Sri Lanka Institute of Information Technology

# Hotel Management System for the Royal Holiday Resort

Project Proposal

Information Technology Project (IT2080)
2020

KDY_2020_WD09

Submitted by:


1. IT19176802– S.S.Eeswar
2. IT19244044 – J.S.Samaratunga
3. IT19114590 – G.Nivethika
4. IT19207728 – Y.G.S.M.Gunathilaka
5. IT19207896 – D.T.Baddegama
6. IT19155494 – S.H.Wijethunga
7. IT19119090 – R.R.Sachinthani


October 22, 2020

Submitted to:

Ms.Sharmila

……..
(Supervisor's signature)

# Declaration

We declare that this project report or part of it was not a copy of a document done by any organization, university any other institute or a previous student project group at SLIIT and was not copied from internet or other sources.

Project Details

| | |
|---|---|
| Project Title | Royal Resort Hotel Management System |
| Project ID | KDY_2020_WD09 |

Group Members

| Registration Number | Name | Signature |
|---|---|---|
| IT19176802 | S.S.Eeswar | |
| IT19244044 | J.S.Samaratunga | |
| IT19114590 | G.Nivethika | |
| IT19207728 | Y.G.S.M.Gunathilaka | |
| IT19207896 | D.T.Baddegama | |
| IT19155494 | S.H.Wijethunga | |
| IT19119090 | R.R.Sachinthani | |

# Abstract

Royal Resort Hotel Management System was developed following essential system development stages in order to meet our client specifications. While gathering requirements we found that Royal resort is currently using a manual file-based system for their processes as their previously used system failed due to various reasons such as lack of user-friendliness and poor interaction between functions. After collecting client requirements and information gathering process on several hotel management systems, we decided to cover the essential functionalities by converting all the manual processes within this hotel into an automated process increasing the hotel's productivity. We have divided our system into several departments based on system users and each group member developed their own scope while maintaining a quality connection with other functions.

The newly introduced system will provide secure access for verified users and it contains user friendly interfaces with essential functions. Our system will solve the problem of handling a large scale of physical files and the errors occurred in manual calculations while keeping up with the requirements specified by the client. The system is well designed and tested to guarantee maximal efficiency of the tasks done by Royal Resort Hotel management and staff.

# Acknowledgement

The success and result of this project required a strong supervision and support from many people and we are extremely fortunate to have got this all along the completion of our project. The success we have achieved is only due to such valuable guidance and assistance and we would not forget to thank them.

First and foremost, we would like to thank our client Mr. Upul Nishantha the Managing Director of The Peradeniya Royal Rest for providing us the golden opportunity to initiate our project and providing us with the required information.

We would like to earnestly acknowledge Ms. S. Sivabalan for providing us all the support and guidance which made us complete the assignment on time, we are extremely grateful to her for providing such a strong support and guidance and being available all the time to clear our doubts.

We also would like to extend our gratitude to Mrs. Chathurika Koswatte for providing us such valuable guidance and support throughout the assignment and a special thanks goes to all the lecturers at Sri Lanka Institute of Information Technology for their sincere efforts and sharing wisdom with us to achieve success.

Last but not least, we would like to express gratitude towards our parents who helped us a lot by providing their fullest support in finalizing this project within the limited time we were given.

# Table of Contents

# List of Figures

# List of Tables

# 1. Introduction

## 1.1 Background of the Company

The Peradeniya Rest House is a 200-year-old colonial type bungalow that was once Captain Dawson's home. The hotel is 600 meters from the Royal Botanic Garden in Kandy. today, the place is popular with locals and tourists alike.

The specialty of this place is that it located in the center of Kandy and visitors can easily walk around the city.

The hotel has luxury accommodation and the working staff very friendly to their customers.

## 1.2 Problem Statement

- All bookings are done by manually.
- Data redundancy
- Accounting errors occur when calculating profit and loss at the end of the month/year.
- Large cost due to file-based system.
- Data loss, unauthorized viewing makes it difficult to properly and securely mange data, and also can't collect the data at the time.
- (Update, Search, Delete, Edit) these methods are inaccessible and do not change with the file-based system.
- Past data cannot be analyzed.

## 1.3 Solution

As the company has performed its operations manually throughout the past few decades, the newly proposed system will be able to reduce the rate of maintaining inaccurate data and the redundancy of data. Also, it will increase the efficiency of the employees with their day to day performances whereas the company will be able to manage and monitor the employees more efficiently.

## 1.4 Benefits

- Reduce time spent on administrative tasks by generating reports from the system.
- Ability to edit functional data will let the system change according to the modifications in hotel environment.
- Having a better interaction between functions will optimize the overall performance of the hotel.
- Capability to search real time availability and adjusting rate according to the demand will help the hotel to get more revenue of every room and hall.
- System data will provide a better understanding of the guest behavior.
- Having system entries and automated billing will gain customer trust towards the hotel.

## 1.5Product Scope

### 1.5.1 Overall Scope

The system is designed to transform all manual processes inside the organization into an automated process to make the day to day tasks undergone in each department of the organization much more convenient and efficient to use which will reduce the number of fraudulent activities within the hotel.

Figure 1 illustrates the system flow once a valid user logs in to the system. They are capable of operating the functions as they login to the respective section.

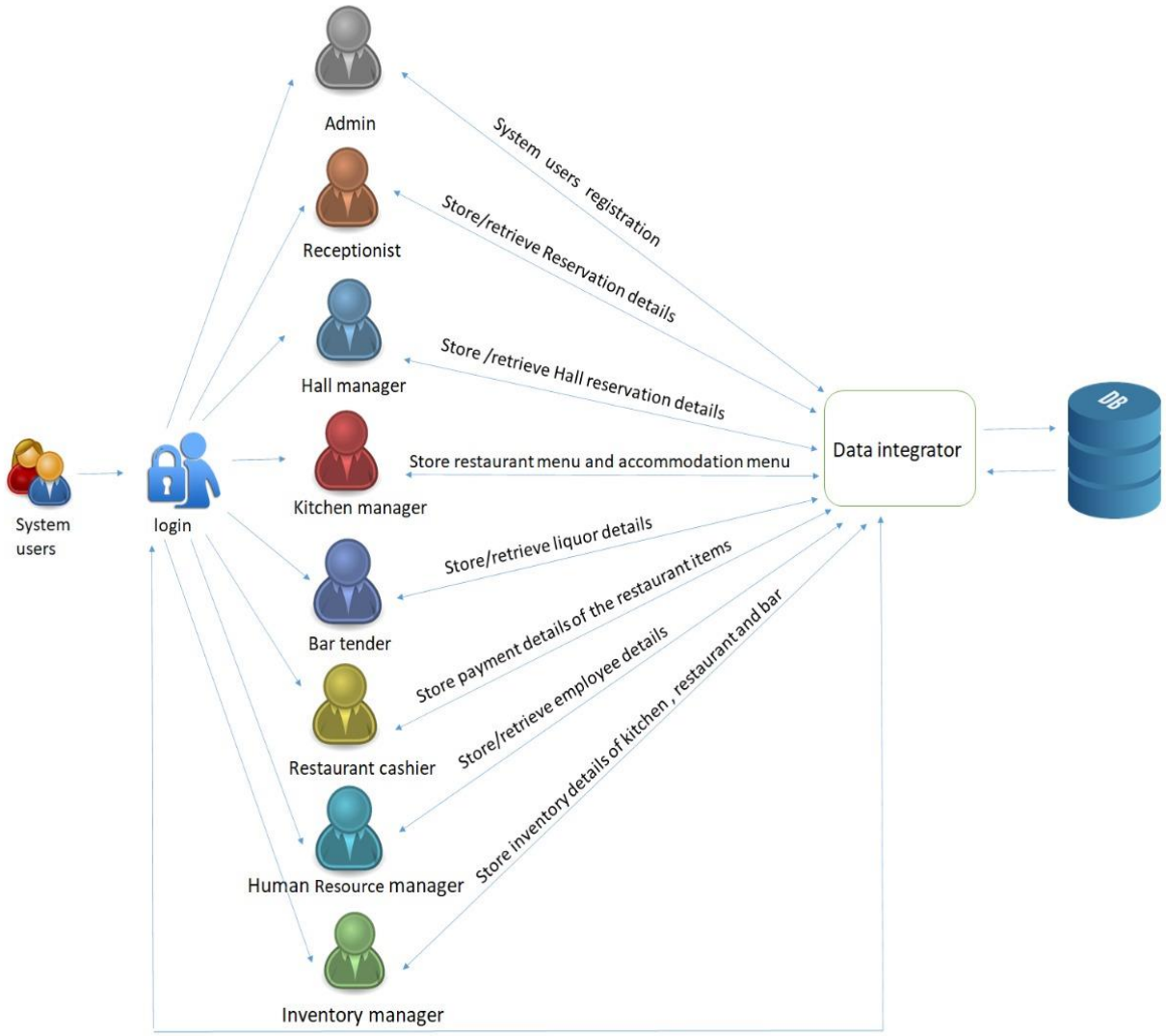


*Figure 1: system overview*

### 1.5.2 Individual Scope

Table 1 illustrates the function scope of each member.

