Interim Report

Level 2

Hotel Management System

SudoCoders

Name	Index number
Pitigala P.K.D.G.D	**204161D**
Aysha M.R.S	204013D
Kaushalya D.M.G	204097K
Gunasiri G.C.S	204064H
Pemasiri M.P.T.B.S	204152C

Faculty of Information Technology
University of Moratuwa

2022

Interim Report

Level 2

Hotel Management System

SudoCoders

Name	Index number
Pitigala P.K.D.G.D	**204161D**
Aysha M.R.S	204013D
Kaushalya D.M.G	204097K
Gunasiri G.C.S	204064H
Pemasiri M.P.T.B.S	204152C

…….

Supervised by

Dr. (Ms) Ahangama S.

(Department of Information Technology)

Faculty of Information Technology
University of Moratuwa

2022

Abstract

In Order to stay one step ahead of your competitors in modern world, it is extremely important to be sensitive to your customer demands and needs. It is also important to discover vulnerabilities and remedy them as soon as possible. The customer is the main player in the current setting, and you should reach out to them in your own terms

Hotel Management is an extremely complicated process. The absence of comprehensive and user-friendly hotel management websites only amplifies this complexity. Even the few software that exists are outdated and only cater to the few needs of the hotel. Hotels are forced to use these software or use multiple of these to cover the wide field of services and fields that exist within the hotel system. Since the staff are not extremely tech-savvy, they are forced to learn about these from scrap because of the complicated UI s.

Considering all these factors, we decided to design and develop a hotel management software for the Circlebook software development company. With this software, we will be able to cater to most needs in almost all the fields of a hotel including but not limited to Guest registration, Reservation and Housekeeping.

Before the development phase we analyzed the requirements and designed UML diagrams, EER diagrams and UI/UX designs. We chose PHP, Laravel, Bootstrap as technologies for our HM software.

Also, we are going to reuse some code segments from the company's earlier works in order to maintain compatibility with the company's main platform.

Table of Contents

Chapter 1 Introduction **1**
1.1. Introduction 1
1.2. Background motivation 2
1.3. Aim and objective 2
1.3.1. Aim 2
1.3.2. Objectives 3
Chapter 2 Background Research **4**
Chapter 3 Technologies **5**
3.1. Introduction 5
3.2. Technologies 5
3.2.1. PHP 5
3.2.2. Laravel 5
3.2.3. jQuery 5
3.2.4. Figma 5
3.2.5. Maria DB 5
3.3. Summary 6
Chapter 4 Approach: Efficiently managing Hotel armed with Realtime information **7**
4.1. Introduction 7
4.2. Functionalities of Admin Panel 7
4.2.1 User Management 7
4.2.2. Assign privileges through job roles 7
4.2.3. Room details management 8
4.2.4. Locator Management (to identify where the guest is) 8
4.2.5. Hotel Chain Management 8
4.2.6. Supplier Management 8
4.3. Main Functions: Front office front operations 8
4.3.1. Creating GRC 8
4.3.2. Creating Guest Profile 9
4.3.3. Share guest profile 9
4.3.4. Front Office Remarks 9
4.3.5. Locators Management 9
4.4. Front Office: Back operations 10

4.4.1. Creating reservation 10
4.4.2. Add meal plan 10
4.4.3. Add billing person 10
4.4.4. Other details 10
4.4.5. Add discounts 11
4.4.6. Interfaces with other components 11
4.5. House Keeping 11
4.5.1. Supervisor Dashboard 11
4.5.2. Cleaning staff Dashboard 11
4.5.3. Interfaces with other components 11
4.6. Billing 12
4.6.1. Update 12
4.6.2. View 12
4.6.3. Grouped bill (generated for multiple guests) split option View 12
4.7. Software Process Model 12
4.7.1. Users, Input, Output of the system 13
4.7.2. Process 14
Chapter 5 Analysis and Design 17
5.1. Introduction 17
5.2. Analysis 17
5.3. Design 17
5.3.1. Use Case Diagrams 17
5.3.2. Activity Diagrams 20
5.3.3. Class Diagram 28
5.3.4. Sequence Diagrams 30
5.3.5. Component Diagram 36
5.3.6 EER 37
5.4. Summary 37
Chapter 6 Implementation 38
6.1. Introduction 38
6.2. System and Software 38
6.3. Summary 43
Chapter 7 Discussion 44

7.1. Introduction 44
7.2. Evaluation and testing 44
7.3. How the solution differs from other solutions 44
7.4. Further development 45
Chapter 8 References 46
8.1. EER Diagram 46
8.2. OOAD diagrams 46
Chapter 9 Appendixes 47
Appendix A - Individuals Contribution to the Project 47
Appendix B: Action Plan 52
Appendix C: UI Designs 53
Appendix D: Software Requirements Specifications 65

Table of Figures

Figure 4.1: Context Diagram 13
Figure 5.1: Use Case Diagram 19
Figure 5.2 : Activity Diagram 1- Creating Guest registration card 20
Figure 5.3 : Activity Diagram 2 - Creating and Merging guest profile 21
Figure 5.4: Activity Diagram 3 - Creating a new user 22
Figure 5.5 : Activity Diagram 4 - User Login 23
Figure 5.6 : Activity Diagram 5 - House Keeping 24
Figure 5.7: Activity Diagram 6- Reservations 25
Figure 5.8: Activity Diagram 7- Room Management 26
Figure 5.9: Activity Diagram 8- Billing 27
Figure 5.10: Class Diagram 29
Figure 5.11 : Sequence Diagram 1-Creating Guest registration card 30
Figure 5.12 : Sequence Diagram 2 - Creating and merging guest profile 31
Figure 5.13: Sequence Diagram 3 - Reservations 32
Figure 5.14 : Sequence Diagram 4 - House Keeping 33
Figure 5.15: Sequence Diagram 5- Room Management 34
Figure 5.16: Sequence Diagram 6- Billing 35
Figure 5.17 : Component Diagram 36
Figure 5.18: EER 37
Figure 6.1: php artisan serve 39
Figure 6.2: localhost:8000 39
Figure 6.3 : MCV Model 40
Figure 6.4 : Controllers in CRUD 41
Figure 6.5: Routes 42
Figure 6.6: Laravel Output 42
Figure 9.1: Action Plan 52
Figure 9.2: Administration Portal 1 53
Figure 9.3: Administration Portal 2 53
Figure 9.4: Administration Portal 3 54
Figure 9.5: Administration Portal 4 54
Figure 9.6: Administration Portal 5 55
Figure 9.7: Administration Portal 6 55
Figure 9.8: Front Office Back Ops 2 56
Figure 9.9: Front Office Back Ops 1 56
Figure 9.10: Front Office Back Ops 3 57
Figure 9.11: Front Office Back Ops 4 57
Figure 9.12: Front Office Back Ops 5 58
Figure 9.13: Front Office Back Ops 6 58
Figure 9.14: Housekeeping 1 59
Figure 9.15: Housekeeping 2 59
Figure 9.16: Housekeeping 3 60
Figure 9.17: Billing 1 61

Figure 9.18: Billing 2 61
Figure 9.19: Billing 3 62
Figure 9.20: Room Management 62
Figure 9.21: Front Office Front Ops 1 63
Figure 9.22: Front Office Front Ops 63
Figure 9.23: Front Office Front Ops 2 64

Chapter 1 Introduction

1.1. Introduction

AWAKAZA is an already existing platform owned by CircleBook(Pvt) Ltd. We are tasked with creating a hotel management software for this platform.

Tourism is one of the emerging giants in the world economy. With the pandemic going away and the world slowly escaping from the chaos it caused, Tourism and hotel industry is gradually approaching its normalcy again. Although this is an extremely good thing, old problems that were associated with the hotel management are also reemerging

Even though there are several hotel management software in the market today, all these lack one or several requirements that clients (i.e., hotels) are expecting.

Guest registration and reservation management is one of the most important aspects of the whole process. Most of the software in the current market is outdated and has very old-style UIs. Also, their outdated design means that they don't ask for all the necessary details that are required by the hotel. These software are also not focusing on creating guest profiles thus they will treat even customers who frequent the hotel as new ones every time.

Housekeeping employs the highest number of employees in a hotel. So, it is important to maintain the whole process like a well-oiled machine. But the problem is that current hotel management software gives little or no attention to this whole side, which in turn creates many problems in a process that requires preciseness and smoothness, unlike any other in order to work properly.

Billing is also immensely important as it is what brings in revenue for the hotel. So, it's important that the hotel provides customer-friendly options like opting to splitting the bill and splitting the bill based on services and goods. But almost all HM software do not have this option.

One of the most persistent problems is that these software only focus one or two sectors at a time. So, the hotel is forced to employ multiple software with their own costs in order to remedy this issue.

So considering all these factors, we can clearly see that there is a market for a hotel management software which encompasses solutions to all these problems.

1.2. Background motivation

Circlebook (Pvt) Ltd is a leading software company in Sri Lanka. They have been developing applications for organizations around the world. Right now, they are engaging in various projects ranging from medium scale to complex projects. Currently, we cannot find a system which includes an all-inclusive hotel management platform infused with a user-friendly UI and a comprehensive system.

Awakaza hotel management SAAS platform is an advanced hotel management software which will enhance every aspect of hotel management including guest registration, engineering, and housekeeping.

Some of the core values a customer expects from a hotel are punctuality and Quality of service. Traditional by hand methods are not going to cut it. The traditional method demands much more labor costs and is not practical when the hotel is scaling out. We need a Hotel Management system to efficiently manage all the information with increased security.

1.3. Aim and objective

1.3.1. Aim

The aim of this project is to introduce a hotel management software which can be used to increase efficiency of the whole management process with emphasis on user-friendliness and comprehensiveness

1.3.2. Objectives

- Facilitates guest registration
- Facilitates billing process
- Facilitates housekeeping process
- Facilitates administration control
- Facilitates kitchen and inventory handling

Chapter 2 Background Research

We had to have a conversation with the end users to better understand their wants and requirements and to maximize user experience. For improving the UI and user experience, we also had to attend many review meetings with the company's technical staff and the end users of this admin dashboard.

To find the most appropriate technologies and make changes to the currently used technologies, we had to study the existing website, its architecture, and the technologies that had been incorporated into the development process.

Chapter 3 Technologies

3.1. Introduction

In this chapter, we will focus on the technologies we have used to build this system.

3.2. Technologies

3.2.1. PHP

PHP takes less time in loading web pages. The faster loading time helps boost speed. So we chose PHP to develop the website.

3.2.2. Laravel

Laravel is a free and open-source PHP web framework. We chose this based on ease of integration with other Circlebook systems.

3.2.3. jQuery

This JavaScript library makes it easier to develop in JavaScript by offering a simplified syntax that is straightforward and quite simple to use. Using jQuery provides us with an excellent deal of power and flexibility.

3.2.4. Figma

We chose Figma to design UI, because this is free software and one of the most feature-rich tools. The work is much simpler with Figma because we can share our project file and get input from our team as well.

3.2.5. Maria DB

Since we have many relational data regarding the application and the users associated, we had to choose a SQL database since Maria DB offers Seamless Scaling and greater business agility.

3.3. Summary

Laravel uses the MVC model which stands for Model, View, Controller. We plan to develop the view using Bootstrap and Model would work with a relational database such as MariaDB.

Chapter 4 Approach: Efficiently managing Hotel armed with Realtime information

4.1. Introduction

In the everchanging business world, businesses need to stay competitive. Higher the competition, higher the efficiency must be. To be efficient, the processes should be coordinated correctly and quickly. Hotels are no different. In a country like Sri Lanka, tourism sector is one of the major foreign income earners. So, there is a huge competition in the Hotel industry. When the hotel becomes larger, coordination becomes harder, efficiency drops, and costs keep rising. Property Management Software (PMS) is the key to declutter the organizational mess and to coordinate your hotel processes correctly and quickly.

4.2. Functionalities of Admin Panel

4.2.1 User Management

1.Add user

2.Delete user

3.Disable user

4.Edit User profile

4.2.2. Assign privileges through job roles

1.View each user's role

2.This restricts users' access to their relevant dashboard

1.Super Admin

2.Admin

3.Receptionist

4.Food and Beverage Management

5.Housekeeping manager

4.2.3. Room details management

1.View Room details

2.Add new room

3.Edit Room

4.Delete Room

4.2.4. Locator Management (to identify where the guest is)

1.View Locator details

2.Add locator

3.Edit locator

4.Delete locator

4.2.5. Hotel Chain Management

4.2.6. Supplier Management

4.3. Main Functions: Front office front operations

4.3.1. Creating GRC

4.3.1.1. Add GRC

1.Validate input fields

2.Validate user input

3.Validate NIC

4.Validate Passport number

5.Validate Email

6.Validate Room number

4.3.3.2. Edit GRC

1.Validate All the above details

2.Delete GRC

3.Use soft delete

Note: GRC needs to be Form e customized

4.3.2. Creating Guest Profile

4.3.2.1. Add Guest Profile

1.Create a guest profile under the GRC Number

2.Validate all input fields

3.Validate all user inputs

4.3.2.2. Edit Guest profile

Validate all above the details

4.3.2.3. Delete Guest profile

Use soft delete

Note: The guest Profile Form needs to be customized

4.3.3. Share guest profile

Share the guest profiles with their hotel chain

4.3.4. Front Office Remarks

1.Add

2.Edit

3.Delete

4.3.5. Locators Management

1.Mark locator

2.Edit marked locator

3.Remove locator

4.4. Front Office: Back operations

4.4.1. Creating reservation

4.4.1.1. Select reservation mode

Select number

1.Individual

2.Group

Select confirmation

1.Confirmed

2.Tentative Select allotment

4.4.1.2. Add reservation

1.Validate input fields

2.Validate user input

3.Validate contact details

4.Validate pax count

5.Validate room number

6.Display available discounts (System)

4.4.2. Add meal plan

View meal options update meal plan

4.4.3. Add billing person

1.Validate name

2.Check whether an advance has been paid

4.4.4. Other details

Validate input fields Validate user input

4.4.5. Add discounts

Add discount list

4.4.6. Interfaces with other components

Housekeeping

-Departure date should be checked

4.5. House Keeping

4.5.1. Supervisor Dashboard

4.5.1.1. View clean status of each room

4.5.1.2. View rooms to be checked

4.5.1.3. Update section: status after checking

4.5.1.4. Mini Bar Dashboard:

1.View which rooms (mini bar) to check (system generated)

2. Update section: update item count in each mini bar

3.(system) calculate bill

4.(system) update the bill in billing database for the customer.

4.5.2. Cleaning staff Dashboard:

1.(system) View rooms allocated to each staff member (No supervisor needed)

2.Update: rooms after cleaning.

4.5.3. Interfaces with other components

4.5.3.1 Front Office front Operations/ back operations

Departure date should be checked and auto update rooms to be cleaned list.

Locator should be checked to identify when guest is out and update the rooms to be cleaned list (with higher priority, should be cleaned before guest return).

4.6. Billing

4.6.1. Update

Update billing database food and beverages for a guest. Role: Food and Beverages operator

4.6.2. View

Role: reception, Food and Beverages operator (default)total bill for a customer. View bill in multiple layouts (max 6)

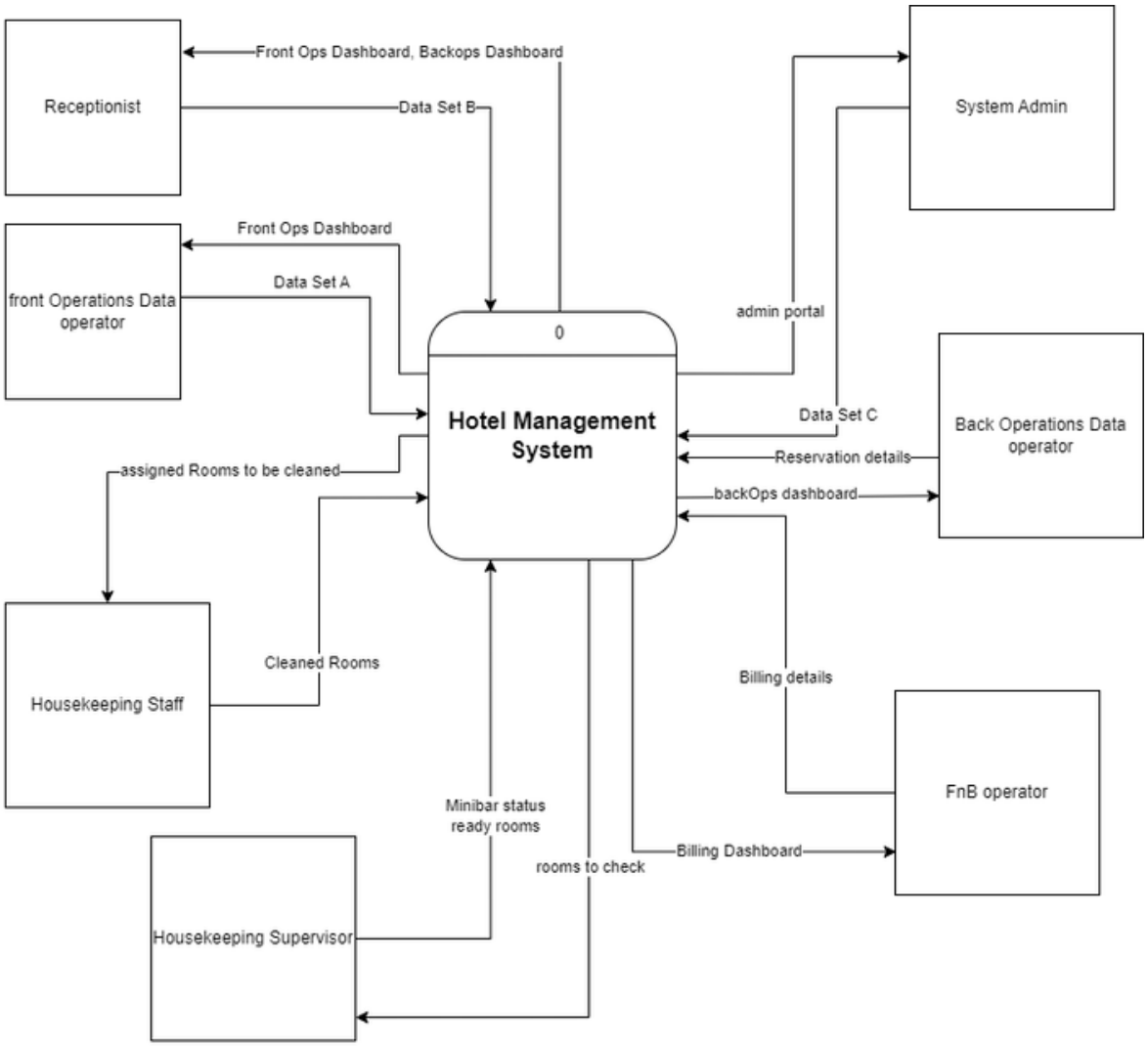
4.6.3. Grouped bill (generated for multiple guests) split option View

Split bill generation for two guests

4.7. Software Process Model

We are using the Agile Scrum methodology for our project development. Since this is not a dummy project but a real customer from the everchanging tourism industry is providing requirements, they are dynamic and not fixed. We must work with evolving requirements. The company who gave us the project is also using this methodology in most of their projects as well.

4.7.1. Users, Input, Output of the system



Data Set A

- Guest Profile updates
- Black listings
- Data shares with hotel chain
- Hotel Chain data
- Room change data

Data Set B

- Guest Details
- Locator updates
- Meal Plans
- Room selections
- Arrival and Departure dates

Data Set C

- User Details
- Locators
- Room details
- Access Privileges

Figure 4.1: Context Diagram

4.7.2. Process

4.7.2.1. Admin Portal

Admin Portal is used to manage users (add, edit, delete) and to add and edit other configuration items such as Locators, Rooms.
User privilege management is achieved by assigning roles. As an example, a user with "House Keeping staff" role cannot access either the Billing dashboard or Front Ops dashboard. Admin portal could be used by a user with "System Admin" role.
Admin Portal could be used to add/edit locators and rooms as well. Locator specifies where the guest is currently at. (Pool area, Room, outside the hotel).

4.7.2.2. Room Change

In some cases, guests need to change rooms. It may happen due to their wish or lack of room space in the hotel. This can happen in two ways. That is, like exchanging rooms on the same level or moving from a lower level to an upper-level room. Any kind of change from this will get the reason for it by the receptionist.
The change in the bill due to change of rooms will be decided by an authorized person of the hotel

4.7.2.3. Billing

Here, the food and beverages operator first update the database. Then the entire bill is created. Here the customer has two options to choose to pay the bill, the entire bill is paid by an individual or the group of people chooses to share the bill among themselves.
If a group chooses the second method, they can split the entire bill according to their wishes. Here the bill can be divided into six layouts for two customers.

