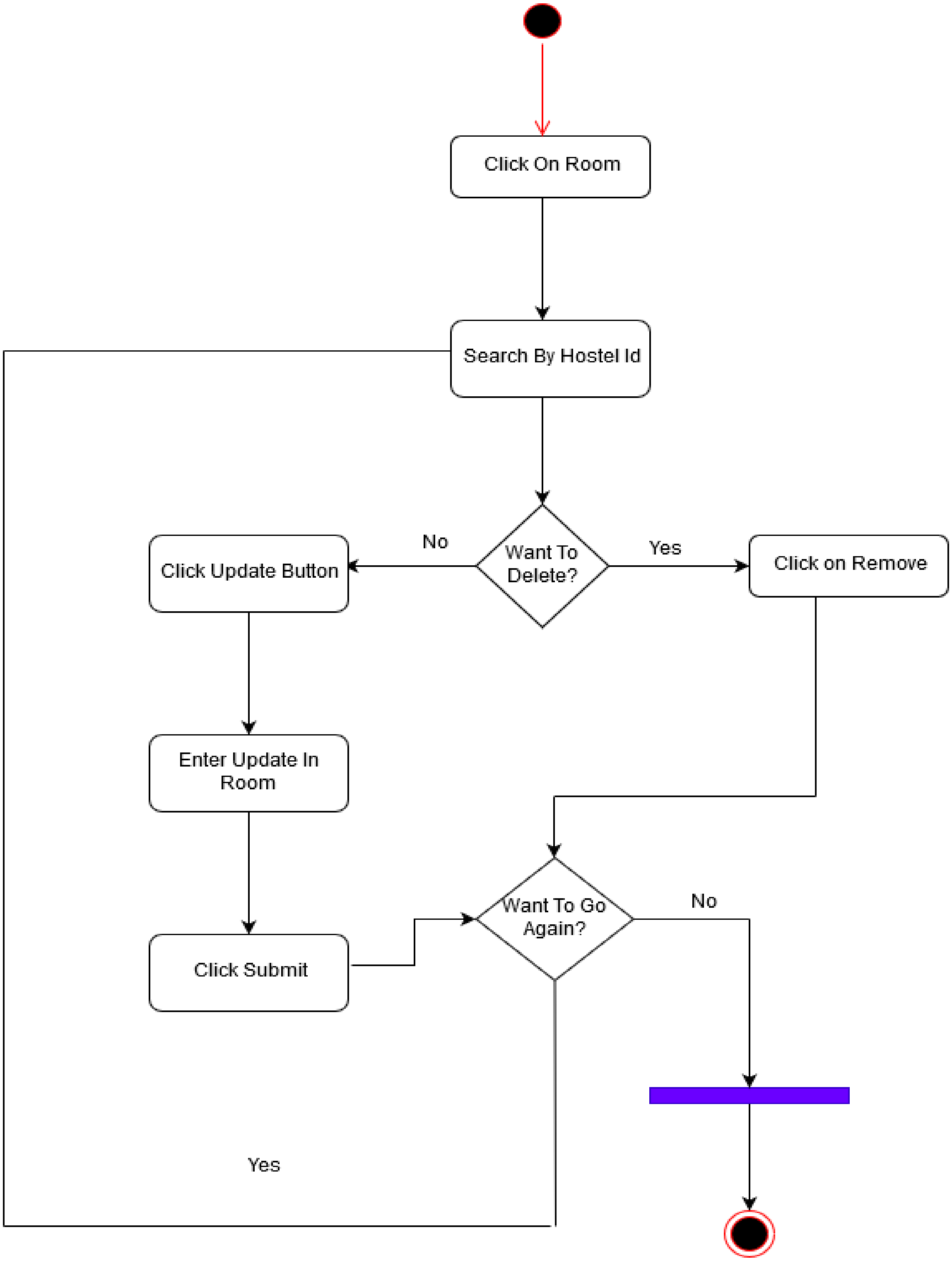


Figure 3.2.8: Activity diagram for Manage Room Info

3.2.9 Add Food Info

Managers can upload food info for available hostels.

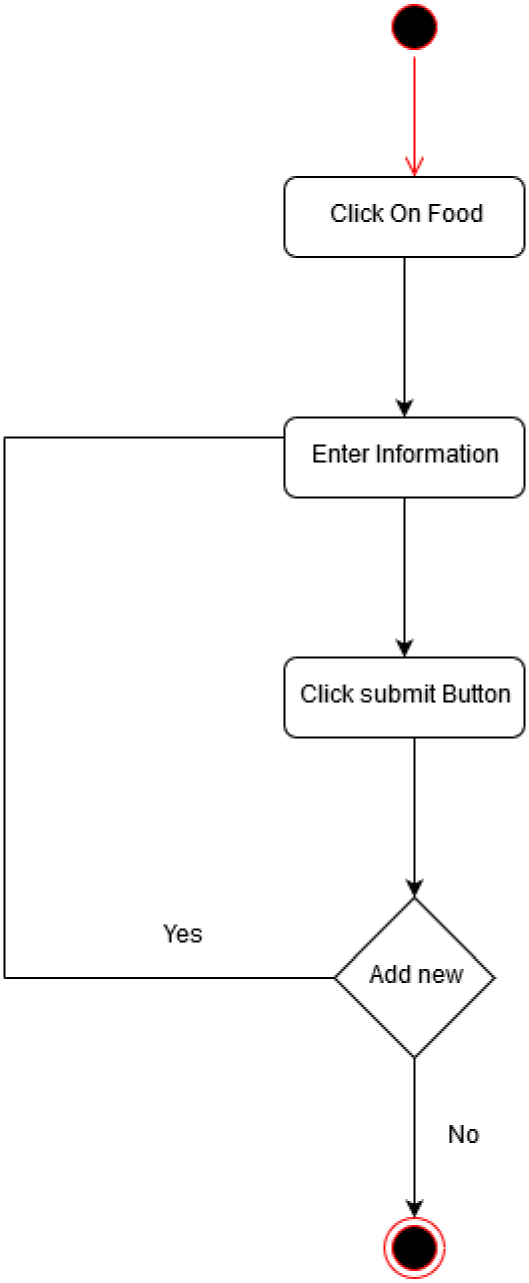


Figure 3.2.9: Activity diagram for Add Food Info

3.2.10 View Food Info

Client or user can see all type of food information.

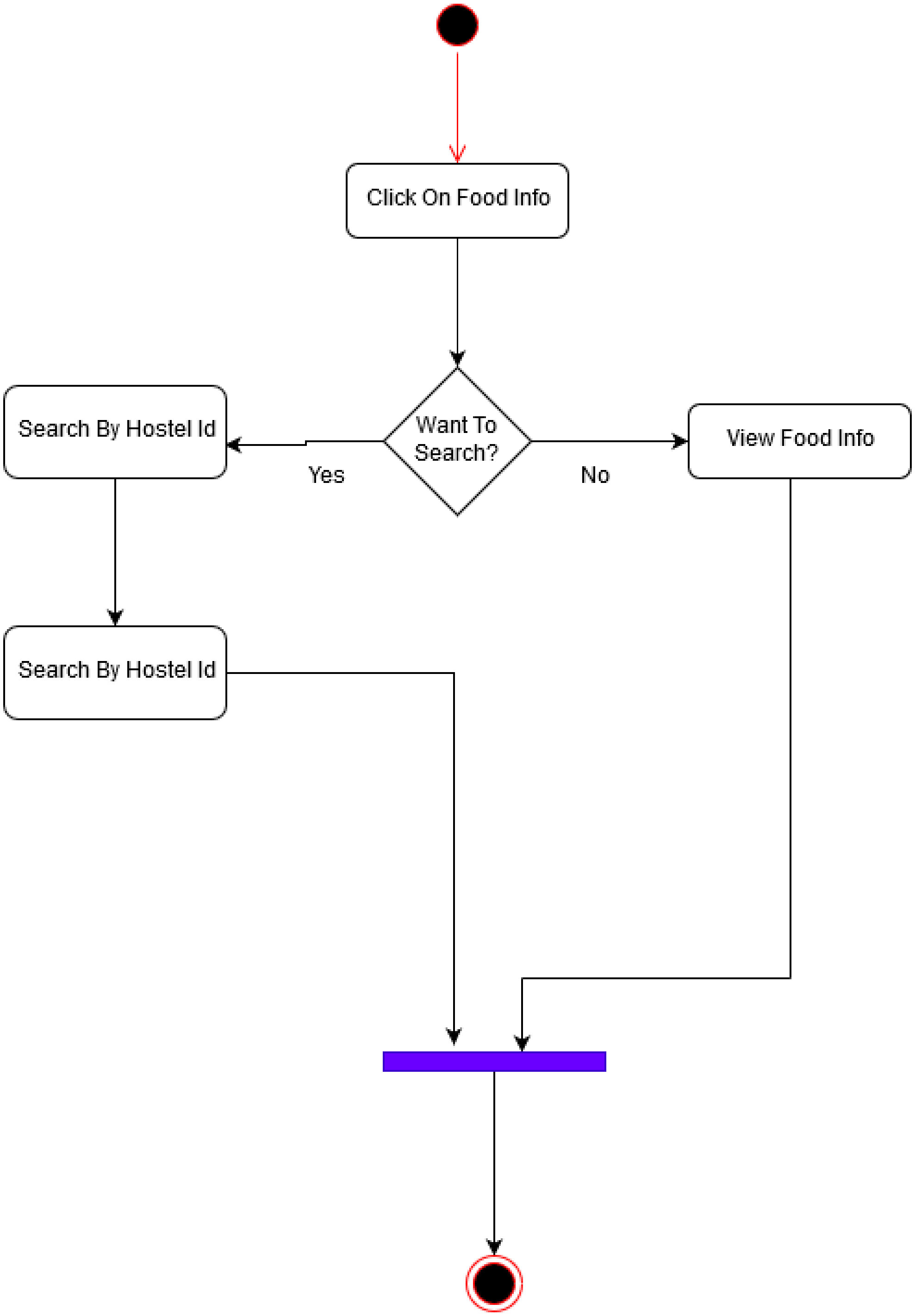


Figure 3.2.10: Activity diagram for view Food Info

3.2.11 Manage Food Info

Hostels Manager can manage his food info. He or she can insert, update his hostels food info.

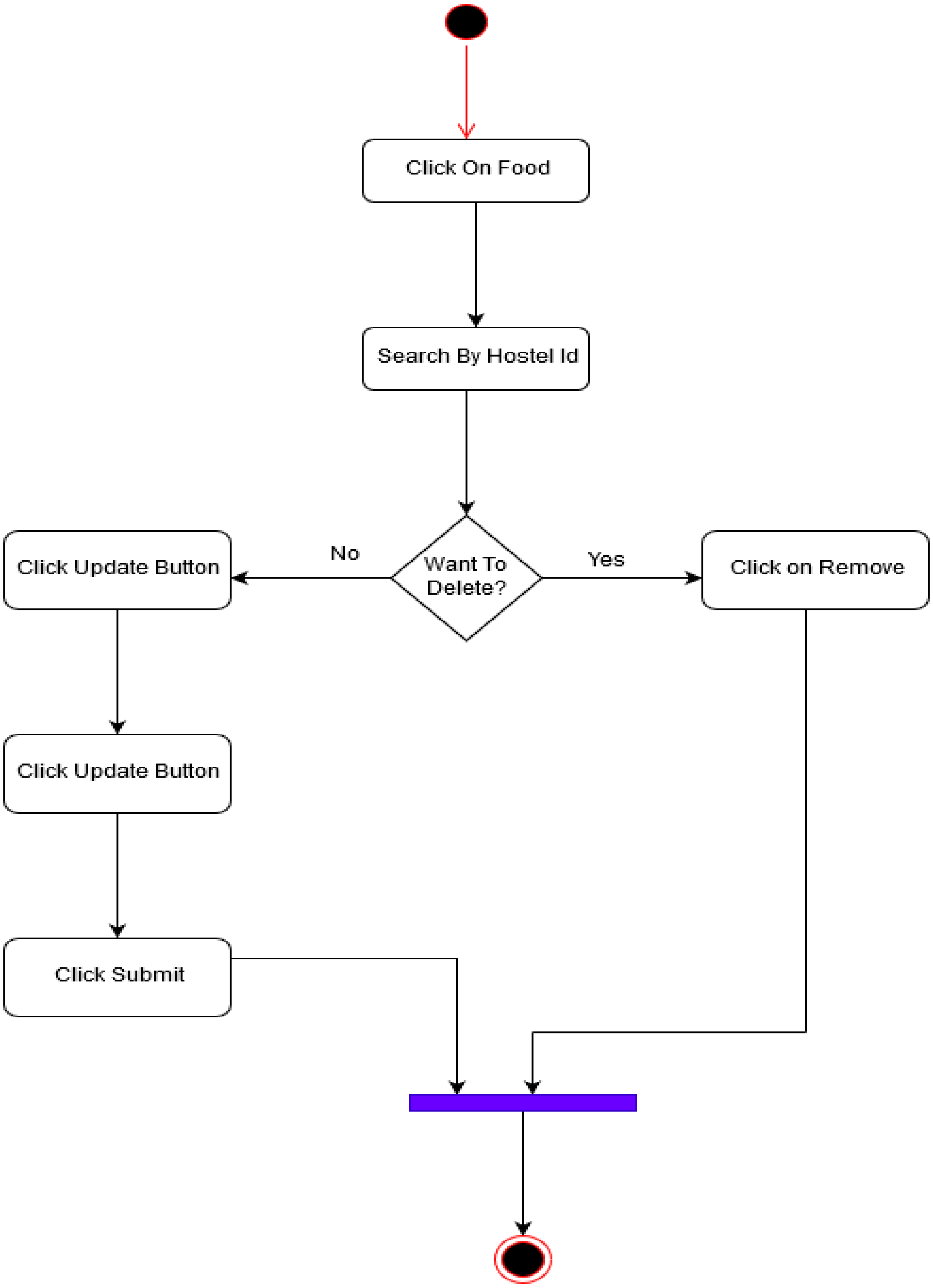


Figure 3.2.11: Activity diagram for Manage Food Info

3.2.12 View Review

All type of registered who can account he or she can see all review of hostels.