*Table 1 individual functions*

| Name with Initials | Brief Description of the Function |
|---|---|
| S.S.Eeswar | ***Inventory Management***<br>• Adding, updating, deleting the items of the Consumable stock<br>• Adding, updating, deleting the items of the Permanent stock<br>• Calculating the remaining stocks.<br>• Search items by its name |
| J.S.Samaratunga | ***Administration***<br>• Adding and updating the accommodation details<br>• Validating the identity of user.<br>• Generating income report.<br>• Search user by username |
| G.Nivethika | ***Reservation and booking***<br>• Adding, updating, deleting reservation details<br>• Adding, updating, deleting booking details<br>• Calculating the total bill amount<br>• Search rooms by room type |
| Y.G.S.M.Gunathilaka | ***Bar cashier***<br>• Adding, updating, deleting the liquor details |

| | |
|---|---|
| | • Adding, updating, deleting menus<br>• Calculating bills.<br>• Search the price by liquor name |
| D.T.Baddegama | ***Hall reservation***<br>• Adding, updating, deleting event details<br>• Adding, updating, deleting event menus<br>• Calculating the bill amount for an event<br>• Generating the report<br>• Search an event by date |
| S.H.Wijethunga | ***Kitchen and Restaurant***<br>• Adding, updating, deleting the Restaurant menu<br>• Adding, updating, deleting the accommodation menu<br>• Calculating the Restaurant bill<br>• Search meals by meals id |
| R.R.Sachinthani | ***Employee management***<br>• Adding, updating, deleting the employee details<br>• Calculating the salary<br>• Generating the employee monthly salary report<br>• Search salary by employee id |

## 1.6 Project Report Structure

The rest of the report is created in order to expose the software development lifecycle of the Hotel management system developed for “Royal Holiday Resort”. Initially, the identified requirements of the system are illustrated in the form of a use case diagram. The requirements of each function are further explained along with an activity diagram per each function.

Secondly, it’s explained the design of the system. The database design is represented using an ER diagram while the manipulation of object-oriented concepts is demonstrated using a class diagram. Then, sequence diagrams and communication diagrams are included to depict the design of each function further. Also interface designs are attached to present the user friendliness of each function.

To manifest the implementation phase, the choice of DBMS, implementation language, platforms and tools are thoroughly explained. In addition, specially developed codes and algorithms are included to summarize logical operations.

The final phase of testing is denoted by displaying one test case per each function. These test cases are confirmed by attaching screen captures which were taken while driving the test cases.

The latter part of the report consists of the conclusion, references and appendices. The deployment diagram for the system, more test cases and special algorithms are attached to appendices for further reference.

# 2. Methodology

## 2.1 Requirements and Analysis

As the first step in the SDLC we decided to gather the required information, we needed to design our system.
We visited our client and we were able to interact with many of the employees working at the hotel and gather the required information.

To gather the information, we used 2 requirements gathering methods,

- Interviews
- Observations

Since it was initially decided to develop the system according to departments, we were given the opportunity to interview the heads of the departments and get a thorough understanding about the flow of each department.

After the interviews we decided to gather information by observation, we were given permission to observe the work flow of the company and get a solid understanding about how the company works and how each department is important for a successful work flow.

After a few brainstorming sessions with our team members we were able to eliminate all the unnecessary requirements and clear all the doubts we had about the unclear requirements.
And then we moved into the documenting phase of requirements gathering.

Figure 2 Illustrates the functions of different actors of the system using a use case diagram

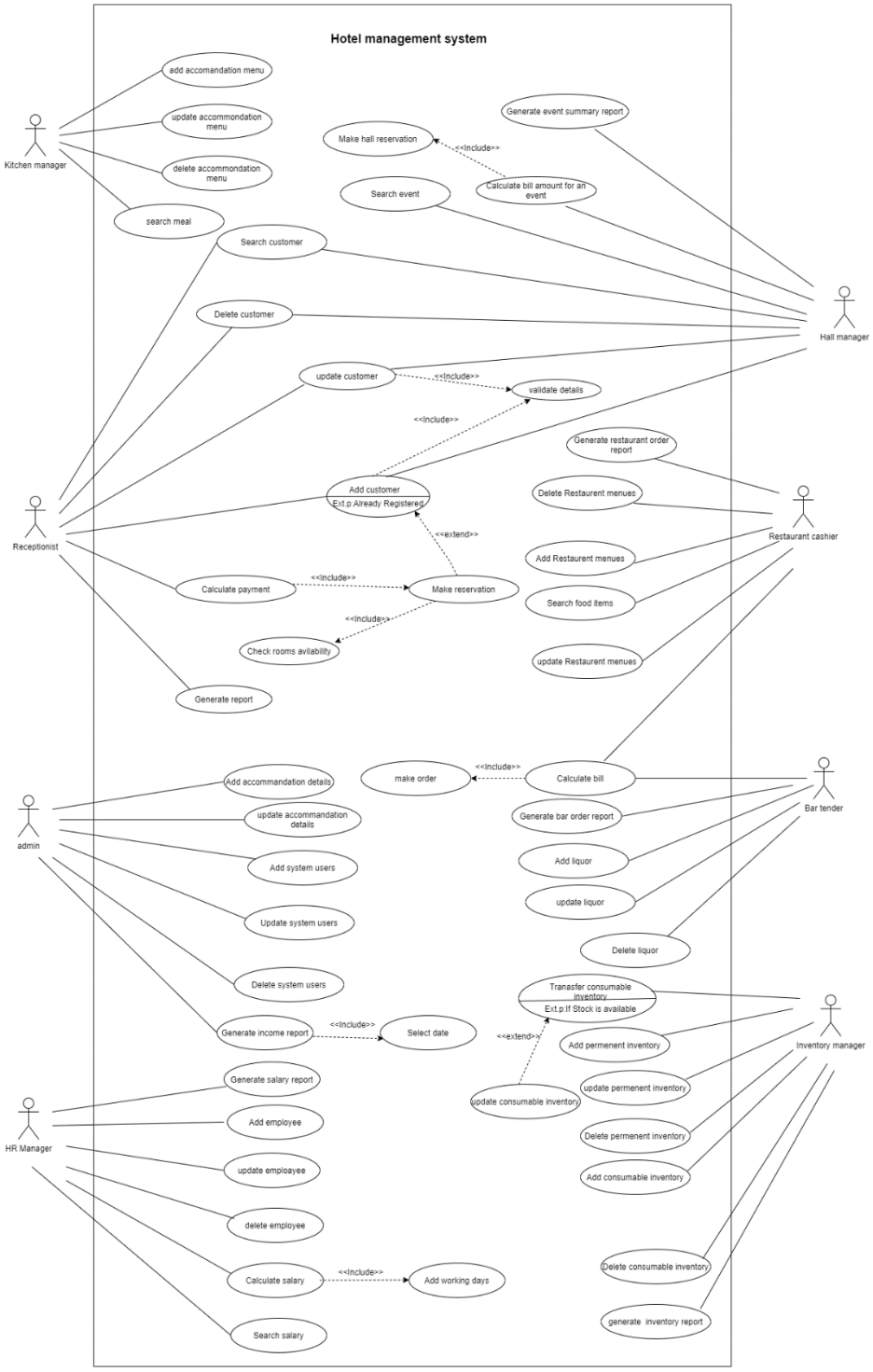


*Figure 2:use case diagram*

### 2.1.1 Administration

System functionalities and performance will be overviewed under the administration function. The general manager of the hotel is the one who operates this function. There are few main parts in this function, as validating system users, managing hotel facility details and generating reports. Under user validation, separate login access is provided for each user type of the system and the general manager is responsible for adding, updating and deleting user details from the system database. Managing hotel facility details is done by adding, updating facilities such as room details and hall details. An income report is generated for the given time period in this function by calculating total income of all the departments. Also, administration function has the capability to search data saved in the system database such as searching user details by given username.

Figure 4 Illustrates an activity diagram to depict report generate function of admin.

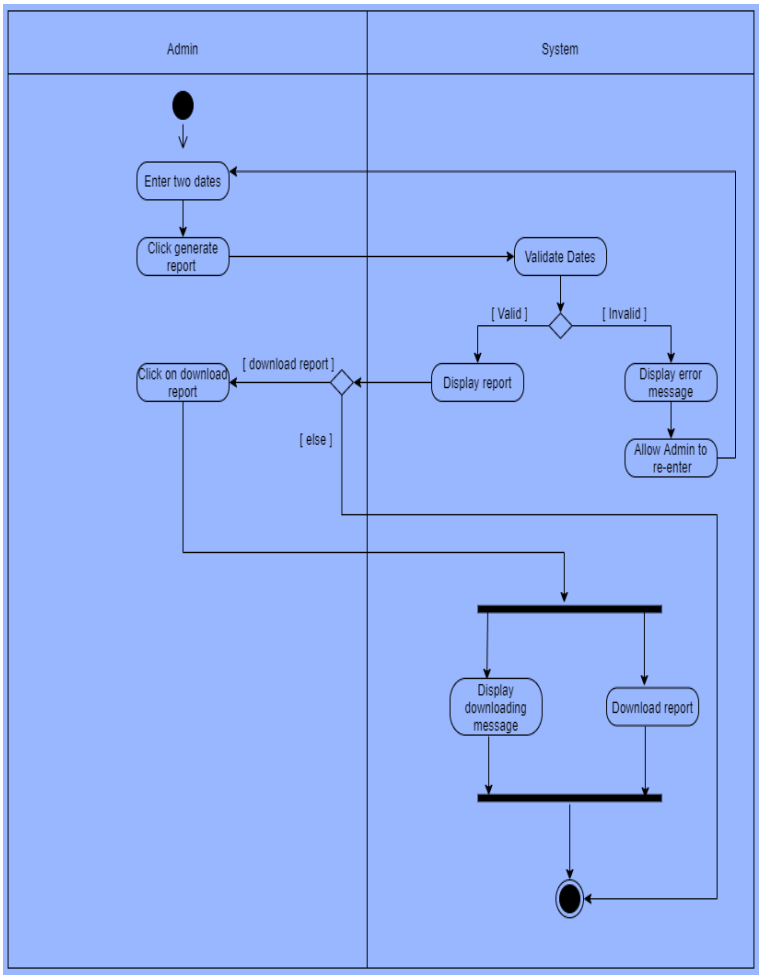


*Figure 3:report generate function of admin*

### 2.1.2 Inventory Management

Under this function inventories of the hotel will be managed. The inventory manager is the person who is in charge of the inventory. He has divided the inventories into two parts. Permanent inventories details will be added to the relevant departments it brought to the hotel. Item name, number of items available, worth of the item will be recorded. If there are any changes in the item details inventory manager able to update the details. Once the item is destroyed or removed that item will be deleted from the system. At the end of each week, inventory managers record the number of items available items in the kitchen, hall, and restaurant. Consumable items will be added batch-wise each time they arrive at the stock. Quantity, arrival date, expiry date, and the details of the supplier details will be recorded accordingly. On requisitions made by kitchen or restaurant inventory manager check for the availability of the stock. Expired items and items with the unusable condition will be deleted from the stock at the end of the week.