4.7.2.4. Reservations

Reservation is mainly focused on room booking and meal planning. The user can make reservations according to the guest's preference.
Two main reservation types are there. Those are confirmed reservations and tentative reservations.

Confirmed: This is used to make confirmed reservations. They cannot change after booking.
Tentative: Guests can make tentative reservations, they may need rooms but are not sure about it, and this may be possible in the future. This type of reservation should not be considered as those rooms are already booked. And there will be a limited date and guests should confirm within the date, if not the reservation will be automatically canceled. This type of reservation could be editable(add/delete).

In the beginning, the user will select the rooms according to the guest's requirements. There will be a section to enter the payer details and if the guests are paying an advance amount, then it should record the advance amount details.
There is a separate section for meal planning. Through that users can input guests' each day meal plans.
There are 3 main meal plans in every hotel
BB - Bed and breakfast
HB - Half board
FB - Full board

When customers make reservation, we need to add their meal plan to day by day.

Example : If the customer reserve room for 4 days we can add like this

1st day - BB
2nd day - FB
3rd day - HB
4th day - FB
This is customer choice. If the customer wants to reserve FB for their all day, they can select it as their wish.

4.7.2.5. Guest Registration

The Guest will have to initiate guest registration by providing details to the receptionist. The receptionist will then input these details to the system and the system will validate the input fields. If the validation is successful, information will be saved, and a guest registration card will be created with an automated GRC number. However, if the validation failed, the system would inform the receptionist and the receptionist will ask for information again.

When creating the guest profile, the system will acquire details from the GRC. But there are several additional fields that need to be filled. The receptionist can ask for the details and complete the guest profile. After adding hotel-exclusive details, the user can decide if they want an additional field to enter new type of information. If he does, there is an option to add a new field. After filling the above field, the guest profile will be created. If he does not, the profile will be created without additional fields. If the guest visits the hotel multiple times, each time a new profile will be created. After a while (preferably a month), the profiles will be merged and will become a single guest profile. If the same guest decided to visit the hotel in future, guest profile will be instrumental in increasing efficiency of the whole registration process.

4.7.2.6. Housekeeping

This Housekeeping section allows housekeepers to easily manage their daily tasks, confirm room conditions and know when a room needs maintenance.

After a room is completely cleaned, the supervisor can provide the latest updates related to the cleaning status, which makes work easier for the workers and the supervisor.

With the Minibar system, the moment a customer leaves, a supervisor comes and checks the minibar in the room. Then a quantity survey is done, and it is calculated by entering it into the system and it is included in the customer's bill. The supervisor restocks the minibar. By setting up the system to operate in this way, the department can continuously improve and grow.

Chapter 5 Analysis and Design

5.1. Introduction

After identifying the problem, we gathered the requirements of our client and started developing a solution. We created UML diagrams to acquire a better understanding of the system. Both the structural and behavioral diagrams were designed and included here.

5.2. Analysis

Before designing the system, we identified the functional and non-functional requirements in the system as our client mentioned. The functional and non-functional requirements are mentioned in the SRS document.

5.3. Design

In this section, we will focus on the diagrams that we designed to visualize the functional and non-functional requirements. We have used draw.io and lucid charts to draw the diagrams. There are several UML diagrams we have used,

1. Use case Diagram
2. Activity Diagram
3. Class Diagram
4. Sequence Diagram
5. ER Diagram

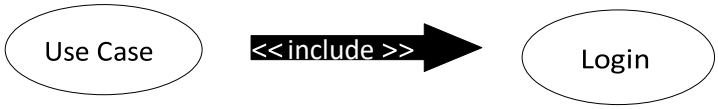
5.3.1. Use Case Diagrams

Use-case diagrams depict a system's high-level functions and scope. The interactions between the system and its actors are also depicted in these diagrams. In use-case diagrams, the use cases and actors define what the system does and how the actors interact with it, but not how the system works internally.

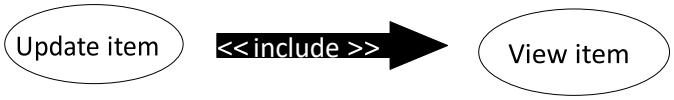
Additional Notes:

Following details are abstracted from the diagram for the sake of simplicity.

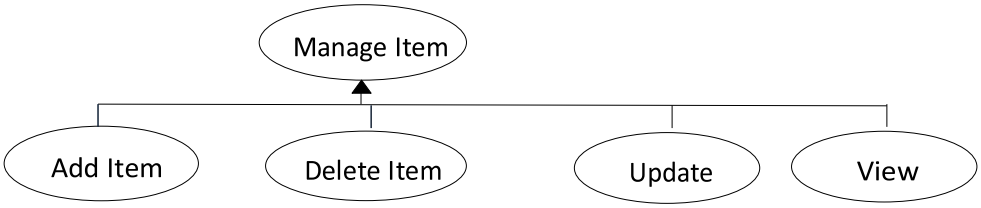
- All Use Cases have a <<include>> relationship with Login use case.



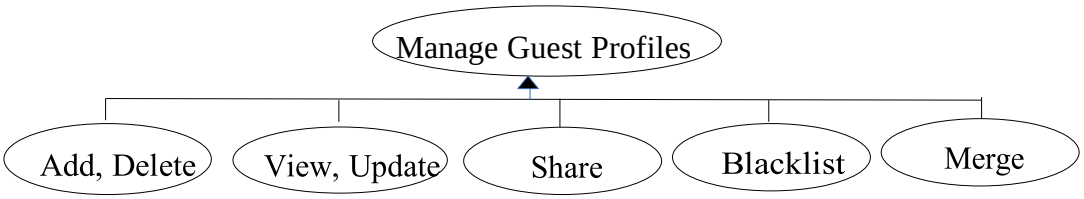
- All update action use cases have a <<include>> relationship with each relevant View action use case.



- All "manage" use cases have specializations as "add, delete, modify, view".



- Guest Profile Management as additional specializations



- Manager of each domain(ex: Housekeeping Manger – Room Clean status summery) have a use case of getting a summery.

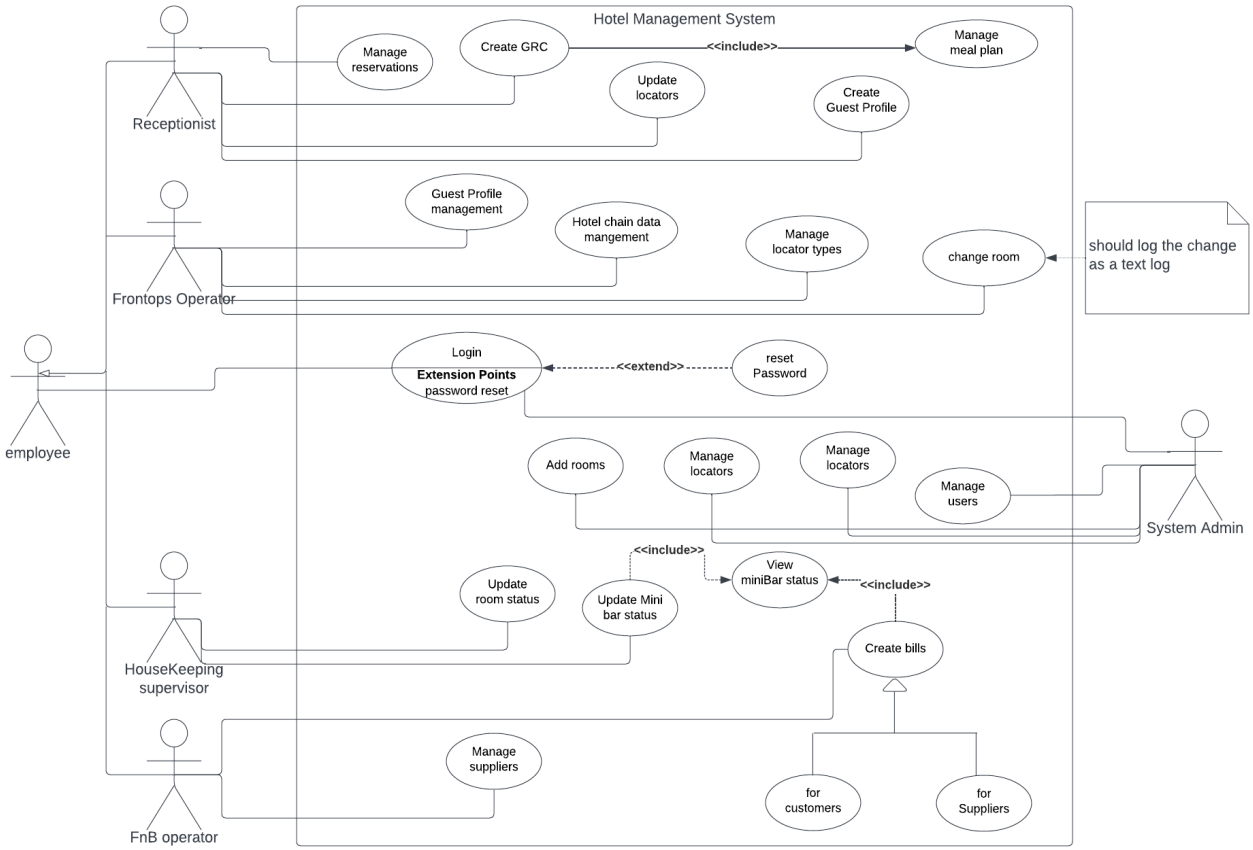


Figure 5.1: Use Case Diagram

5.3.2. Activity Diagrams

To demonstrate important activities in our system, we generated many activity diagrams.