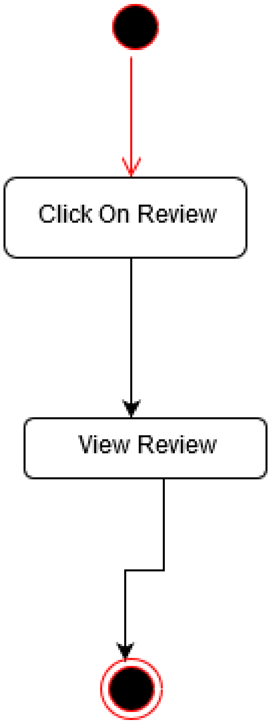


Figure 3.2.12: Activity diagram for view Review

3.2.13 Submit Comment

Registered user who can experience with available hostel they can add their comments.

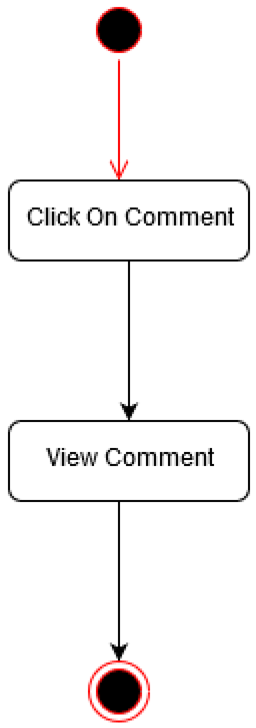


Figure 3.2.4: Activity diagram for Submit Comment

3.3 System Sequence Diagram

Mainly sequence diagrams understand us how the data will be followed in any application. Now I am going to show some sequence diagrams.

3.3.1 .Create Account

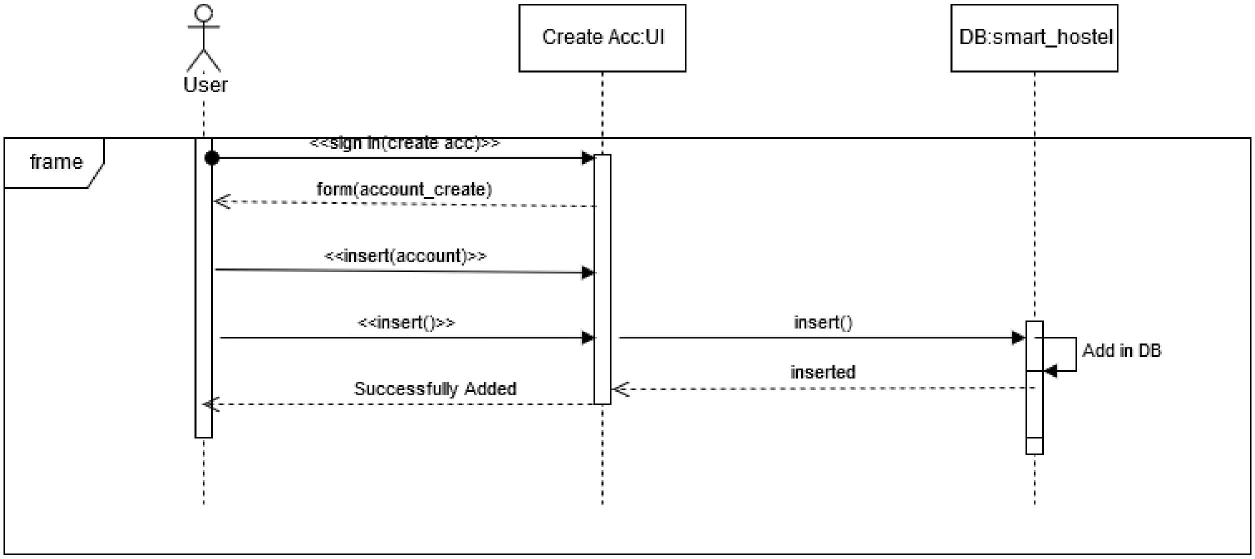


Figure 3.14: Sequence Diagram for Create account

3.3.2 .Manage Account

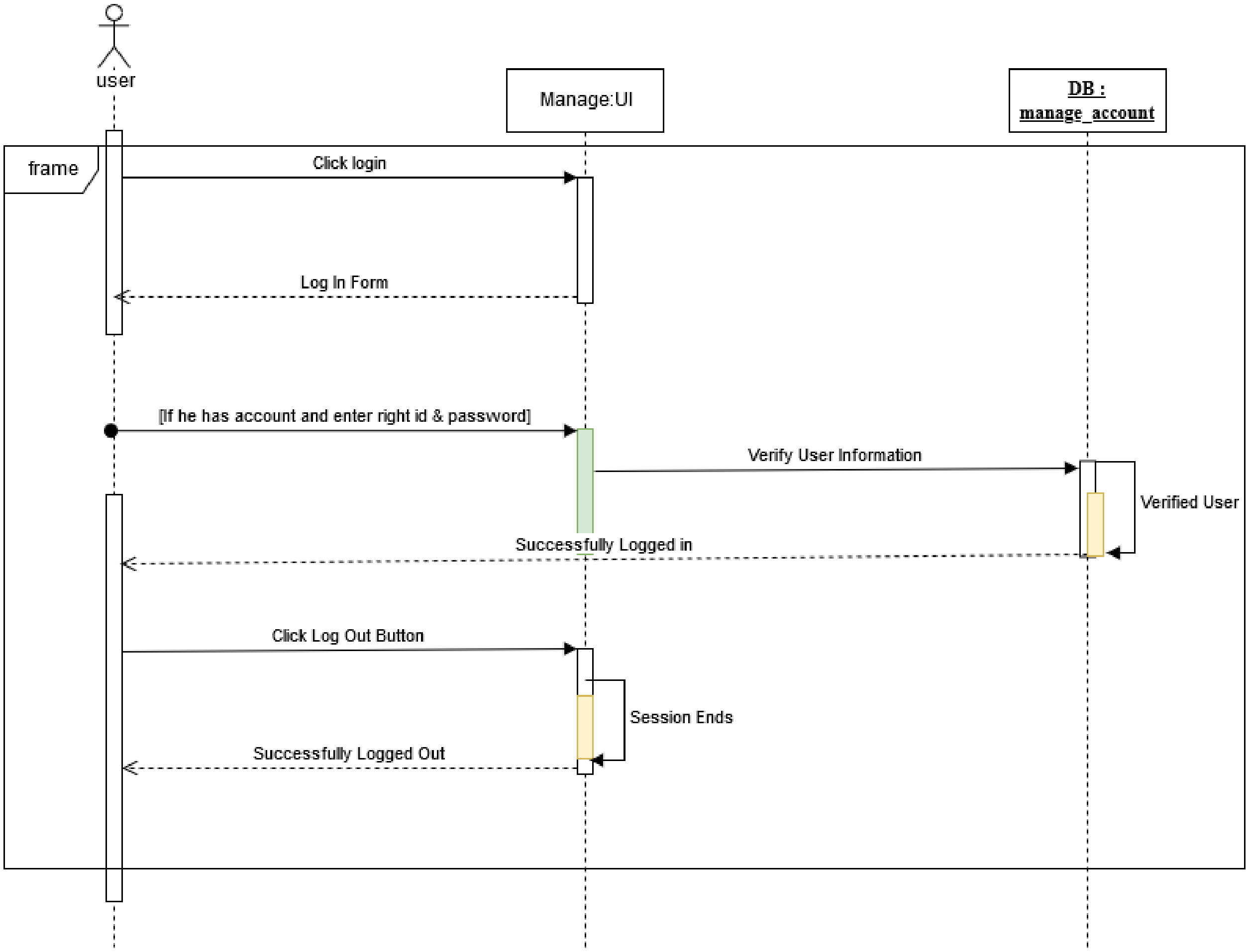


Figure 3.15: Sequence Diagram for Manage Account

3.3.3 .Add Hostel Info

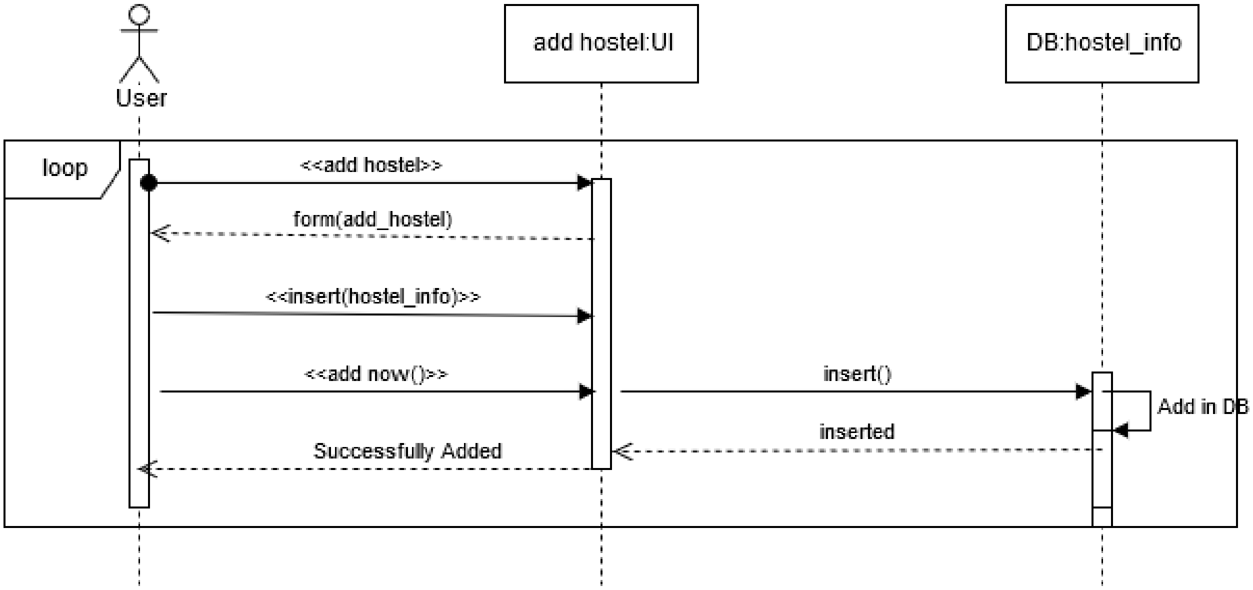
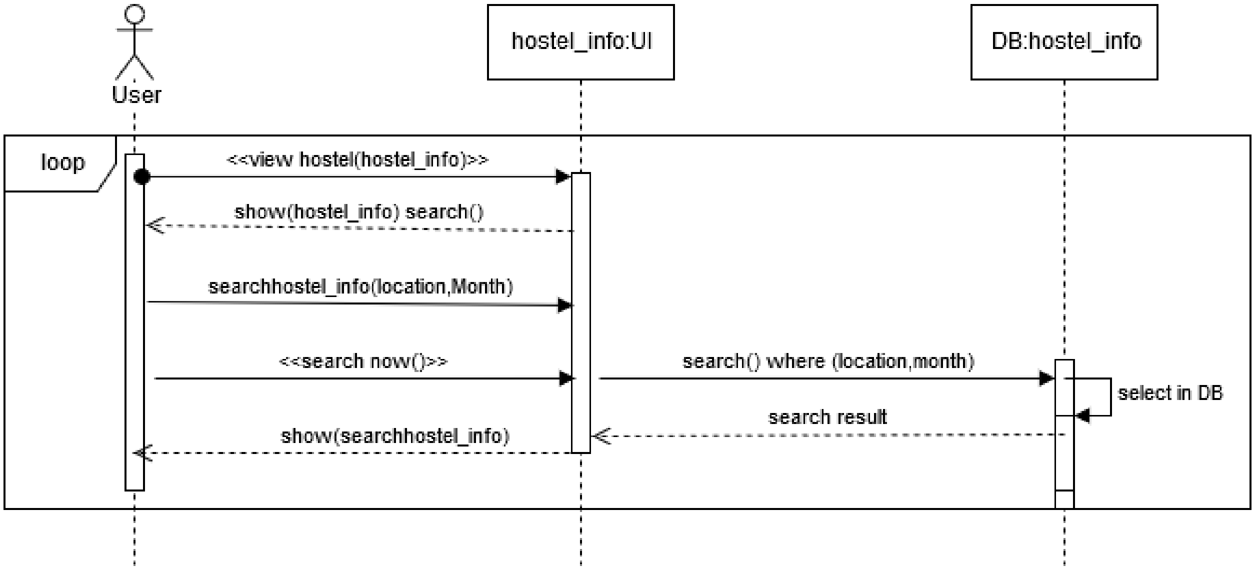


Figure 3.16: Sequence Diagram for Create account

3.3.4 .View Hostel Info



User
hostel_info:UI
DB:hostel_info
loop
<<view hostel(hostel_info)>>
show(hostel_info) search()
searchhostel_info(location,Month)
<<search now()>>
search() where (location,month)
select in DB
search result
show(searchhostel_info)