Figure 4 Illustrates an activity diagram to depict the process of transfer consumable inventory from inventory in a form of activity diagram

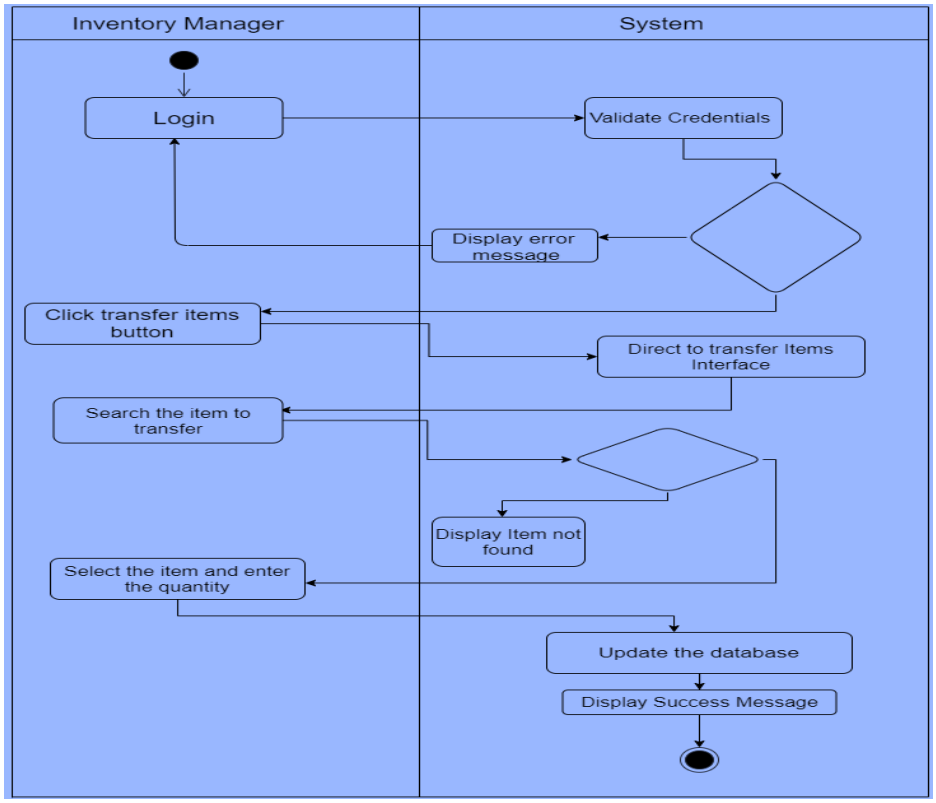


*Figure 4:process of transfer consumable inventory*

### 2.1.3 Room Reservation function

The objective of this function is to handle room reservation related activities easily and reliably using the system. The Room Reservation related activities are handled by the Receptionist. There are few main parts in this function, which are Customer Registration, Booking, calculate payment and Generate Report. Once the customer requests to make a reservation, Receptionist can enter the particular customer's personal details in the system. These details will be stored in the database. And also, the system allows to enter the booking details also. The customer can select his/her preferable meal plan and room type. System will be able to display the available rooms of a particular room type and reduce the workload of the receptionist. So that the Receptionist can allocate a room for the customer with his preferred room type. And also, system automatically displays total price for each booking and the total payment of the bookings for a particular customer. The system can print bill for the customer too. So that the payment process will be easier to handle by the receptionist.

In generating report, the system gives options to select current date or two dates. So, the system will generate total meal price, total room rent and total income for the given option along with all the booking details for the given option. And the Receptionist has the facility to print these details also.

Figure 5 Illustrate Room Reservation Function in the form of activity diagram.

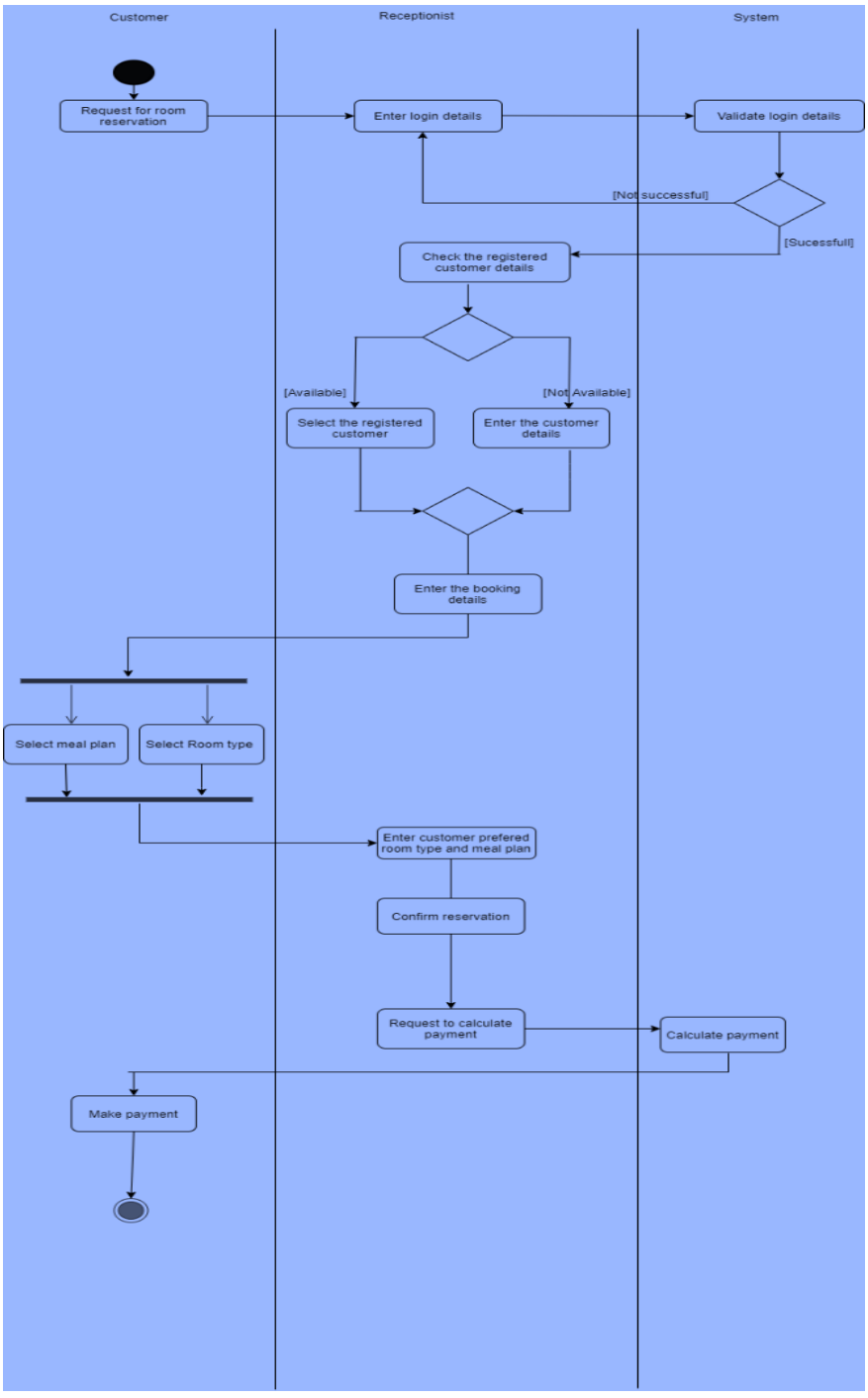


*Figure 5:Room Reservation Function*

### 2.1.4 Employee Management

The purpose of this function is to allow the system to manage all the records of the employees in the company. Employee management function allows the HR Manager to keep track of the employee's personal details, salary details, designation details as well as the reports. The HR Manager is able to add employees, update employee details, delete employee details from the system. Also, the HR Manger is able to calculate the salary of each employee at the end of each month. The HR Manger is also able to generate reports too.

Figure 6 Illustrates Employee Management in the form of an activity diagram.

