



**“SE-231: System Analysis & Design Project”
in-charge**

Smart Hostel

Project Final Documents

PREPARED FOR

Khalid Been Md. Badruzzaman Biplob

Lecturer

Department of Software Engineering

Daffodil International University

PREPARED BY

Meher Durdana Khan Raisa

ID:192-35-2818

Sec-C

Department of Software Engineering

K.M.Mohiuddin

ID:192-35-2894

Sec-C

Department of Software Engineering

Acknowledgment

First of all, we are grateful to The Almighty Allah for giving us the ability to complete this project. Today we are feeling proud of ourselves. Because to be a student of Daffodil International University. And we are thankful to Daffodil International University for giving us a chance to prove ourselves by showing this project.

We are thankful to our Department Head In-Charge **Dr. Imran Mahmud**. And we want to thank our respected class teacher **Khalid Been Md. Badruzzaman Biplob** for supporting and giving your guideline and valuable advice.

“Smart hostel” is not a new system. But this software will give you a better experience. We hope every person who will use this system will be happy after completing it.

To build this software we have collected many types of information from different online sources. And also our friends help to build this software. At last, we also thank our parents for their unceasing encouragement and support. On record, our sense of gratitude to one and all who, directly or indirectly, have lent their helping hand in this project.

Thank You

Table of Content

Acknowledgment	1
Table of Content	2
Abstract	5
CHAPTER 1: INTRODUCTION	6
1.1. Overview	7
1.1.1. Background	7
1.1.2. Objectives	7
1.1.3. Scope	8
1.1.4. Assumptions and Constraints	9
1.1.5. Dependencies and Risks	9
1.2. Project Delivery	10
1.2.1 Deliverables	10
1.2.2 Timescales	10
1.2.3 Work Distribution	10
1.2.4 Project Resources	10
1.3. Summary	12
1.4. References	12
CHAPTER 2: SOFTWARE REQUIREMENT & SPECIFICATION	13
1.1 Functional Requirements	14
Hosteller feature:	14
Hostel Incharge feature:	16
1.2 Performance Requirements	21
1.2.1 Speed and Latency Requirements	21
1.2.2 Precision and Accuracy Requirements	21
1.2.3 Capacity Requirements	21
1.3 Dependability Requirements	21
1.3.1 Reliability and Availability	22
1.3.2 Robustness and Fault Tolerance Requirements	22
1.3.3 Safety-CriticalRequirements	22
1.4 Maintainability and Supportability	22
1.4.1 Maintenance Requirements	23
1.4.2 Supportability Requirements	23
1.4.3 Adaptability Requirements	23

1.5	Security Requirements	23
1.5.1	Access Requirements	24
1.5.2	Integrity Requirements	24
1.5.3	Privacy Requirements	24
1.6	Usability and Human Integrity Requirements	25
1.6.1	Ease of Use Requirements	25
1.6.2	Understand-ability and Politeness Requirements	25
1.6.3	Accessibility Requirements	26
1.6.4	User Documentation	26
1.7	Look and Feel Requirements	26
1.7.1	Appearance Requirements	27
1.7.2	Style Requirements	27
1.8	Operational and Environmental Requirements	27
1.8.1	Expected Physical Requirements	27
1.8.2	Requirement for Interfacing with Adjacent System	28
1.8.3	Release Requirements	28
1.9	Legal Requirements	28
1.9.1	Compliance Requirements	28
1.9.2	Standard Requirements	28
CHAPTER 3:	USE CASE DIAGRAM	29
3.1	Use case diagram for Smart hostel	30
CHAPTER 4:	USE CASE DESCRIPTION	31
4.1	Use case description: Login	32
4.2	Use case description: Add New Hosteller	33
4.3	Use case description: View Hosteller	34
4.4	Use case description: Meal Portal	35
4.5	Use case description: View Menu	36
4.6	Use case description: Add Meals	37
4.7	Use case description: Cancel Meal	38
4.8	Use case description: Payment Portal	39
4.9	Use case description: Update Payment Portal	40
4.10	Use case description: Payment ledger	41
4.11	Use case description: Payment scheme	42
4.12	Use case description: Online payment	43
4.13	Use case description: Query and Complaint	44
CHAPTER 5:	ACTIVITY DIAGRAM	45
5.1	Activity diagram for LOGIN	46

5.2 Activity diagram for View Hosteller	47
5.3 Activity diagram for Add Hosteller	48
5.4 Activity diagram for Meal Portal	49
5.5 Activity diagram for view Menu	50
5.6 Activity diagram for Add meals	51
5.7 Activity diagram for Cancel meal	52
5.8 Activity diagram for Payment Portal	53
5.9 Activity diagram for Payment scheme	54
5.10 Activity diagram for Payment Ledger	55
5.11 Activity diagram Online Payment	56
5.12 Activity diagram Update Payment Portal	57
5.13 Activity diagram Query & Complaint	58
CHAPTER 6: SEQUENCE DIAGRAM	59
6.1 Sequence diagram Login:	60
6.2 Sequence diagram Add Hosteller:	61
6.3 Sequence diagram View Hosteller:	62
6.4 Sequence diagram Meal Portal:	63
6.5 Sequence diagram View Menu:	64
6.6 Sequence diagram Add Meals:	65
6.7 Sequence diagram Cancel meal:	66
6.8 Sequence diagram Payment Portal:	67
6.9 Sequence diagram Payment Ledger:	68
6.10 Sequence diagram Payment Scheme:	69
6.11 Sequence diagram Online Payment:	70
6.12 Sequence diagram Update Payment Portal:	71
6.13 Sequence diagram Query & Complaint:	72
CHAPTER 7: ENTITY RELATIONSHIP DIAGRAM	73
7.1 Entity Relationship Diagram (ERD) for “Smart hostel”	74

Abstract

This project entitled “Smart Hostel” is a web-based application to manage hosteller, hostel meal systems, and Payment. There is a large number of hostels in our country. The hostel meal system is developed for automating the activities of the hostel meal process and rent payment. The software will be a great relief to the hostel owner. This software will help the hostel in-charge in case of managing hostel activities and maintaining reports. Admin/Hostel in-charge can handle all the processes so easily.

CHAPTER 1

INTRODUCTION

1.1. Overview

1.1.1. Background

A hostel by definition is an establishment that provides cheap food and lodging for a specific group of people, it is also seen as a home for students when staying away from their home. It has large well-ventilated dormitories and single rooms and is situated in the establishment. Providing clean and calm hostel accommodation is one of the keys responsible for hostel management. They are staffed 24 hours and most have security cameras installed. If you are ever uncomfortable or don't feel safe, then you have to tell the staff and ask for a room change. Sleeping in a hotel room is challenging enough for most of us, but it's nothing compared to hostels. Hostels introduce a whole new variety of elements that make you acutely aware you're not in your bed. You're in a strange city, sleeping in a room full of strangers. Sometimes we don't know who lives in the room next to us. Sometimes hostellers face many problems like what's on the menu or to cancel a meal or keeping track of the monthly meal cost. Many times we need to contact our service provider directly to find out any complaints or information. If there are a lot of people in the dormitory then a lot of staff is needed to take care of all the facilities. If the number of staff is less then the maintenance of the hostel becomes much more complicated.

1.1.2. Objectives

The smart hostel is a software-based system that will resolve the traditional problem of maintaining Hosteler records, queries, meal management, and communication. As the cost of staff and maintenance are increasing day by day also the number of web surfers is increasing day by day, a web-based solution will be certainly more accurate than a traditional desktop-based solution.

In this project, the users will be-

1. Hostel incharge
2. Hosteller

The list of operations that the system will provide are-

1. This software allows hostels to view the food portal by looking at what is on the menu today, canceling a meal, or requesting an extra meal.

-
2. The payment portal will help you pay your rent and food bills online. Also, helps to see the status of payment like payment scheme and payment ledger.
 3. Add a new hostel, create and manage its contract, edit the dormitory information feature for the hostel in-charge.
 4. Provide the Hosteller ID to registered users.
 5. Warn and report any hosteller about his/her rental and meal payment by sending feedback.
 6. Send any query or complaint through this system feature for hostellers.
 7. This system will provide some special features for hostel charges such as review food requests, review meal cancellation requests, review payment reports and alert any hosteller, send review inquiries and complaints and respond to each review.

1.1.3. Scope

Smart Hostel will help maintain an ideal and modern environment inside a hostel. The hostel in-charge will use the system to keep track of the hostellers attached to them. This system will provide different facilities to a manager for maintaining the entire Hostel. Any hosteller can send any kind of query or complaint or maintenance request directly to the manager or caretaker. The hostel in-charge can review the query or complaints and take steps. By using this software Hostellers can view the menu from the menu portal, can request an extra meal or any special meal, or cancel a meal. From the admin site, the hostel in charge can add meals, review meal requests, and send feedback about confirmation or other information. The hostel in charge can cancel meals for anyone to everyone. Another main feature of this system is the payment portal. In this portal, any hosteller can view his/her Payment Scheme and Payment ledger. In a word, he will get all the information related to his payment from here, and also the hosteller can pay online by using the software. The hostel in-charge will regularly update any hosteller's payment scheme or update payment information and send notifications about payment confirmation.

All the information about hostellers admins will be automatically updated into a database. So again a smart hostel software will be very user-friendly. It will reduce the maintenance cost.

1.1.4. Assumptions and Constraints

It is assumed that the user is comfortable with the computer. Every user should know how to use Smart Hostel.

The user interface is in English as a result people lacking English skills will face difficulty in using the system. Login and password are used for the identification of users and there is no facility for guests.

1.1.5. Dependencies and Risks

The user must have web access to use the system. The main risk behind implementing the project is security. If somebody hacks the system then it will be a total disorder. So during development, it will be one of our major concerns. Another concern is having common bugs such as the common users are having the same functionalities as the hostellers and hostel in-charge.

1.2. Project Delivery

1.2.1 Deliverables

The following contents will be delivered with the project:

- a) Project CD
 - i. Project Demo
 - ii. User manual along with Tutorial
- b) Documentation

1.2.2 Timescales

The time frame for implementing the project is given in Figure 1.2.1.

1.2.3 Work Distribution

The work distribution of the project is given in Table 1.2.1.

1.2.4 Project Resources

The resources required to finish the project are given in Table 1.2.2.

Duration									
	7 days	7 days	7 days	7 days	7 days	7 days	7 days	7 days	7 days
Project Proposal									
SRS									
Designing									
Coding									
Testing									

Finalization									
--------------	--	--	--	--	--	--	--	--	--

Figure 1.2.1: Time frames for project implementation

Project Proposal	K.M.Mohiuddin Meher Durdana Khan	7days
Software Requirement Specification	K.M.Mohiuddin Meher Durdana Khan	7 days
Software Design	K.M.Mohiuddin Meher Durdana Khan	21 days
Coding	K.M.Mohiuddin Meher Durdana Khan	14 days
Software Testing	K.M.Mohiuddin Meher Durdana Khan	7 days
Project Finalization	K.M.Mohiuddin Meher Durdana Khan	7 days

Table 1.2.1 Work Distribution

Hardware Requirements		
Processor	RAM	Hard Disk Space

Pentium II or higher	64 Mb or higher	128 Mb or higher
----------------------	-----------------	------------------

Software Requirements	
Operating System	Database
For users no specific OS is required. The server machine must have Windows XP/Vista/7 along with .NET framework 4 and IIS.	SQL Server 2008

Table 1.2.2 Project Resources

1.3. Summary

By using this software, the hostel incharge will always be connected with all the members of a crowded hostel and members will also be able to inform the incharge in case of any inconvenience. There is no need to keep an account of the cost of the hostel. This software will keep all the accounts by itself. Hostel Incharge will be able to address all the issues by using this system by staying at home.

1.4. References

1. HowTo: Write a project proposal [Online] URL:

[How to Write a Project Proposal \(With Example\)](#)

2. Homie -smart hostel living [Online] URL:

[Homie - Smart Hostel Living](#)

3. Hostel management system [Online] URL:

[Hostel Booking,Hostel Management System Software: School,College](#)

CHAPTER 2

SOFTWARE REQUIREMENT & SPECIFICATION

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

1.1 Functional Requirements

The Functional Requirements Specification is designed to be read by a general audience. Readers should understand the system, but no technical knowledge should be required to understand the document.

Hosteller feature:

FR-001	Log-In
Description	After entering the URL in any internet browser, in the software index page, the user must have to register himself first. After every time of access, he must log-in first before entering the main software. All users are mandatory for login. Only authorized users can log in the software.
Stakeholder	Hosteller

FR-002	Select Meal Portal
Description	It allows hostellers to go through some features like add an extra meal, request a special meal, cancel the meal.
Stakeholders	Hosteller

FR-003	Add Extra Meal
Description	Hostellers can request for extra meals in the meal portal through the system.
Stakeholders	Hosteller

FR-004	Request Special Meal
Description	Hostellers can request for a special type of meal in the meal portal through the system.
Stakeholders	Hosteller

FR-005	Cancel Meal
Description	Hostellers can cancel his/her meal by using this system.
Stakeholders	Hosteller

FR-006	Select Payment Portal
Description	It allows hostellers to go through some features like payment ledger, payment scheme and can view the current payment status & also pay online.
Stakeholders	Hosteller

FR-007	View payment status
Description	From the payment portal page, hostellers can view his/her payment status.
Stakeholders	Hosteller

FR-008	View Payment Scheme
Description	Here hostellers can see all the information about how much money has been spent in any sector of the hostel.
Stakeholders	Hostellers

FR-009	View Payment Ledger
Description	Hostellers will be able to see when and how much money has been deposited in any sector
Stakeholders	Hostellers

FR-010	Online Payment
Description	Hostellers can pay online by going through the online payment feature in the payment portal.
Stakeholders	Hosteller

FR-011	Select Query and complaint
Description	If any Hosteller has any information or report, Hosteller can report it from here, or if Hosteller has any complaint against any other member, he can say it directly to the authorities.
Stakeholders	Hosteller

FR-012	Send Query
Description	Hostellers can send queries for any information through this system.
Stakeholders	Hosteller

Hostel Incharge feature :

FR-013	Log-In
Description	After entering the URL in any internet browser, in the software index page, the user must log in. After every time of access, he must log in first before entering the main software. All users are mandatory for login. Only authorized users can log in the software.
Stakeholder	Hostel InCharge

FR-014	Search hosteller
Description	Hostel InCharge can search the details of the students and the system displays the specific member
Stakeholders	Hostel InCharge

FR-015	View hosteller Details
Description	Hostel InCharge, as well as members, can view the entire details of the students or members who are registered.
Stakeholders	Hostel InCharge

FR-016	Add new hosteller
Description	Hostel InCharge can add a new hosteller and its details in the system.
Stakeholders	Hostel InCharge

FR-017	Select Edit Hosteller
Description	This module helps Hostel InCharge to update hosteller information or report the hosteller. Hostel InCharge can update the details of the members and we store these details in a database.
Stakeholders	Hostel InCharge

FR-018	Edit Hosteller information
Description	This module helps Hostel InCharge to update hosteller information. Hostel InCharge can update the details of the members and we store these details in a database.
Stakeholders	Hostel InCharge

FR-019	Report & Warn the Hosteller
Description	If someone makes a complaint against that member, InCharge can warn him and inform the complainant about his complaint's consequences.
Stakeholders	Hostel InCharge

FR-020	Review Query and Complaints
Description	Hostel InCharge can review any query or can review Complaints made by other hostellers
Stakeholders	Hostel InCharge

FR-021	Send feedback
Description	Hostel inCharge can send feedback after reviewing someone's review or query or Complaints.
Stakeholders	Hostel InCharge

FR-022	Report & Warn Any Hosteller
Description	If someone makes a complaint against another member, InCharge can warn that member and inform the complainant about his complaint's consequences.
Stakeholders	Hostel InCharge

FR-023	Select Payment Portal
Description	It allows Hostel InCharge to go through some features like update payment portal review payment
Stakeholders	Hostel InCharge

FR-024	Select update Payment Portal
Description	It allows Hostel InCharge to go through some features like update payment ledger, update payment scheme .
Stakeholders	Hostel InCharge

FR-025	Update Payment Scheme.
Description	All the information about how much money has been spent in any sector of the hostel will be given here
Stakeholders	Hostel InCharge

FR-026	Update Payment Ledger
Description	Can show when and how much money has been deposited in any sector
Stakeholders	Hostel InCharge

FR-027	Review payment
Description	If any member has any problem or inquiry regarding his deposit or if the hostel in-charge wants to increase or change any member's deposit, he can do so from here.
Stakeholders	Hostel InCharge

FR-028	Send payment confirmation
Description	If a member submits money and accepts it by the authority and updated on the portal, submission confirmation will be through this option.
Stakeholders	Hostel InCharge

FR-029	Select Meal Portal
Description	It allows hostellers to go through some features like add an extra meal, request a special meal, cancel the meal.
Stakeholders	Hostel InCharge

FR-030	View Menu
Description	Hostel InCharge as well as members can view the menu list from the meal portal.
Stakeholders	Hostel InCharge

FR-031	Review meal requests
Description	If anyone craves extra food, the hostel InCharge can check it out.
Stakeholders	Hostel InCharge.