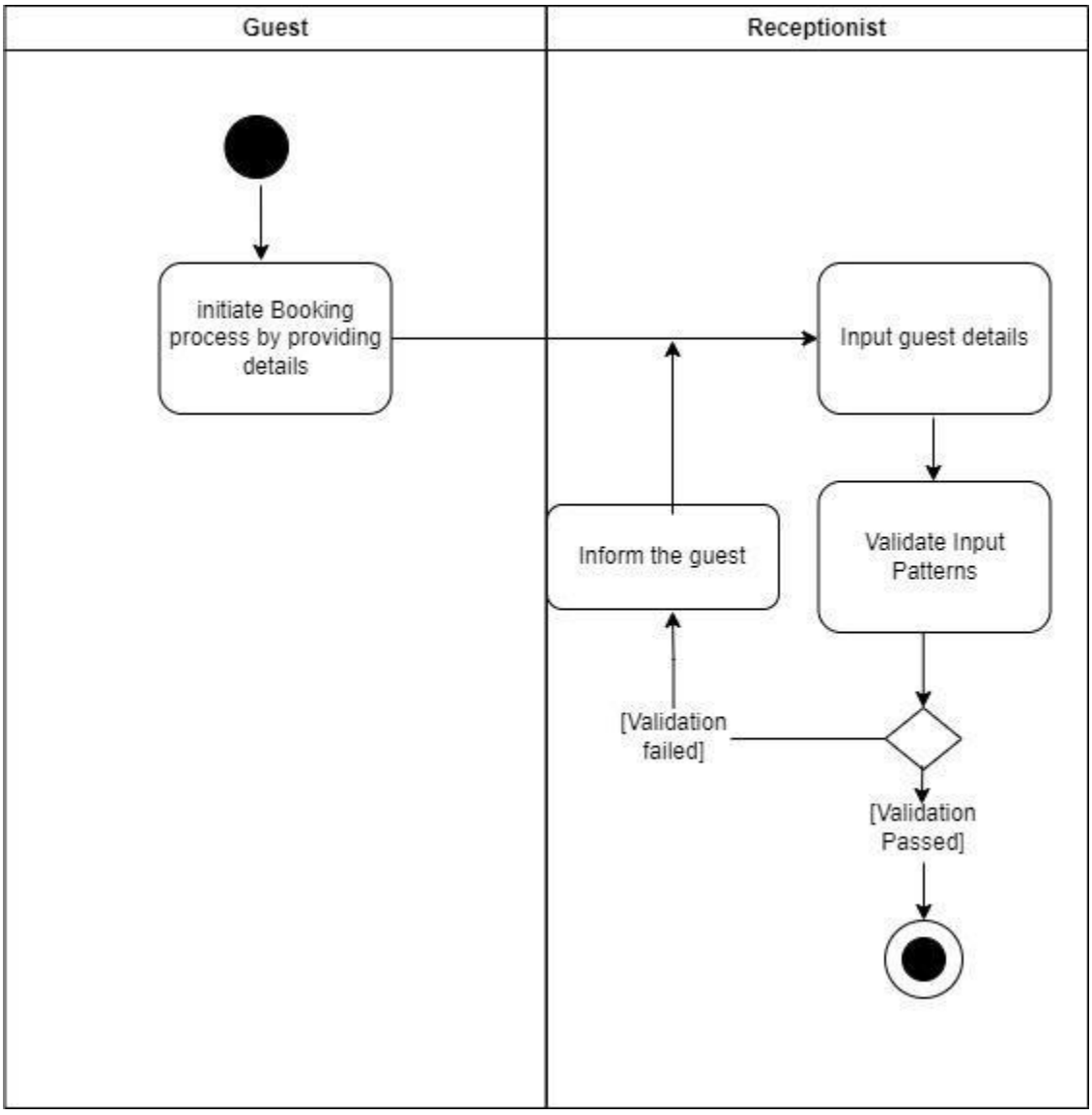


Figure 5.2 : Activity Diagram 1- Creating Guest registration card

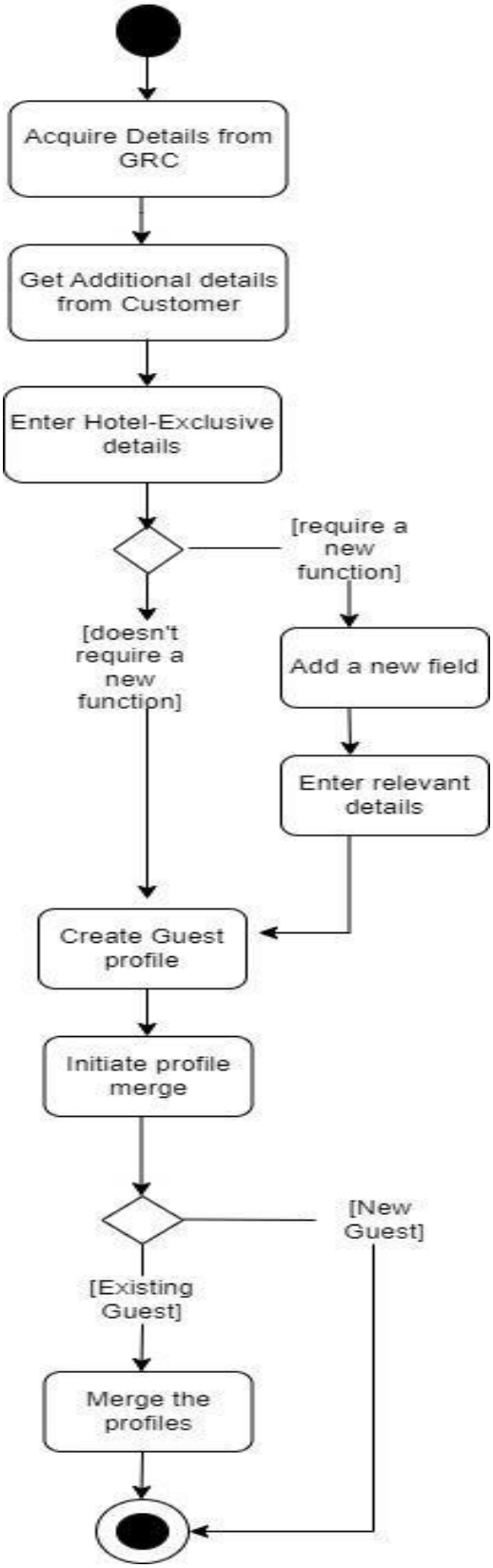


Figure 5.3 : Activity Diagram 2 - Creating and Merging guest profile

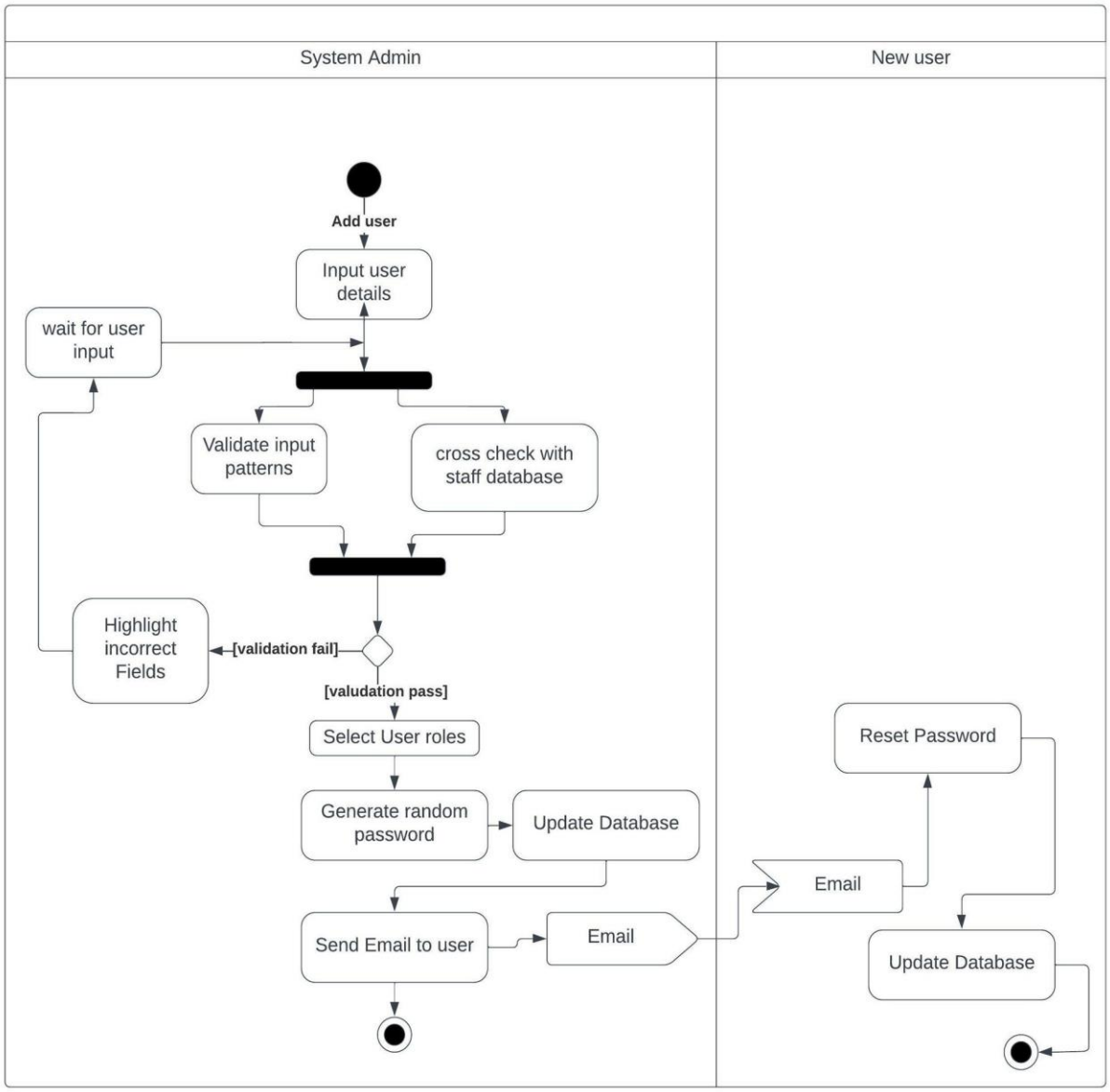


Figure 5.4: Activity Diagram 3 - Creating a new user

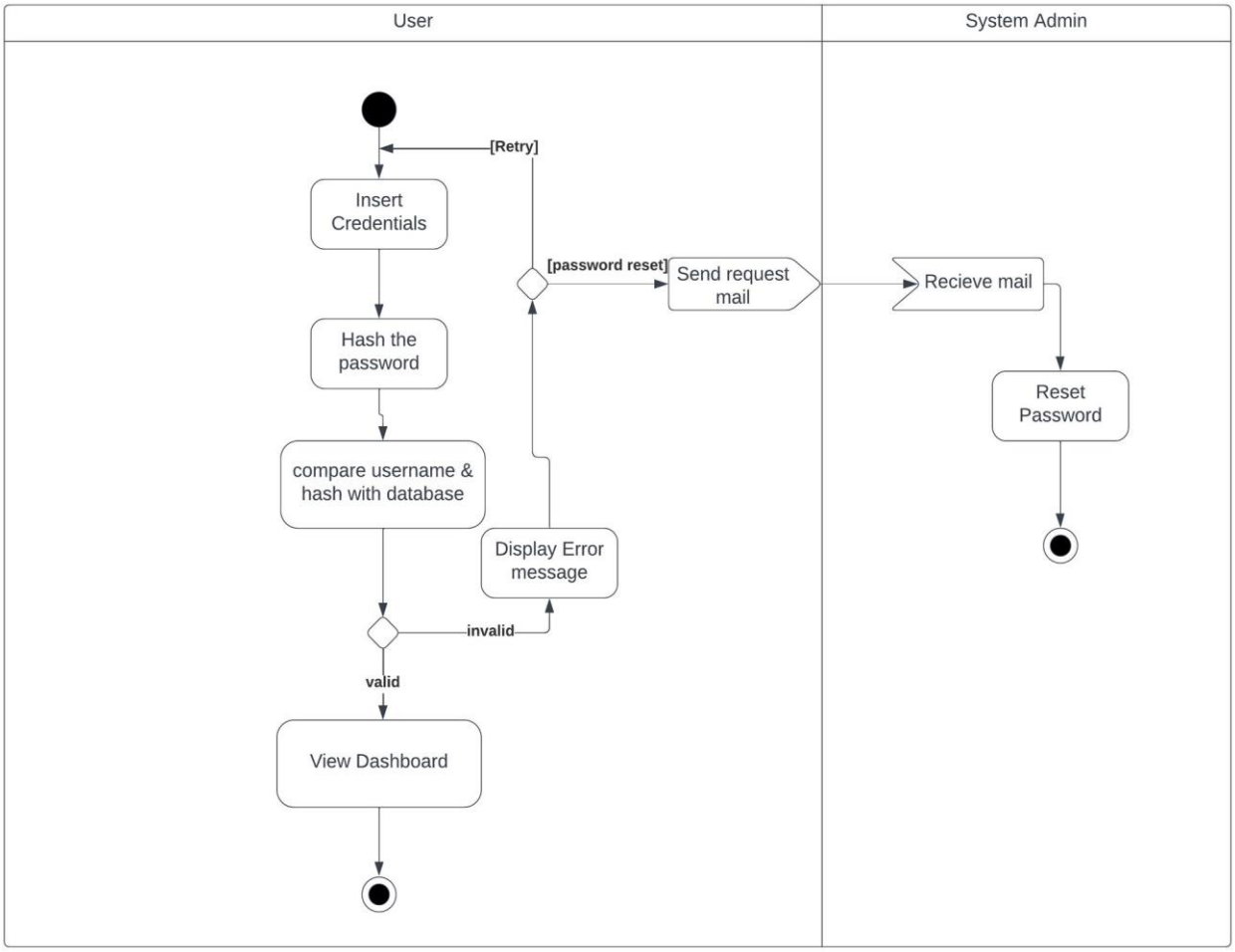


Figure 5.5 : Activity Diagram 4 - User Login

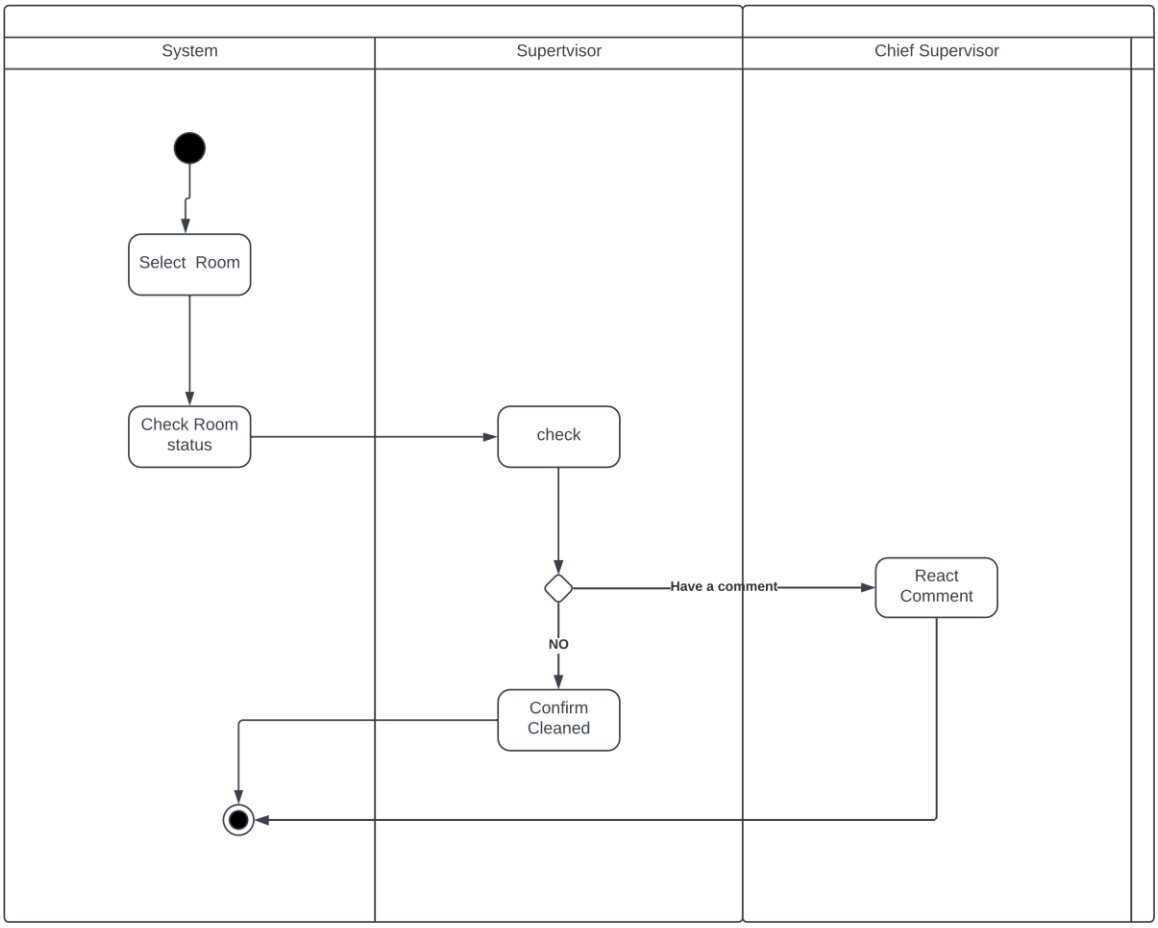


Figure 5.6 : Activity Diagram 5 - House Keeping

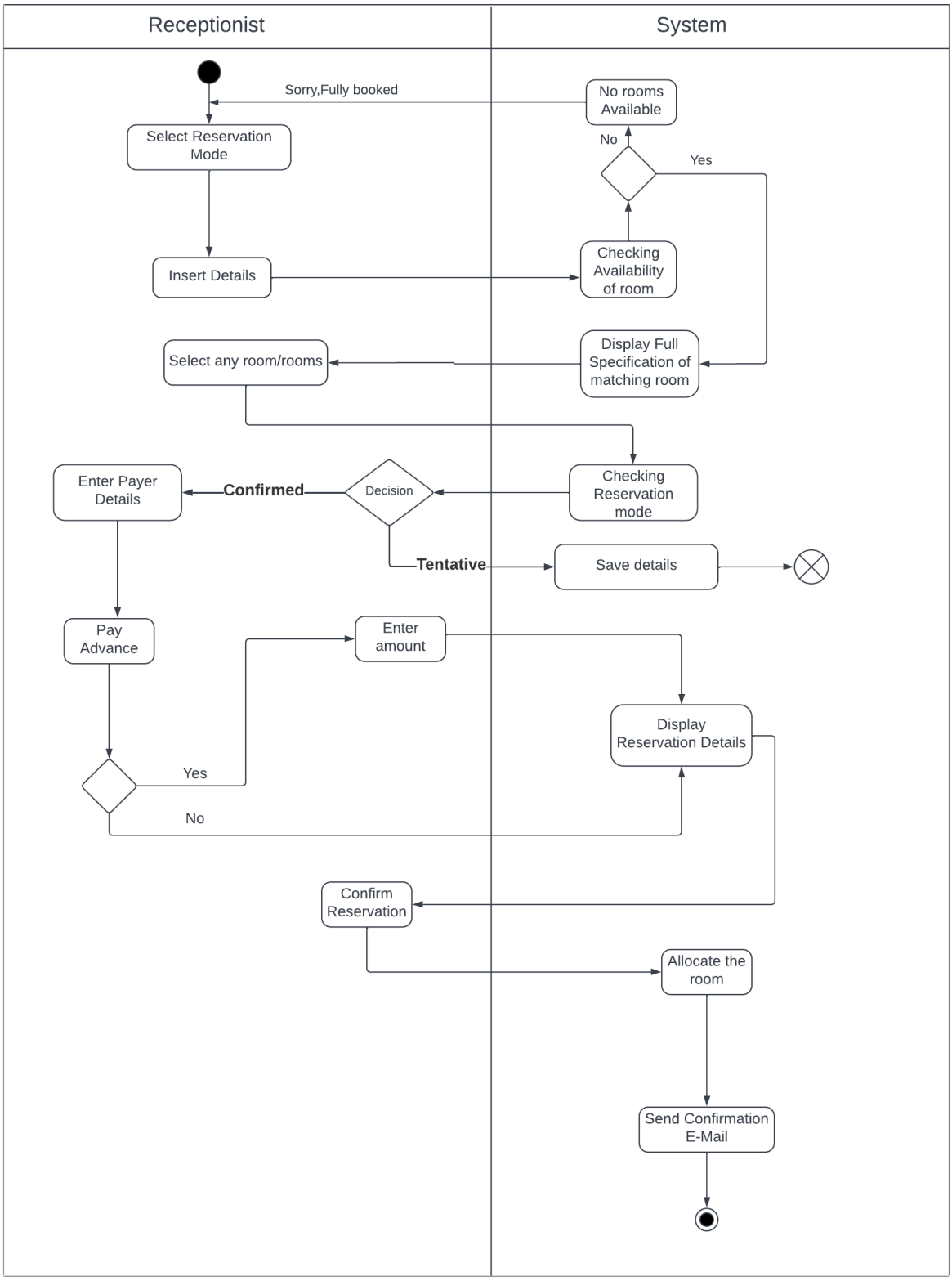


Figure 5.7: Activity Diagram 6- Reservations

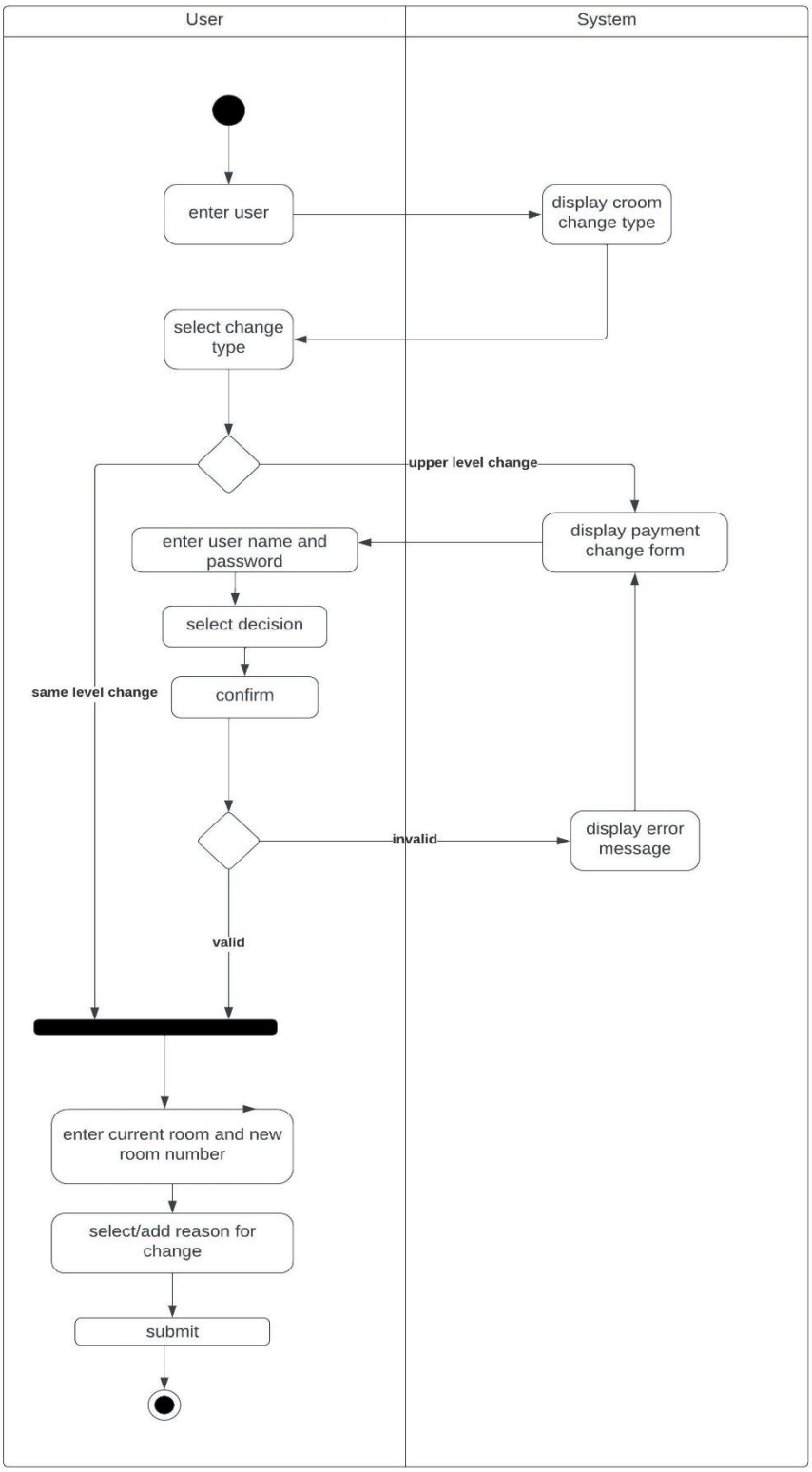


Figure 5.8: Activity Diagram 7- Room Management

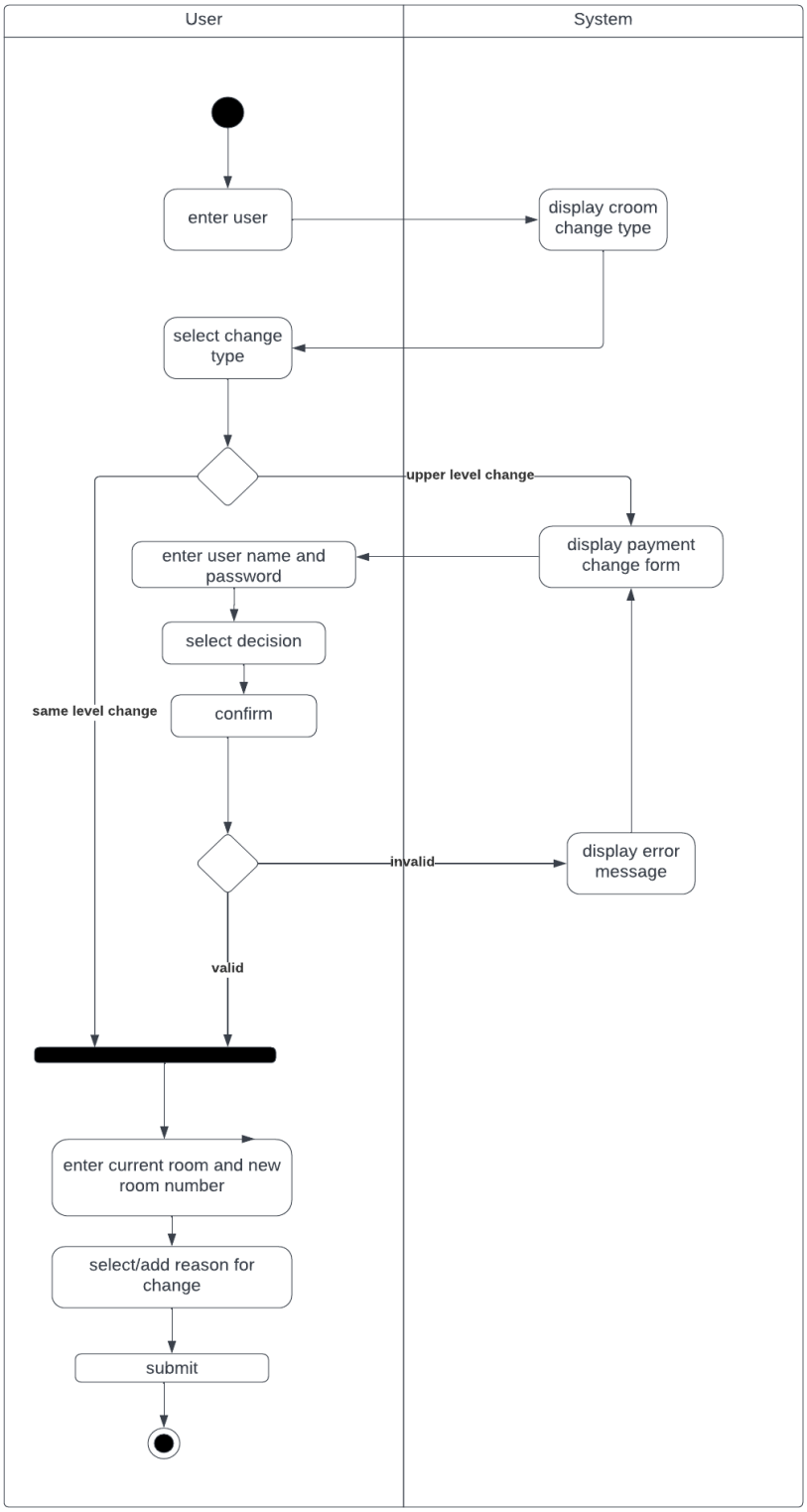


Figure 5.9: Activity Diagram 8- Billing

5.3.3. Class Diagram

A class diagram is a sort of static structure diagram that portrays the structure of a system by demonstrating the system's classes, their properties, methods, and the relationship among objects

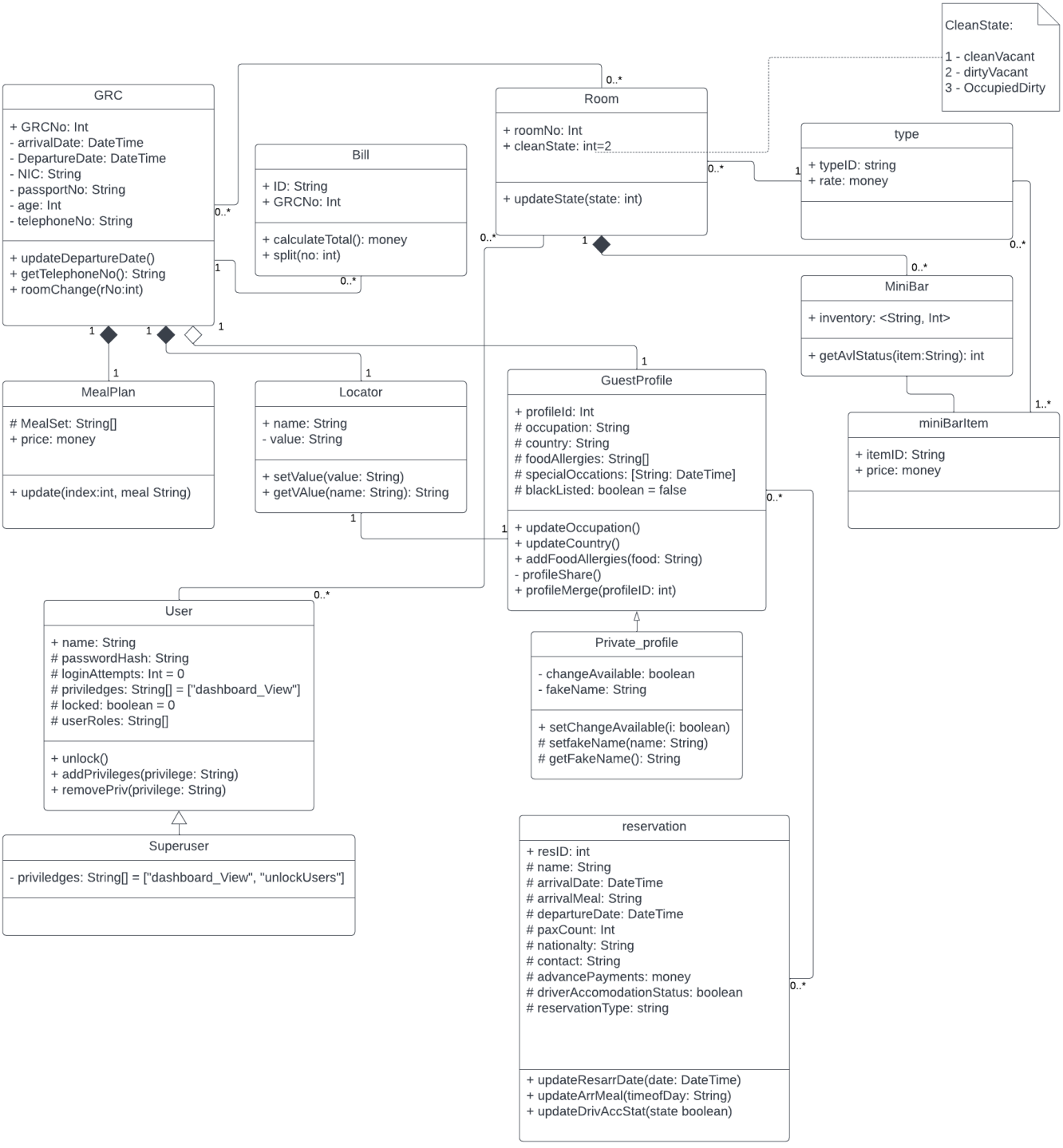


Figure 5.10: Class Diagram

5.3.4. Sequence Diagrams

A sequence diagram depicts item connections in chronological order. It depicts the scenario's objects as well as the sequence of messages that must be passed between them in order for the scenario to be useful. As shown above, we created many sequence diagrams for each activity diagram.