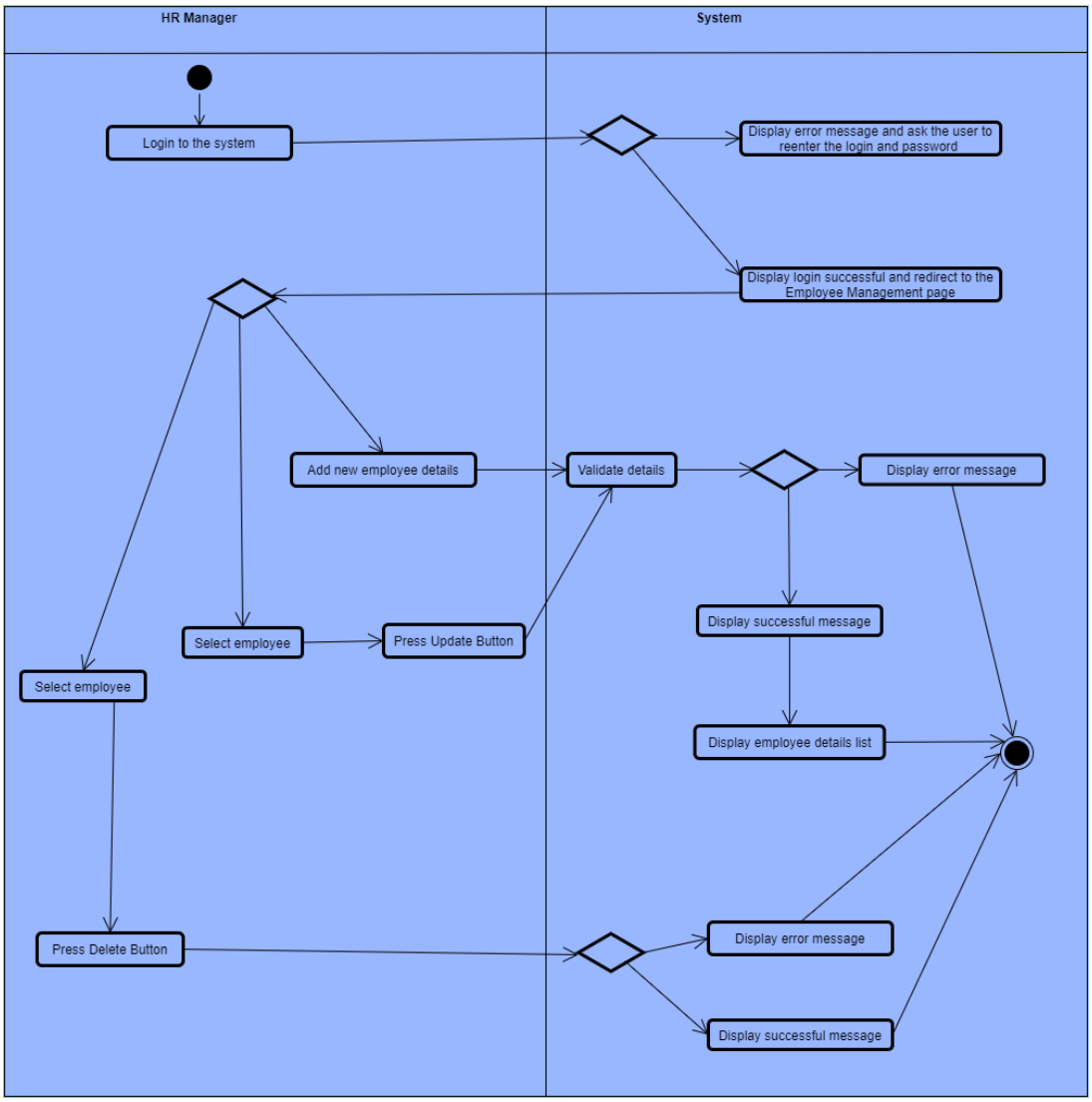


*Figure 6:Employee Management function*

### 2.1.5 Kitchen and Restaurant

The purpose of this function is to manage food items in the restaurant and manage meal plans for accommodation. Here the restaurant manager is capable of adding different food items and the kitchen manager is capable of adding meal plans. Both can update the added details and delete an item/ a meal plan too. Also, the users can search for a particular food item/ meal plan by giving its name. Most Importantly, The Bill calculation for the restaurant cashier is implemented under this function. When user complete an order, a preview of a receipt will be displayed on the screen so he can check it and print the receipt. In this function a report is generated for check the number of items and their details of particular order. Restaurant cashier can generate this report by giving the order id.

Figure 7 Illustrates An activity diagram to depict bill calculation function of restaurant cashier

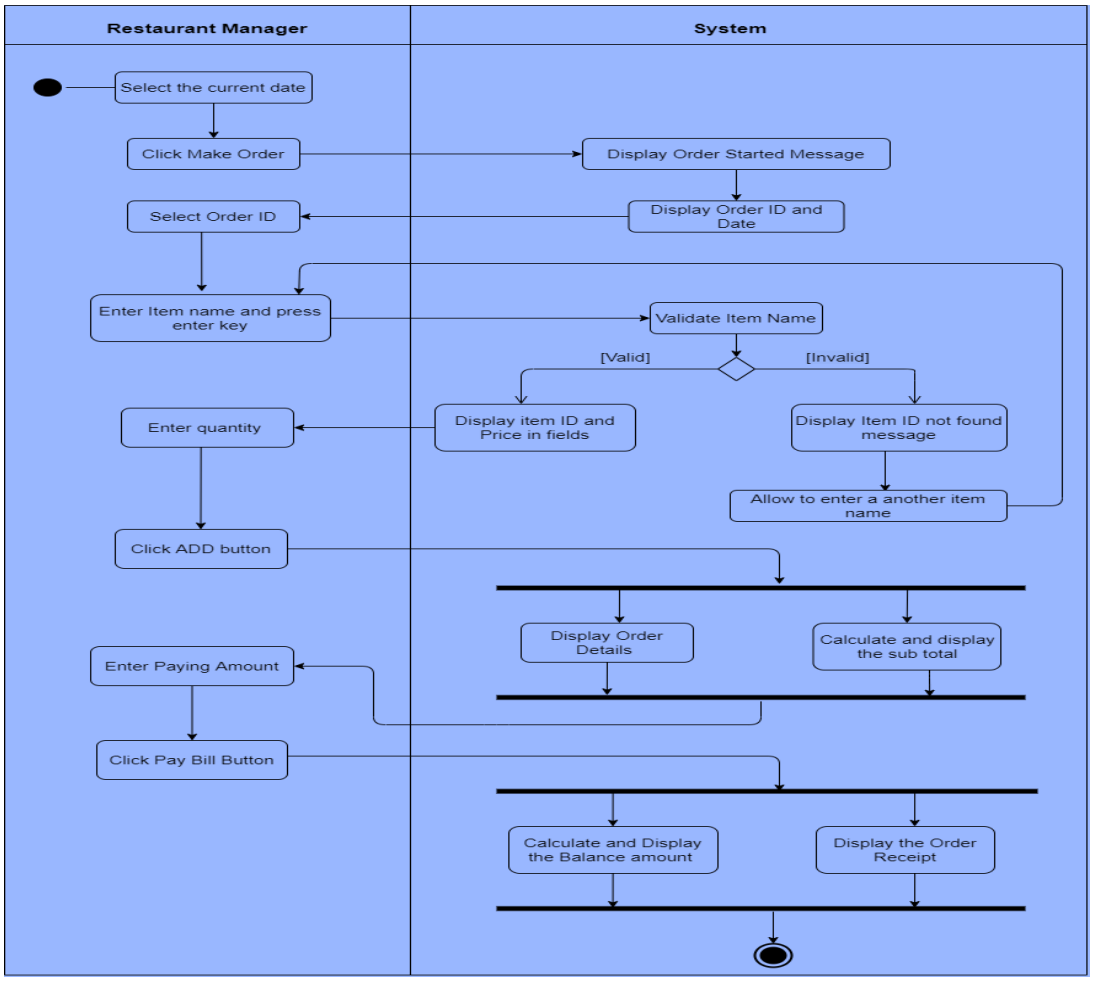


*Figure 7:restaurant bill calculation*

### 2.1.6 Bar Cashier

The objective of this function is to handle bar related activities easily and reliably using the system. This bar related actives are handled by bar cashier. The system allows the bar cashier to store and retrieve details in liquor and snacks table. The liquor table contain fields such as Item id, item no, item type (Ex: -Whisky, Vodka), item name, Bottle size (in liters), Manufacture date, Expired date, Batch no, strength v/v and price. The snack table contain snack id, snack name, description and price. The system will also help the bar cashier to maintain the sales table. The sales table contain item no, item name, description, unit price, sold quantity and total sales. So, the bar cashier needs to enter the item name and sold quantity only. So, the item details will be retrieved from database and the total sales will be calculated by the system automatically. So, it will reduce the workload of the bar cashier. And also, bar cashier will be able to calculate the payment using the system. The payment table contain item id, item no, item type, Quantity, unit price, total price of a particular item and total payment. Bar cashier needs to enter the item name only. System will retrieve all the details relevant to the item mentioned from liquor table and snack table and calculates payment by making the work easier for bar cashier.

Figure 8 Illustrates An activity diagram to depict Order and Billing function of Bar management

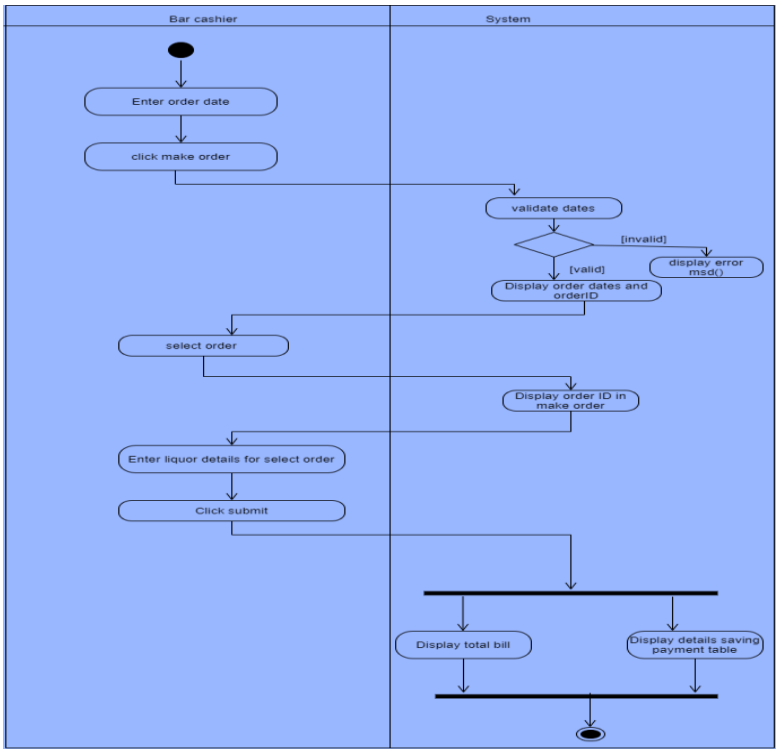


*Figure 8:Order and Billing of Bar management*

### 2.1.7 Hall Reservation

The King, Queen and Princess halls will be managed in this function. The customer must register with the system if he / she wants to make a reservation. Once the event details are attained from the customer, they will be added to the system by creating a new booking. If a customer needs to change the entered details, this software allows user to update or delete it. Customer can customize their food package according to the menu provided by the hotel management. In this system the user can add selected menu items to the menu details form and print it out and give it to the customer as well as the kitchen. After the above process, according to the package details, number of people, hall rent and service charge and the total amount will be calculated. Finally, the user can take a summary of the event details as a report.