Figure 3.16: Sequence Diagram for view Hostel Info

3.3.5 .Manage Hostel Info

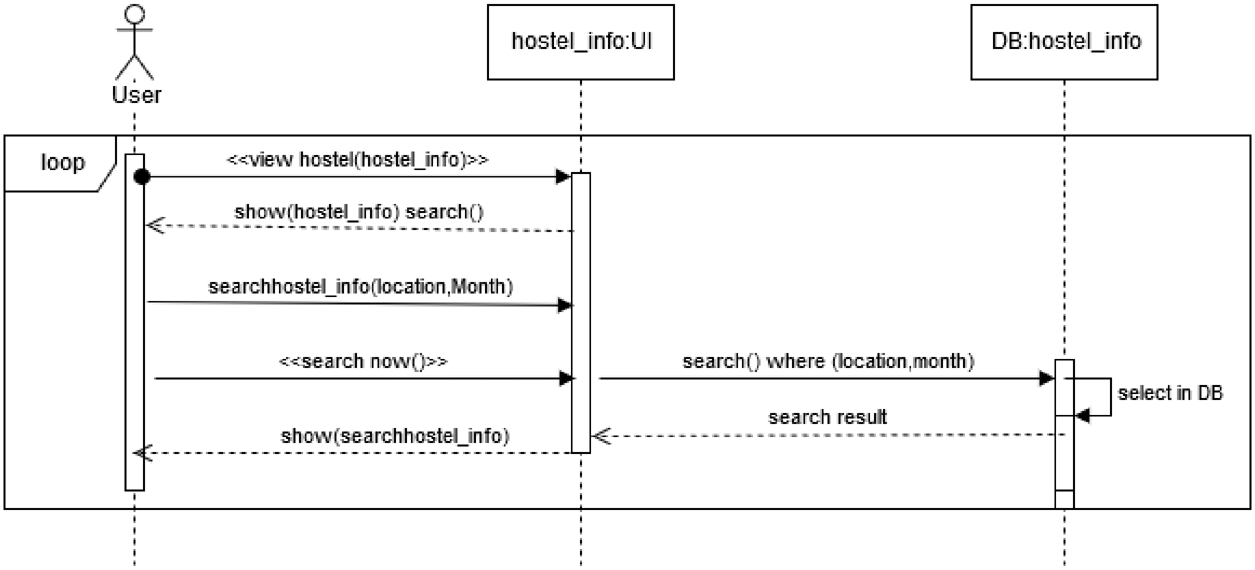

User
hostel_info:UI
DB:hostel_info
loop
<<view hostel(hostel_info)>>
show(hostel_info) search()
searchhostel_info(location,Month)
<<search now()>>
search() where (location,month)
select in DB
search result
show(searchhostel_info)