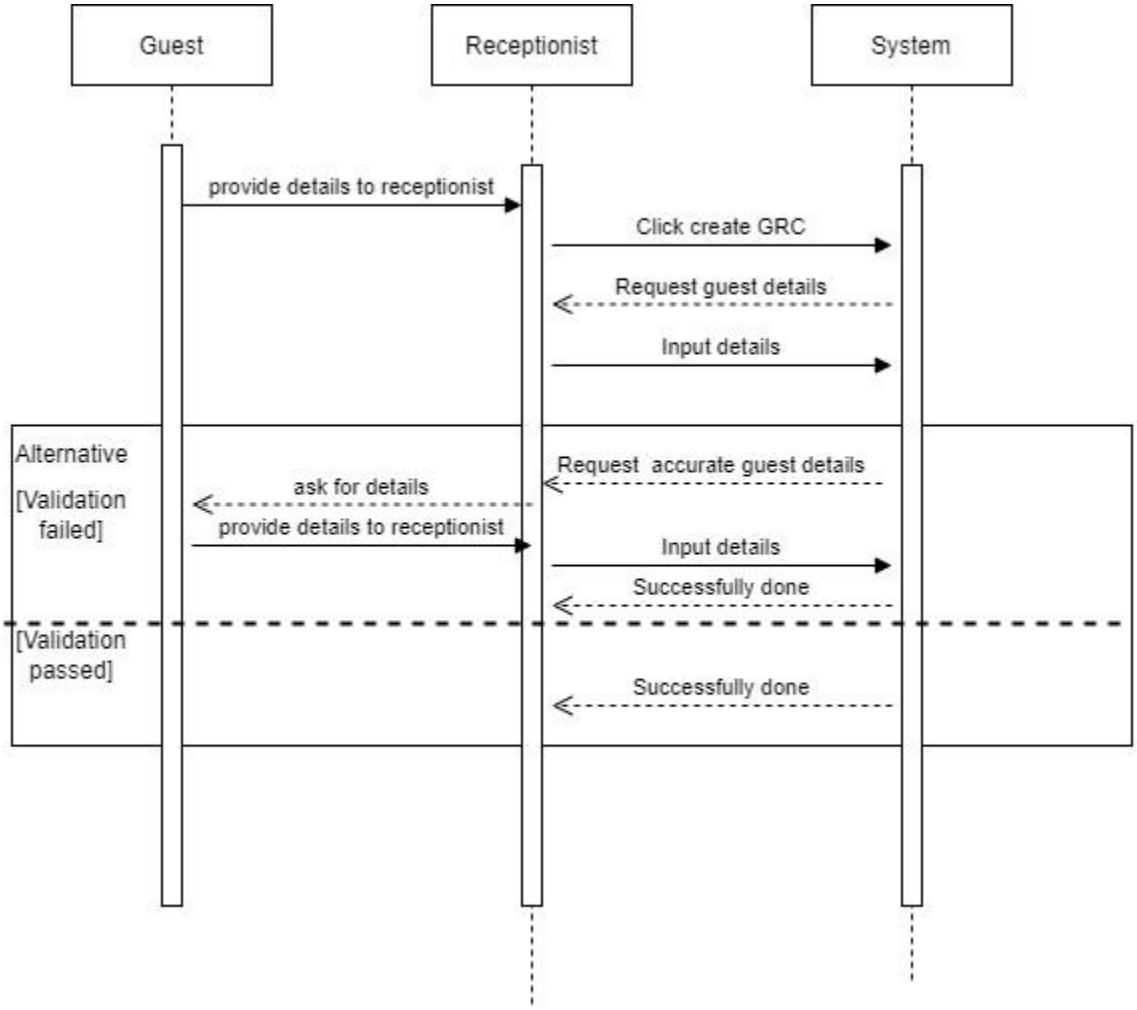


Figure 5.11 : Sequence Diagram 1-Creating Guest registration card

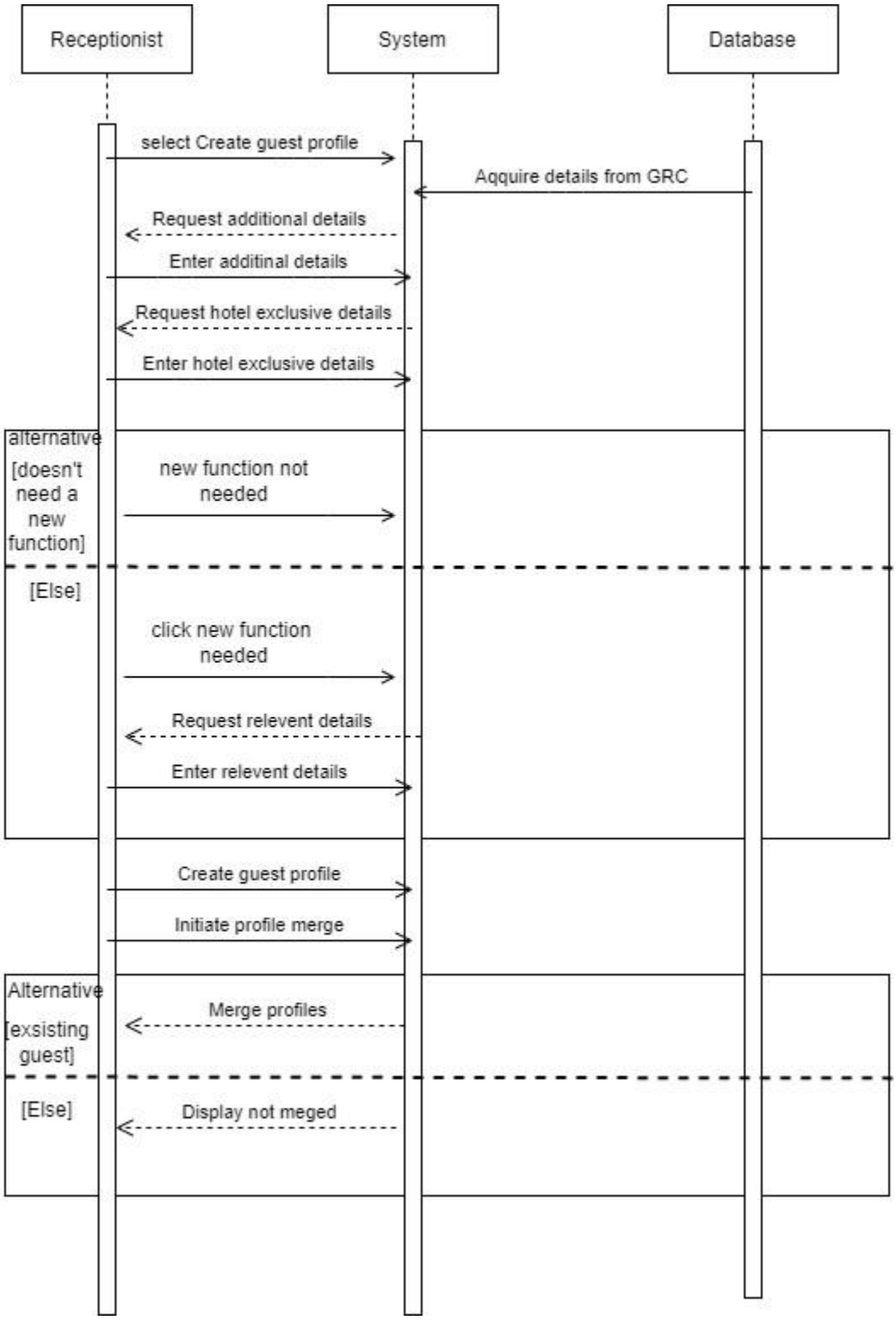


Figure 5.12 : Sequence Diagram 2 - Creating and merging guest profile

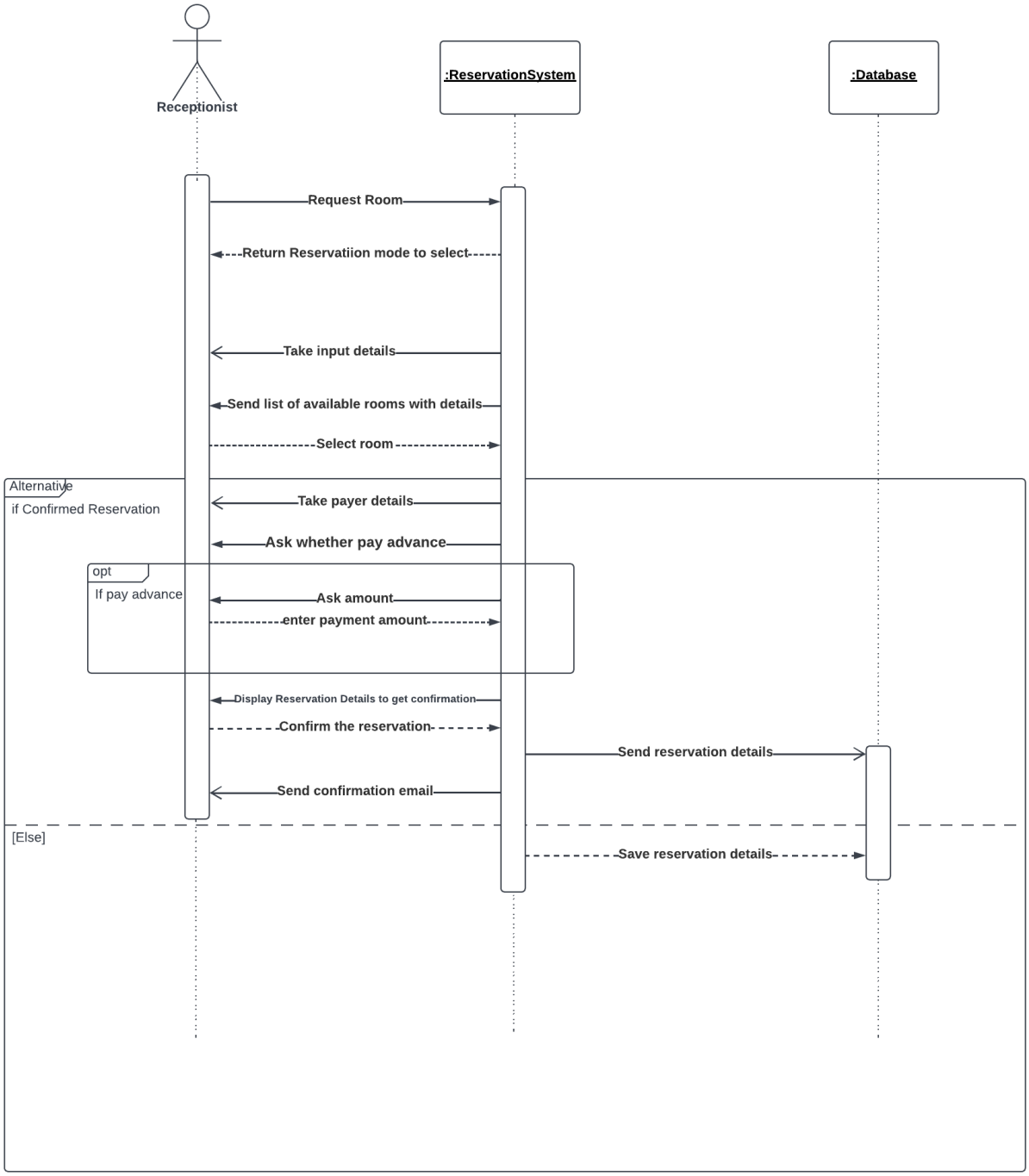


Figure 5.13: Sequence Diagram 3 - Reservations

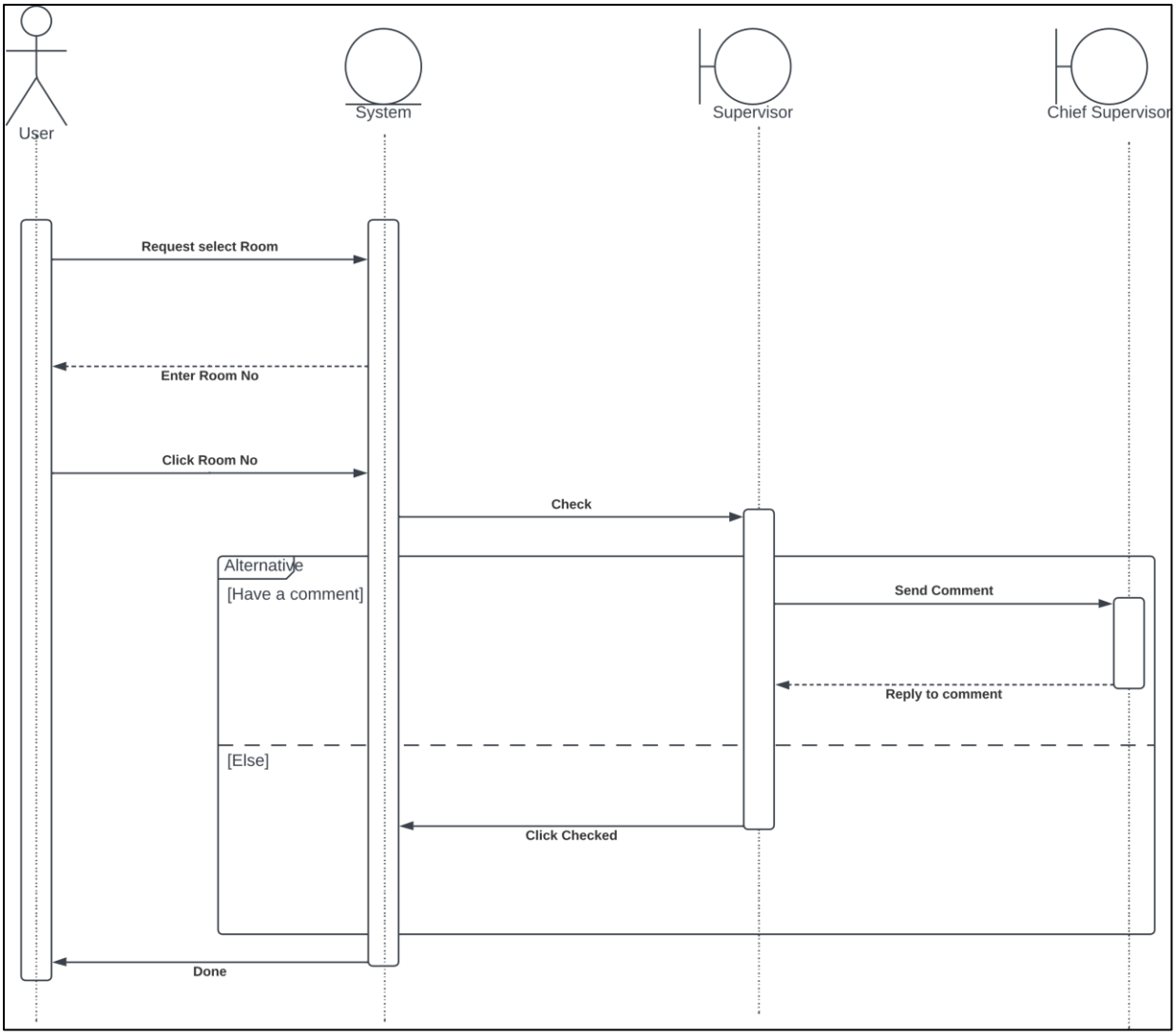


Figure 5.14 : Sequence Diagram 4 - House Keeping

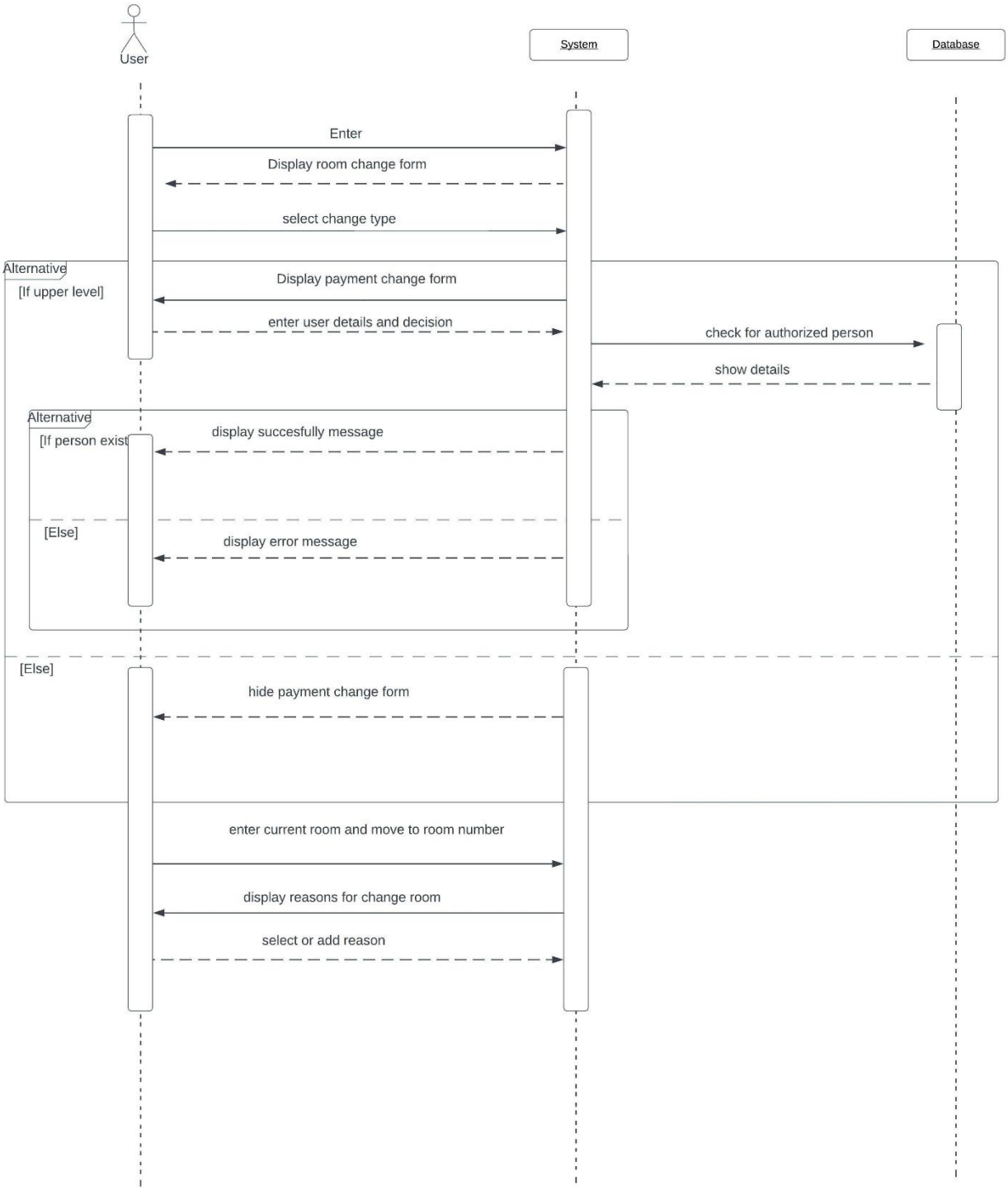


Figure 5.15: Sequence Diagram 5- Room Management

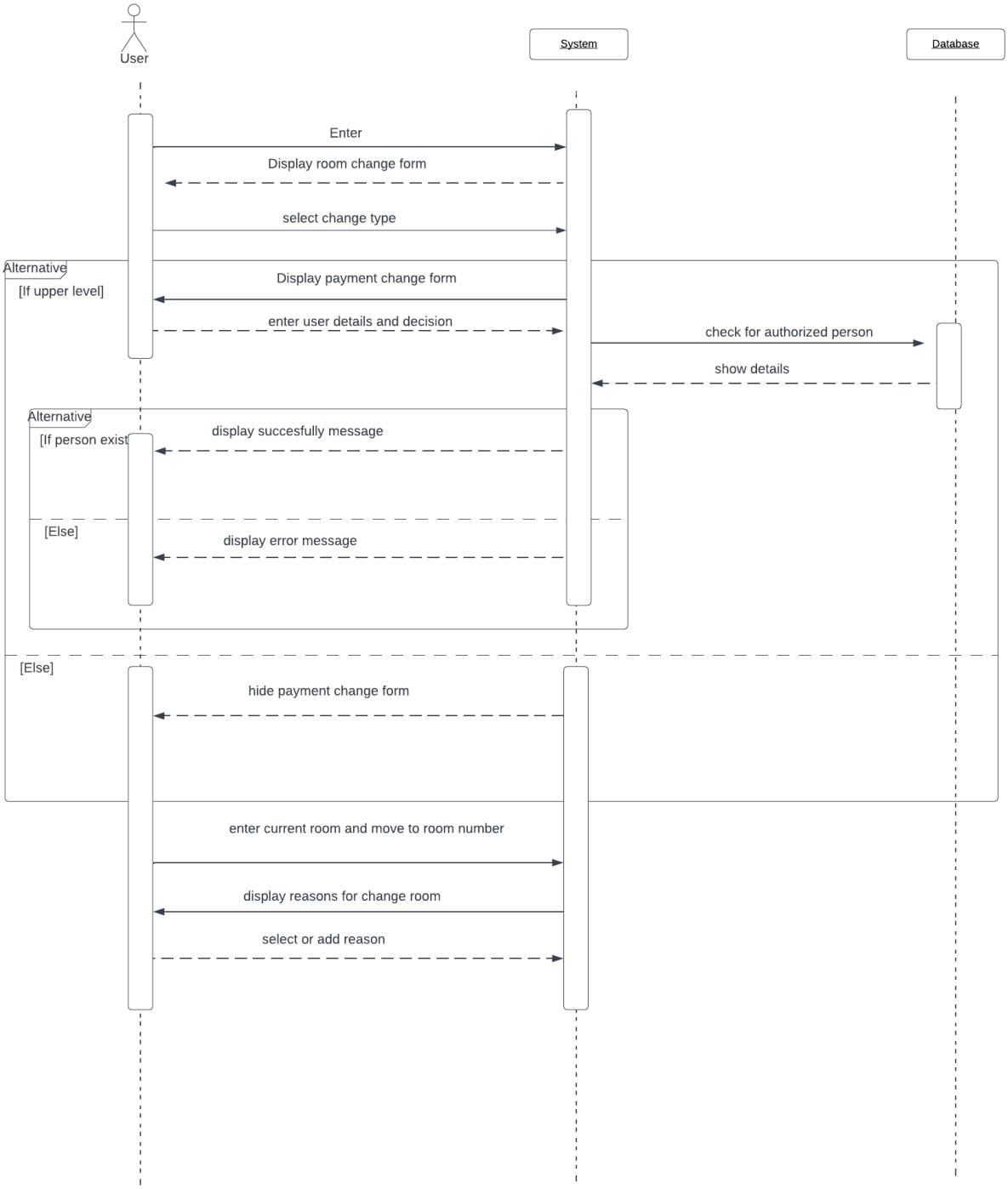


Figure 5.16: Sequence Diagram 6- Billing

5.3.5. Component Diagram

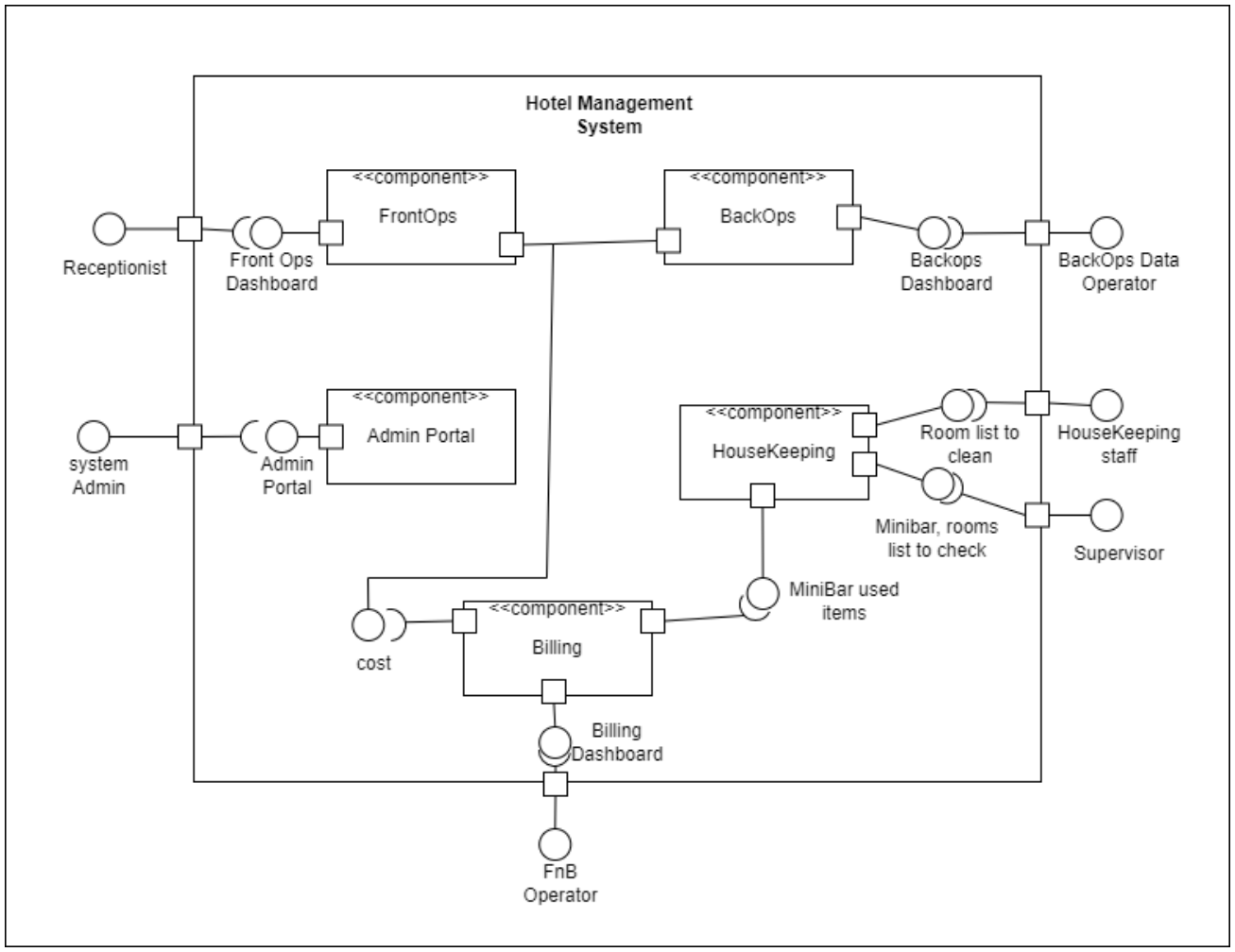


Figure 5.17 : Component Diagram

5.3.6 EER

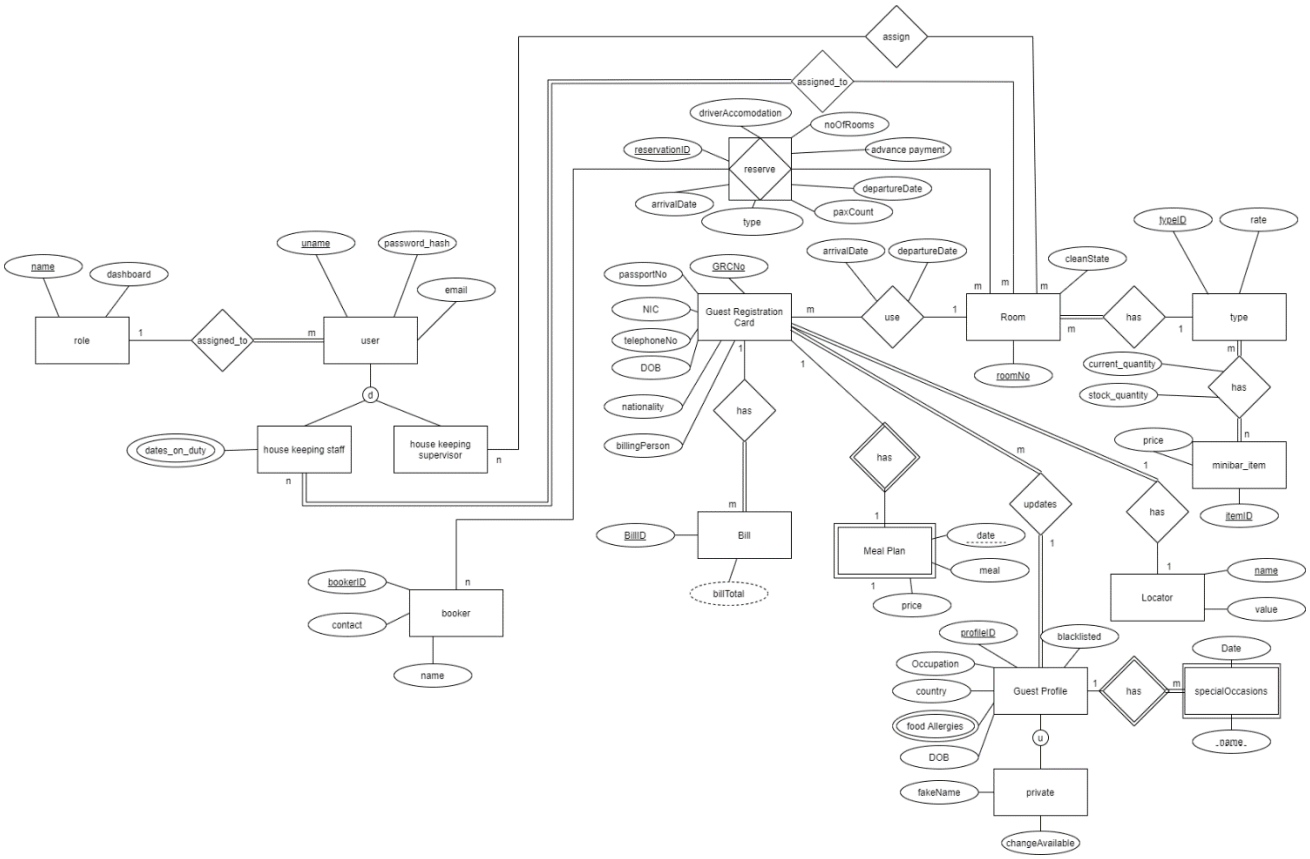


Figure 5.18: EER

EER diagram is also attached as a link at references part since this is not much clear.

5.4. Summary

Through this chapter analysis and design, we have represented diagrams like Use case, Activity, Class, Sequence, Component, Context, and ER which provides a basic understanding of our design and analysis mechanism.

Chapter 6 Implementation

6.1. Introduction

Let us look at the development plan in a more practical way.