FR-032	Add Meal
Description	Hostel InCharge can add meals in the meal portal through the system.
Stakeholders	Hostel InCharge

FR-033	Cancel meal for all
Description	Hostel InCharge can cancel any meal for all
Stakeholders	Hostel InCharge.

FR-034	Review meal cancellation requests
Description	If someone requests to cancel a mail, the in-charge will verify the reason for his or her cancellation request.
Stakeholders	Hostel InCharge

FR-035	Confirm cancellation
Description	From this option, InCharge can confirm cancellation for requested hosteller
Stakeholders	Hostel InCharge

FR-036	Cancel Meal
Description	Hostel InCharge can cancel a meal for any member by using this system
Stakeholders	Hostel InCharge

1.2 Performance Requirements

A requirement that specifies a performance characteristic that a system or system or system component must possess; for example, speed, accuracy, frequency.

1.2.1 Speed and Latency Requirements

The system requires a fair amount of speed.

PR-01	The Landing page will respond within a second
Description	While the user's browsing the system the landing page will show within a second. It also depends on the user's internet connection.
Stakeholders	Hostel InCharge, Hosteller.

1.2.2 Precision and Accuracy Requirements

There are no specific precision and accuracy requirements

1.2.3 Capacity Requirements

The system can manage all the information about hostel members.

PR-02	Initially, the system will store 2000- member information
Description	The information of the hosteller will be stored in a database.
Stakeholders	Hostel InCharge

1.3 Dependability Requirements

The flexibility of current frameworks encourages system architects to enable reconfiguration mechanisms that refocus the available, safe resources to support

the most critical services rather than over-provisioning to build failure-proof systems. Therefore, these requirements are essentials.

1.3.1 Reliability and Availability

To support global and smooth operations the system must be available around the clock. On the other hand, most services in this system are not mission-critical. Even better, the game posting can handle times of downtime as the users usually interact with high availability from third party websites. This system will be able to catch up with their data once it is up and running again.

DR-01	The system must be available 24x7
Description	<ul style="list-style-type: none">• The system must be available 24 hours in a day• The system must be updated regularly
Stakeholders	Hostel InCharge.

1.3.2 Robustness and Fault Tolerance Requirements

The system will almost ensure 0% crash in any single minor error and do not give any wrong calculation.

DR-02	The system handles over access and system errors
Description	Sometimes multiple users can over access this system. The system can handle multiple user access
Stakeholders	N/A

1.3.3 Safety Critical Requirements

There are no specific safety critical requirements

1.4 Maintainability and Supportability

Supportability is the degree to which system design characteristics and planned logistics resources meet system requirements. Supportability is the capability of a total system design to support operations and readiness needs throughout the life-cycle of a system at an affordable cost.

1.4.1 Maintenance Requirements

MS-01	The system helps to update any information in any time
Description	The Hostel InCharge can change or update any information in any situation
Stakeholders	Hostel InCharge.

1.4.2 Supportability Requirements

To understand the system's behavior on a technical support required by the system operator. The reason for reading them might be

- System malfunction has occurred and the system operator must find the exact point of time when this happened
- System produces wrong results and the developers must be able to reproduce the data flow through the system
- Hacker tried to breach the system's security mechanisms and the system operator must understand what he did

1.4.3 Adaptability Requirements

There are no specific adaptability requirements

1.5 Security Requirements

There are no access requirements beside those that have been outlined in the below:

- The software must validate all user input to ensure it does not exceed the size specified for that type of input

-
- The server must authenticate every request accessing the restricted Web pages
 - After authenticating the browser, the server must determine whether that browser is authorized to access the requested restricted Web pages
 - The system must have security controls to protect against denial-of-service attacks
 - The system must encrypt sensitive data transmitted over the Internet between the server and the browser

To get access to this system or a specific module the system must provide a central authentication mechanism. To prevent anyone from being stolen all user's passwords must be encrypted in the hash process.

1.5.1 Access Requirements

To get access to the system, the system provides authorization/authentication. This system uses various modules.

SR-01	The system provides security strategies.
Description	The system is designed in a way that allows all modules to access a mechanism that provides security services.
Stakeholders	Hostel InCharge, Hosteller

1.5.2 Integrity Requirements

To protect credentials of users from being stolen, all passwords are stored in encrypted form. The Requirements significantly reduces the value of stolen user credentials, it is not easy to decrypt the password.

1.5.3 Privacy Requirements

The system provides a protection of the database in the server. However, the system will have to increment this level of protection because of the personal data mode available on the system & the larger share of people that will be having

access to it through the system's registration. The user's privacy will be granted by the limited access that the login process is going to give to the database.

SR-02	All data will be protected
Description	The main requirement in the context is the generation of hostel member's data for analysis.
Stakeholders	Hostel InCharge, Hosteller

1.6 Usability and Human Integrity Requirements

These Requirements define how to meet the physical and cognitive needs of the intended users of your website or application

1.6.1 Ease of Use Requirements

The system is easy to use and can easily be understandable.

UH-01	The system must be usable for Smart Hostel members with all associate stakeholders.
Description	The system indicates the several possibilities that the hostel member must go on in using the system. The hostel members can undo any of the operations.
Stakeholders	Hostel InCharge, Hosteller

1.6.2 Understand-ability and Politeness Requirements

This section describes more requirements of Smart Hostel to add more features in future

UH-02	The features of Smart Hostel system
--------------	-------------------------------------

Description	The system is more efficient and easier to use with more added features. The system is understand-ability for both users. The system will not use any term that is not specified in this system.
Stakeholders	Hostel InCharge

1.6.3 Accessibility Requirements

There are no access requirements beside those that have been outlined in the below:

AR-1: Log in as a Hostel InCharge

AR-2: Log in as a Hosteller

AR-3: Log out as a Hostel InCharge

AR-4: Log out as a Hosteller

To get access to this system or a specific module the system must provide a central authentication mechanism. To prevent anyone from being stolen, all user's passwords must be encrypted in the hash process.

1.6.4 User Documentation

UH-03	The system developer documentation
Description	To develop this project, we have specified requirements for user documentation. The teams are involved in this project documentation.
Stakeholders	System Developer

1.7 Look and Feel Requirements

The look and feel requirements describe the intended spirit, the mood, or the style of the product's appearance. These requirements specify the intention of the appearance, and are not a detailed design of an interface.

1.7.1 Appearance Requirements

It should be clear to the Hostel InCharge and alumni which fields need to be filled and which can be left blank in this system.

LF-01	Labels of mandatory fields must be bold
Description	Labels of mandatory fields must be bold to identify them as being mandatory.
Stakeholders	Hostel InCharge.

1.7.2 Style Requirements

We will provide a web-based user interface. This requirement does not only define the necessity to use a CSS but although the requirements regarding the CSS's content as well as CSS framework like bootstrap.

LF-02	The look and feel must be controllable using a style sheet.
Description	The styling of the elements of the web-based user interface will be defined using CSS, JS and bootstrap.
Stakeholders	Hostel InCharge.

1.8 Operational and Environmental Requirements

This requirement focuses on how the users will operate the system, including interfaces and interoperability with other systems. The requirements establish how well and under what conditions the system must perform.

1.8.1 Expected Physical Requirements

There are no specific expected physical requirements

1.8.2 Requirement for Interfacing with Adjacent System

There is no specific interfacing with adjacent system requirements

1.8.3 Release Requirements

There are no specific release requirements but in the project schedule section it was described briefly.

1.9 Legal Requirements

These requirements consider any violence of rules and regulation and which rules should be followed to maintain this system

1.9.1 Compliance Requirements

There are no specific compliance requirements

1.9.2 Standard Requirements

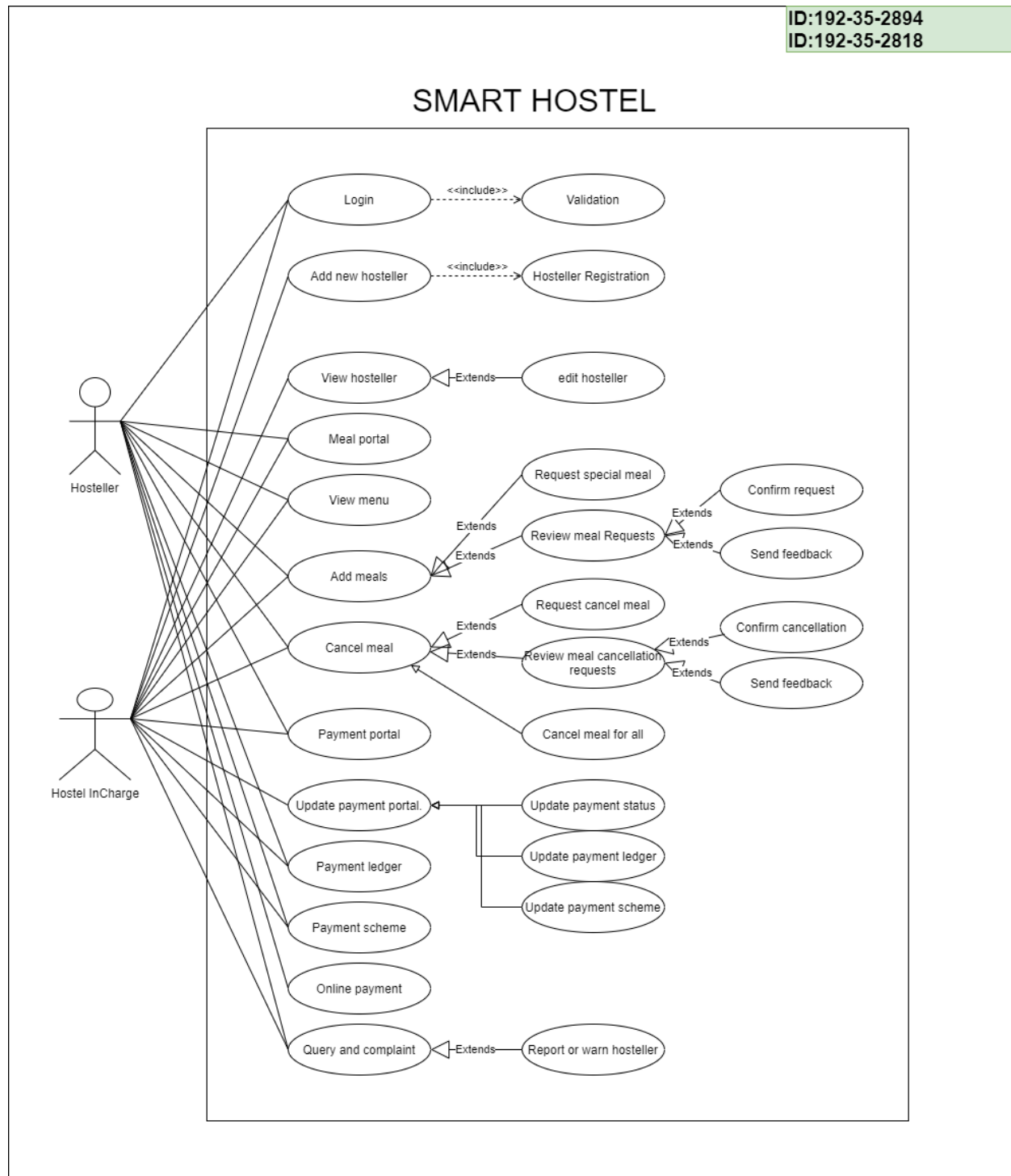
There are no specific standard requirements

CHAPTER 3

USE CASE DIAGRAM

This is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

3.1 Use case diagram for Smart hostel



CHAPTER 4

USE CASE DESCRIPTION

This is a text-based narrative of a functionality comprising detailed, step-by-step interaction between the actor and the system. It describes the outcomes of an action taken to accomplish a specific goal.

4.1 Use case description: **Login**

Use Case	Login	
Goal	To access the system	
Preconditions	Must have an account on this system	
Success End Condition	Successfully logged into the system	
Failed End Condition	Failed to login to	
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:	None	
Trigger	To enter the system	
Description / Main Success Scenario	Step	Action
	1	View log-in window.
	2	Click on „Log In
	3	Input User name or User ID & Password.
	3.1	Use case: 'Validation'
	4	Proceed to the next activity.
Alternative Flows	Step	Branching Action
	3a	User name or password incorrect
	4a	Login again
Quality Requirements	Step	Requirement
	3a	Provide username & password information correctly
	3a1	pop up message will stay no later than 3 seconds after it is sent by the System

4.2 Use case description: **Add New Hosteller**

Use Case	Add New Hosteller	
Goal	Add a new hosteller in the system	
Preconditions	Before adding a new hosteller hostel InCharge need to login into the portal	
Success End Condition	The hosteller bus is now added to the system	
Failed End Condition	The new hosteller did not add in the system	
Primary Actors:	Hostel InCharge	
Secondary Actors:		
Trigger	System registers new hosteller in the system	
Description / Main Success Scenario	Step	Action
	1	Hostel InCharge login in the portal
	1.1	Hostel InCharge choose the add hosteller option
	1.2	Hostel InCharge provide all information for the new hosteller to add to the system
	2	System verifies the given hosteller information
	3	System confirms the action
	4	Use case 'Hosteller Registration'
	5	System adds a new hosteller in the system
Alternative Flows	Step	Branching Action
	1a	The given information is already in the system
	1a1	The system doesn't add the new hosteller
	2a	The given information is not correct
	2a1	The system doesn't add the new hosteller
	3a	Hostel InCharge didn't give all the required information
	3a1	The system doesn't add the new hosteller
Quality Requirements	Step	Requirement
	3a	Provide new hosteller information correctly
	4a	Properly register new hosteller in the system

4.3 Use case description: **View Hosteller**

Use Case	View Hosteller	
Goal	Hostel InCharge selects to see the designated hosteller details, expect all information about it	
Preconditions	To view the hosteller, the first hostel InCharge need to select the hosteller menu	
Success End Condition	The system provides the designated hosteller	
Failed End Condition	The system doesn't have any information on the selected hosteller	
Primary Actors:	Hostel InCharge	
Secondary Actors:		
Trigger	Expected hosteller found	
Description / Main Success Scenario	Step	Action
	1	Hostel InCharge selects the menu to view hosteller details
	2.1.1	Hostel InCharge search Hosteller from the search bar
	2.1.2	Hostel InCharge select the Hosteller
	2.2.1	Hostel InCharge select Hosteller from the suggestions list
	3	System shows Hosteller information
Alternative Flows	Step	Branching Action
	1	System doesn't have any records of the Hosteller
	2	Again, select another Hosteller
	3	Give selected Hosteller information
Quality Requirements	Step	Requirement

4.4 Use case description: **Meal Portal**

Use Case	Meal Portal	
Goal	Hostel InCharge doesn't as well as hosteller, can access the features in the meal portal do choose or change anything	
Preconditions	User must log in to the system	
Success End Condition	User can access the meal portal successfully	
Failed End Condition	The user was unsuccessful to access the meal portal	
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:		
Trigger	The features in the meal portal are shown	
Description / Main Success Scenario	Step	Action
	1	User logged in to the system
	2a 2b	User chooses the meal from the other portals in the system The system shows the meal portals features
	3	By selecting the meal portal, its features are accessible
	4	User can access meal portals features only from the meal portal
Alternative Flows	Step	Branching Action
	2a	User selects the wrong portal
	2a1	System shows wrong portals features
	3a	User couldn't get access to meal portals features from elsewhere
Quality Requirements	Step	Requirement
	1a	Must confirm the request
	2a	Review the request properly

4.5 Use case description: **View Menu**

Use Case	View Menu	
Goal	User selects the meal portal to see the meal menu	
Preconditions	For to see the meal catalog, first we need to select the meal portal	
Success End Condition	The system provides the designated menu	
Failed End Condition	The system didn't show the meal menu	
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:		
Trigger	The desired result found	
Description / Main Success Scenario	Step	Action
	1	User select the meal portal
	2.1	User selects the view menu
	2.2	System provides user the meal menu
Alternative Flows	Step	Branching Action
	1	System doesn't have any records of the meal list
	2	Again, user reload system
Quality Requirements	Step	Requirement
	1	
	2	