Figure 3.17: Sequence Diagram for Manage Hostel Info

3.3.6 .Add Room Info

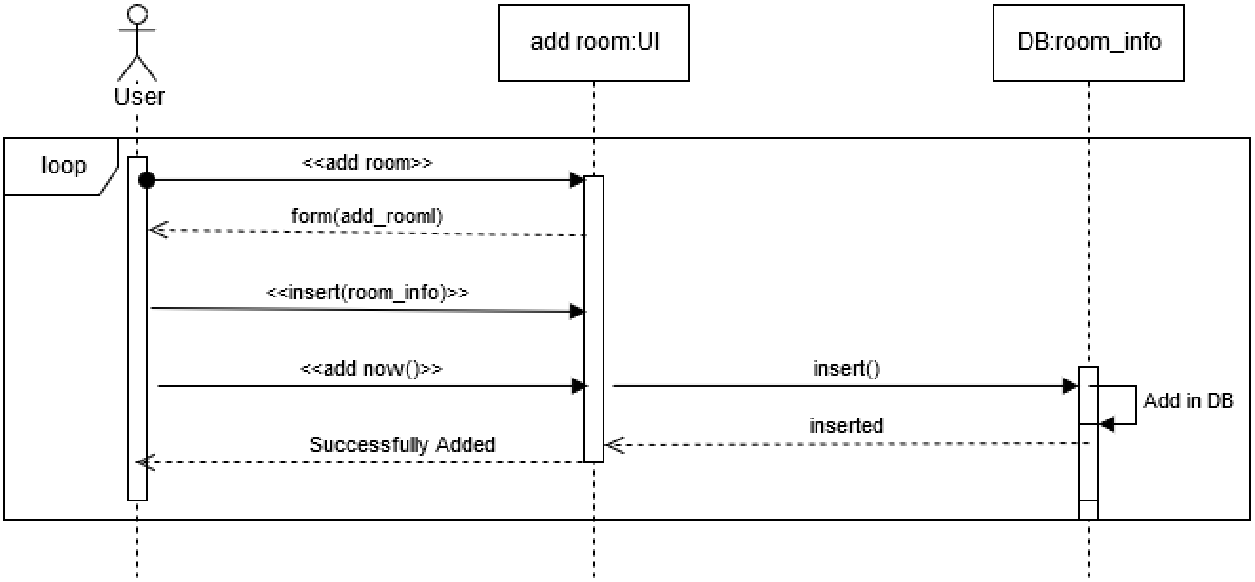


Figure 3.18: Sequence Diagram for Add Room Info

3.3.7. View Room Info

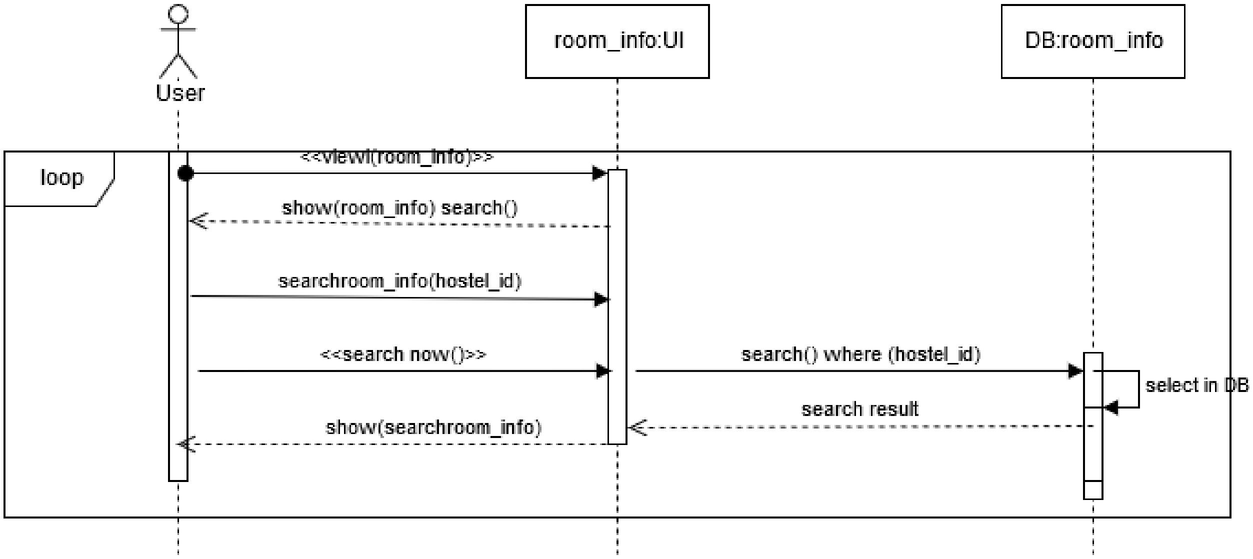


Figure 3.19: Sequence Diagram for View Room Info

3.3.8. Manage Room Info

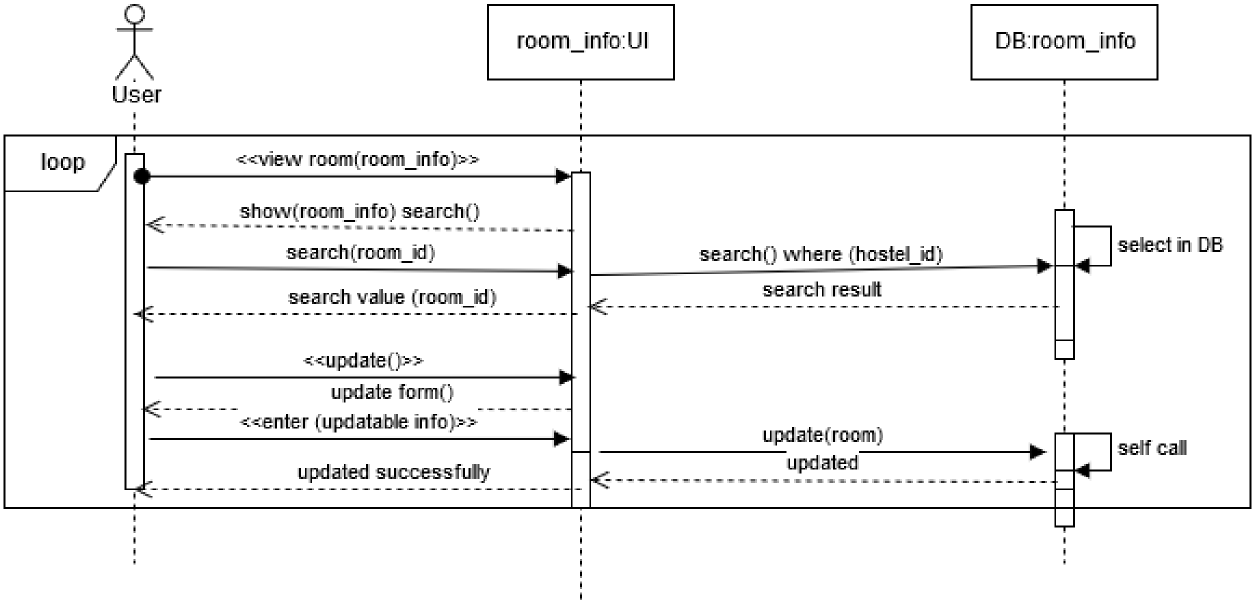


Figure 3.20: Sequence Diagram for Manage Room Info

3.3.9. Add Food Info

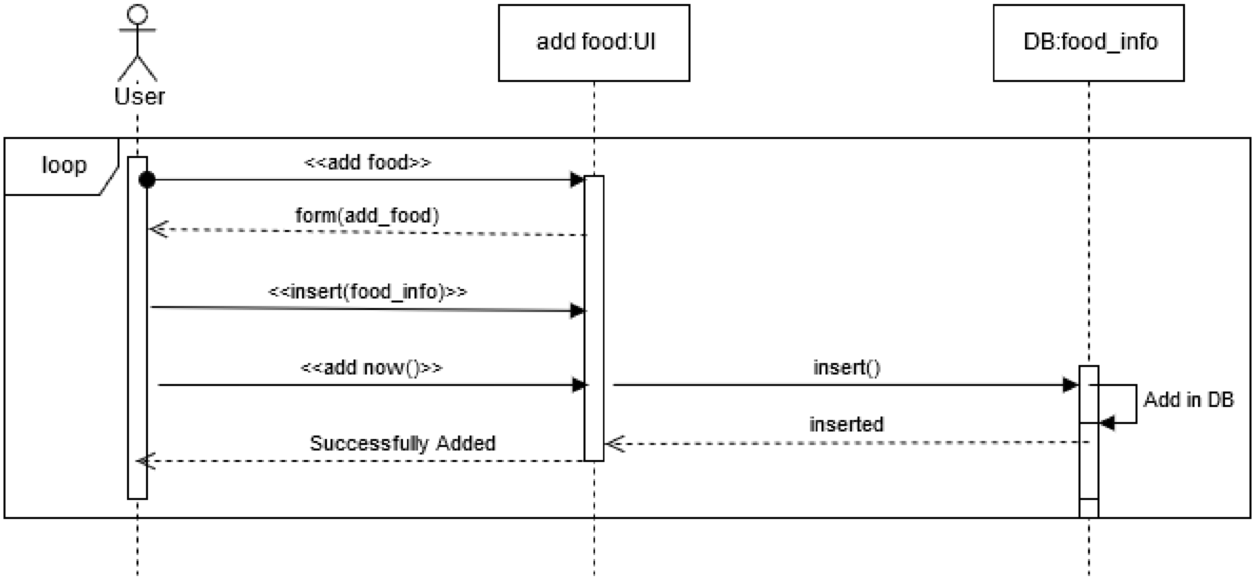


Figure 3.21: Sequence Diagram for Manage Add Food Info

3.3.10. View Food Info

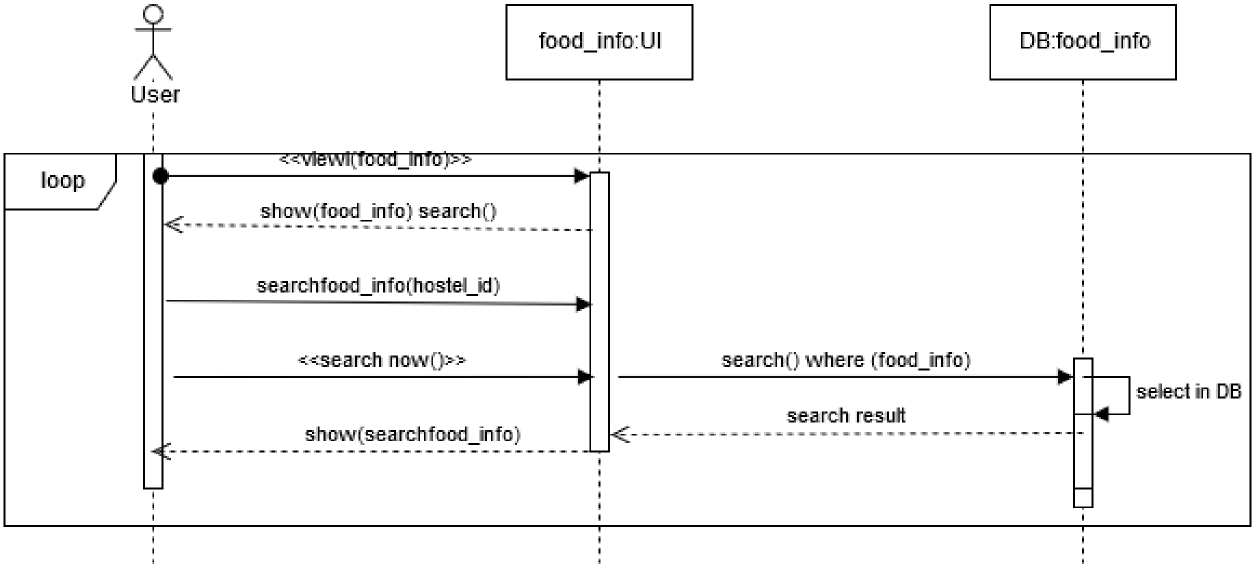


Figure 3.22: Sequence Diagram for Manage View Food Info

3.3.11. Manage Food Info

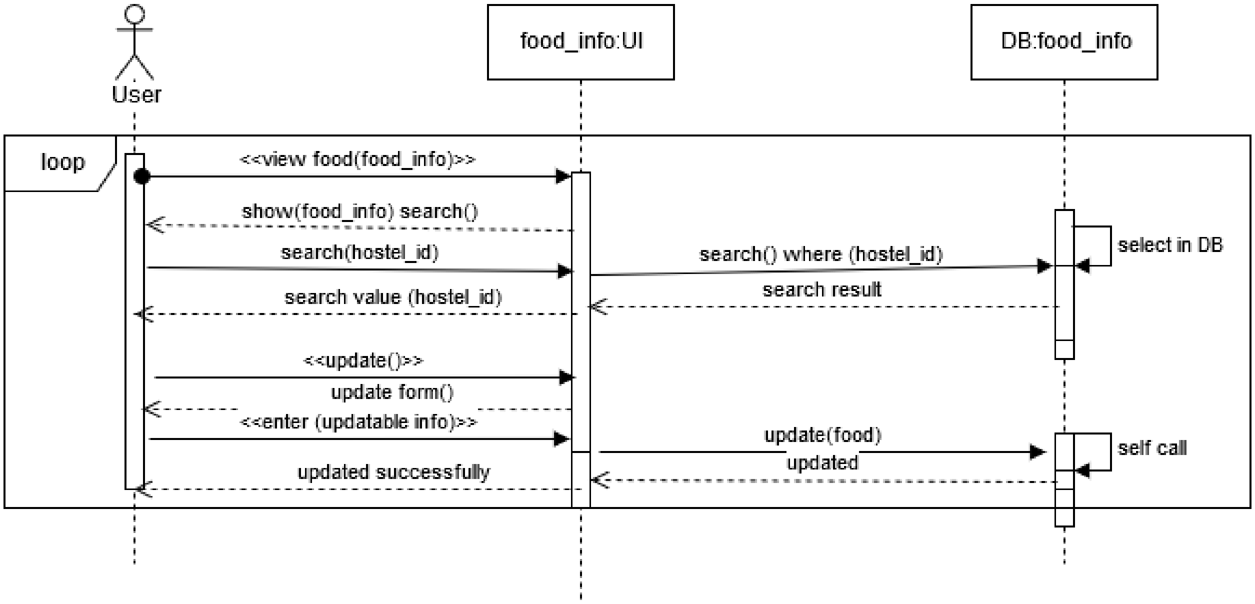


Figure 3.23: Sequence Diagram for Manage Food Info

3.3.12. View Review

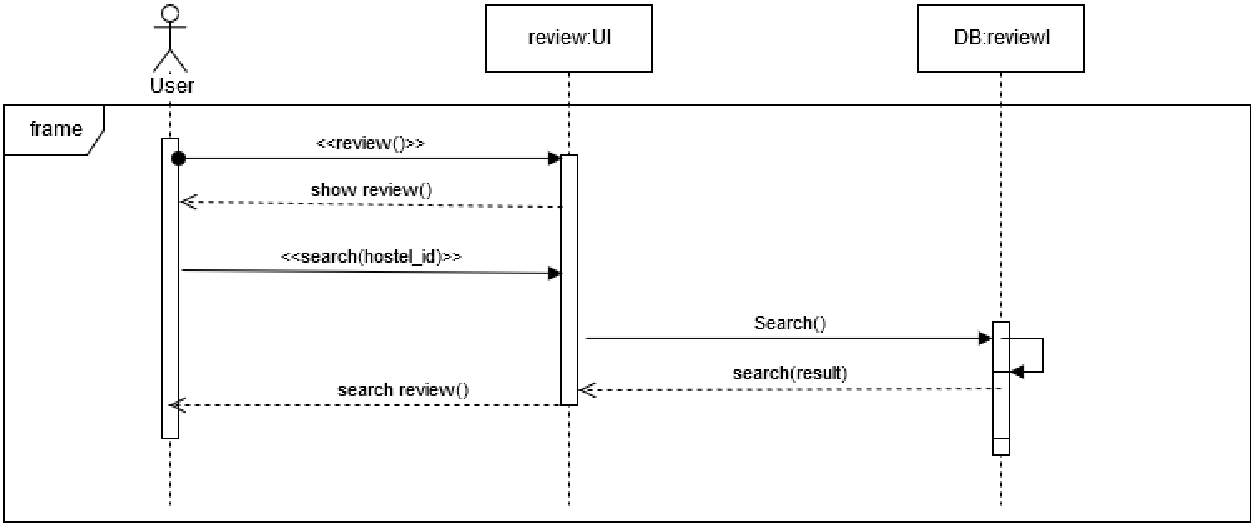


Figure 3.24: Sequence Diagram for View Review

3.3.13. Comment

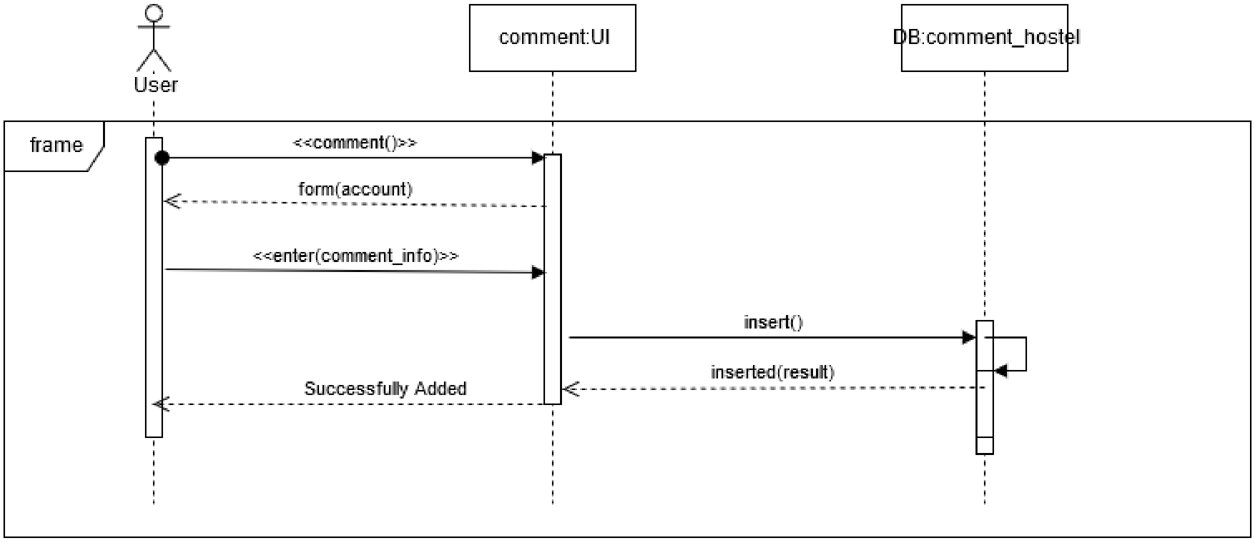


Figure 3.25: Sequence Diagram for Comment

Chapter 4

System Design Specification

4.1 Development tools and technology

Software development tools are used to develop software. There are different kinds of tools for software developers to make the process of software development very smoothly. But now I will mention some of tools that I am using to develop my project

4.2 Class Diagram

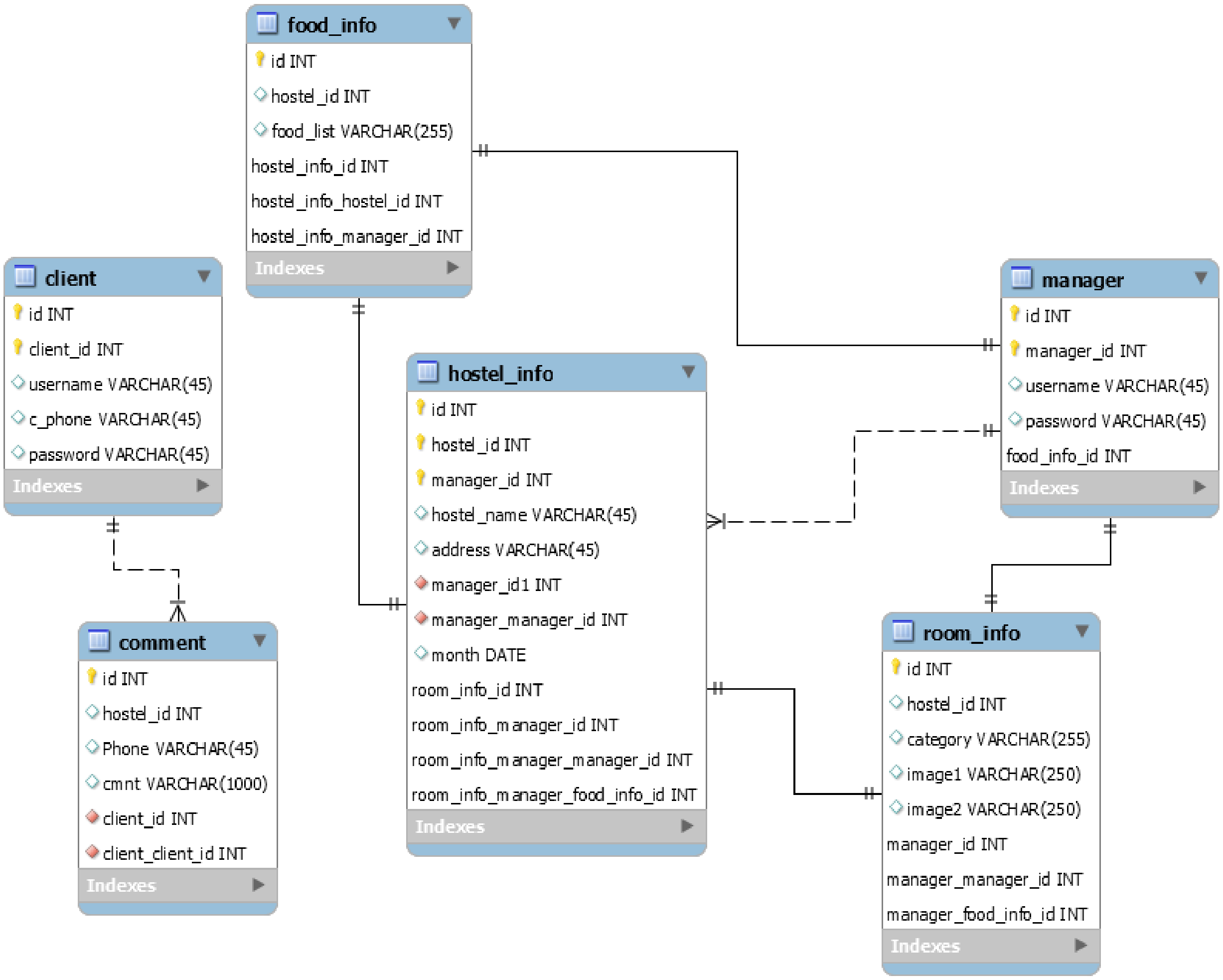


Figure 4.1: Class diagram (Smart Hostel Finder)