We selected the Agile Scrum methodology for our development process. We are using Trello to manage tasks more efficiently as well.

6.2. System and Software

System should be able to accommodate a PHP Laravel server to test while developing. We must use our laptops for the development process. Hosting could be done on Azure cloud free tier given to University of Moratuwa undergraduates.

Minimum:

2GB memory

Core I3 Processor

HDD 20GB

- PHP Laravel version: 8
- MariaDB >v8.0
- Visual Studio Code/ PHP Storm for Coding
- Figma for UI design

A test server could be started using the command “php artisan serve”.

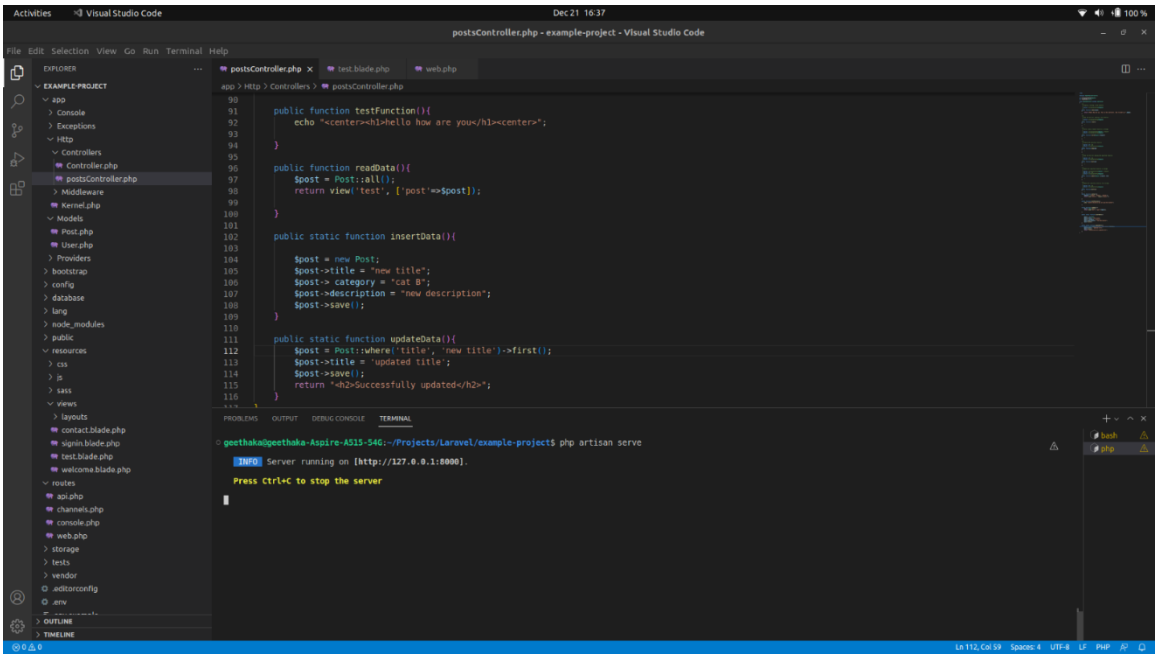


Figure 6.1: php artisan serve

We could access the server on localhost:8000.

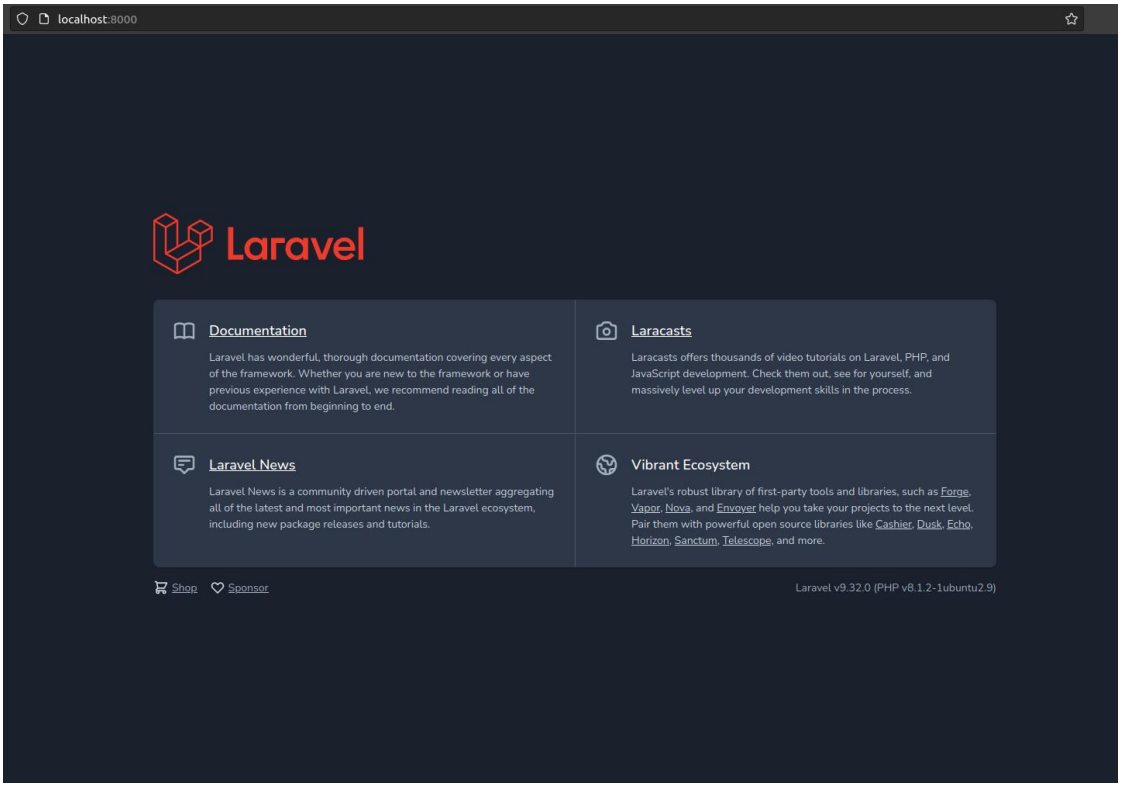


Figure 6.2: localhost:8000

PHP Laravel behaves according to the MVC model.

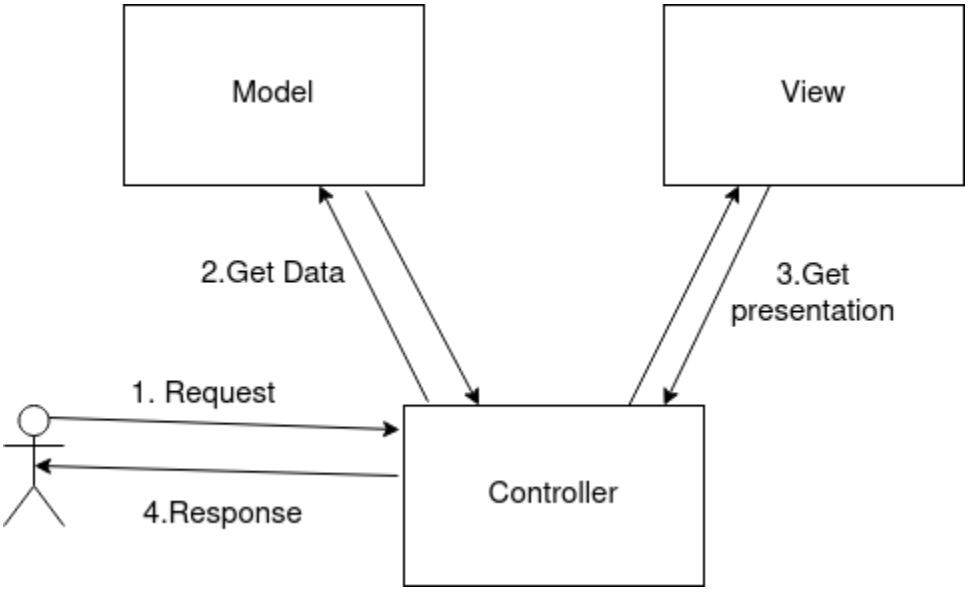


Figure 6.3 : MCV Model

We learned CRUD (Create, Read, Update, Delete) operations in Laravel.

Controllers should get the data from the model and pass it to the View upon receiving user request through a specified route.