Figure 9 Illustrates event management in the form of an activity diagram.

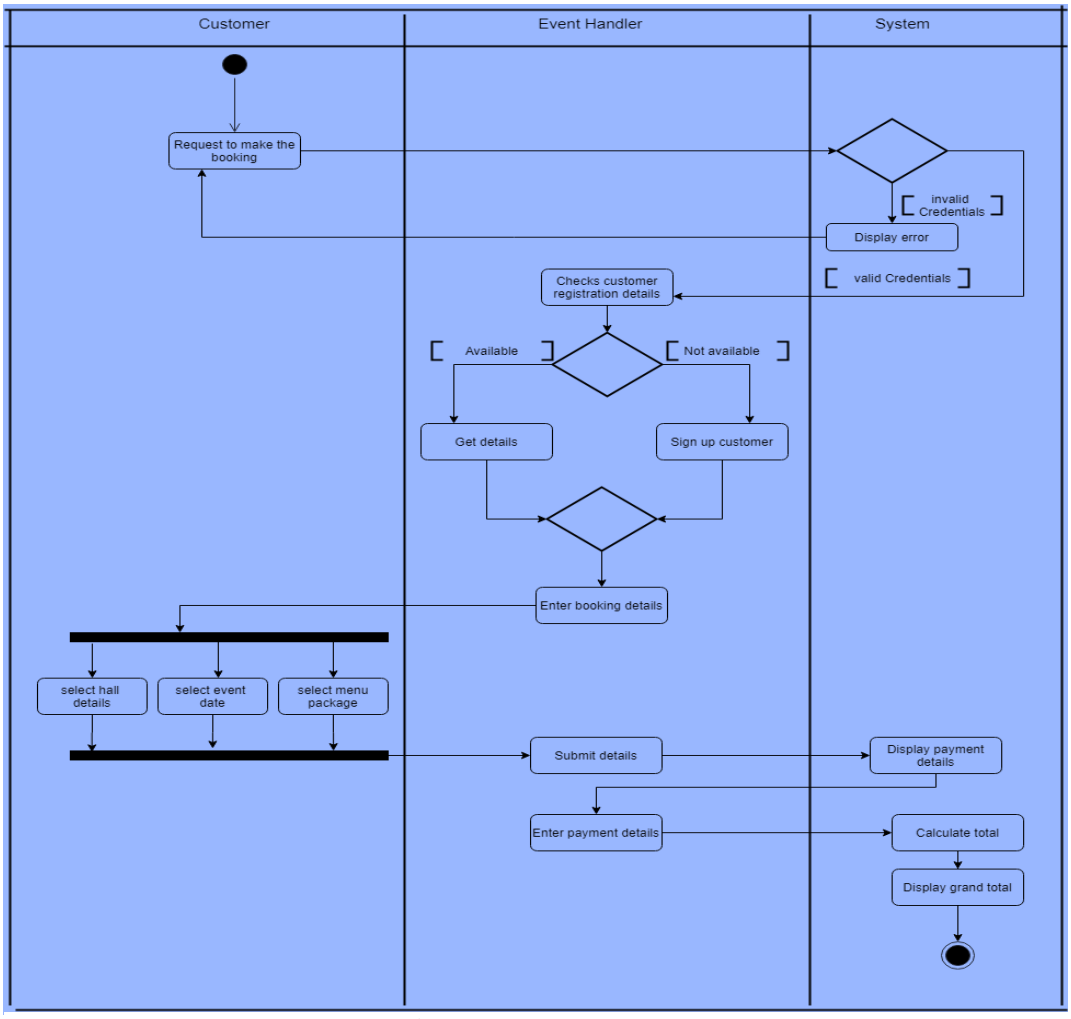


*Figure 9:event management*

## 2.2 Design

### 2.2.1 Class Diagram

Figure 10 Illustrate the complete Class diagram of the system

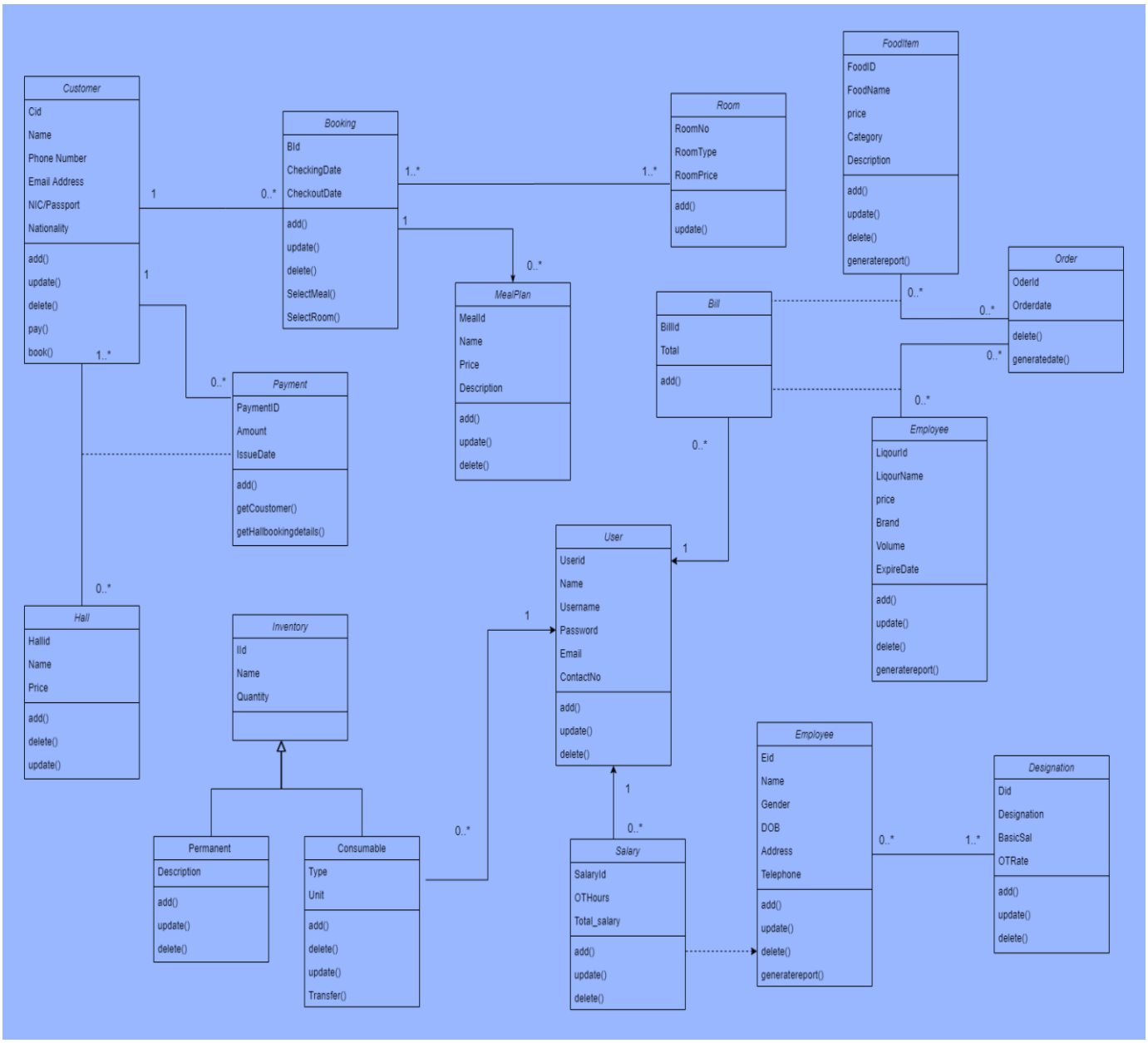


*Figure 10:Class diagram of the system*

### 2.2.2 Er Diagram

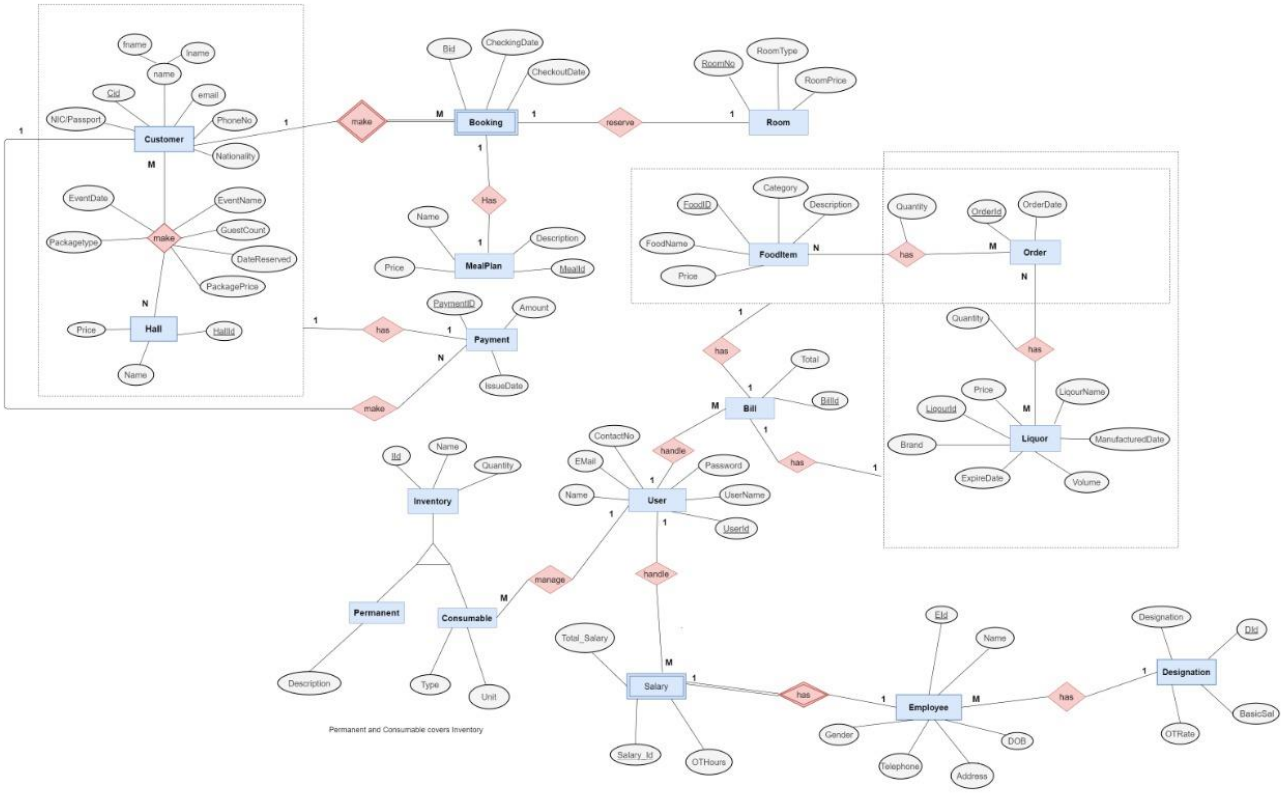


*Figure 11:Er diagram*

### 2.2.3 Sequence Diagrams

Figure 12 Illustrates the ref tag used in the sequence diagrams for user login.