4.2 Database schema

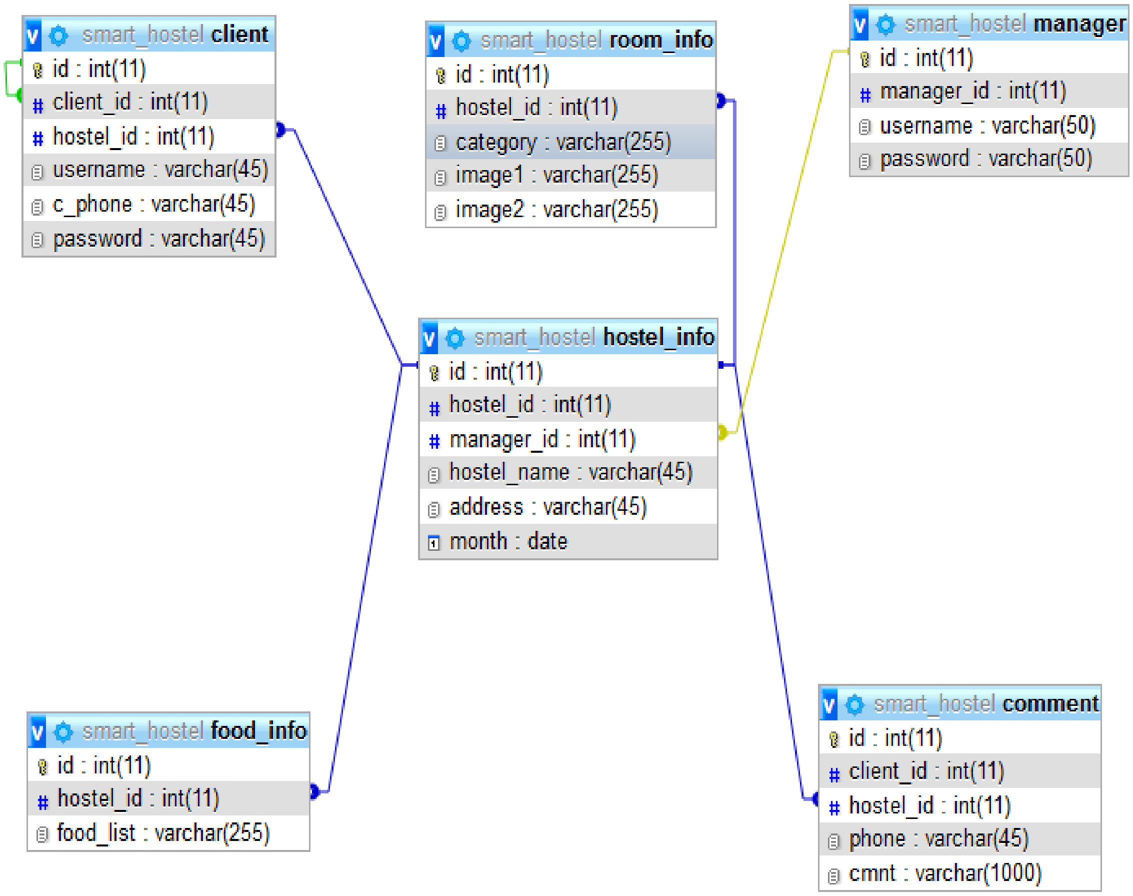


Figure 4.3: Database Diagram (Smart Hostel Finder)

4.4.1 User Interface Technology

First of all, after landing any application, interface of that software is appeared to the user. So, the importance of user interface is very high. For success of any software application, a good looking user interface plays a vital role. User interface includes using good image, graphics, typography, stylesheet scripting etc.

4.4.1.1 Programming Language

For developing any application system minimum one programming language is essential. In my application, there are two different programming language is used. One is for front-end side. And another is for server side. The front-end language that I have used to my application is HTML. And the server side programming language name is PHP which stands for Hypertext Preprocessor. Both of them are open source general purpose scripting language.

4.4.1.2 JQuery UI

JQuery UI is a name of library which uses JavaScript as core programming language. It simplifies codes of a programming language named JavaScript. Generally, it interacts with Graphical User Interface (GUI). It also provides visual effects with animation. It can also override cascading style sheets. It is also a subset of JavaScript. By using AJAX no page reloading is required. It is also compatible with any browser like Google Chrome, Mozilla Firefox, Opera, Safari, Internet Explorer etc

4.4.1.3 CSS framework or Bootstrap

After completing markup, cascading style sheets are designed. It generally explains how HTML elements will display. There are three ways to write stylesheet. One internal CSS, another is external CSS. And last one is inline CSS. But most of time external CSS is used. Because, by using external CSS, all CSS data can be kept in different files. But nowadays CSS framework seems to be very popular. In our application, I have used Bootstrap4. This framework is developed mainly for CSS and JavaScript. It is a free and open source front-end framework. This framework is highly used for designing website and web application also. This framework supports almost every browser. Like Google Chrome, Mozilla Firefox, Opera, Safari, Internet Explorer etc. Bootstrap also provides media query which features responsive layout for different devices with different screen size. Bootstrap provides a set of some files which contains stylesheets which gives basic definition. Bootstrap also provides some JavaScript components also. There are some built in components like JQuery UI. By using Bootstrap framework, we get both CSS and JavaScript facilities with a single platform. But before start designing an application interface with Bootstrap, one may have some basic knowledge about this framework. It will increase the efficiency.

4.4.2 Implemented tools and platform

As I have said before, there are some tools and technologies that need to be used for developing software. It is very important to determine which tools and platforms are the best match of my requirements. After making a proper decision, one need to start using them.

4.4.2.1 Integrated Development Environment

IDE stands for Integrated Development Environment. Programmers write code on IDE. After that IDE provide the feature to execute the source code. For developing my web application, I have used an IDE. To develop my web version, I have used SublimeText which is powered by Sublime HQ Company. It is a commercial IDE for cross platform environment. It is able to suggest code to the programmers also.

4.4.2.2 Database Server (MYSQL Server V-10.1.36-MariaDB)

For developing our whole project, I have followed Relational Database Management System or RDBMS. And I find that MySql provides the feature of RDBMS. So I should not have any issue to use MySql database. It is also very easy to use. It can also ensure the security, scalability, high performance and many things.

4.4.2.3 Web Server

I have used Xampp apache server. It is free and open source software to use. It can be used on cross platform. It supports a wide range of features and most of them are already implemented as compiled modules. This module can extend the main features or core functionality.

Chapter 5
System Test

5.1 Testing Features

Feature testing can be considered as making change to add or modify the new functionalities to the existing project. To test the features and functionality, a new test set is to be written for testing purpose. Almost every feature and functionality has different characteristics. Those are designed to make the application more useful, intuitive, reliable, secured, scalable, effective and efficient.

5.1.1 Features to be tested

Features	**Priority**	**Description**
Login	1	User must be authenticated by login
Logout	1	User must logout log out
Search hostel by location	3	Giving location properly
Insert hostel info	3	Hostel info must insert properly
Update hostel info	1	Hostel info must be update properly
View hostel info	3	Hostel info view properly
View room info	1	Room info view must be properly
Update room info	2	Room info must be update properly
View food info	1	Food info must be view properly
Update food info	2	Food info must be update properly
View Review	1	Review must be view properly
Comment	2	Comment done must be properly
Technological Features		
Database	1	Database will be used at almost every operation. So this is why, this part must be controlled tightly.

Here, 1 = Low Priority, 2 = Medium Priority, 3 = High Priority

5.2 Testing Strategy

Testing strategy is to be considered as a general purpose of testing process. Testing object, testing function methods, total available resources are included to the testing strategy. It is also the indicator of test levels that are to be performed on the whole software development life cycle. Those strategies are prepared by the quality assurance team that should be reviewed by the developers of the application. After that it should be reviewed by the test team leads. Different kinds of testing strategies can be performed according to the type of application system need to be tested.

5.2.1 Test approach

To complete the test process, testers must take some approach. There are mainly two type of test approach.

- **Automation testing:** Automation testing is a name of testing technique by which test engineers prepare some scripts according to test plan and after that they use suitable tools to perform testing of the software. Nowadays, almost every software company follow the approach of automation testing.
- **Manual testing:** Manual testing is also a name of technique of testing by searching out the bugs or vulnerability in an application. In this process, test engineers manually test and execute the test cases without having any automation tools.

5.2.1.1 Black Box Testing

Black box testing is a software testing method by which test engineers test the software without having knowledge of the internal architecture of the application that need to be tested. It is also known as behavioral testing. Black box testing can be either functional or non-functional. It ignores the internal mechanism of a system. We have decided to perform the equivalent class partitioning and Boundary value analysis techniques to implement.

5.2.1.2 Equivalent Class Partitioning

Equivalent class partitioning is also known as equivalent partitioning. In this technique input data is to be divided into groups. Those groups of data are expected to perform as similar behavior. Each group works as same as other groups also.

The main advantage of following the approach of equivalent class partitioning is to reduce the total number of test cases from infinite to finite. Another advantage is that it can be applied to all testing levels also.

For legal input value output will produce a meaning data. But for illegal input value, output will not produce meaningful data.

5.2.1.3 Boundary Value Analysis

Boundary value analysis is a name of software testing by which input data can be included within a boundary range. Following this strategy, test engineers firstly define the boundary value. After that they take a specific boundary and then write the test cases that will be go through the application for testing purpose. After completing those tasks, testers use test cases to test applications software.

5.2.1.4 White Box Testing

White box testing is also a name of testing approach which is also known as clear box testing, glass box testing, open box testing, transparent box testing, code based testing or structural testing. It is opposite to the black box testing. In black box testing, the internal architecture or algorithm is not known to the testers whether the whole software architecture is known to the testers while white box testing.

Testers can also predict the output of every test cases for white box testing. White box testing can be classified into some levels. Such as:

- Unit Testing
- Integration Testing
- System Testing

The main advantage of white box testing is that testing is more throughout and the testing can be started from the very beginning stage.

5.2.2 Pass / Fail Criteria

Pass or fail criteria will be set by the test engineers. They will prepare the pass / fail criteria on the basis of which input data are worked and which are not works well. Those data that are worked well will considered as pass criteria. And rest of input data will be considered as fail criteria.

Now I will give the pass / fail criteria below.

- System crash will not be considered as pass case.
- If any criteria pass 100% times, then it will be considered as pass criteria only.
- If data can’t be displayed to the application properly, then it is also to be considered as fail criteria.

5.2.3 Testing Schedule

Test Phase	**Time**
Testing plan create	1 week
Test specification	2 week
Unit testing	During development time
Component test	1 week
Test Phase	**Time**
Integration testing	1 week
Validating use cases	1 week
Testing user interfaces	1 week
Load testing	1 week
Performance testing	1 week
Release to production	1 week

5.2.4 Trace Ability Matrix

Project Manager			**Business Analyst Lead**		
QA Lead			**Target Implementation Date**		
BR#	**Category /** **Functionality / Activity**	**Requirement Description**	**Use Case Reference**	**Test Case Reference**	**Comments**
Br-1	Functional	Login	Use case 3.1.2	Test case 5.6.1	
BR-2	Functional	Log out	Use case 3.1.2	Test case 5.6.3	
BR-3	Functional	Search Hostel	Use case 3.1.4	Test case 5.6.4	
BR-4	Functional	Insert Hostel info	Use case 3.1.3	Test case 5.6.5	

5.3 Testing Environment

Testing environment means to prepare the environment with hardware and software so that test engineers can be able to execute test cases as required. Besides hardware and software usage, network configuration might be needed to execute test plans.

For making the environment for testing, some key area need to setup. Those are:

- Test data
- Database server
- Client’s operating system
- Front end running environment
- Browser
- System and application
- Network
- Hardware with server operating system
- Documentation is also required. Like: user manuals, installation guides, configuration guides, documents etc.

5.4 Test Cases

A test case refers to some rules and regulations or conditions by which it can be determined whether a system can be able to meet the works or requirements under test cases properly. We know, there is a chance to have some fault or break holes in any application. This is very common scenario. And those issues are solved usually by software testing approaches. But if we don’t care of those issues, then the full application development might be ruined. So proper testing must be needed. For testing our application, I have prepared some test cases. Now, I am going to provide them.

5.4.1 Log In

<table>
<tr><td colspan="3"><b>Test case #1</b></td><td colspan="3"><b>Test case name:</b> Log in</td></tr>
<tr><td colspan="3"><b>System:</b> Smart Hostel Finding</td><td colspan="3"><b>Subsystem: N/A</b></td></tr>
<tr><td colspan="3"><b>Designed by:</b> Kamruzzamn</td><td colspan="3"><b>Designed date:</b> 2/12/19</td></tr>
<tr><td colspan="3"><b>Executed by:</b></td><td colspan="3"><b>Executed date:</b></td></tr>
<tr><td colspan="6"><b>Short description:</b> The registered users need to login to the application. And before that our application will check the authentication and authorization.</td></tr>
<tr><td colspan="6"><b>Pre-conditions:</b>
• Users are always redirected to the login page whether they have not authenticated by our application while entering into the dashboard.
• Assume that, the Username is ’Kamruzzaman’ and password is ‘SMF’</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td><b>Step</b></td><td><b>Username</b></td><td><b>Password</b></td><td><b>Expected result</b></td><td><b>Pass/Fail</b></td><td><b>Comment</b></td></tr>
<tr><td>1</td><td>Nazmul islam nabil</td><td>112233</td><td>Invalid</td><td></td><td></td></tr>
<tr><td>2</td><td>Kamruzzzaman</td><td></td><td>Password can’t be blank</td><td></td><td></td></tr>
<tr><td>3</td><td></td><td>000000</td><td>Username can’t be blank</td><td></td><td></td></tr>
<tr><td>4</td><td>Admin</td><td>Root</td><td>Successfully login to the application and redirected to the dashboard</td><td></td><td></td></tr>
<tr><td colspan="6"><b>Post-conditions:</b> Users including Smart Hostel Finding will successfully login to the application.</td></tr>
</table>