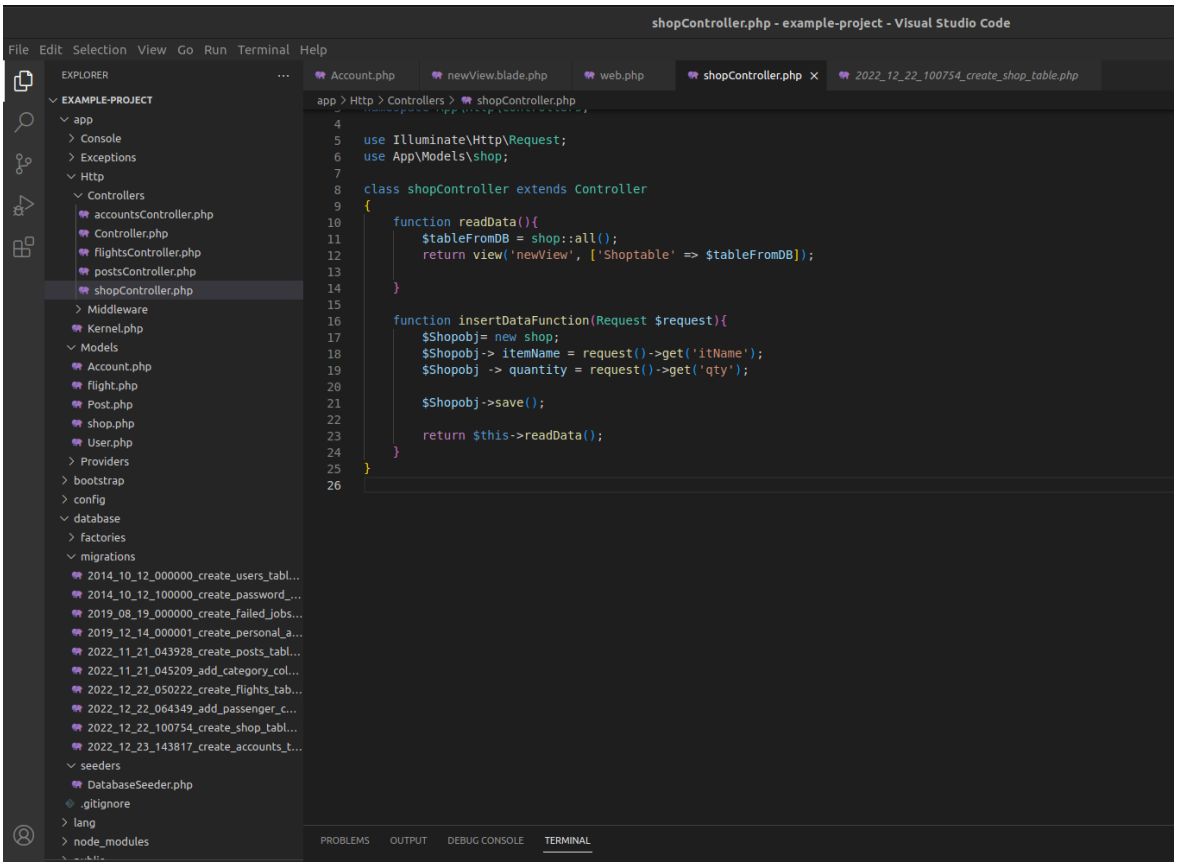


Figure 6.4 : Controllers in CRUD

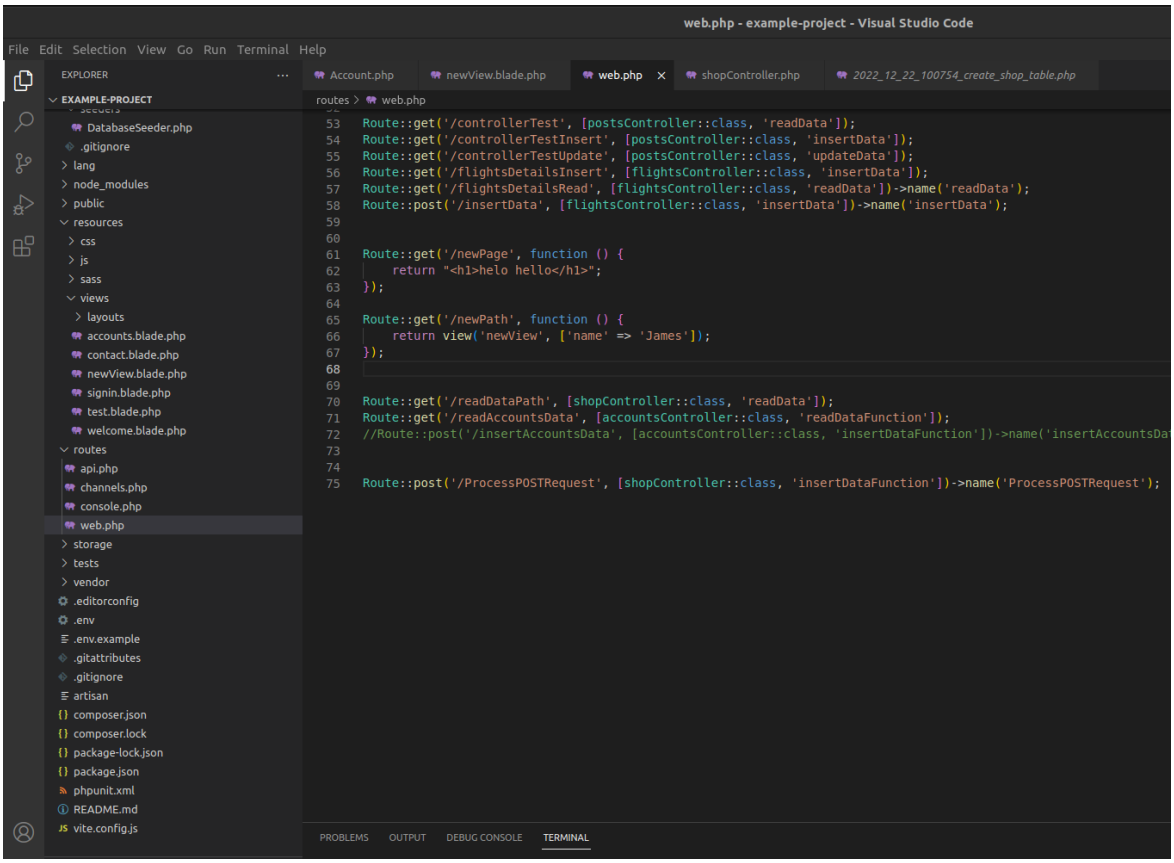


Figure 6.5: Routes

Figure 6.6: Laravel Output

6.3. Summary

We have started the development process in parallel with learning the Laravel framework.

We are familiarizing ourselves with Laravel using the trial-and-error method.

Chapter 7 Discussion

7.1. Introduction

In this chapter, we discuss the evaluation and testing of our solution and how this differs from other solutions in the market along with further developments and implementation.

7.2. Evaluation and testing

We are currently at the initial stage of developments. Therefore, we have not yet tested or evaluated the solution. However, we have a clear plan and timeframe about the testing phase.

7.3. How the solution differs from other solutions

Even though there are few hotel management software in current market, none of them offers a user-friendly UI and all-in-one platform. One or several basic functions are also absent from above software. Most of these software offers outdated systems which were made using outdated frameworks and because most of these focused only one sector of the management process, hotels are forced to pay for multiple software to get the whole process covered. This is an extremely expensive option.

However, our hotel management software is a direct solution for all above mentioned problems. Our software offers an extremely user-friendly UI which can be used by almost anyone with little to no past experience.

Also, we cover every sector in hotel management process including but not limited to guest registration, billing and housekeeping. Since our software includes all the above functions and more, it will be very cost effective for customers to choose our solution which will provide an all-in-one platform with every required function.

7.4. Further development

The system can be further developed to fully automate several functions like guest registration and housekeeping so they will require minimal staff involvement. Also, this can be further optimized to send emails about registration details and promotions done by the hotel to customers.

Chapter 8 References

8.1. EER Diagram

https://uniofmora-my.sharepoint.com/:i:/g/personal/pitigalapkdgd_20_uom_lk/EdCewSJkI4lJliQY3Pe34dIBetQtsfo-jzCpYYLDURxQLg?e=OPL9IY

8.2. OOAD diagrams

https://uniofmora-my.sharepoint.com/:f:/g/personal/pitigalapkdgd_20_uom_lk/EkzNMO20BEpHulgNGpikUFEBvVuwmjrQx5C9KXMUSwEBxA?e=5DmGcW

Chapter 9 Appendixes

Appendix A - Individuals Contribution to the Project

Name of student: 204161D Pitigala P.K.D.G.D

As the team leader, I had to bear the responsibility of the project manager. We were lucky enough to receive a Hotel Management system dev project from the company "CircleBook Pvt Ltd".

- I had to coordinate with the mentor from the company to get the project going. And had to act as the point person from the team when communicating with the company. And had to take decisions on deciding the project management plan.
- Went to the company physically for the requirements gathering part.
- Finding resources to learn the technologies needed to implement the project was also my responsibility. I managed to find a suitable Udemy course to learn Laravel.
- Learning Laravel and Bootstrap using the Udemy course and YouTube.
- Main Component I am responsible for is the Administration portal. I studied the admin portal of systems like Ansible Tower to an idea.
- Learned basic GitHub operations.
- Designed the UI for the Admin portal using "Figma" . They are attached in Appendix C.

Name of student: 204013D Aysha M.R.S

- I helped the team to gather the requirements provided by Circlebook (Pvt) Ltd.
- I attended the mentoring sessions about PHP Laravel.
- Learned UI/UX Design principles
- Learned GitHub operations for collaborative coding.
- I learned PHP Laravel, and bootstrap through the internet.
- I learned UI design using Figma and I designed the UI diagram for the front office-back operations part. They are attached in Appendix C.
- I had to conduct some research regarding reservations that happen at a hotel and gather data about the main types of reservations.

Name of student: 204097K Kaushalya D.M.G

- Learning PHP Laravel framework using official documentation and Udemy course.
- Learned to use Figma for UI/UX designing.
- Designed housekeeping dashboard and support pages using Figma. They are attached in Appendix C.
- Learned CSS and Bootstrap for front end development.
- I did research about the housekeeping process inside a hotel.
- Did research on how the minibar system works.
- Reached out to contacts working in hotels to verify the details.
- Learned basic GitHub operations.

Name of student: 204064H Gunasiri G.C.S

We got a chance to work on this project with a great mentorship from “Circlebook (Pvt) Ltd.” .

- I had to attend Knowledge sessions about technologies of the CircleBook company to understand the workflow of the company .
- Learning PHP Laravel framework using official documentation and Udemy course.
- Learned to use Figma for UI/UX designing.
- Designed housekeeping dashboard and support pages using Figma. They are attached in Appendix C.
- Learned CSS and Bootstrap for front end development.
- I did research about the room change and upgrade process inside a hotel.
- Did research on how the billing system and layouts.
- I looked at similar systems to get an idea.
- Learned basic GitHub operations for collaborative coding.

Name of student: 204152C Pemasiri M.P.T.B.S

- I had to attend Knowledge sessions about principles and technologies of the CircleBook company to understand the workflow of the company
- I attended few workshops that were conducted by our mentors to get an idea about UI/UX design and PHP Laravel framework
- I learned about the UI/UX designing and designed the GRC and guest profile UIs. They are attached in Appendix C.
- I learned Laravel and used that knowledge for coding purposes.
- I also learned about Bootstrap and CSS.
- I conducted research about front office operations in hotels and problems they are facing.

Appendix B: Action Plan

Task	December	January				Februrary				March				April			
	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Learning PHP Laravel																	
interim report creation																	
interim presenations																	
Normalization, mapping , Database Creation																	
Main dashboard Development																	
Front End finetuning																	
Add and Update function 1 dev - sprint																	
Add and Update function 2 dev - sprint																	
Add and Update function 3 dev - sprint																	
unit Testing																	
System Finetuning																	