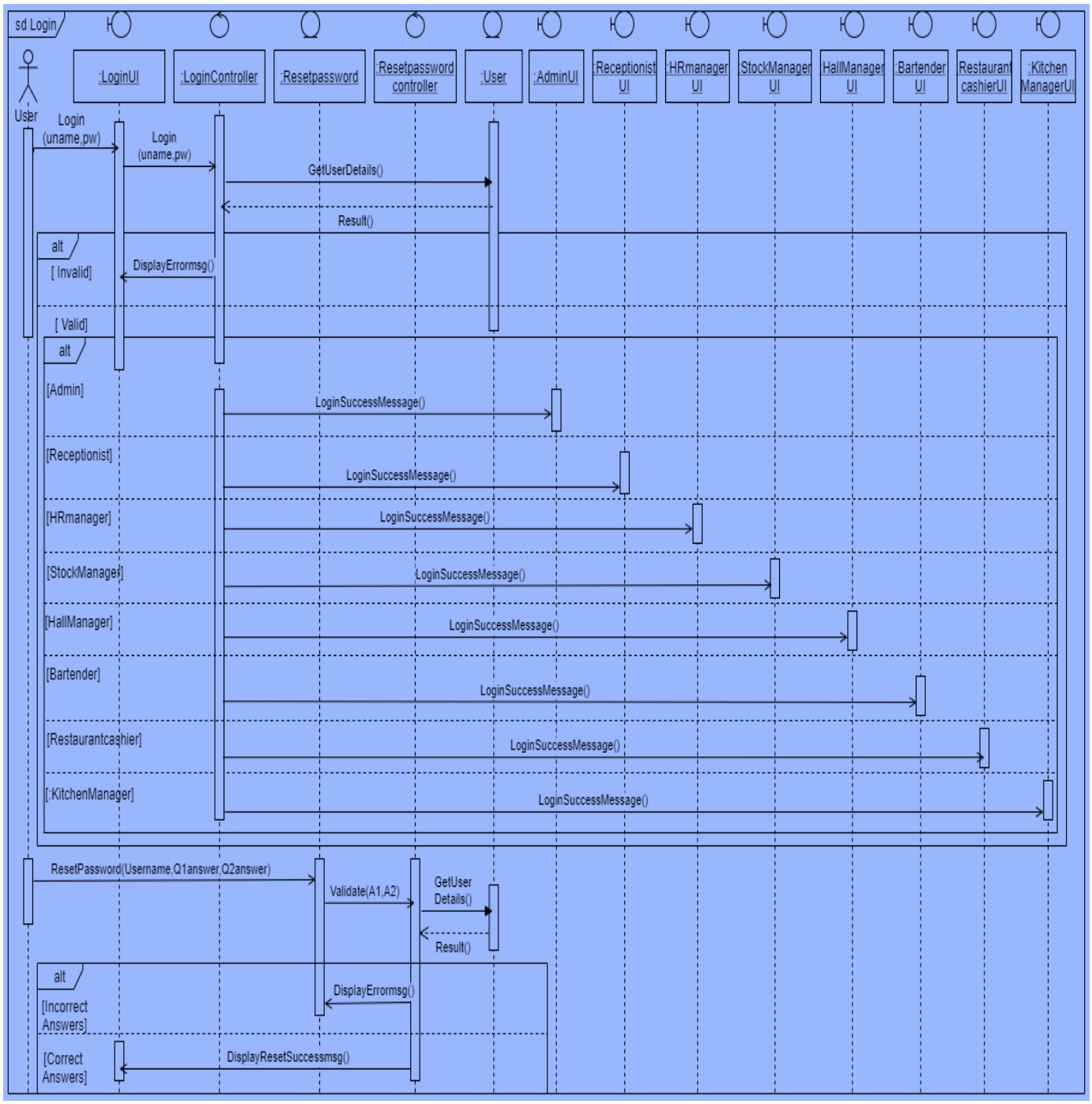


*Figure 12:login management*

Figure 13 Illustrates the process of searching, adding, updating and deletion of the users in a form of a sequence diagram.

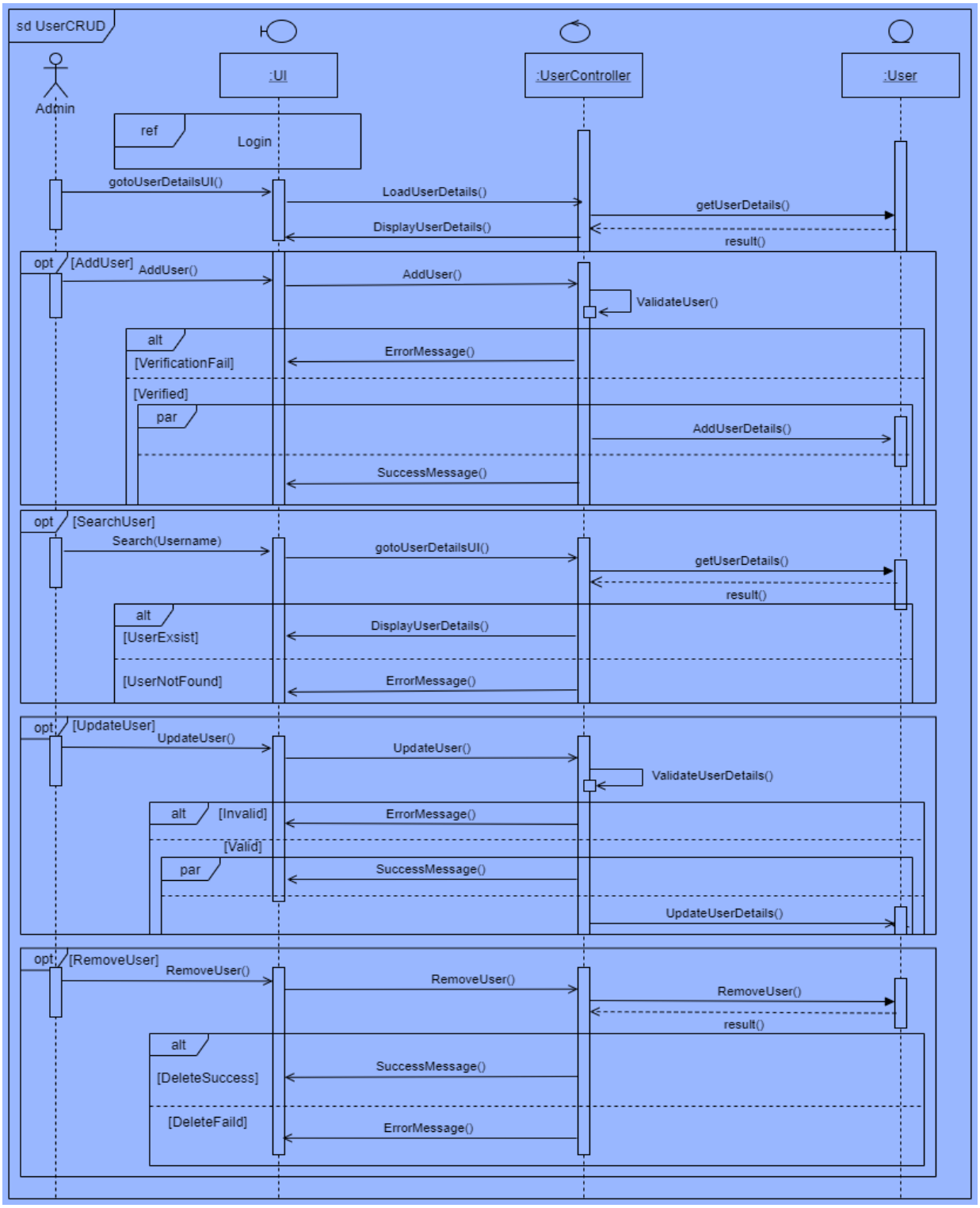


*Figure 13:user management*

Figure 14 Illustrates Room Reservation function in the form of a sequence diagram. It includes Adding, updating and deleting customer details, calculate payment, Adding, updating and deleting Booking details, make payment.