4.6 Use case description: **Add Meals**

Use Case	Add Meals	
Goal	Hosteller request meal in the system Hosteller InCharge accept meal request and send feedback	
Preconditions	Before requesting a new meal, the hosteller needs to choose to add a meal option Hosteller needs to review the request to accept them and send feedback	
Success End Condition	Hosteller was successful in requesting a meal Hosteller reviewed the request and sent feedback	
Failed End Condition	Meal request wasn't sent Request & feedback wasn't sent	
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:		
Trigger	System sends notification about the meal request	
Description / Main Success Scenario	Step	Action
	1	User choose the add meal feature
	1a	Use case 'Request extra meal'
	1b	Use case 'Review meal request'
	2	Hosteller choose the Use case 'Request extra meal'
	2a	Hostel InCharge choose Use case 'Review meal request'
	3	System confirms the action
	4	Use case 'Confirm request'
	4a	Use case 'Send Feedback'
	5	Hostel InCharge choose to confirm the request and send feedback
Alternative Flows	Step	Branching Action
	2	Meal request sending was failed
	2a	Meal request was cancelled
	3a	The meal request wasn't reviewed
	2a1	The system doesn't add the new meal
	4	Meal request was ignored
	4a	Confirmation failed
Quality Requirements	Step	Requirement
	1	Provide new meal information correctly
	2	Proper required information for the new meal

4.7 Use case description: **Cancel Meal**

Use Case	Cancel Meal	
Goal	Hostel InCharge or Hostel InCharge can cancel meals.	
Preconditions	Must be logged into the system Cancel Meal Before the deadline	
Success End Condition	Hosteller requested to cancel a meal Hostel InCharge canceled the meal	
Failed End Condition	Hostel InCharge can not cancel the meal	
Primary Actors:	Hostel InCharge, Hosteller.	
Secondary Actors:		
Trigger	Hosteller wants to cancel a meal	
Description / Main Success Scenario	Step	Action
	1	Select Meal portal
	2	Select cancel meal
	3.1	Use case 'Request cancel meal'
	3.1.1	Select which meals want to cancel
	3.1.2	Add reasons to cancel
	3.2.	Use case 'Review meal cancellation requests'
	3.2.1	Use case 'Confirm request'
	3.2.2	Use case 'Send feedback'
Alternative Flows	Step	Branching Action
	3a	Cannot request to cancel meal due to time limit
	3a1	Already canceled
	3b	Request declined
Quality Requirements	Step	Requirement

4.8 Use case description: **Payment Portal**

Use Case	Payment Portal	
Goal	This portal allows accessing some other feature related to the payment	
Preconditions	Must be logged into the system	
Success End Condition	Can access the portal	
Failed End Condition	Cannot access the portal	
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:		
Trigger	Want to access the portal	
Description / Main Success Scenario	Step	Action
	1	User logged in to the system
	2	System provides some portal to choose from
	3	User chooses payment portal
	4	By choosing the payment portal user can now access use its feature
Alternative Flows	Step	Branching Action
	3a	User couldn't log in
	3a1	System couldn't provide the portal options
	4a	User selects the wrong portal
	4a1	Its features were unavailable
Quality Requirements	Step	Requirement

4.9 Use case description: **Update Payment Portal**

Use Case	Update Payment Portal	
Goal	Hostel InCharge can update payment portal features through the system	
Preconditions	Before updating payment portals features, the hostel incharge must select the payment portal	
Success End Condition	Payment portals features were successfully updated	
Failed End Condition	Updating the payment portal features were failed	
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:		
Trigger	After selecting update the payment portal, the system shows the other features in the payment portal	
Description / Main Success Scenario	Step	Action
	1a	Select payment portal
	2a	Use case 'Update Payment Status' Use case 'Update Payment ledger' Use case 'Update Payment Scheme'
	3a	User choose Use case 'Update Payment Status' to solve any payment status issue for any hosteller in the system User choose Use case 'Update Payment ledger' to solve any ledger related issue for any hosteller User choose Use case 'Update Payment Scheme' to solve any payment scheme issue for any hosteller
	3b	
	3c	
	4a	System confirms the action
	5a	System updates the chosen features
Alternative Flows	Step	Branching Action
	1a	Payment portal wasn't provided
	2a	The system didn't present the updated payment portal features
	4a	System didn't confirm the actions
	5a	System failed to update chosen feature
Quality Requirements	Step	Requirement
	4a	Review all the information for the designated feature to update
	4b	Approve all the actions to update

4.10 Use case description: **Payment ledger**

Use Case	Payment ledger	
Goal	Payment ledger will provide info on when and how much money has been deposited in any sector.	
Preconditions	Must be logged into the system	
Success End Condition	Can access the ledger	
Failed End Condition	Cannot access the ledger	
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:		
Trigger	Want to access the ledger	
Description / Main Success Scenario	Step	Action
	1	User logged in to the system
	2	System provides some portal to choose from
	3	User chooses payment portal User choose Payment ledger
	4	By choosing the payment ledger user can now access to use its feature
Alternative Flows	Step	Branching Action
	3a	User couldn't log in
	3a1	System couldn't provide the portal options
	4a	User selects the wrong portal
	4a1	Its features were unavailable
Quality Requirements	Step	Requirement
	4	Provide correct username & password
	7	Choose the correct portal

4.11 Use case description: **Payment scheme**

Use Case	Payment scheme	
Goal	Hosteller will be able to know how much money has to be paid in which sector through a payment scheme	
Preconditions	Must be logged into the system	
Success End Condition	Can access the scheme	
Failed End Condition	Cannot access the scheme	
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:		
Trigger	Want to access the scheme	
Description / Main Success Scenario	Step	Action
	1	User logged in to the system
	2	System provides some portal to choose from
	3	User chooses payment portal User choose a Payment scheme
	4	By choosing the payment scheme user can now access use its feature
Alternative Flows	Step	Branching Action
	3a	User couldn't log in
	3a1	System couldn't provide the portal options
	4a	User selects the wrong portal
	4a1	Its features were unavailable
Quality Requirements	Step	Requirement
	4	Provide correct username & password
	7	Choose the correct portal

4.12 Use case description: **Online payment**

Use Case	Online payment	
Goal	To pay money through this system	
Preconditions	Before using this feature, a user must log in	
Success End Condition	Payment is completed through the system	
Failed End Condition	The payment wasn't successful	
Primary Actors:	Hosteller	
Secondary Actors:		
Trigger	Hosteller wants to pay	
Description / Main Success Scenario	Step	Action
	1	Hostel InCharge login in the portal
	1.1	Select the payment portal
	1.2	Select Online payment
	2	Select payment method
	3	Pay online
Quality Requirements	Step	Requirement
	1	Provide new hosteller information correctly
	2	Properly register new hosteller in the system

4.13 Use case description: **Query and Complaint**

Use Case	Query and Complaint	
Goal	User will be able to ask any and also send a complaint through the system	
Preconditions	Before using this feature user need to login into the portal	
Success End Condition	Hosteller can successfully send the query or complaint Hostel InCharge can warn or report hostellers through this system	
Failed End Condition		
Primary Actors:	Hostel InCharge, Hosteller	
Secondary Actors:		
Trigger	System provides choosing the option to send complaint or report	
Description / Main Success Scenario	Step	Action
	1	User logged into the system
	2	Use case 'Report or Warn Hosteller'
	3a	Hosteller can ask query through the system
	3b	Hosteller can Send complain about any issue
	4	Hosteller InCharge can warn another hosteller in whose name the complaint is made
	5	Hostellers are informed by the system if the hostel InCharge send a report to them
Alternative Flows	Step	Branching Action
	2a	Hosteller's query wasn't sent
	3a	Reported Hosteller is not in the system
	3a1	Hostellers' complaint wasn't sent
	7a	Hostel InCharge couldn't send the report due to the hosteller not being in the system
Quality Requirements	Step	Requirement

CHAPTER 5

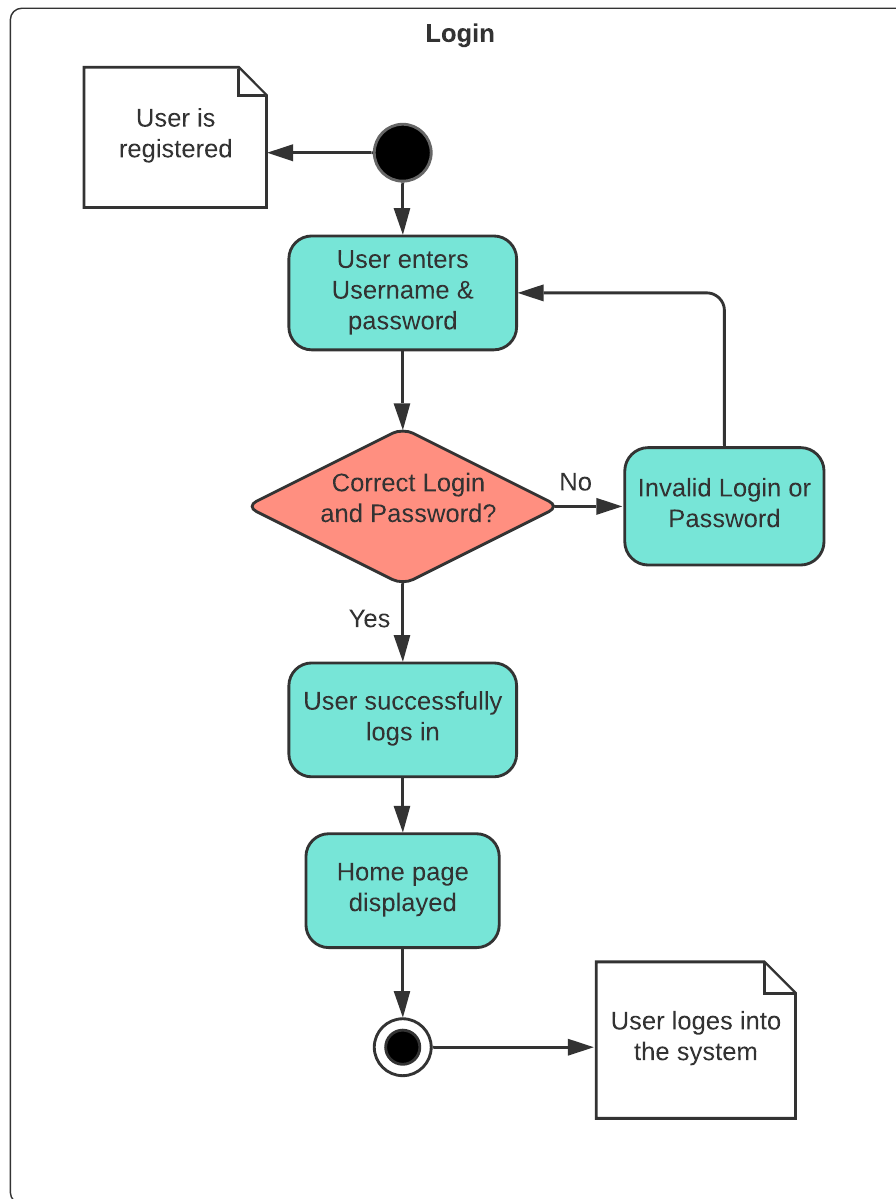
ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency.