Figure 9.1: Action Plan

Appendix C: UI Designs

Name of student: 204161D Pitigala P.K.D.G.D

Administration Portal

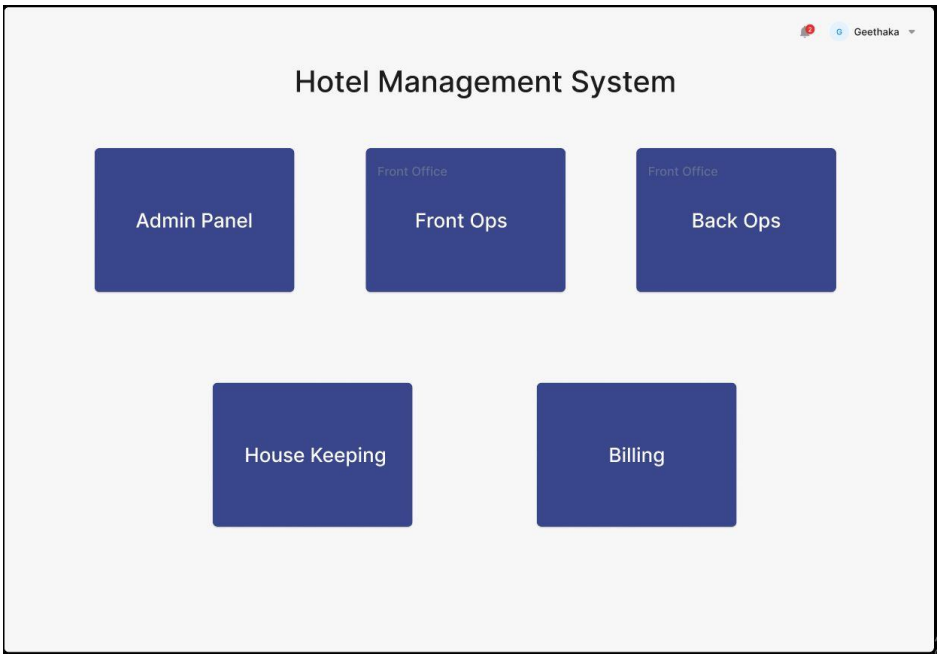


Figure 9.2: Administration Portal 1

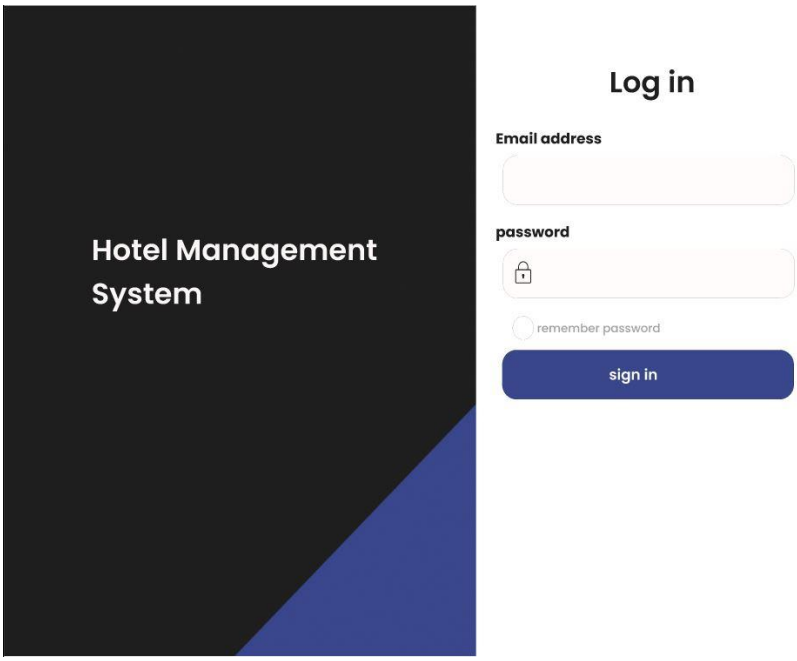


Figure 9.3: Administration Portal 2

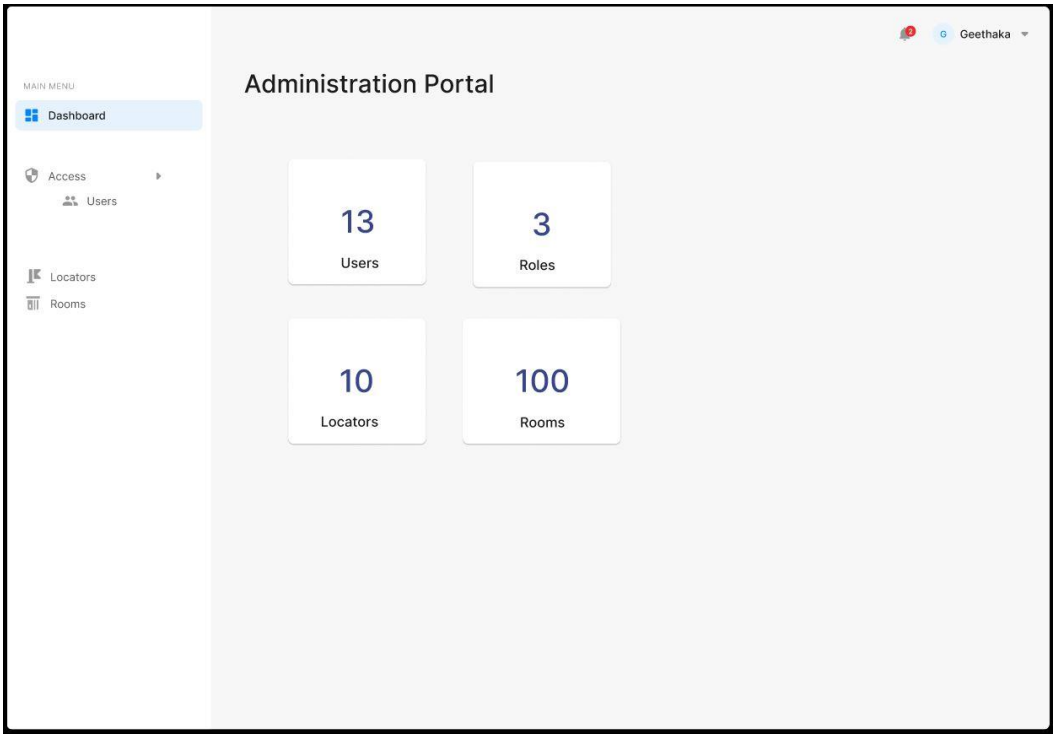


Figure 9.4: Administration Portal 3

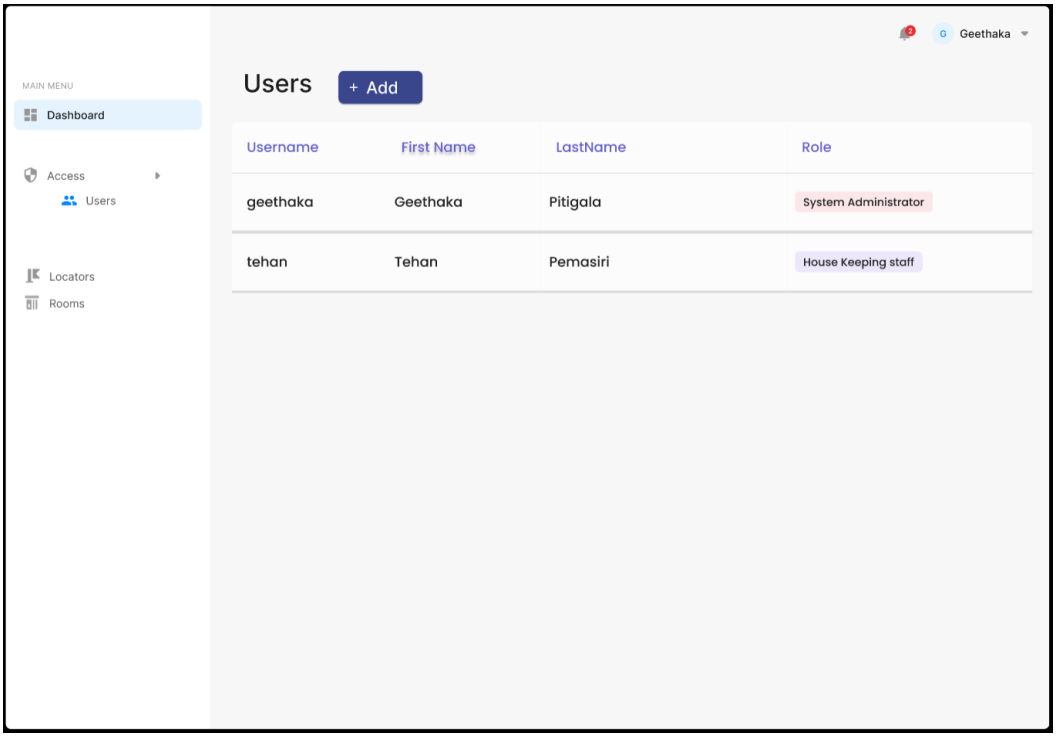


Figure 9.5: Administration Portal 4

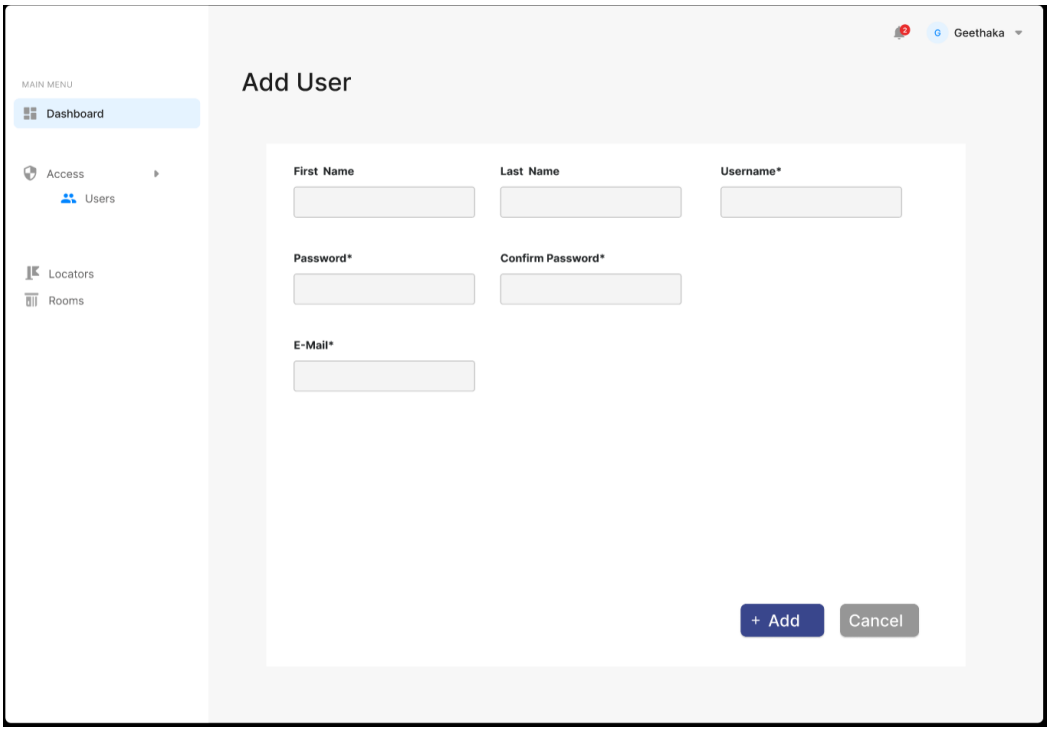


Figure 9.6: Administration Portal 5

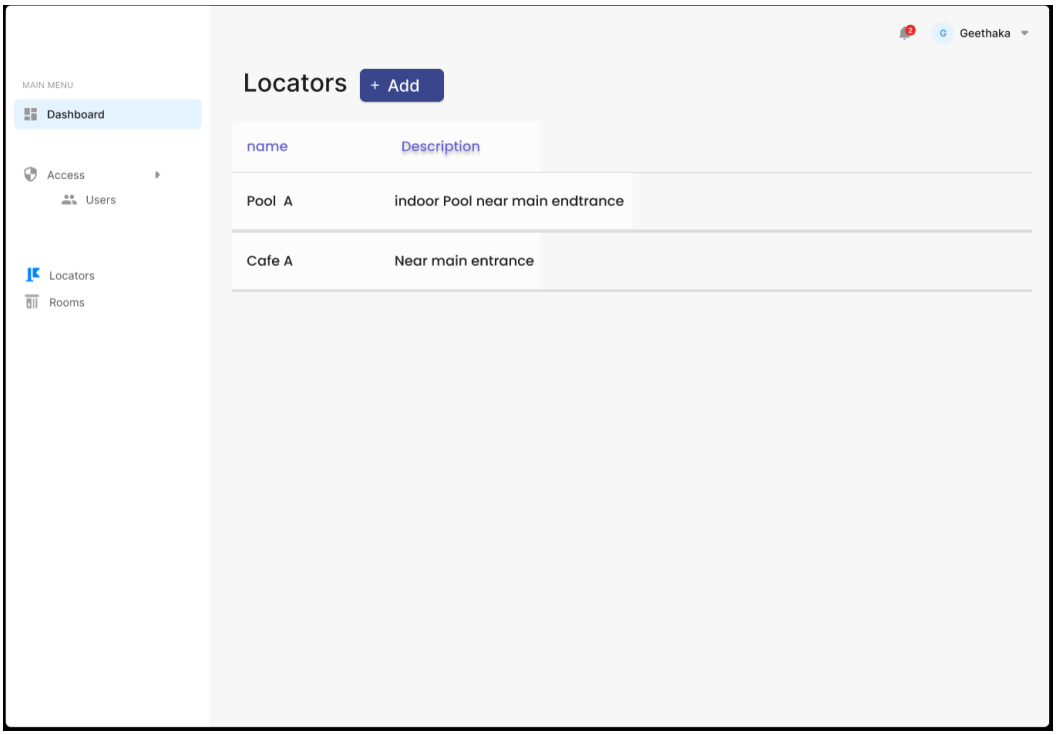


Figure 9.7: Administration Portal 6

Name of student: 204013D Aysha M.R.S

Front Office Back Operations Portal

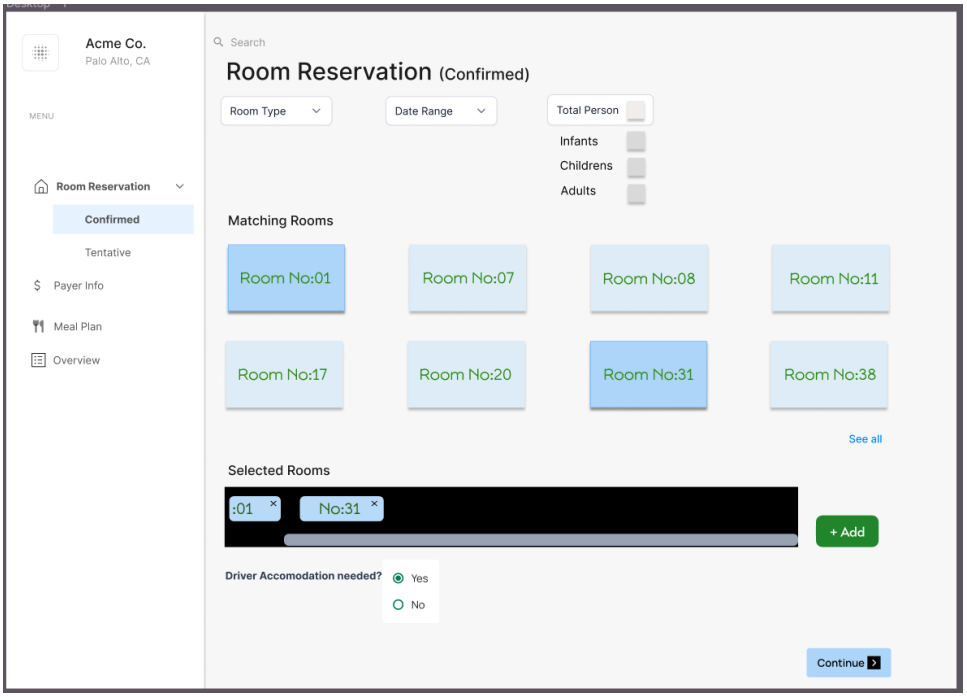


Figure 9.9: Front Office Back Ops 1

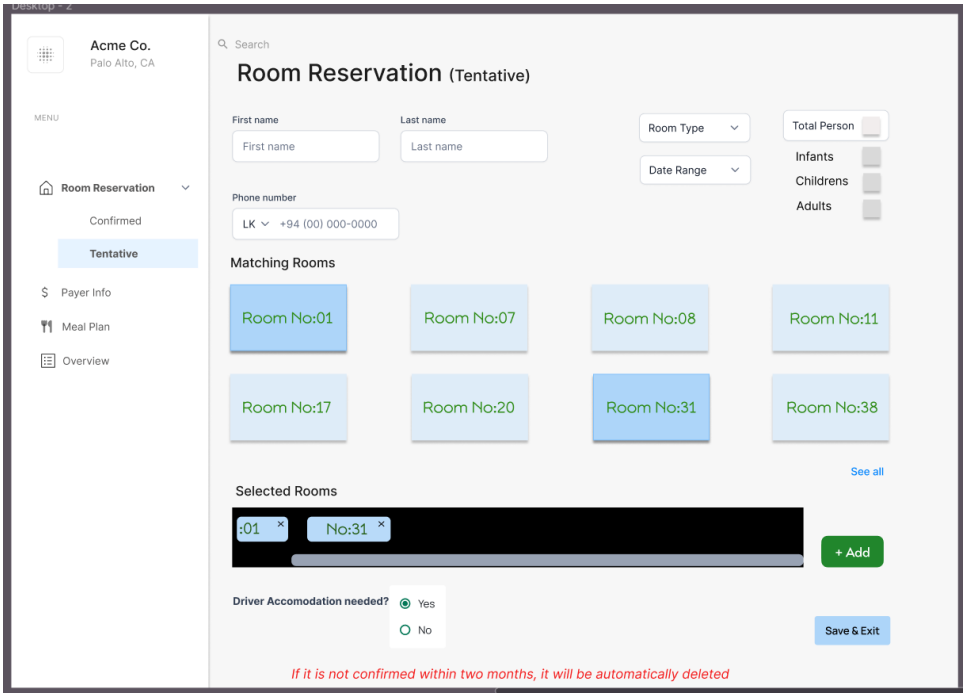


Figure 9.8: Front Office Back Ops 2

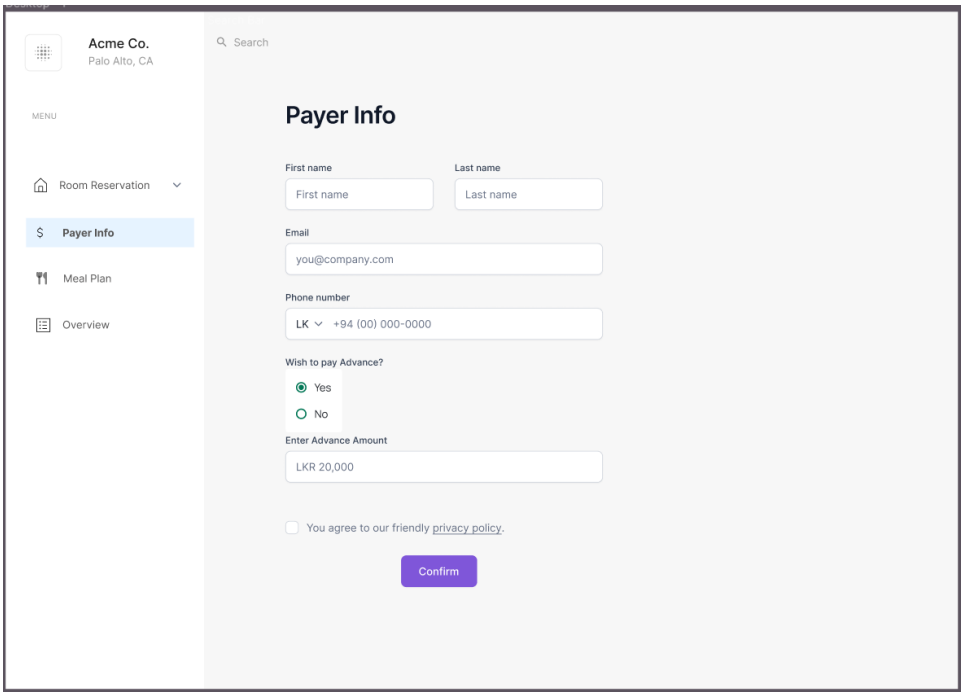


Figure 9.10: Front Office Back Ops 3

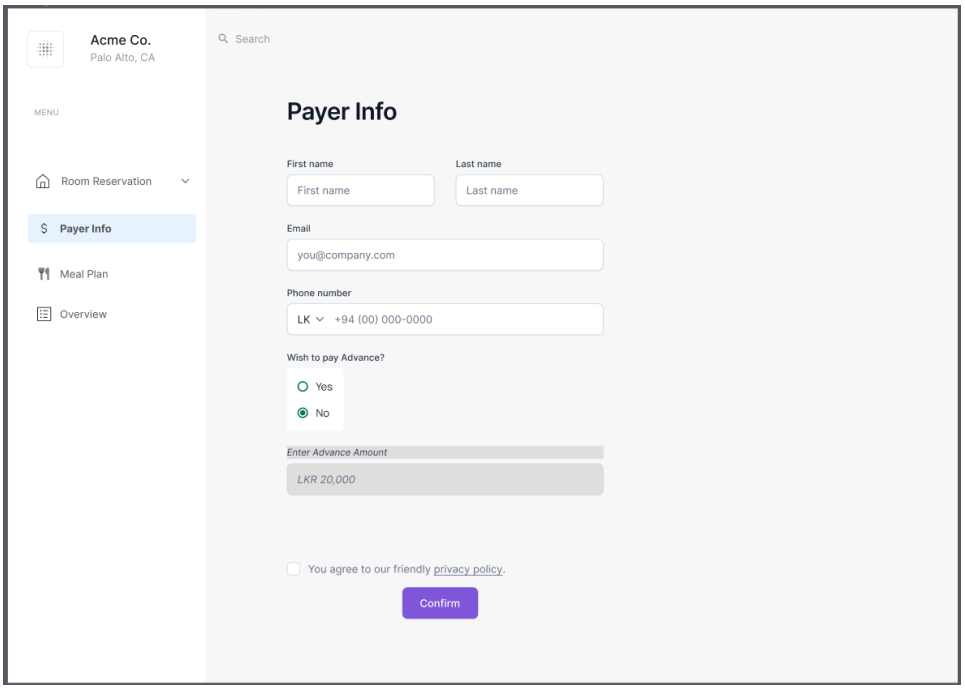


Figure 9.11: Front Office Back Ops 4

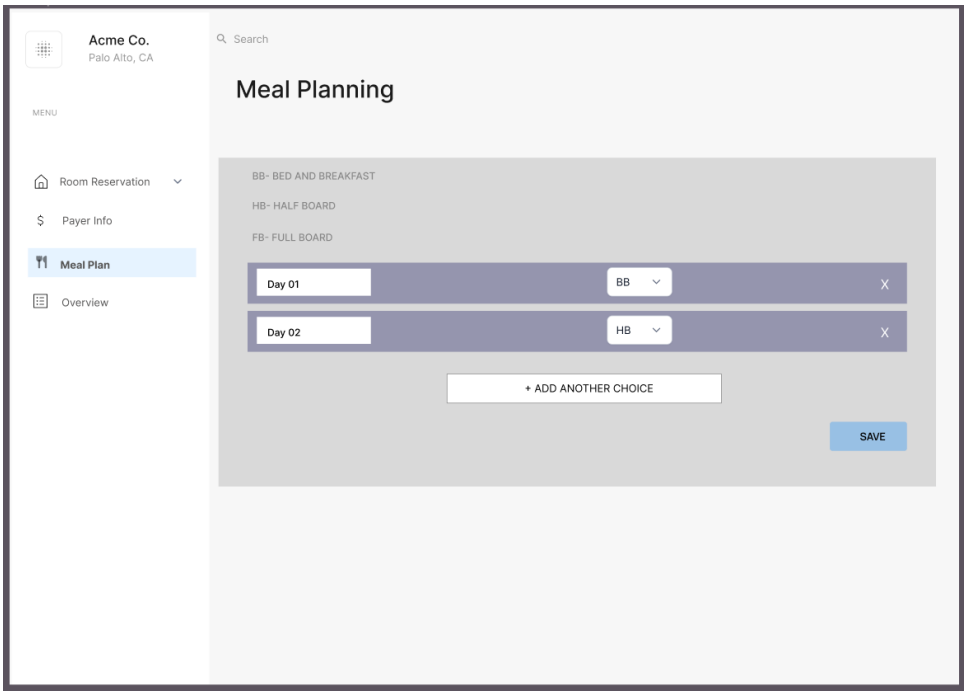


Figure 9.12: Front Office Back Ops 5

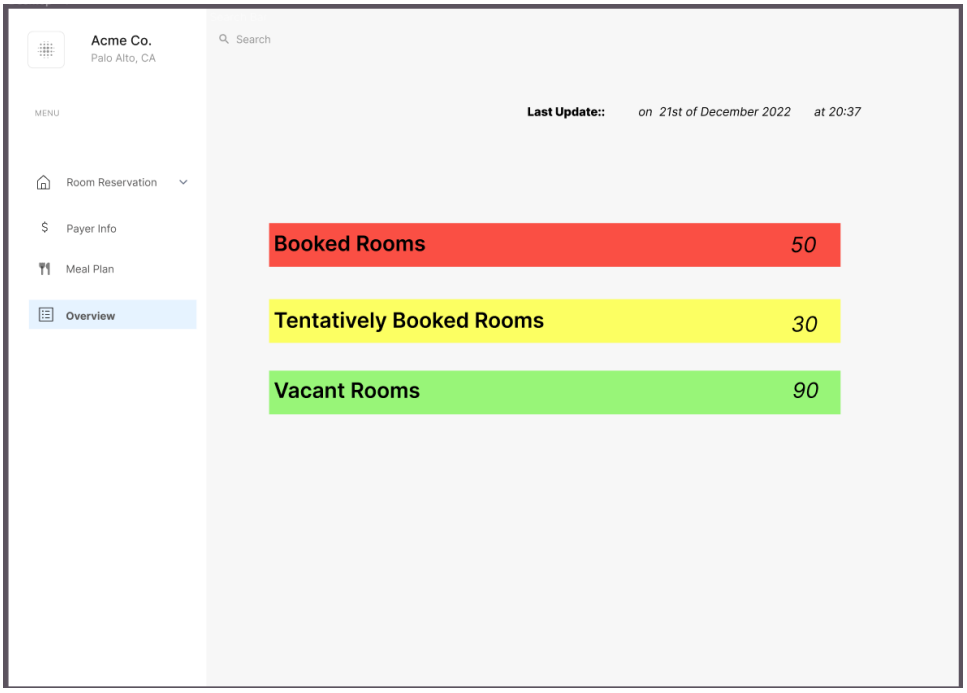