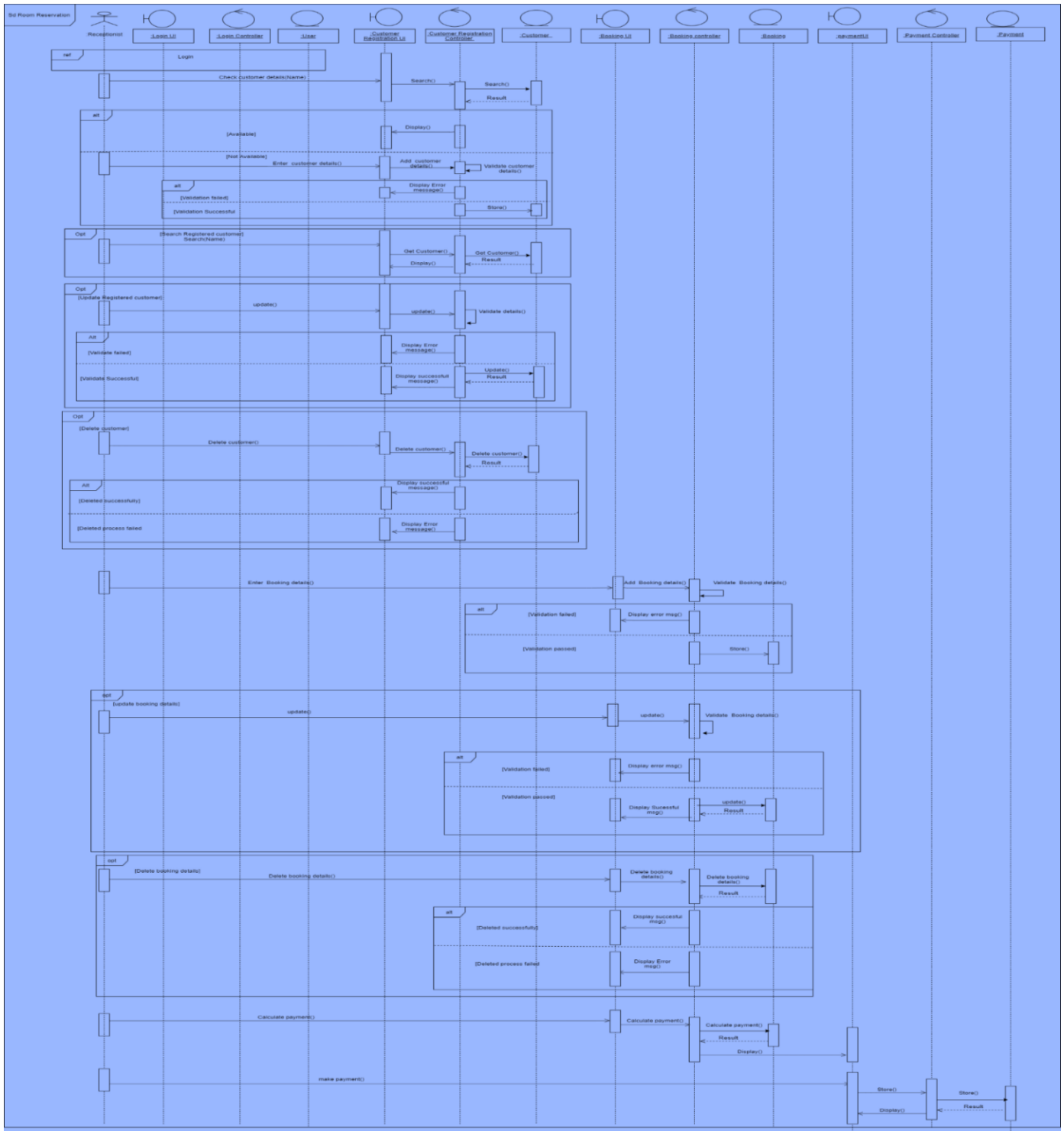


*Figure 14:Room Reservation*

Figure 15 Illustrates Sequence Diagram of Hall Booking function which shows the Add, Update and Delete operations of Event Handler

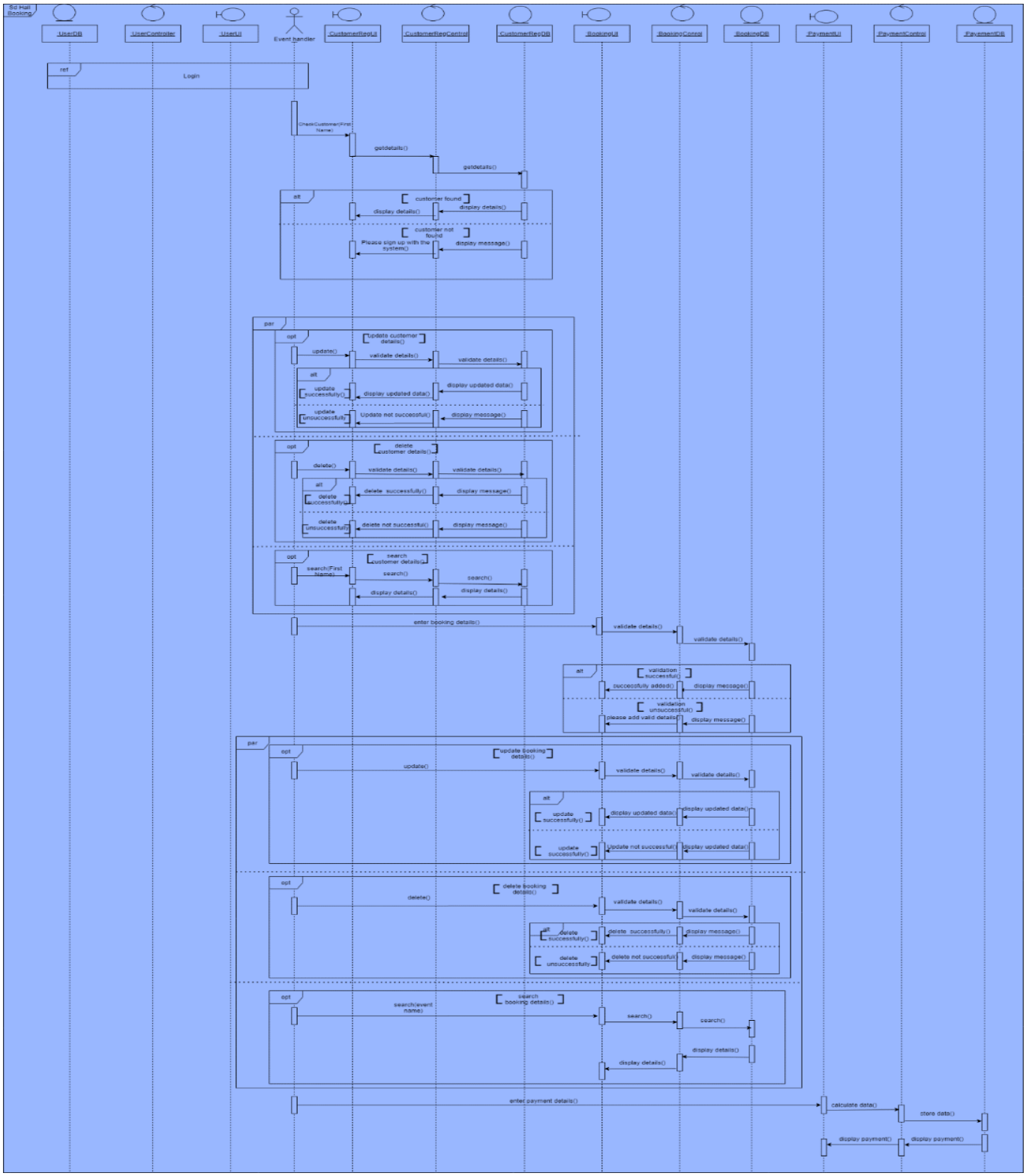


*Figure 15:Hall Booking*

Figure 16 Illustrates the process of adding, updating and deleting employees in the form of a sequence diagram