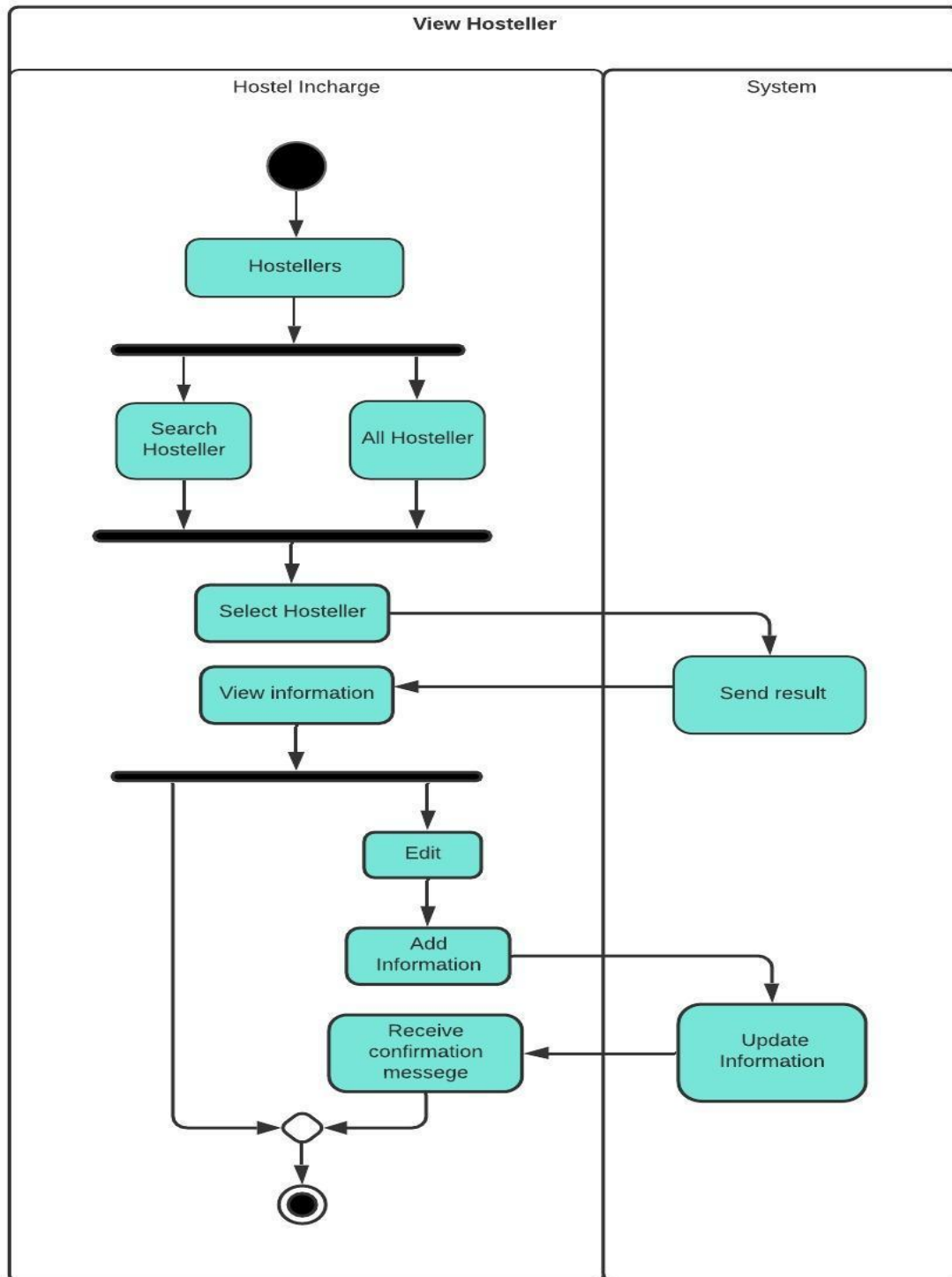
5.1 Activity diagram for **LOGIN**

Smart hostel

Mohiuddin , Raisa | April 15, 2021



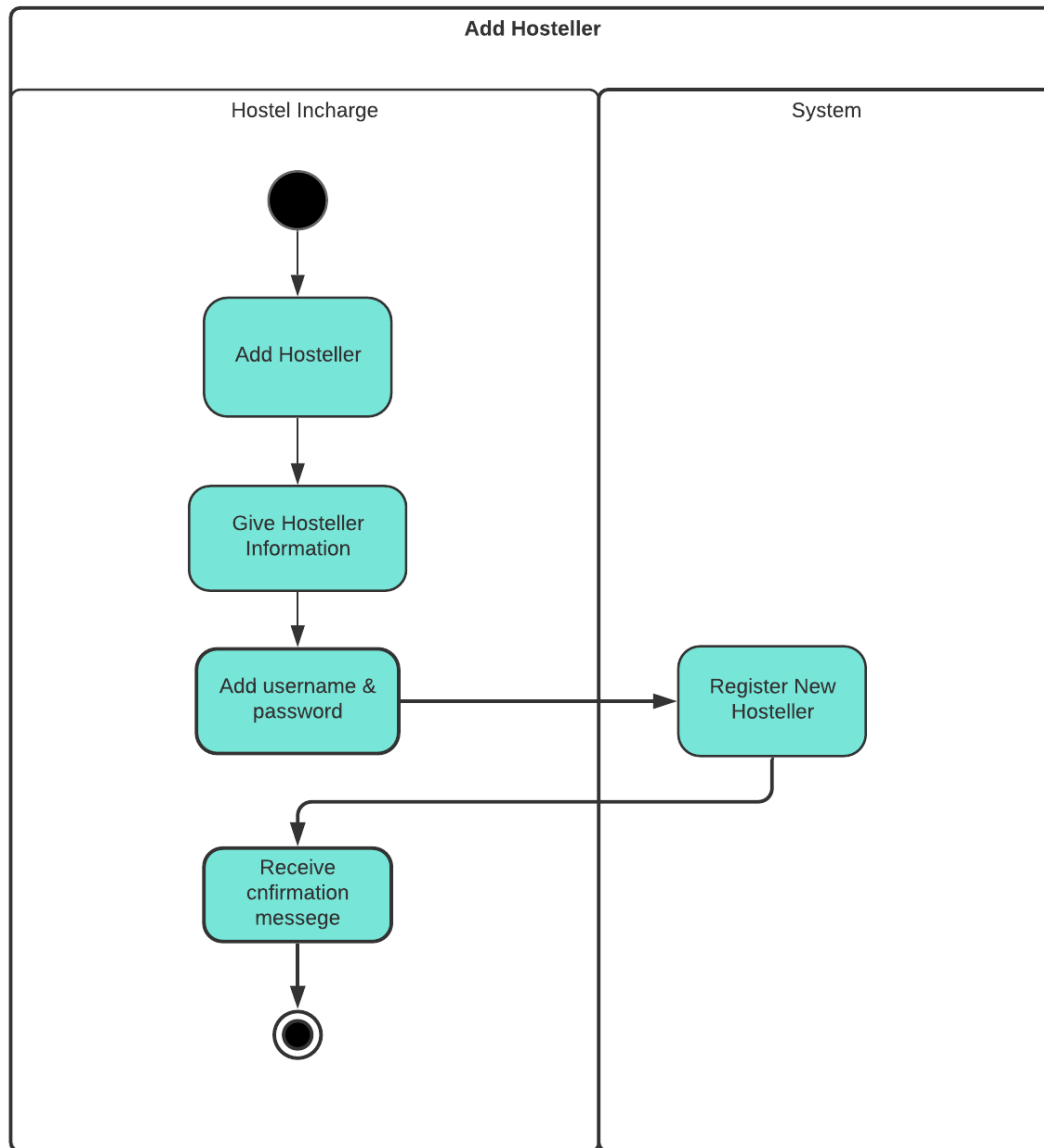
5.2 Activity diagram for **View Hosteller**



5.3 Activity diagram for Add Hosteller

Smart hostel

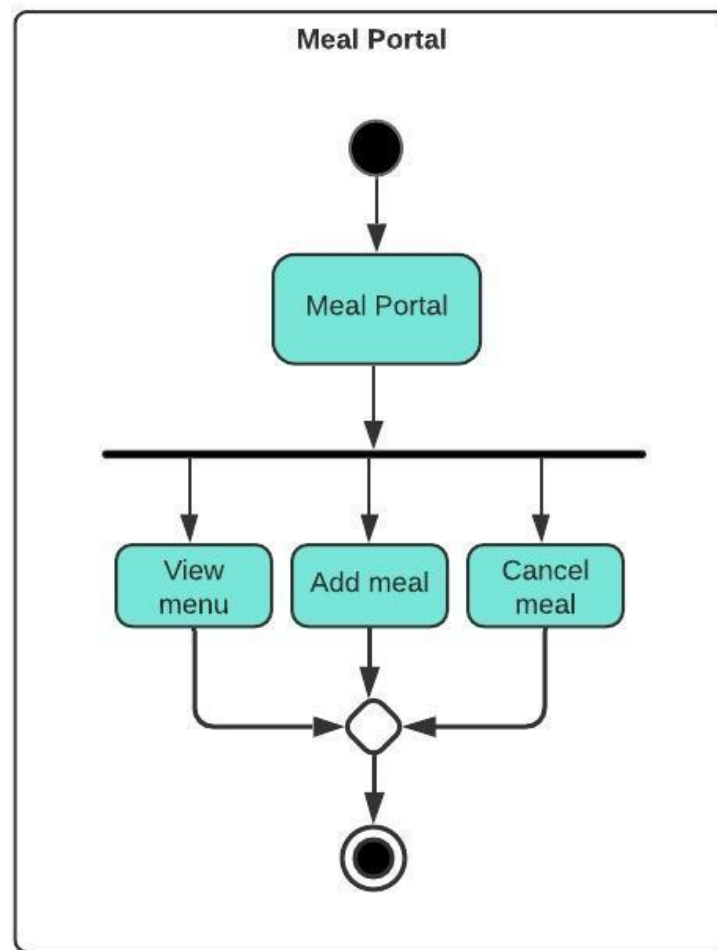
Mohiuddin , Raisa | April 15, 2021



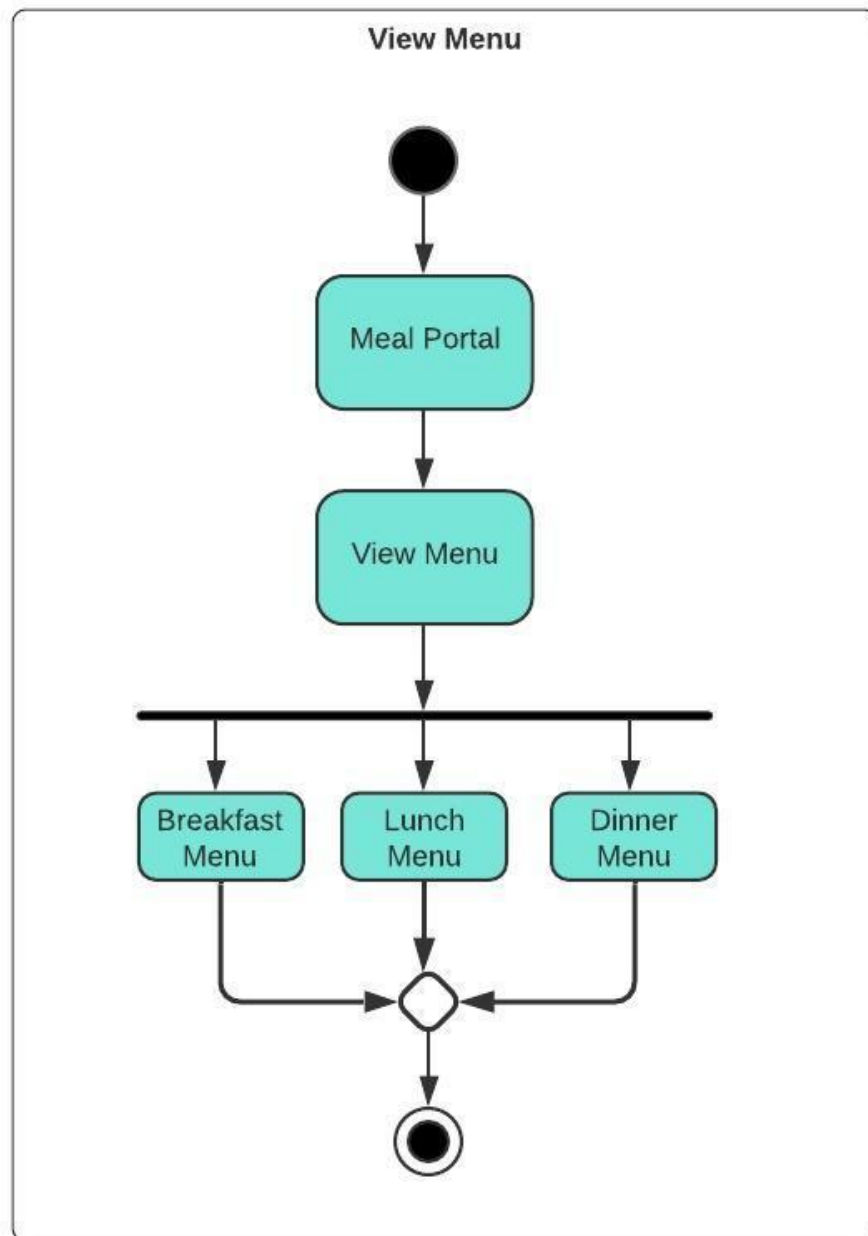
5.4 Activity diagram for **Meal Portal**

Smart hostel

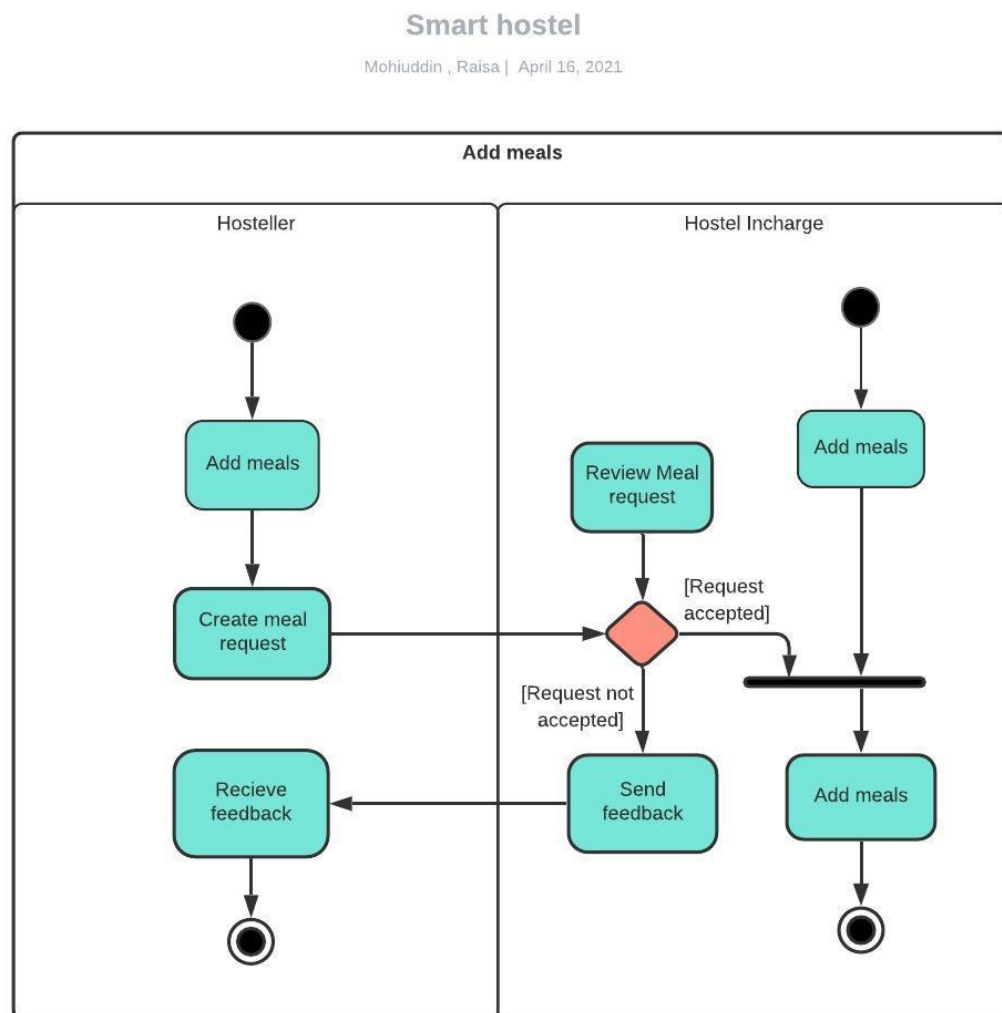
Mohiuddin , Raisa | April 16, 2021



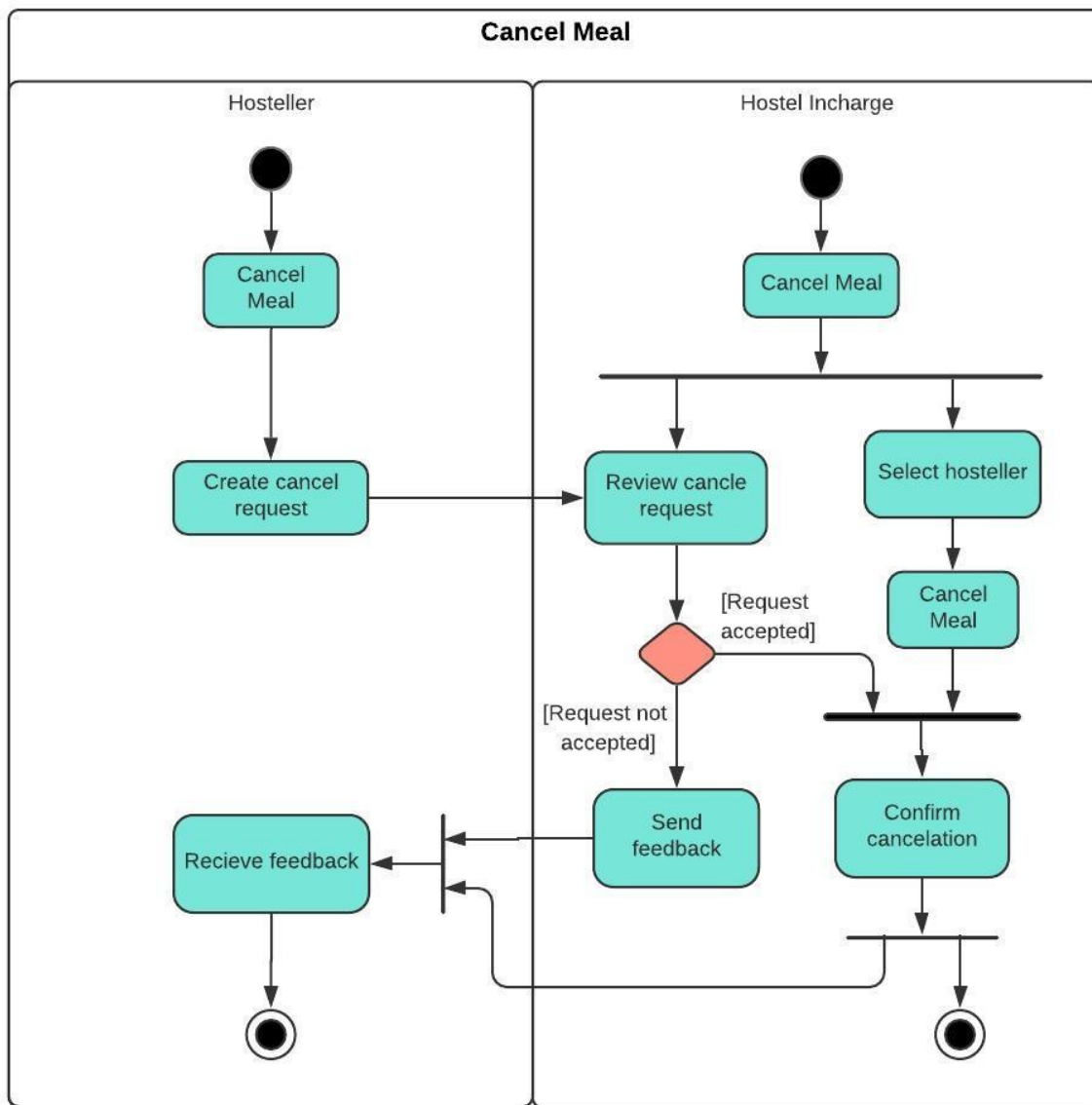
5.5 Activity diagram for **view Menu**



5.6 Activity diagram for Add meals



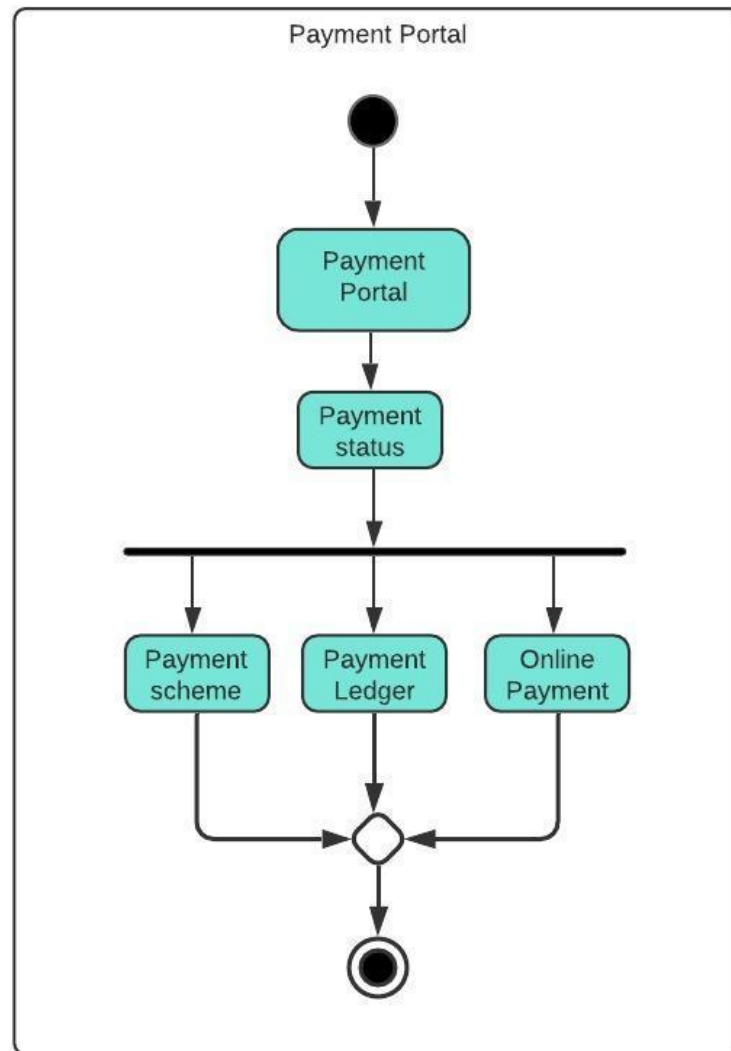
5.7 Activity diagram for **Cancel meal**