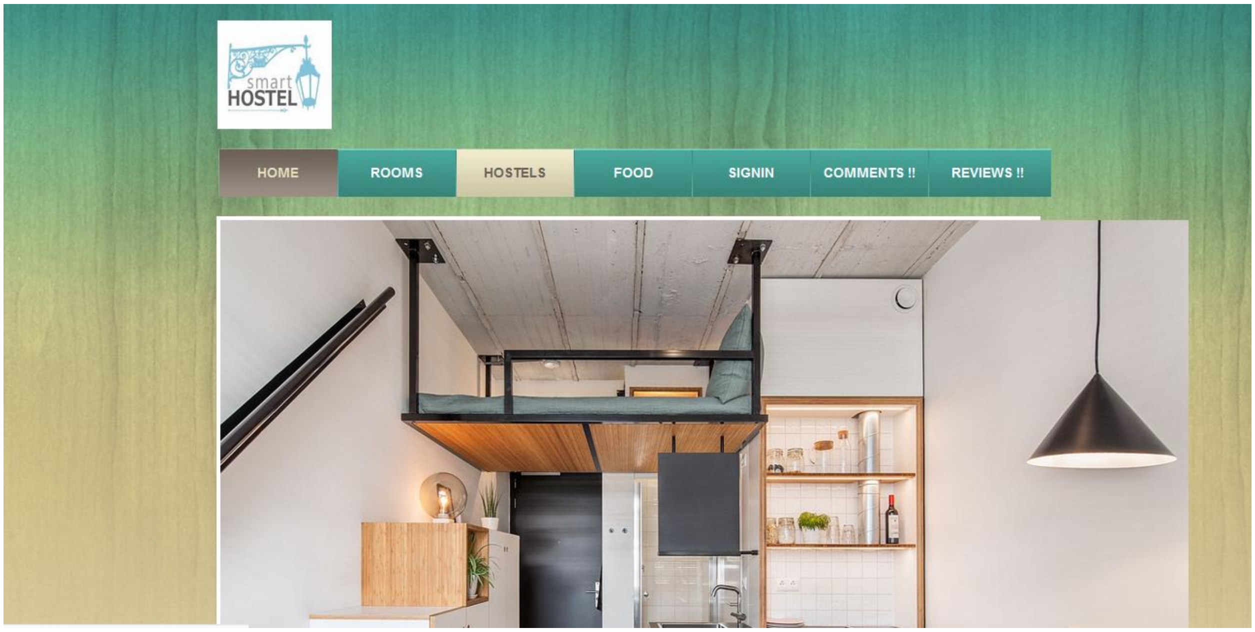
Chapter 6

Requirement Analysis

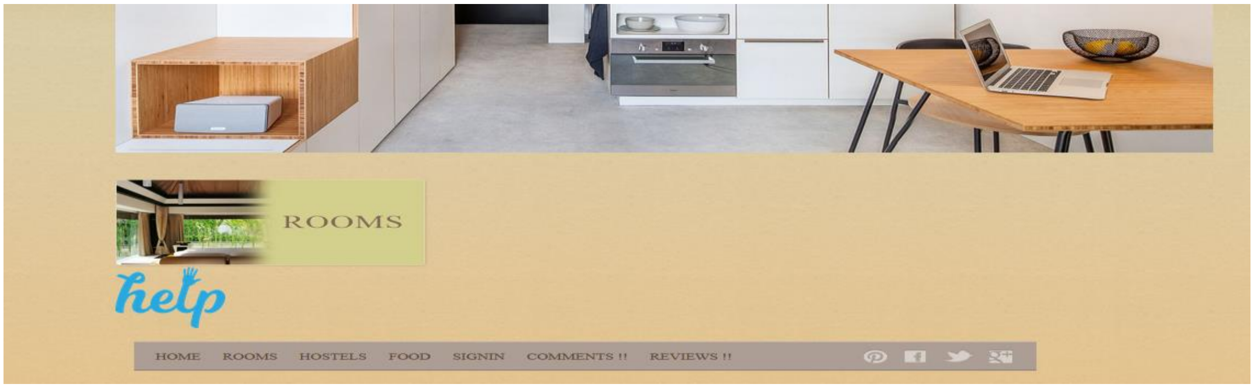
6.1 User Interface

First of all, after landing any application, interface of that software is appeared to the user. So, the importance of user interface is very high. For success of any software application, a good looking user interface plays a vital role.