Figure 9.13: Front Office Back Ops 6

Name of student: 204097K Kaushalya D.M.G
House Keeping Portal

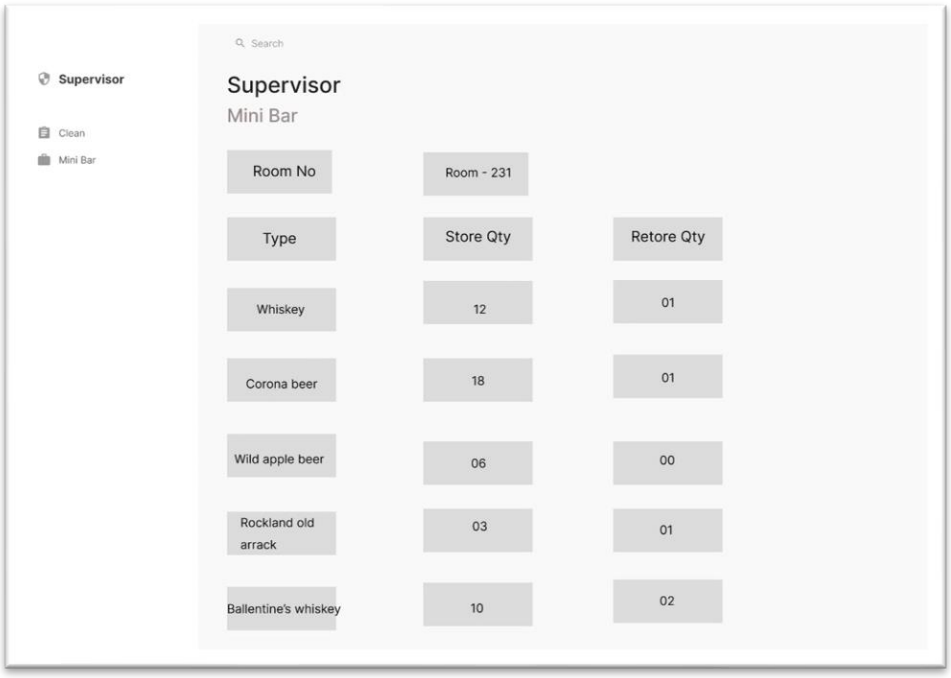


Figure 9.14: Housekeeping 1

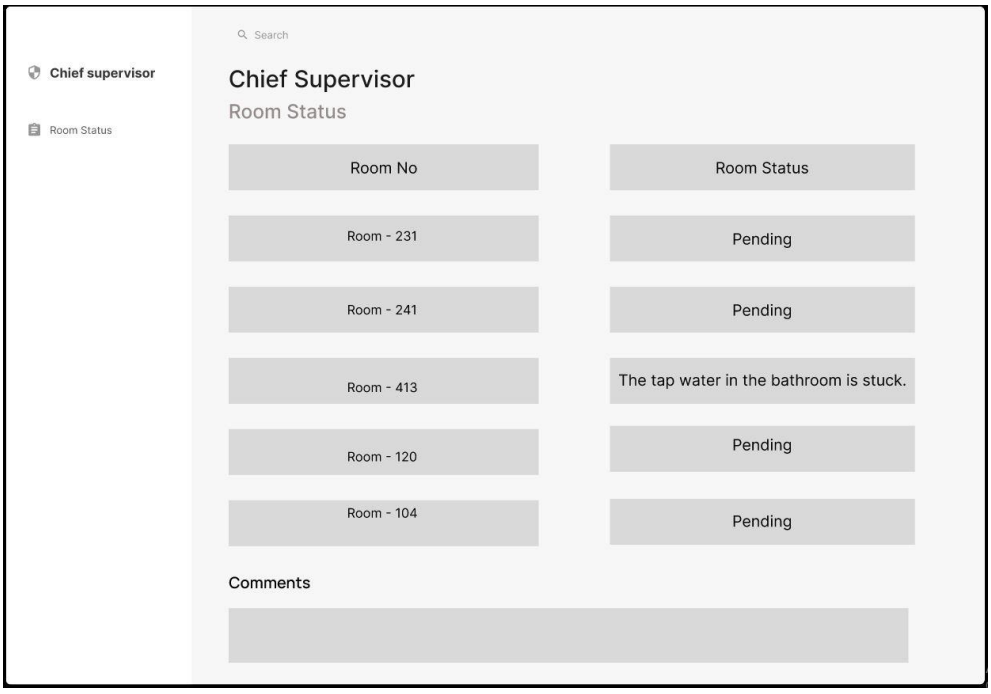


Figure 9.15: Housekeeping 2

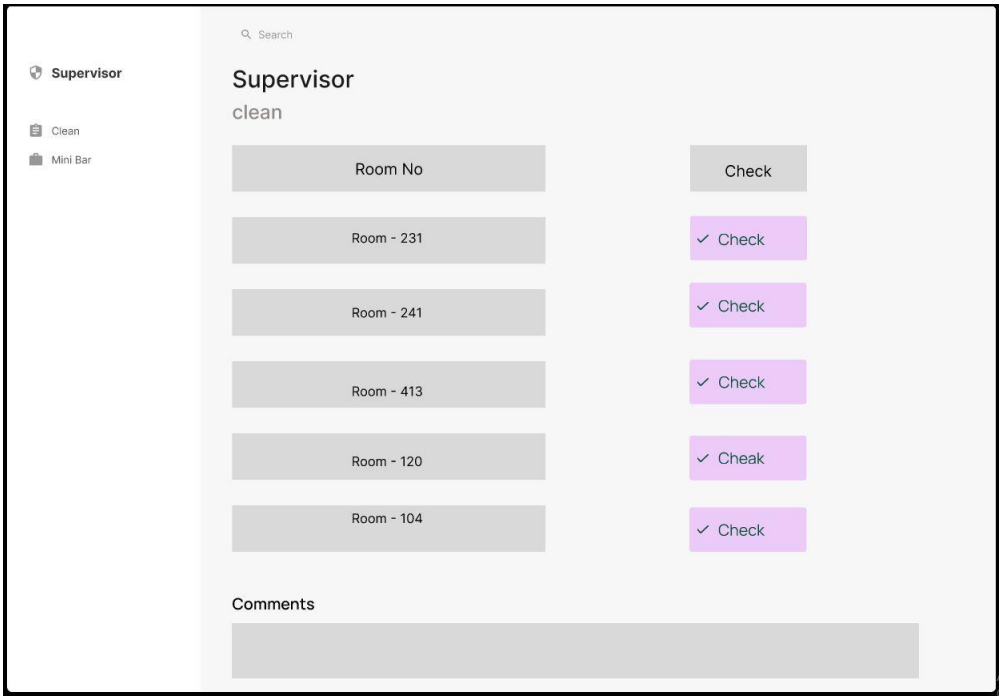


Figure 9.16: Housekeeping 3

Name of student: 204064H Gunasiri G.C.S

Billing and Room Management

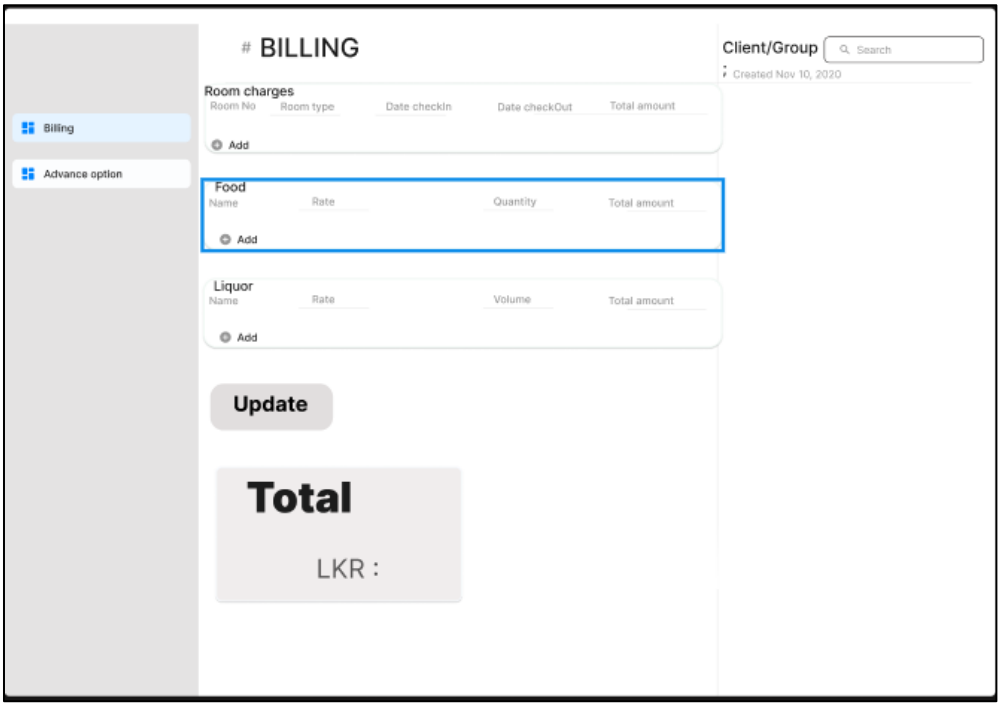


Figure 9.17: Billing 1

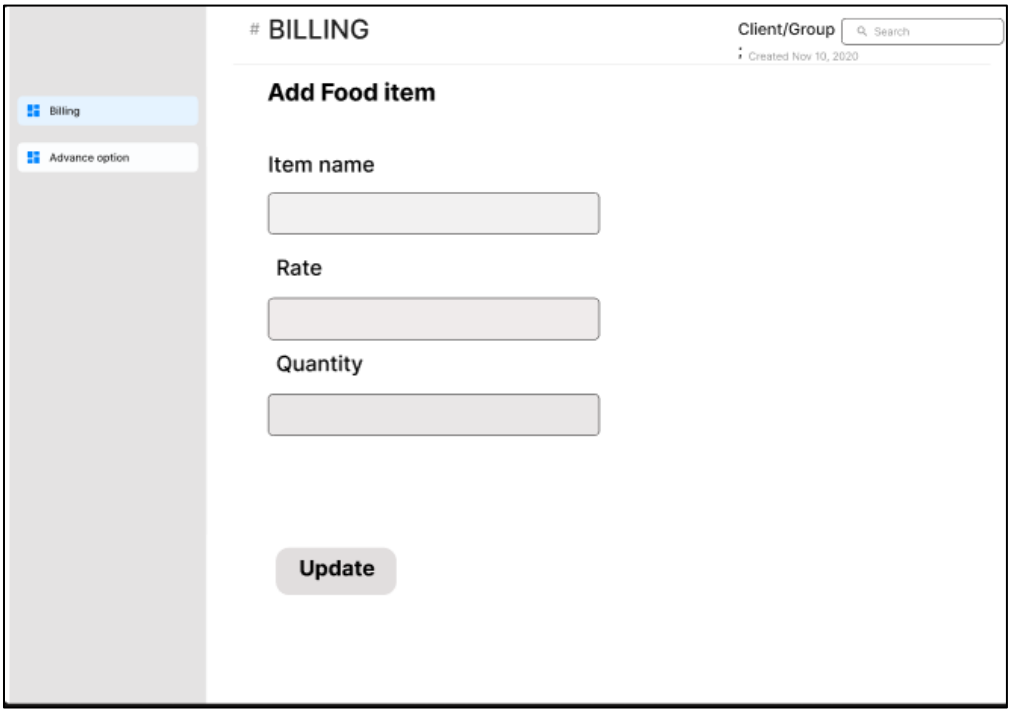


Figure 9.18: Billing 2

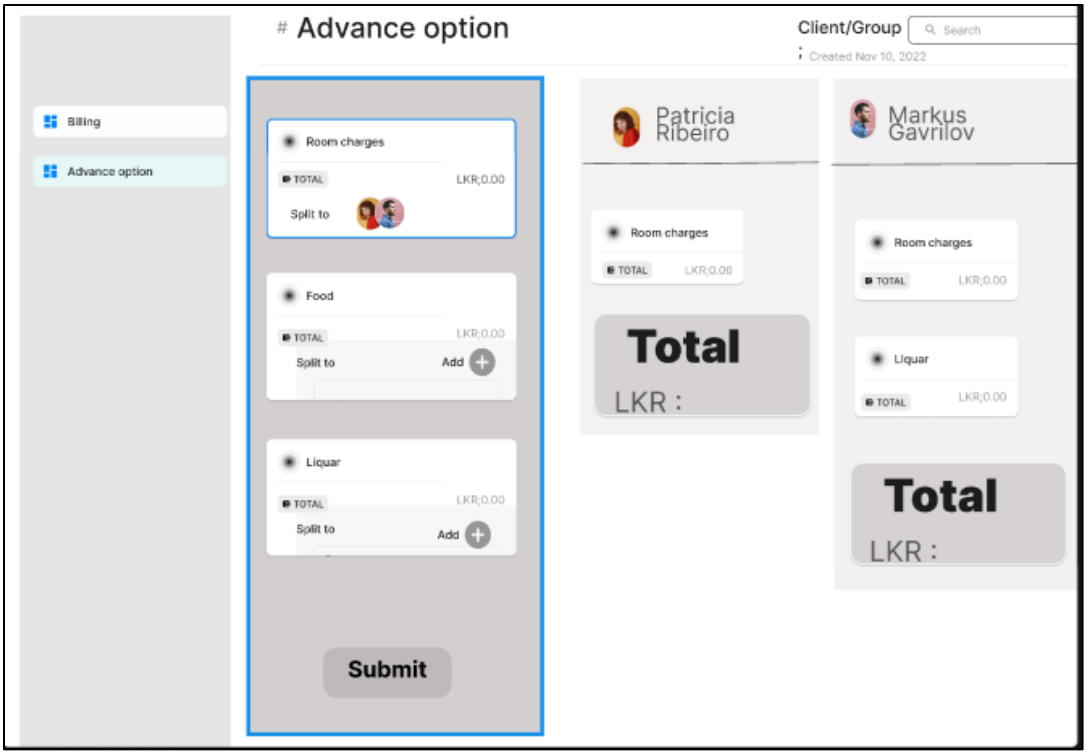


Figure 9.19: Billing 3

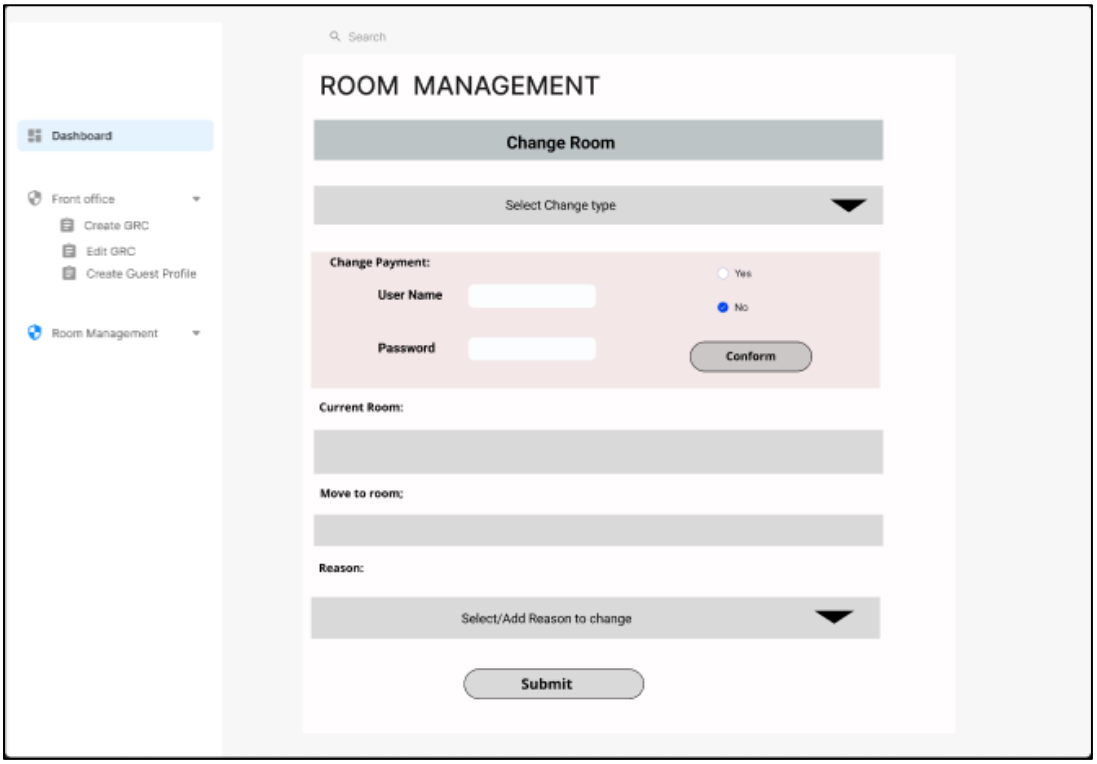


Figure 9.20: Room Management

Name of student: 204152C Pemasiri M.P.T.B.S

Front Office Front Operation Portal

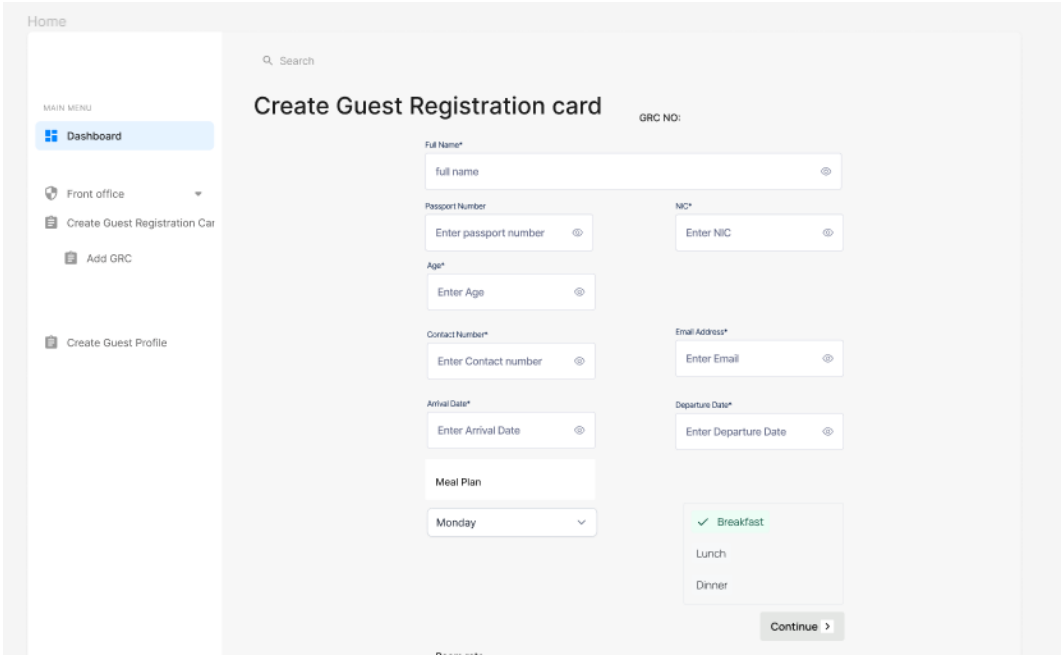


Figure 9.21: Front Office Front Ops 1

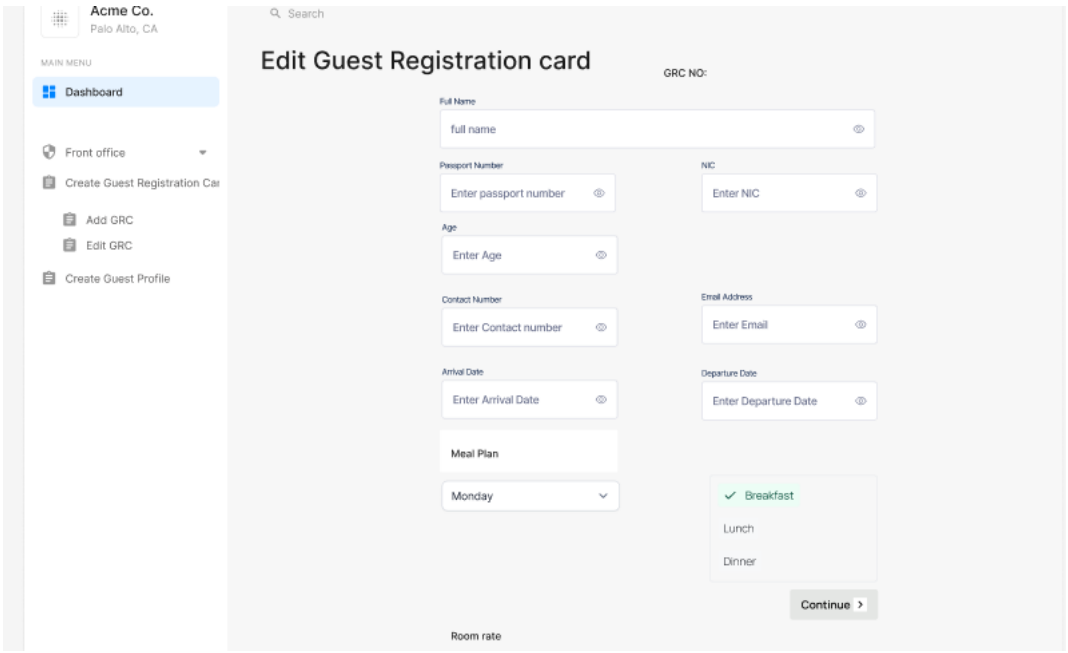


Figure 9.22: Front Office Front Ops

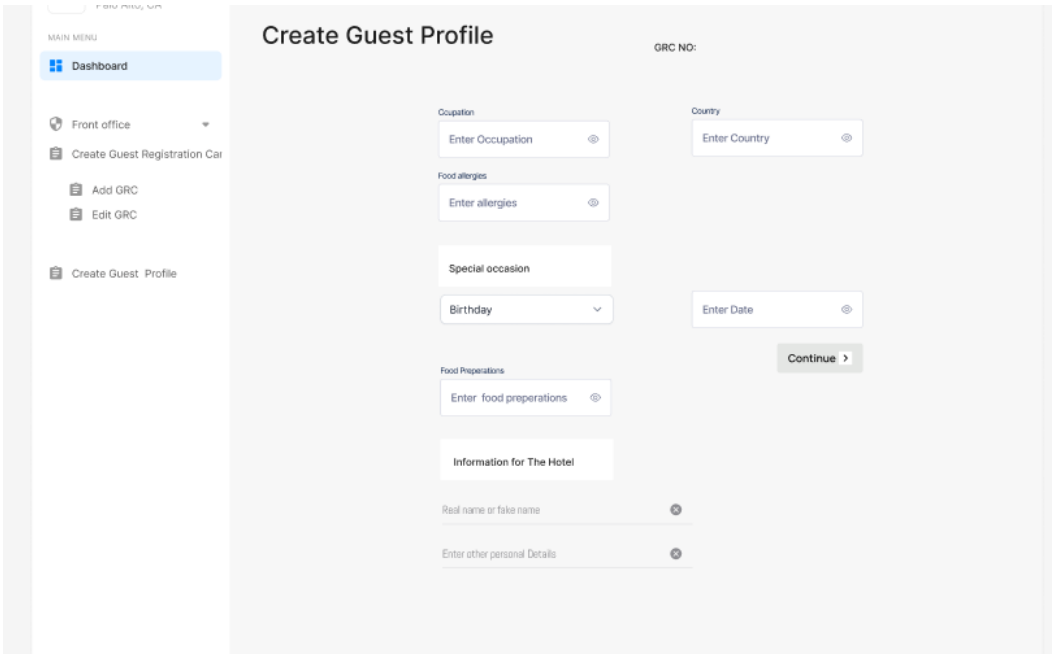


Figure 9.23: Front Office Front Ops 2

Appendix D: Software Requirements Specifications

Software Requirement Specification document is attached separately.