5.8 Activity diagram for **Payment Portal**

Smart hostel

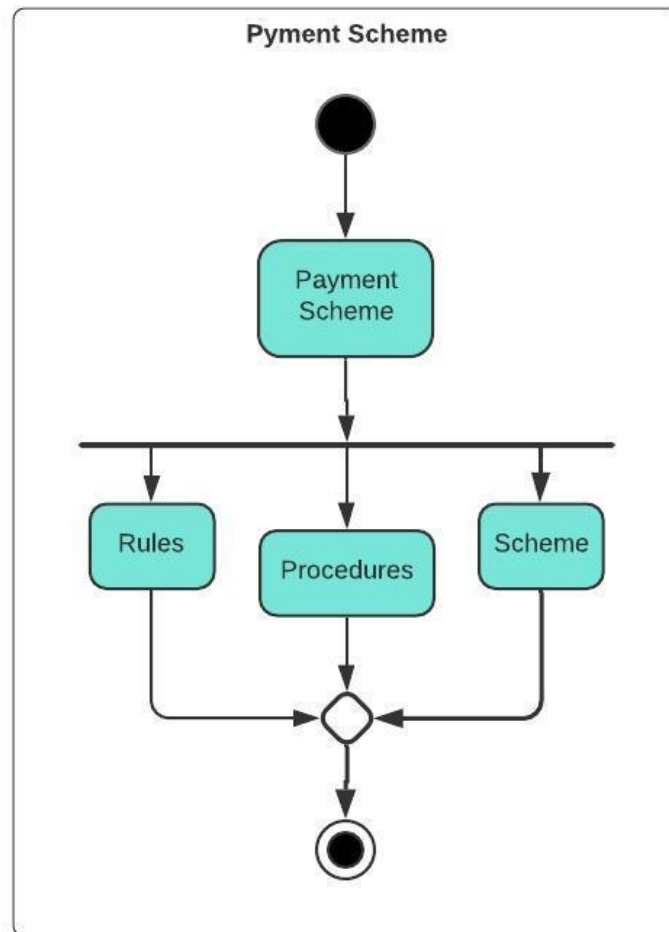
Mohiuddin , Raisa | April 16, 2021



5.9 Activity diagram for **Payment scheme**

Smart hostel

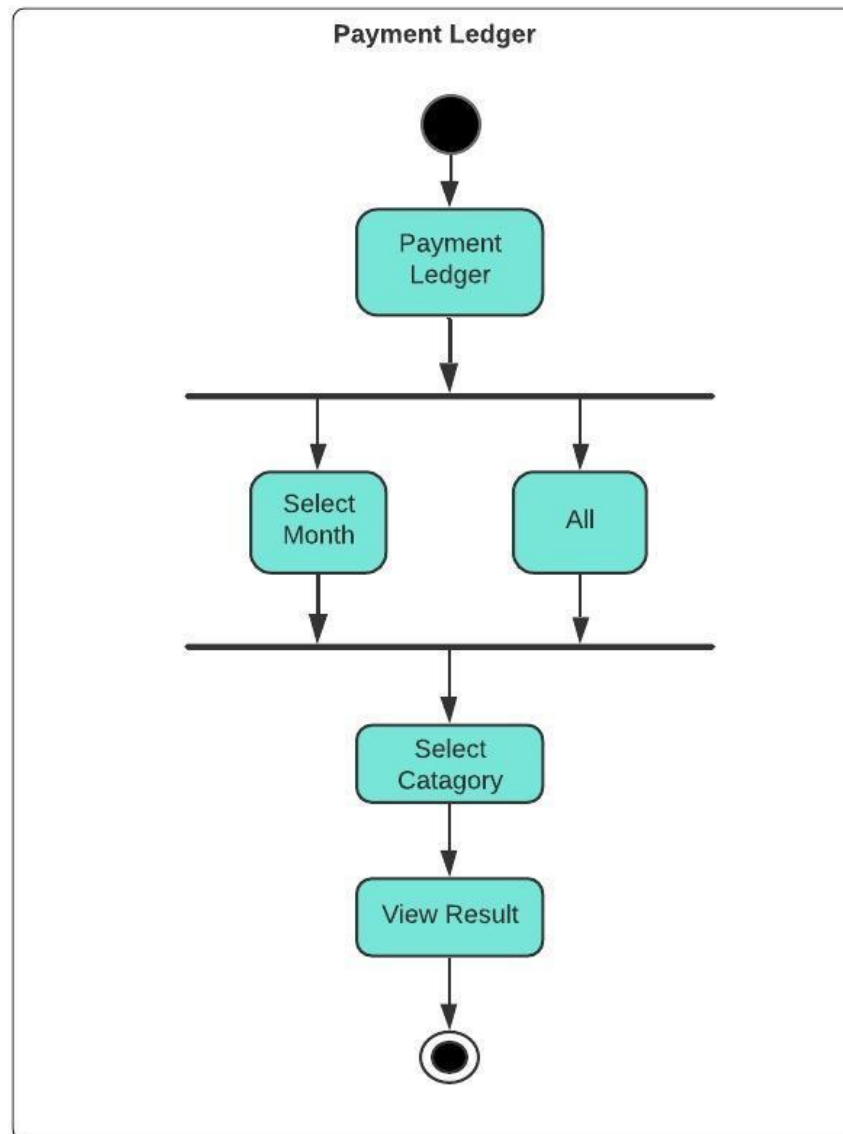
Mohiuddin , Raisa | April 16, 2021



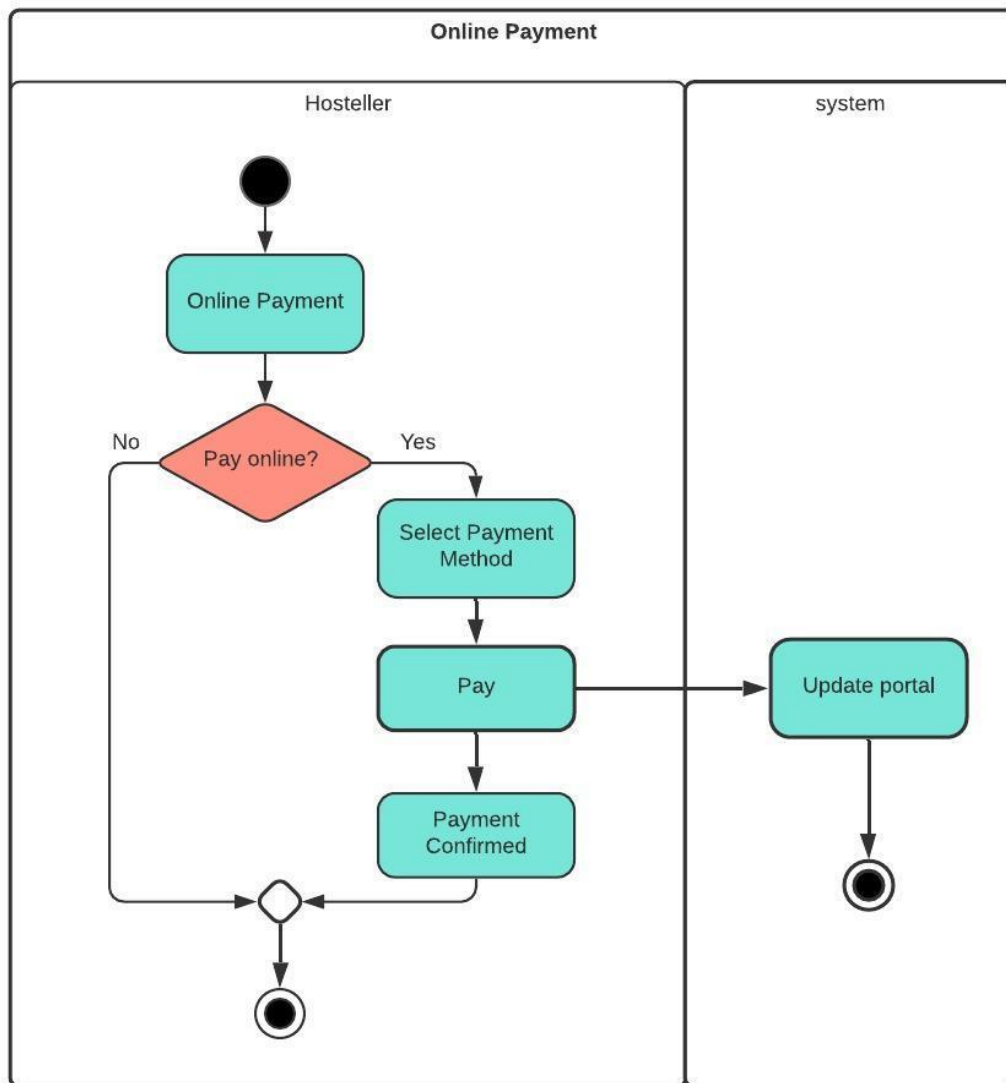
5.10 Activity diagram for **Payment Ledger**

Smart hostel

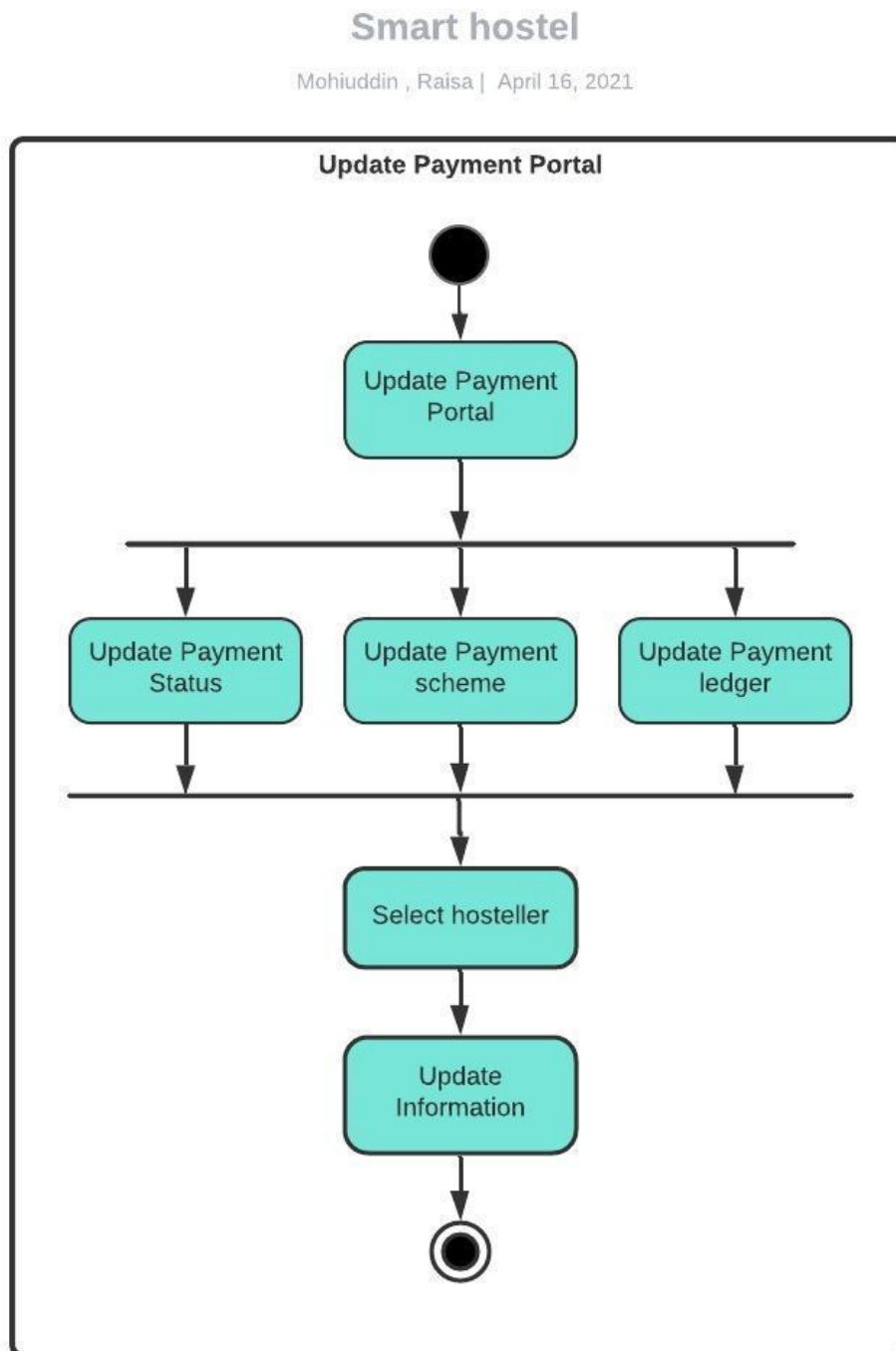
Mohiuddin , Raisa | April 16, 2021



5.11 Activity diagram **Online Payment**



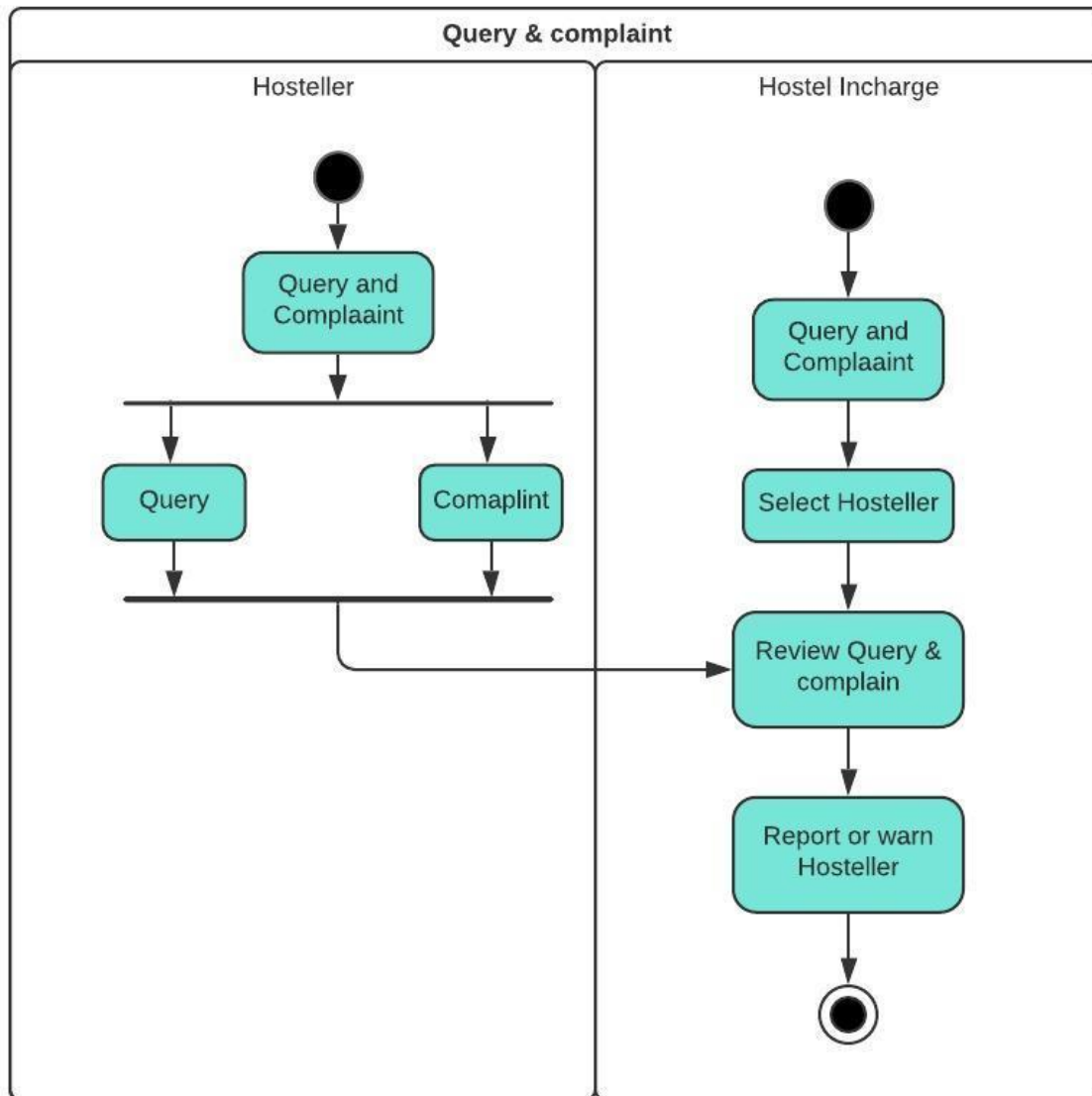
5.12 Activity diagram **Update Payment Portal**