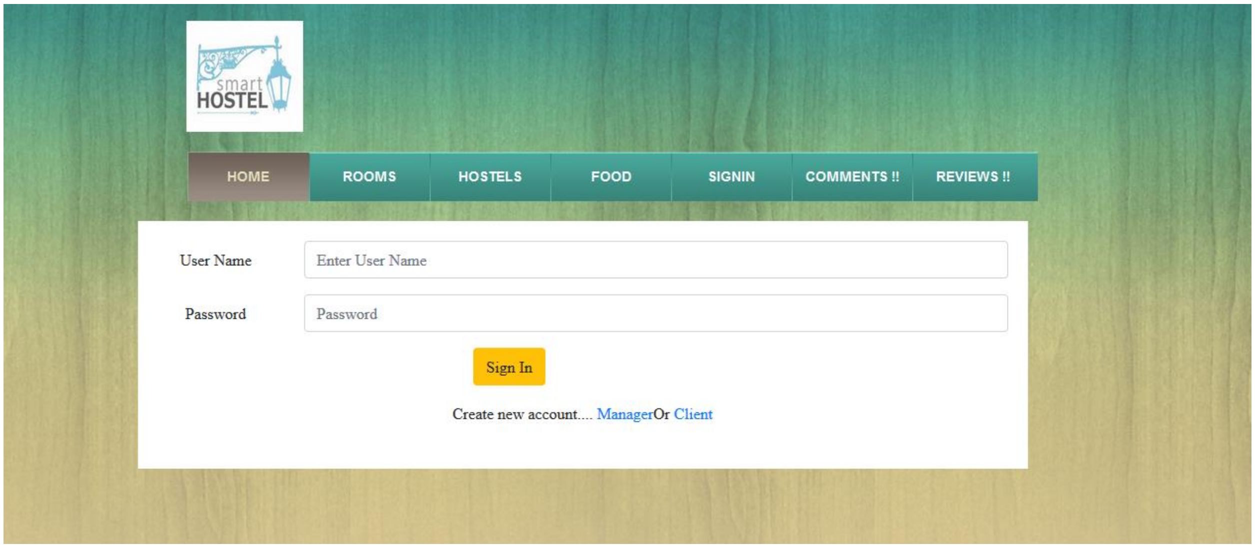
6.1.1 This is my home page



6.1.2 Home Page Footer



6.1.3 Login Page for Both Client and Manager

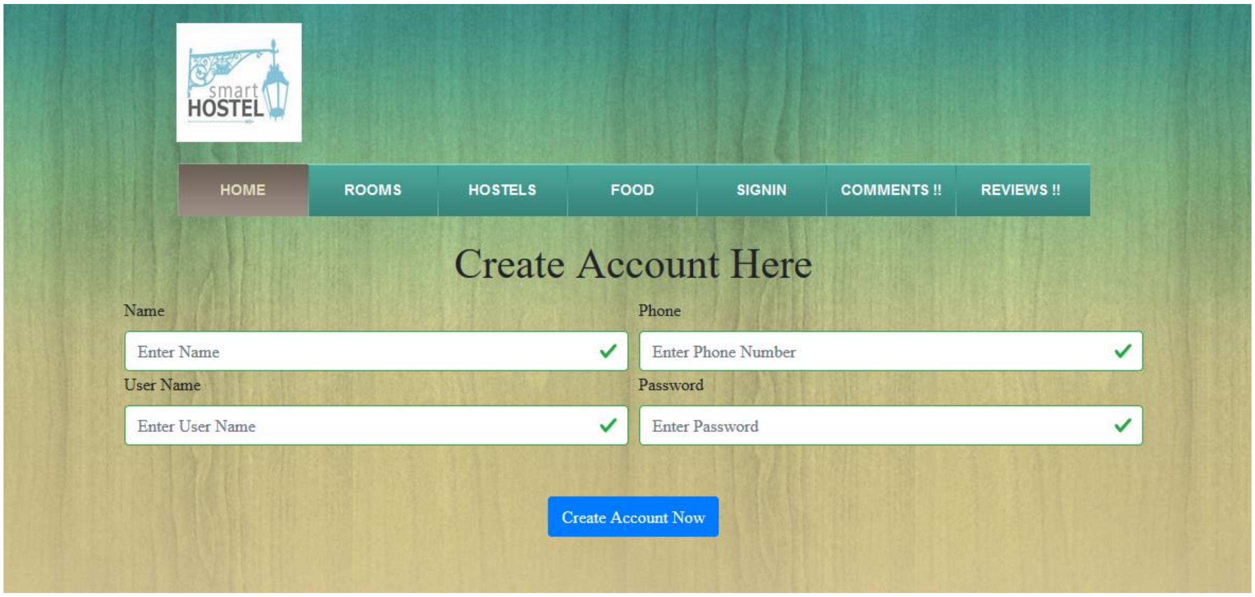


User Name Enter User Name

Password Password

Create new account.... ManagerOr Client

6.1.4 Registration for Manager



Create Account Here

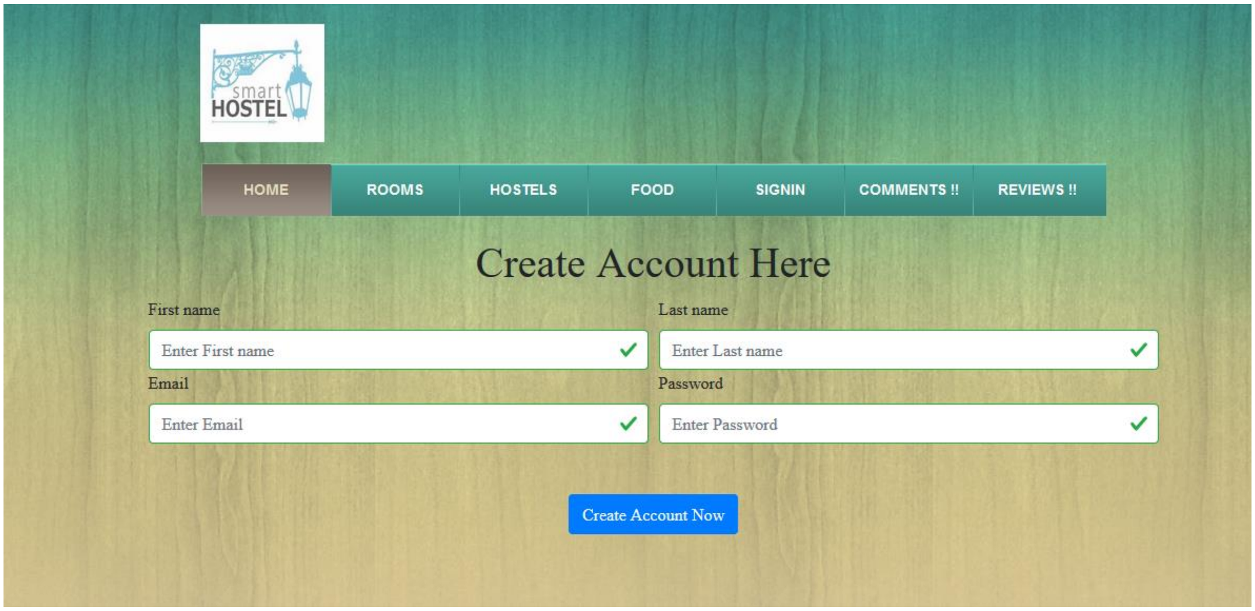
Name Enter Name ✓

Phone Enter Phone Number ✓

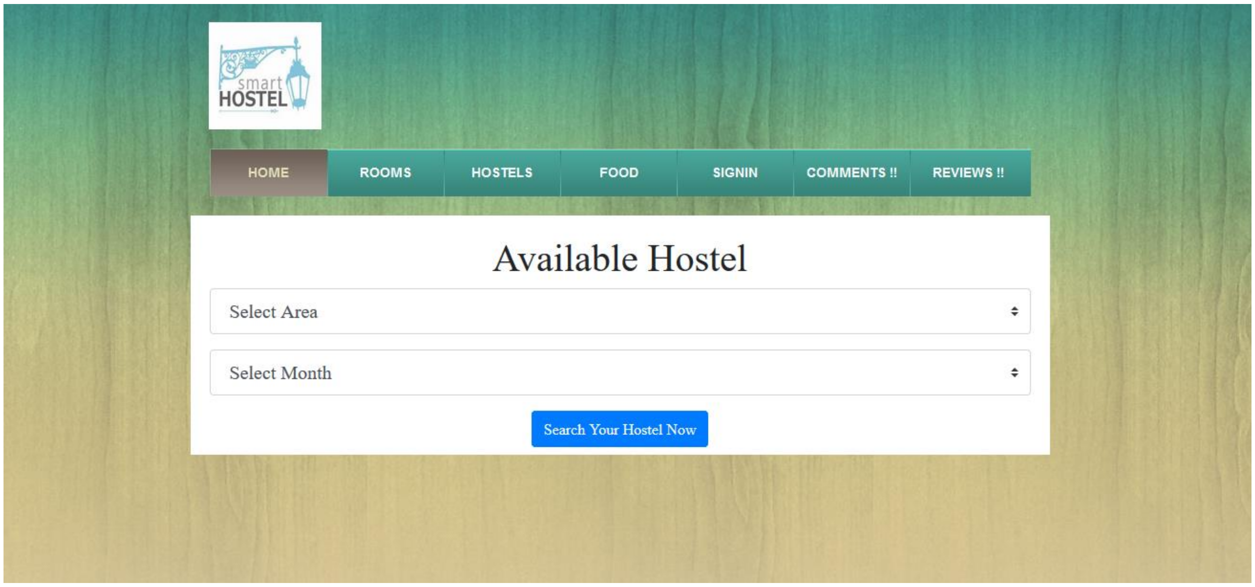
User Name Enter User Name ✓

Password Enter Password ✓

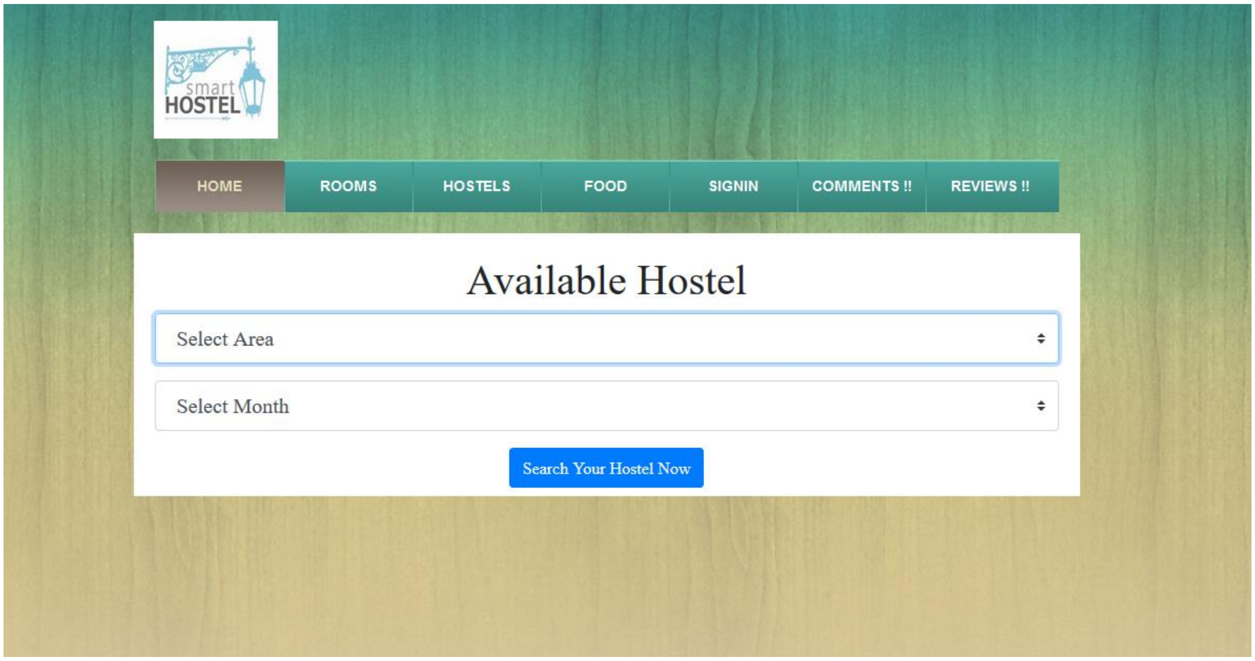
6.1.5 Registration for Client



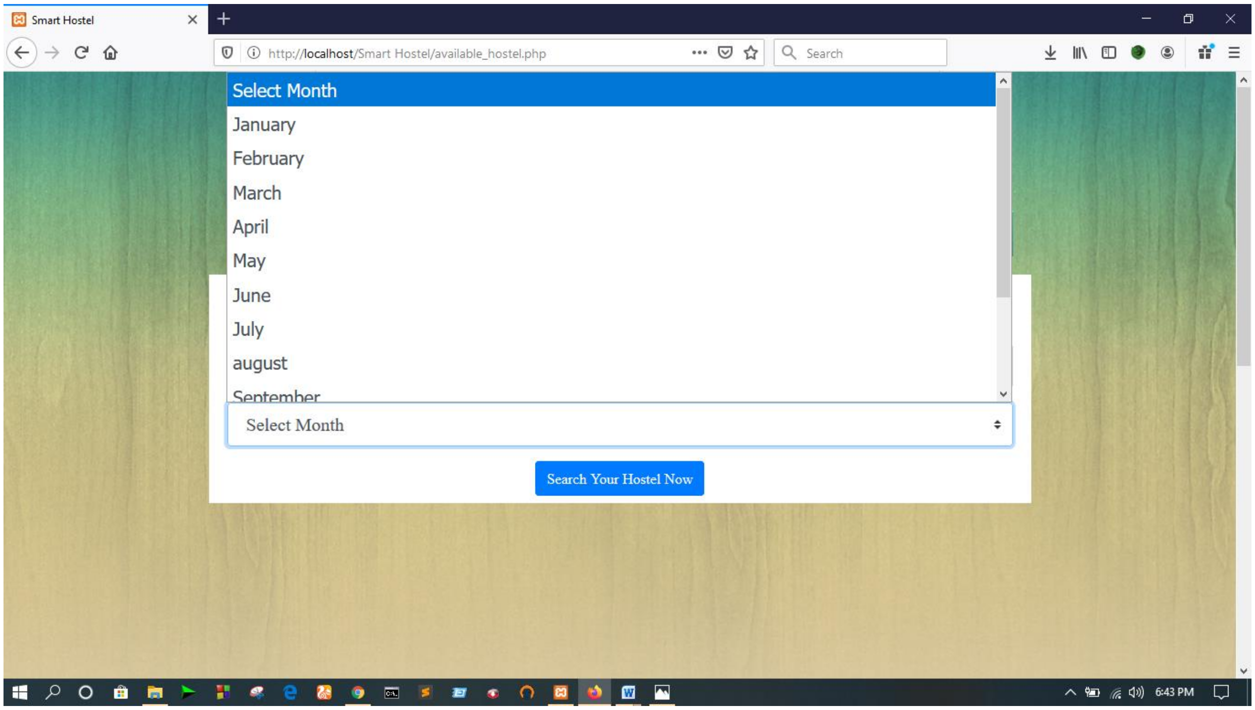
6.1.6 Search available Hostel



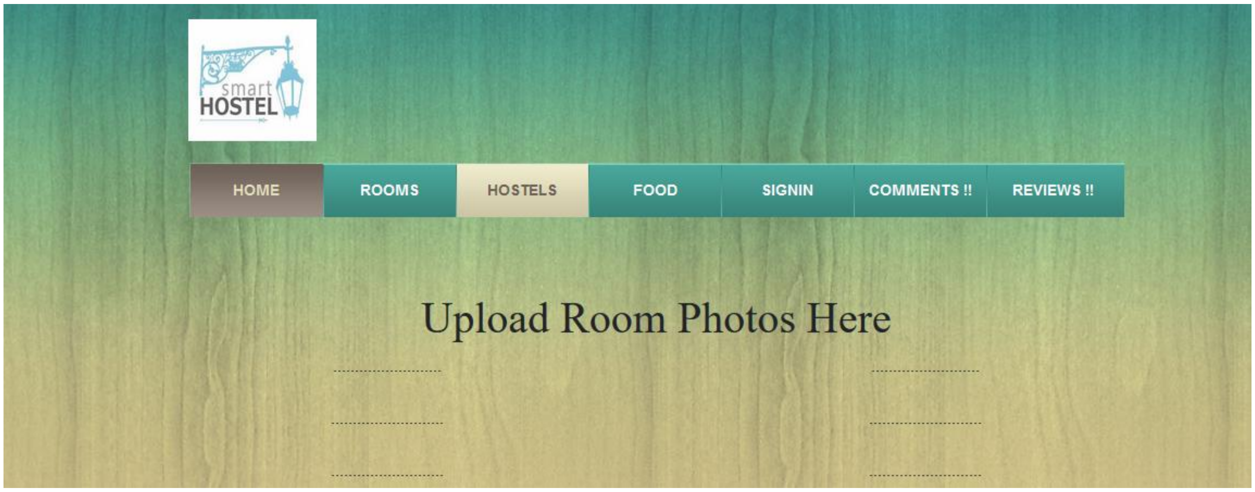
6.1.7 Search available Hostel by location