5.13 Activity diagram **Query & Complaint**

Smart hostel

Mohiuddin , Raisa | April 16, 2021

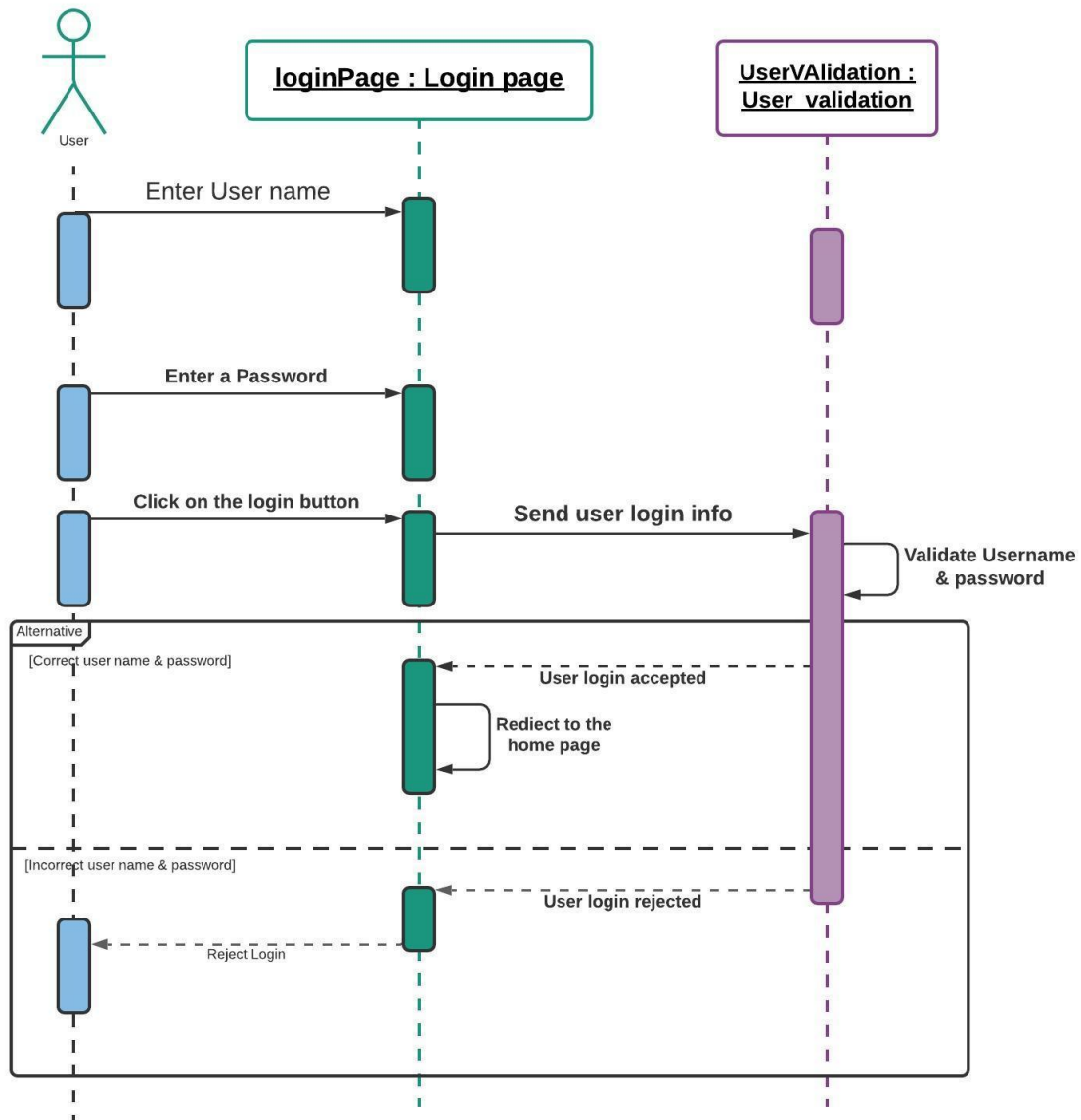


CHAPTER 6

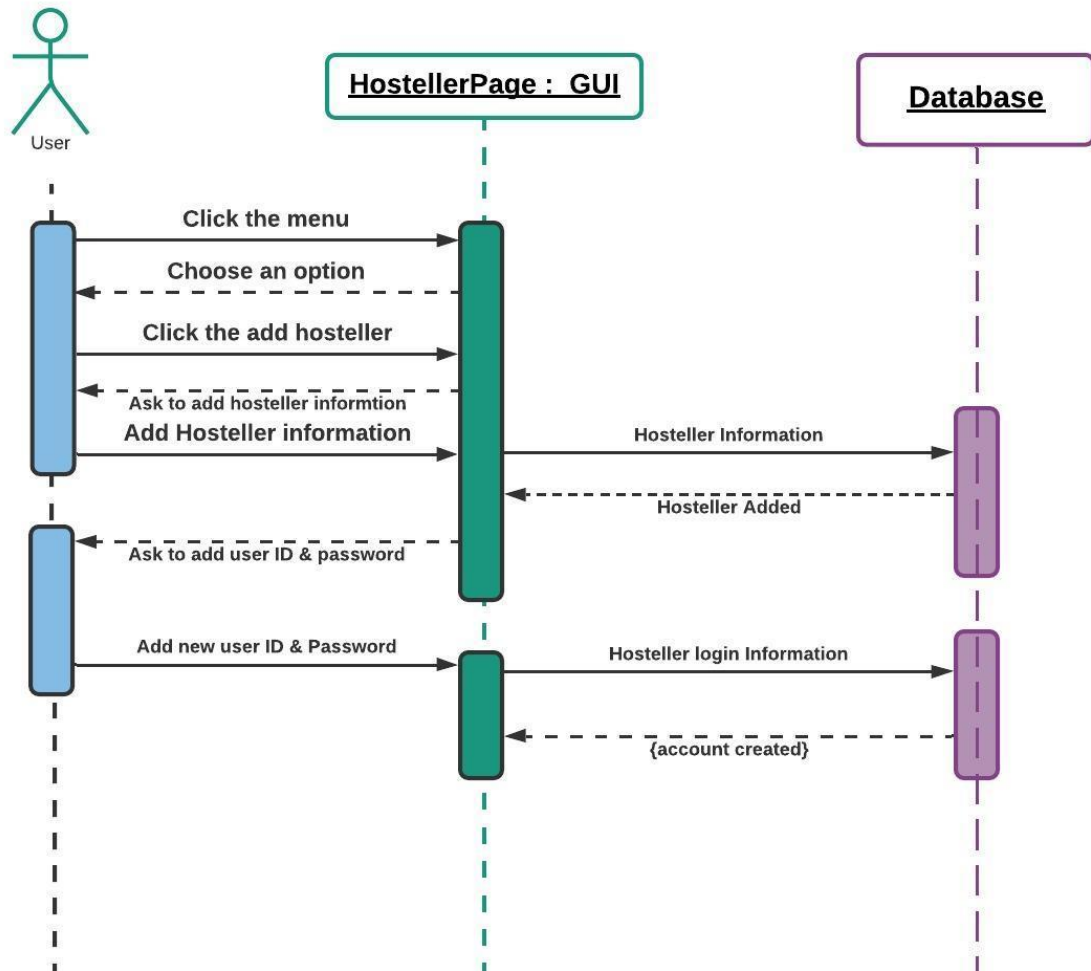
Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

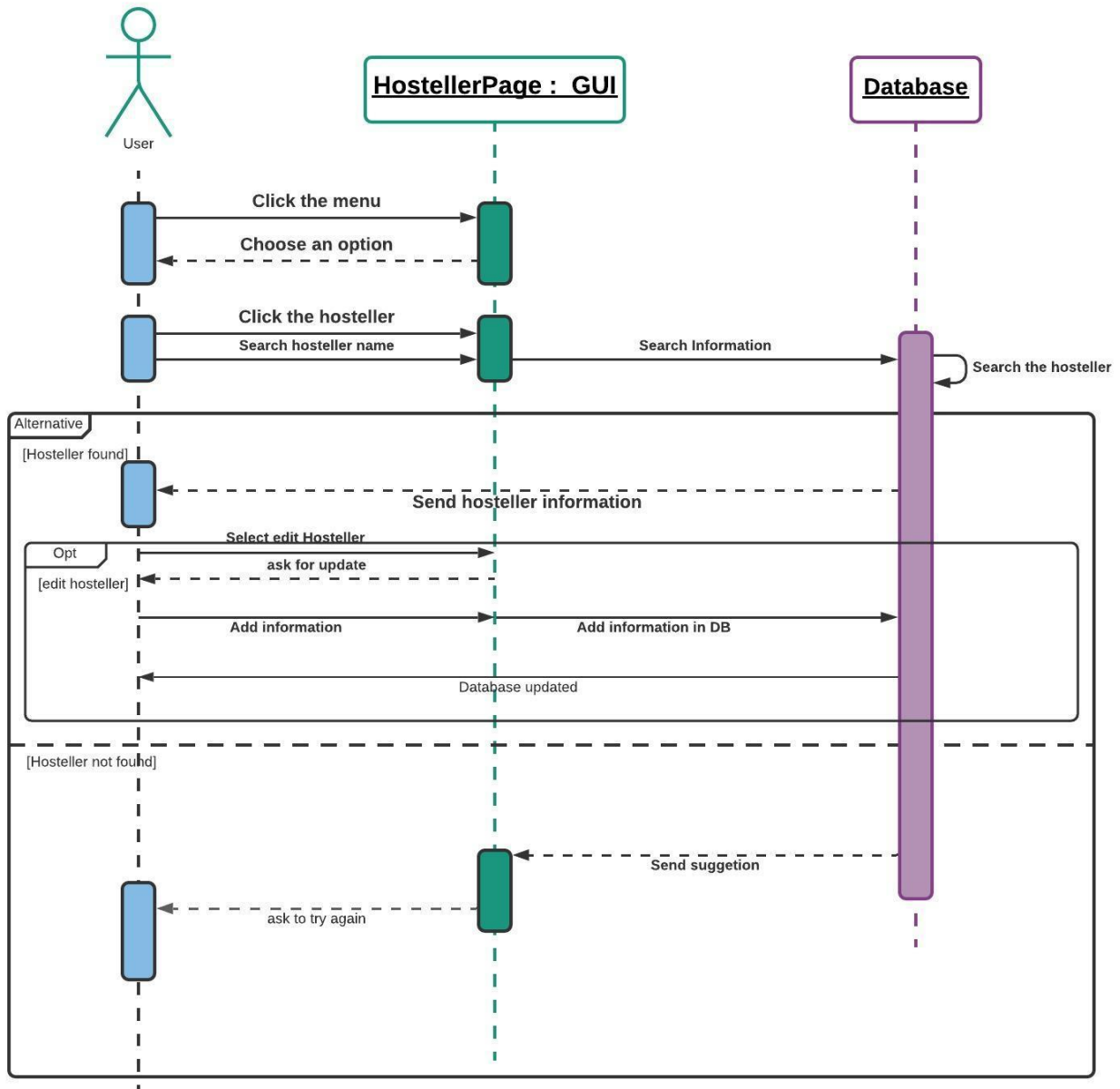
6.1 Sequence diagram **Login:**



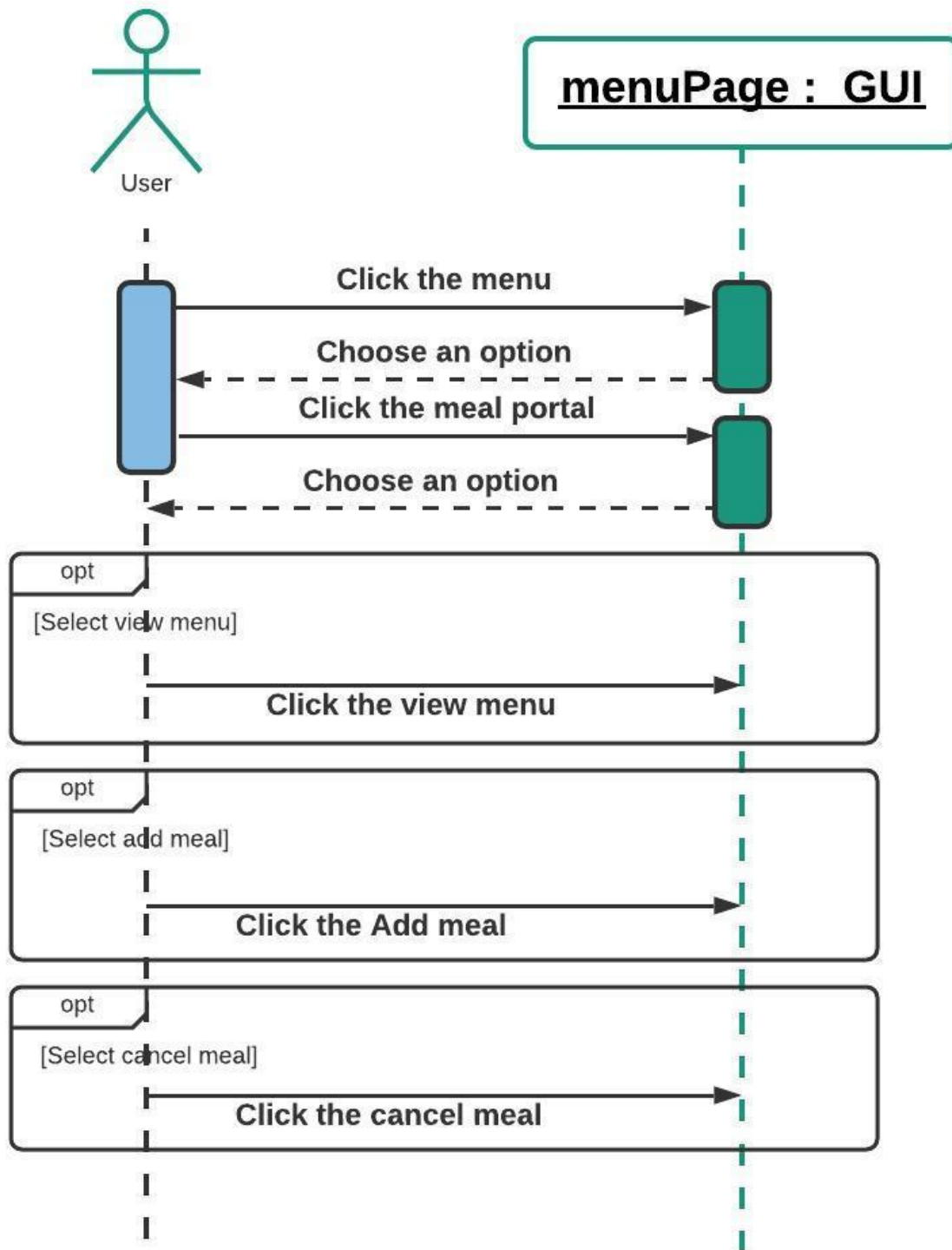
6.2 Sequence diagram Add Hosteller:



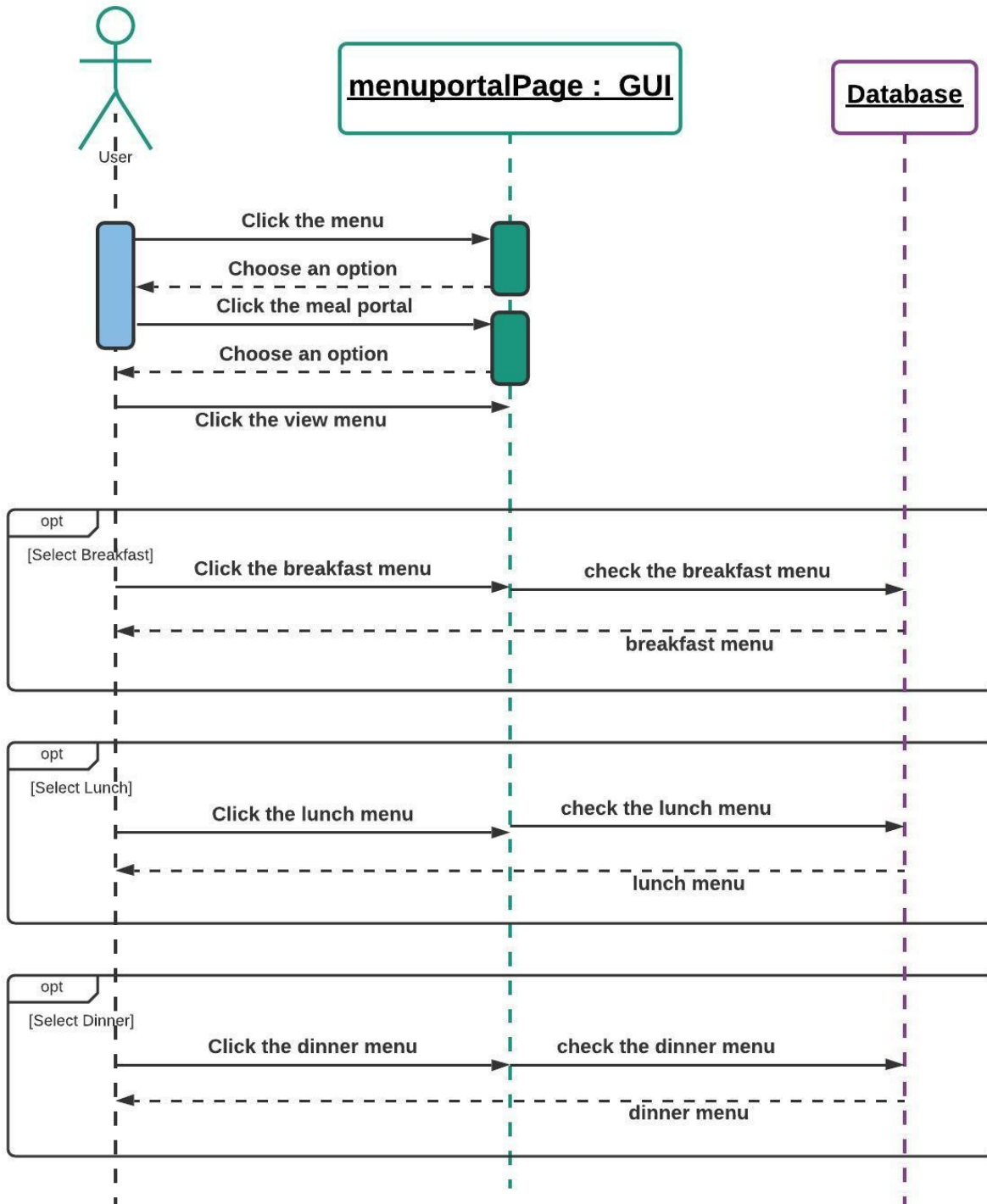
6.3 Sequence diagram View Hosteller:



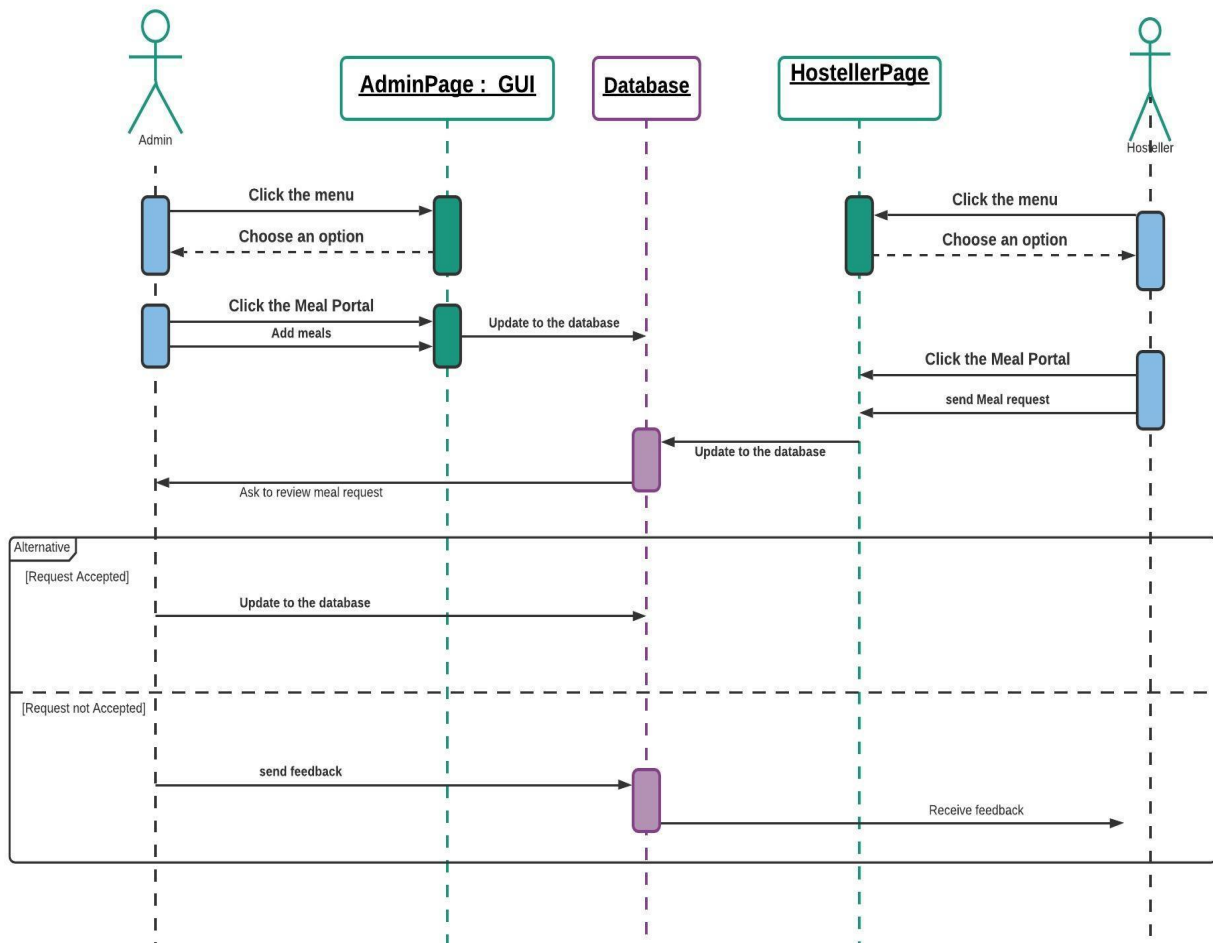
6.4 Sequence diagram Meal Portal:



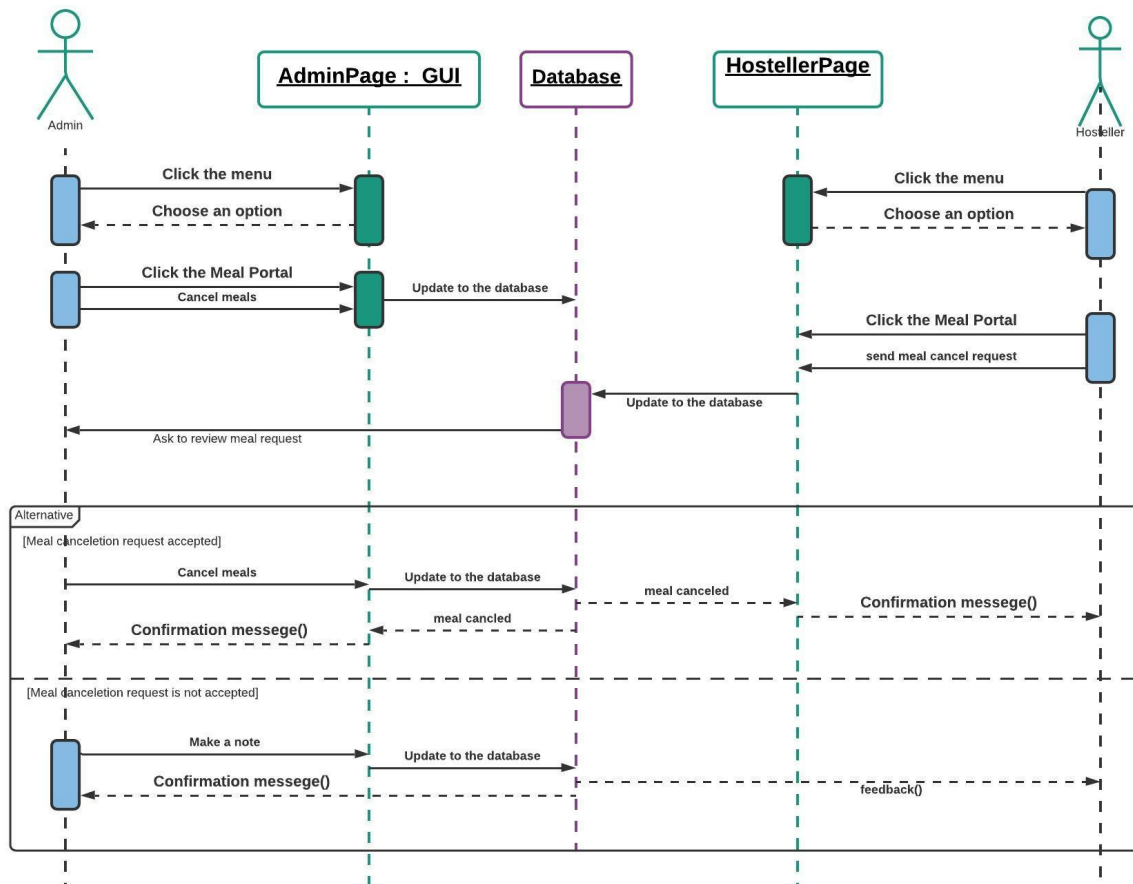
6.5 Sequence diagram View Menu:



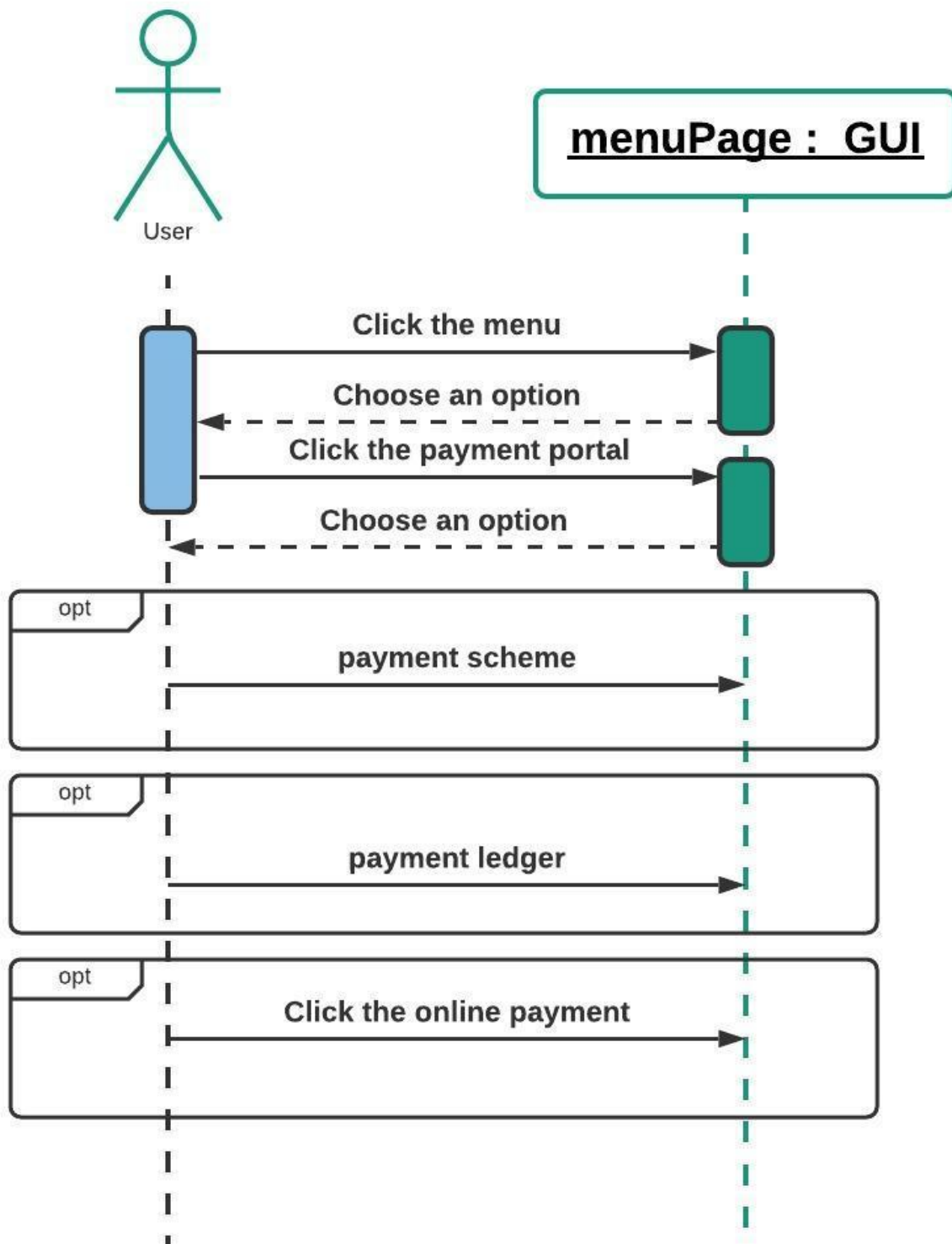
6.6 Sequence diagram Add Meals:



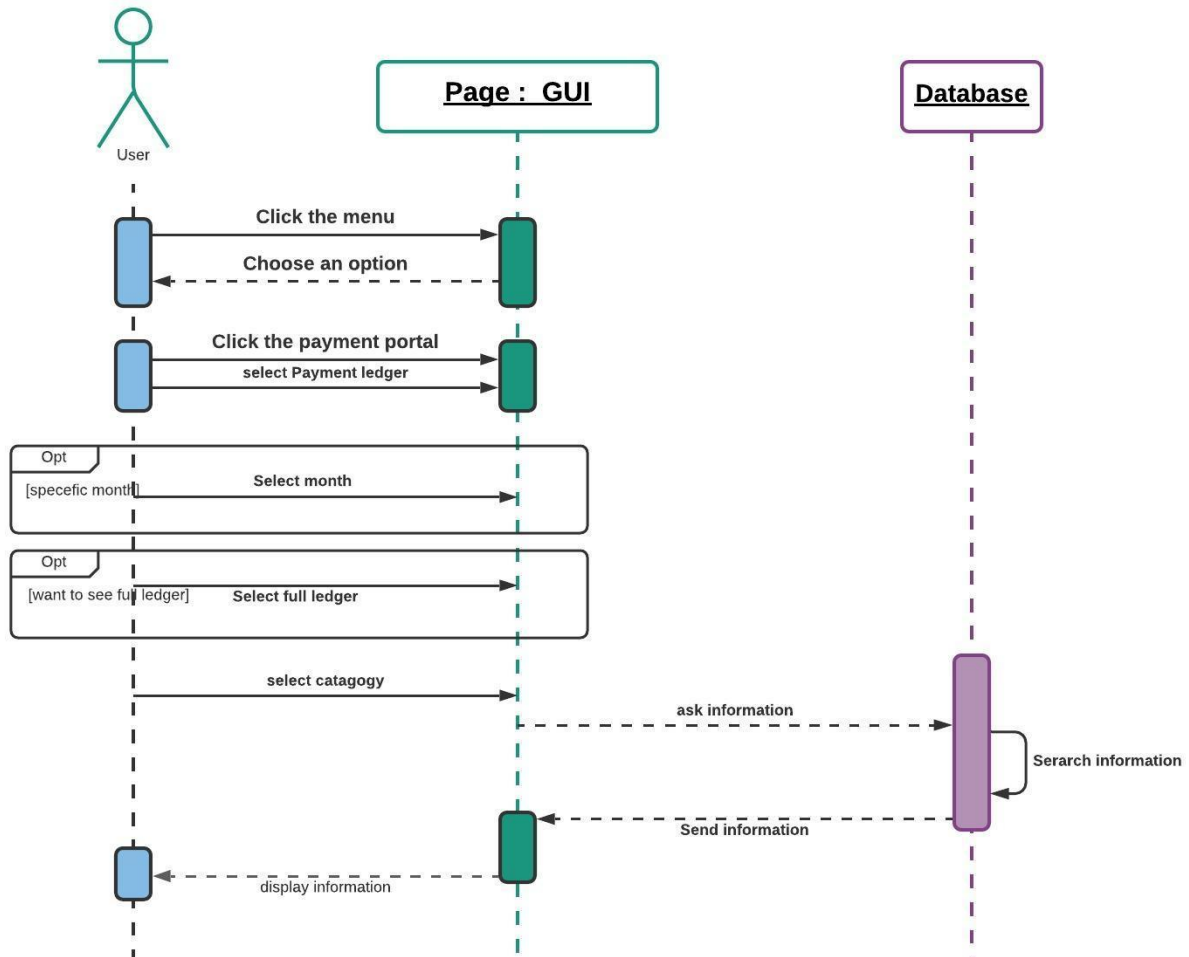
6.7 Sequence diagram **Cancel meal:**



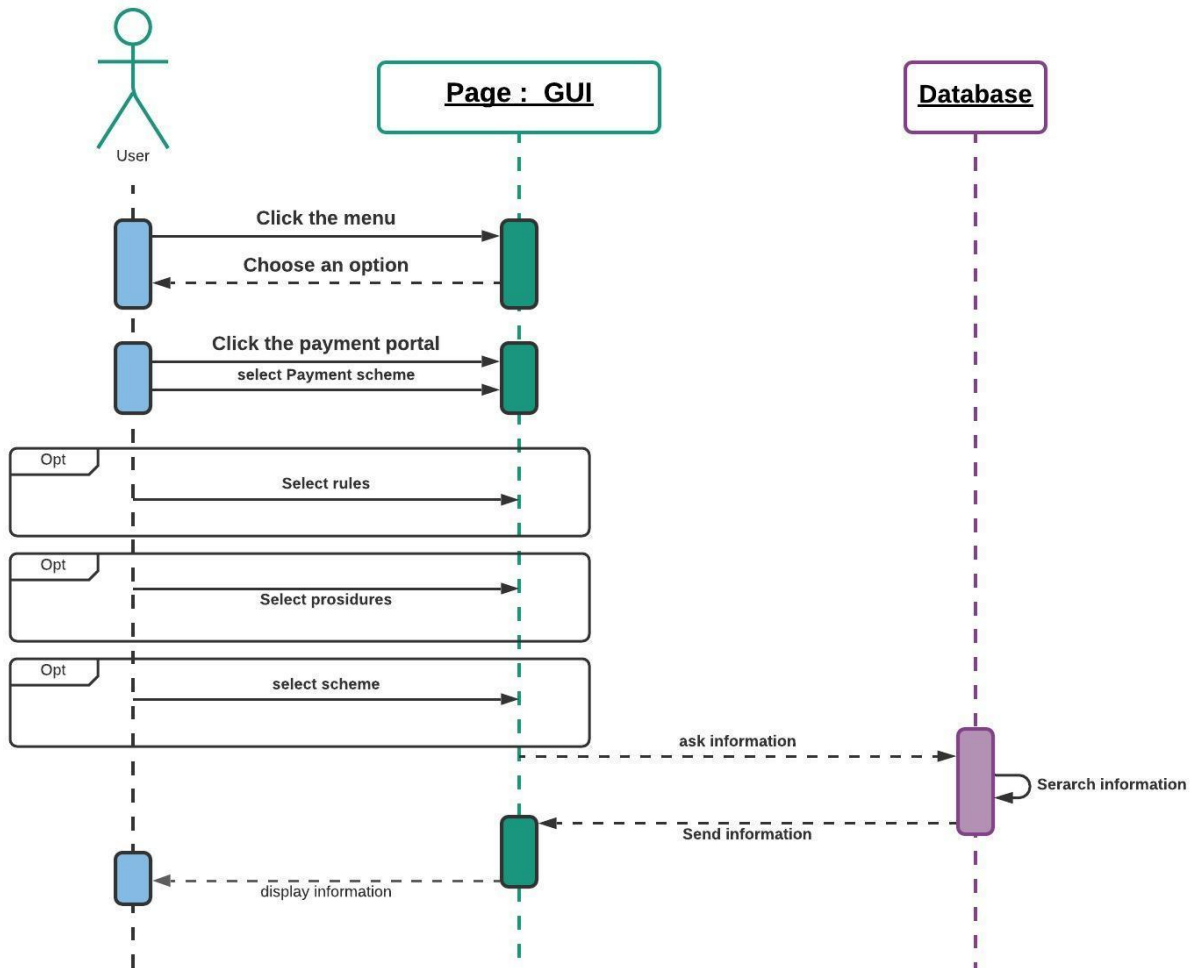
6.8 Sequence diagram **Payment Portal:**



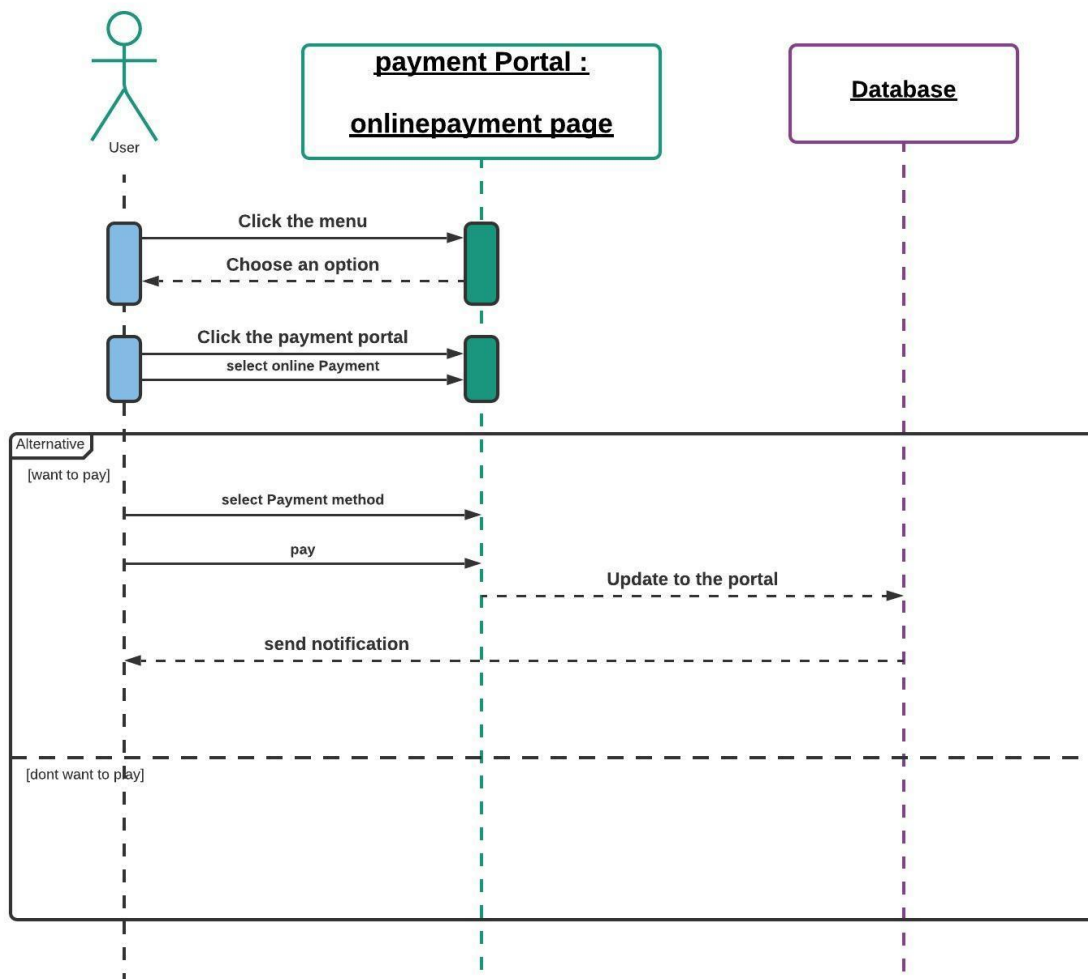
6.9 Sequence diagram **Payment Ledger**:



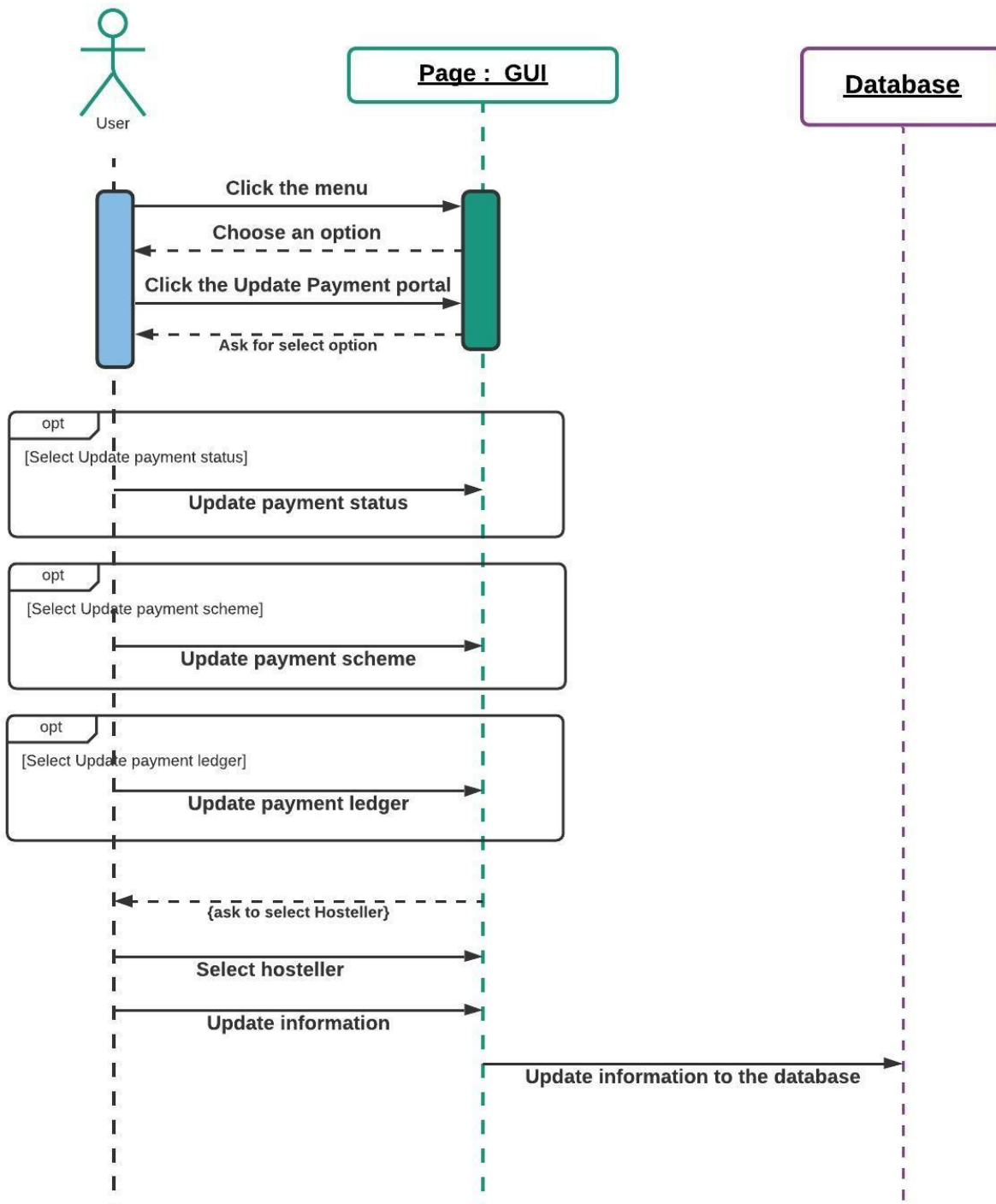
6.10 Sequence diagram **Payment Scheme:**



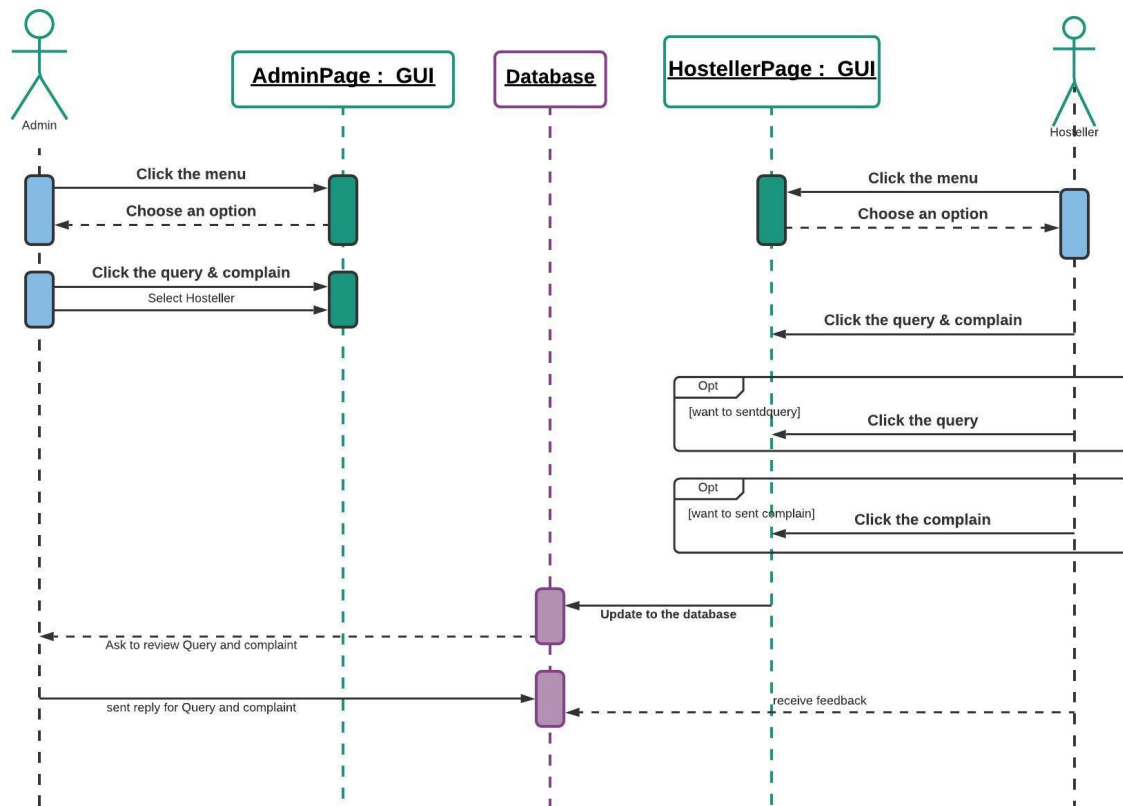
6.11 Sequence diagram **Online Payment:**



6.12 Sequence diagram **Update Payment Portal:**



6.13 Sequence diagram Query & Complaint:



CHAPTER 7

Entity Relationship Diagram

An entity-relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define their properties.

7.1 Entity Relationship Diagram (ERD) for “Smart hostel”