6.1.8 Search available Hostel by Month

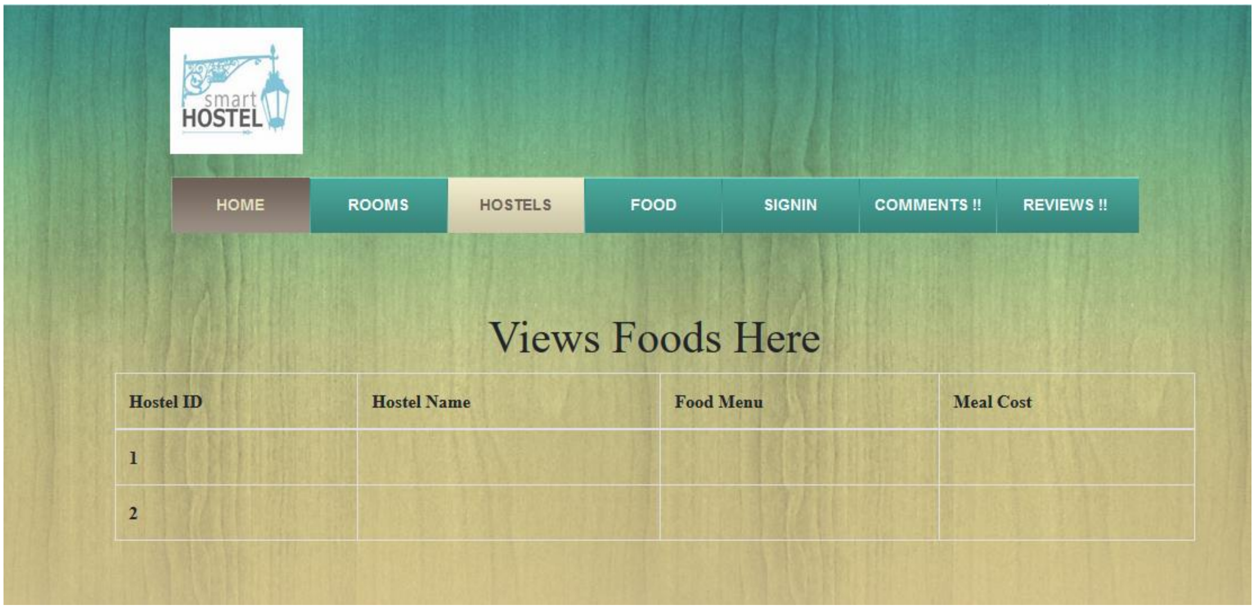


6.1.9 View Room Images



Upload Room Photos Here

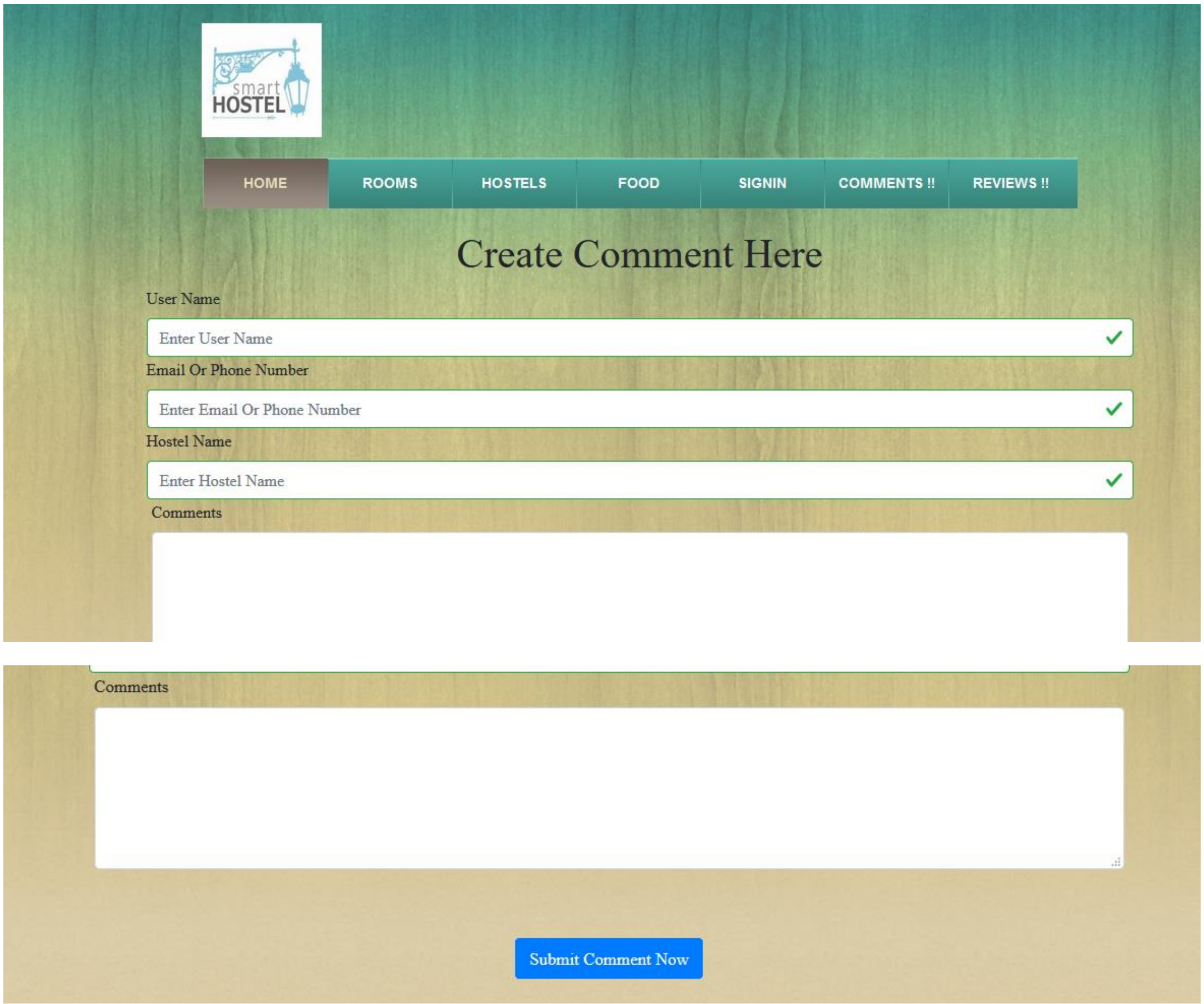
6.1.10 View Food Item



Views Foods Here

Hostel ID	Hostel Name	Food Menu	Meal Cost
1			
2			

6.1.11 Give Review by client



Create Comment Here

User Name

Enter User Name ✓

Email Or Phone Number

Enter Email Or Phone Number ✓

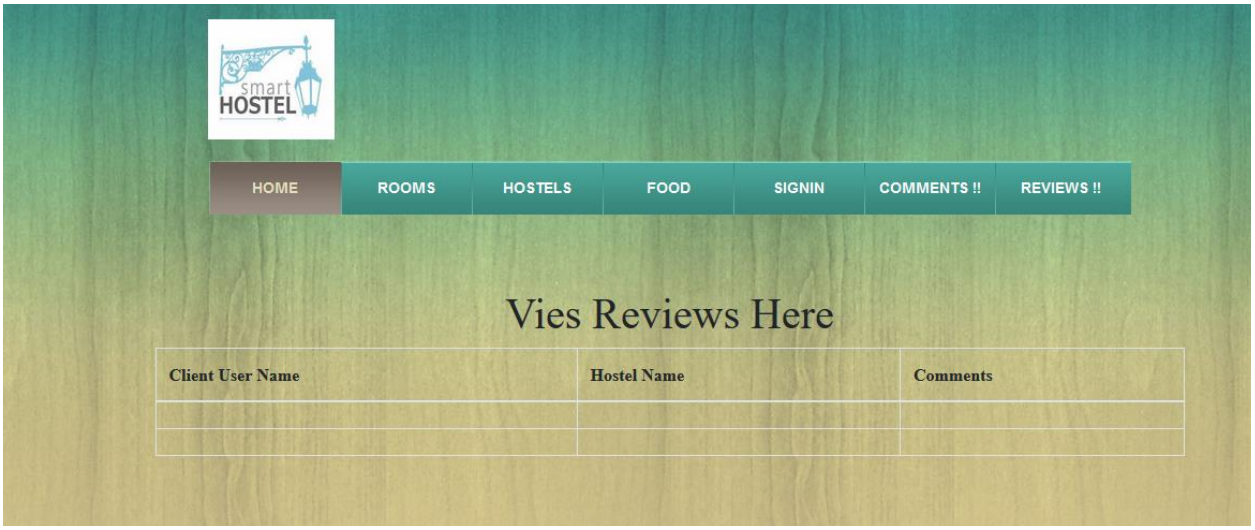
Hostel Name

Enter Hostel Name ✓

Comments

Comments

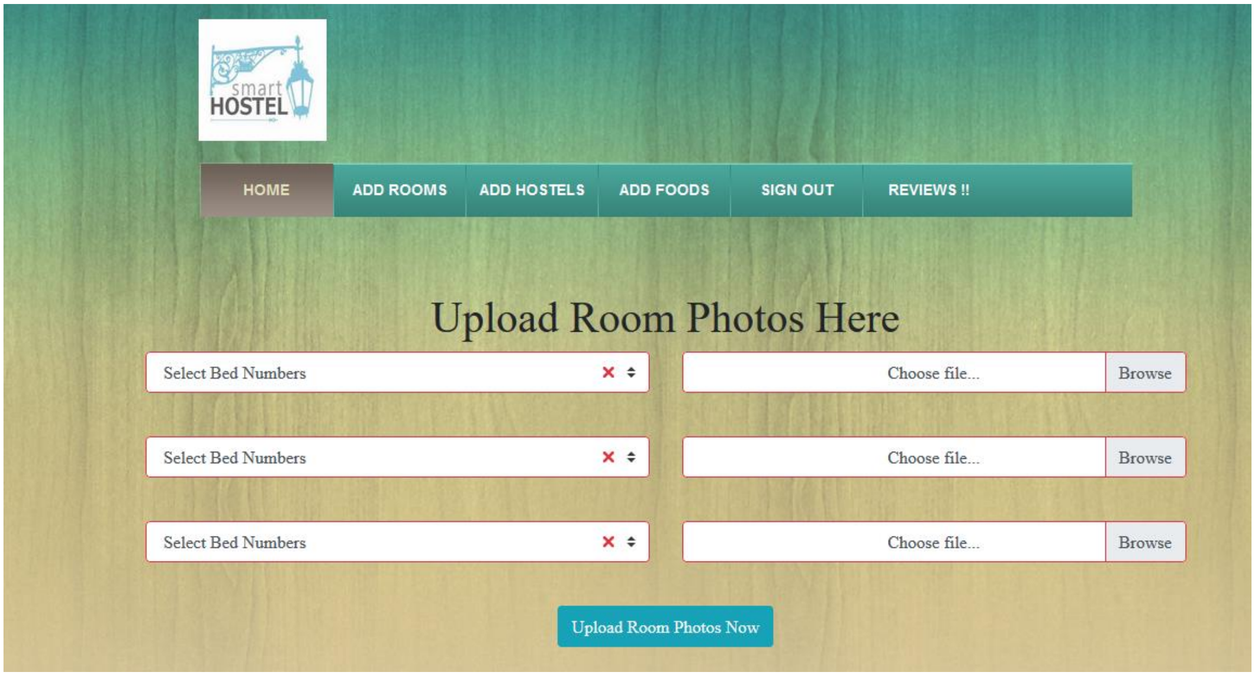
6.1.12 Read Reviews by Client and hostel Manager



Vies Reviews Here

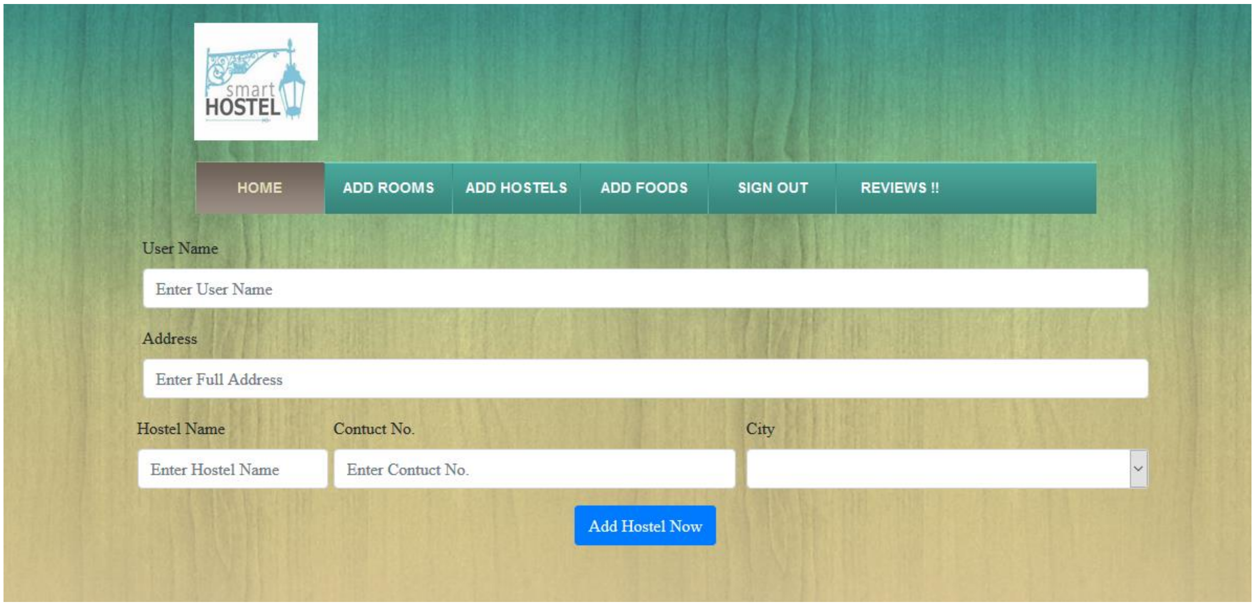
Client User Name	Hostel Name	Comments

6.1.13 Upload room photos by Manager

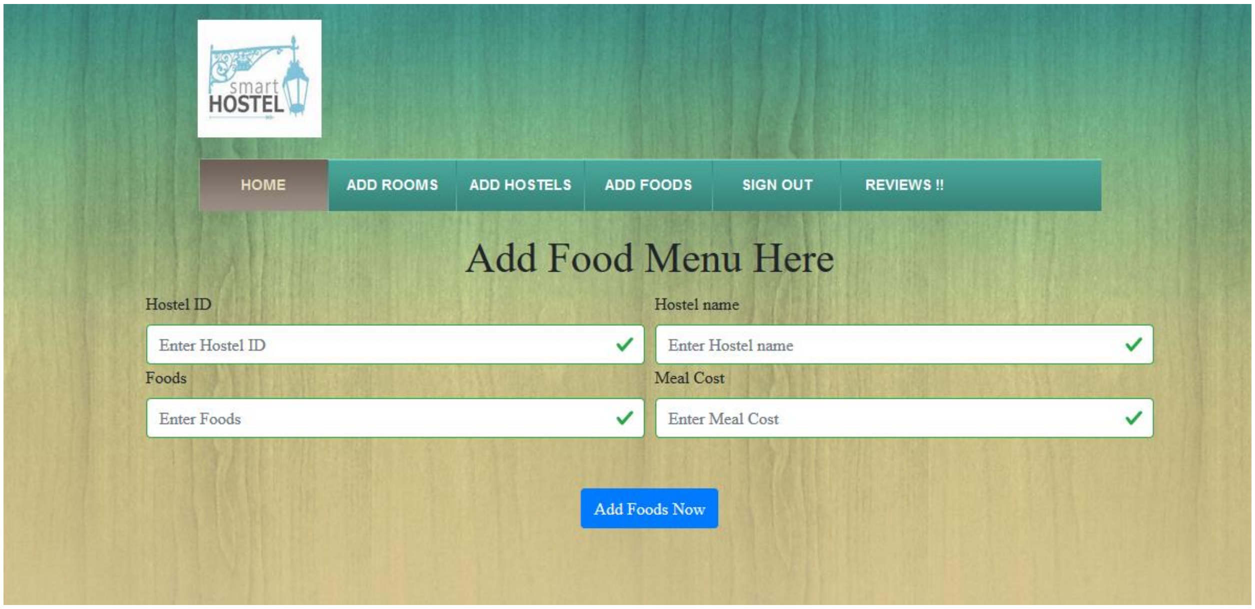


Upload Room Photos Here

6.1.14 Upload Hostel info by Manager



6.1.15 Upload Food info by Manager



6.1.16 Sign Out

Chapter 7

Conclusion

7.1 GitHub Link

7.2 Project Summary

I have started to develop this project from April. From the very starting of development of this project, it demands hard working, patients, persistency to meet the requirements of stakeholders. After that I have proposed the design then started to work.

Database plays a vital role for any application software. And so this is why, I have designed the database diagram having tables with proper relationship. After that, I have prepared the user interface and take their approval to continue to the next part. It is to be said that, the interface of my application is very simple and easy to understand. After completing that, I have started to write the core functionality of the project.

Developing project is not end of all tasks actually. There are some other important tasks to perform. And that is testing. It is also known as quality assurance also. Almost at every software company there have a quality assurance team. Their main responsibility is to find the loop holes or vulnerability of software. If there any bug remains before handover to the stakeholders, there is a change to ruin the whole project. So testing plan is very important. And after developing the project, I have tried to assure the quality of this project.

7.3 Limitations

For developing this project, I have faced some limitations. Now I will describe those in brief.

- **Sending reports by location**: In my application, there is not having any GPS system by which one can contract with others very easily and find location. End user need to find hostel on manual way.

7.4 Obstacles and Achievements

I believe that if there are not any obstacles to develop a project, then there doesn’t have any challenges. Because we know, challenges give us the opportunity to prove ourselves. Obstacles, challenges and achievements are like a path to the success. Before starting this project, I didn’t know the actual flow of software development life cycle. By developing this project, I have learnt to know how to have a row requirement from clients. After that I have learnt system analysis, database design and many things. My supervisor helps me a lot from the very beginning of the development of this project. I am very much lucky to have such kind of supervisor like her. She helped me lot.

There are some other obstacles and achievements also that I will describe below.

- **Lack of Actor's Engagement:** There are different types of stakeholders in my full web application. And each actors uses different functionality.

- **Scope Change:** Sometimes, some features need to be changed or modified. Then I need to follow reverse engineering process and again designed to meet the new requirements. It also made me frustrated sometimes.

7.5 Future Scope

I have learnt a lot throughout the whole development stage of this project. For making this project developed, I have also meets some young entrepreneurs and enthusiasts also. I am very much thankful to all of them as their idea and discussion gave me some opportunities to make my project complete. It will help me to work with similar type project in future also.

7.6 References

I have gained some knowledge from some platforms. Obviously I will mention those references. For making my project successful those resources help me a lot. Not I will mention the names below.

- www.google.com
- www.youtube.com
- www.github.com
- www.wikipedia.com
- www.php.net
- www.mysql.com
- www.mysql.com/products/workbench
- www.getbootstrap.com
- www.w3schools.com
- www.jquery.com
- www.sublimetext.com/3
- www.stakeoverflow.com
- www.notepad++.com
- https://draw.io/