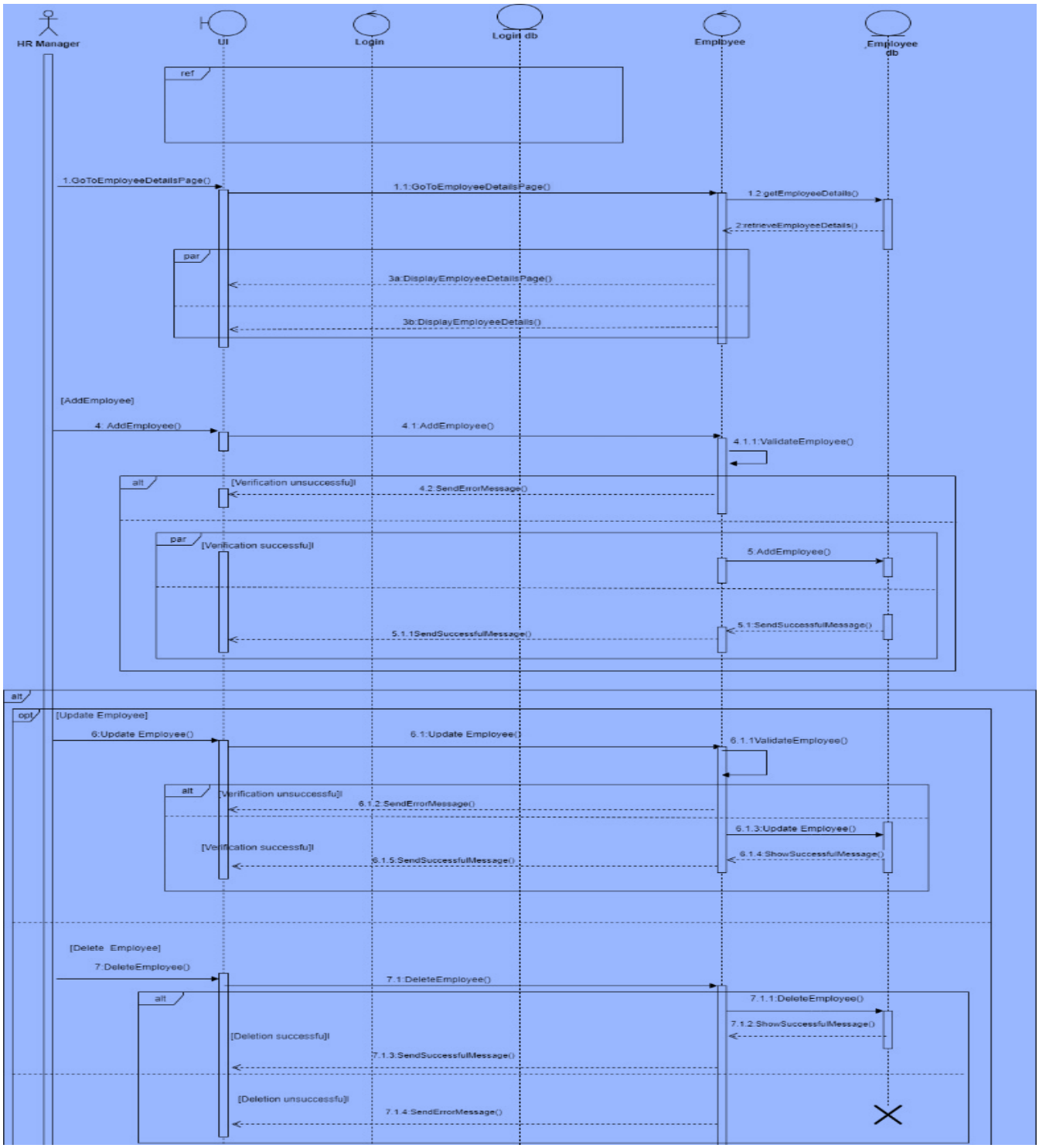


*Figure 16:employee management*

Figure 17 Illustrates the process of adding, updating and deletion of the liquor details in a form of a sequence diagram.

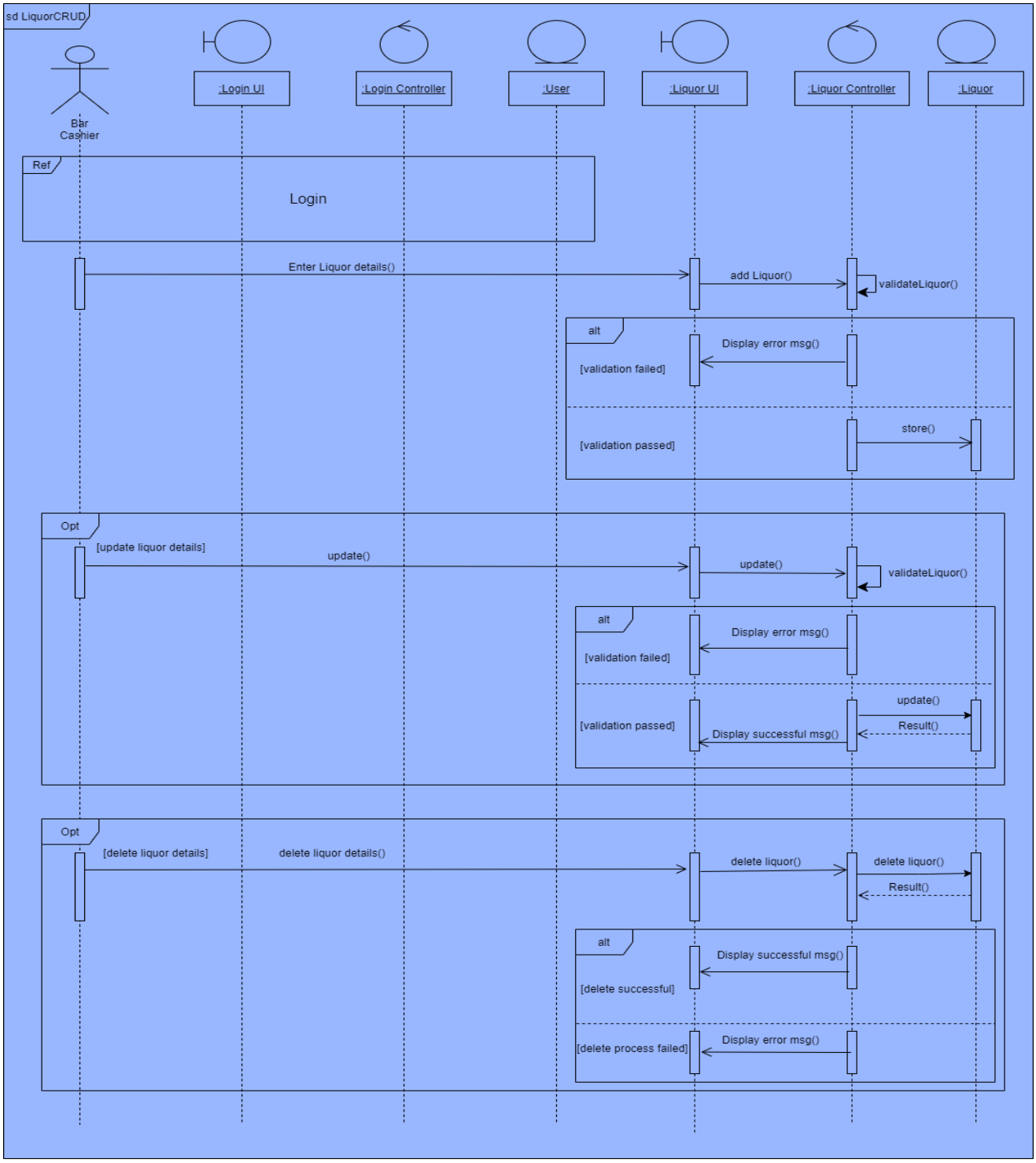


*Figure 17:liquor management*

Figure 18 Illustrates A sequence diagram to depict Add, Update and Delete operations of restaurant manager

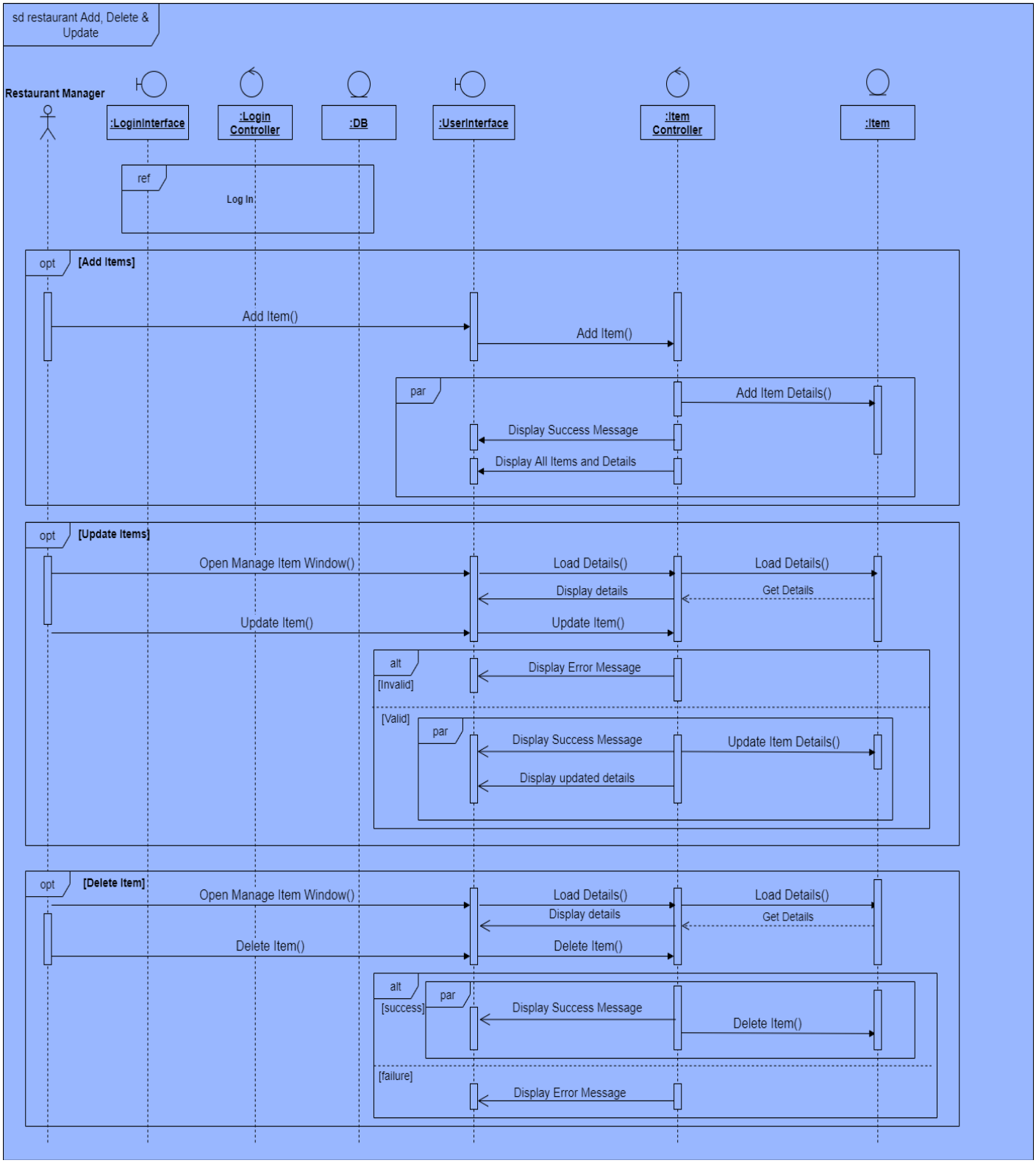


*Figure 18:restaurant management*

Figure 19 Illustrates the process of the transfer a consumable inventory and updating the consumable inventory of consumable table in a form of sequence diagram

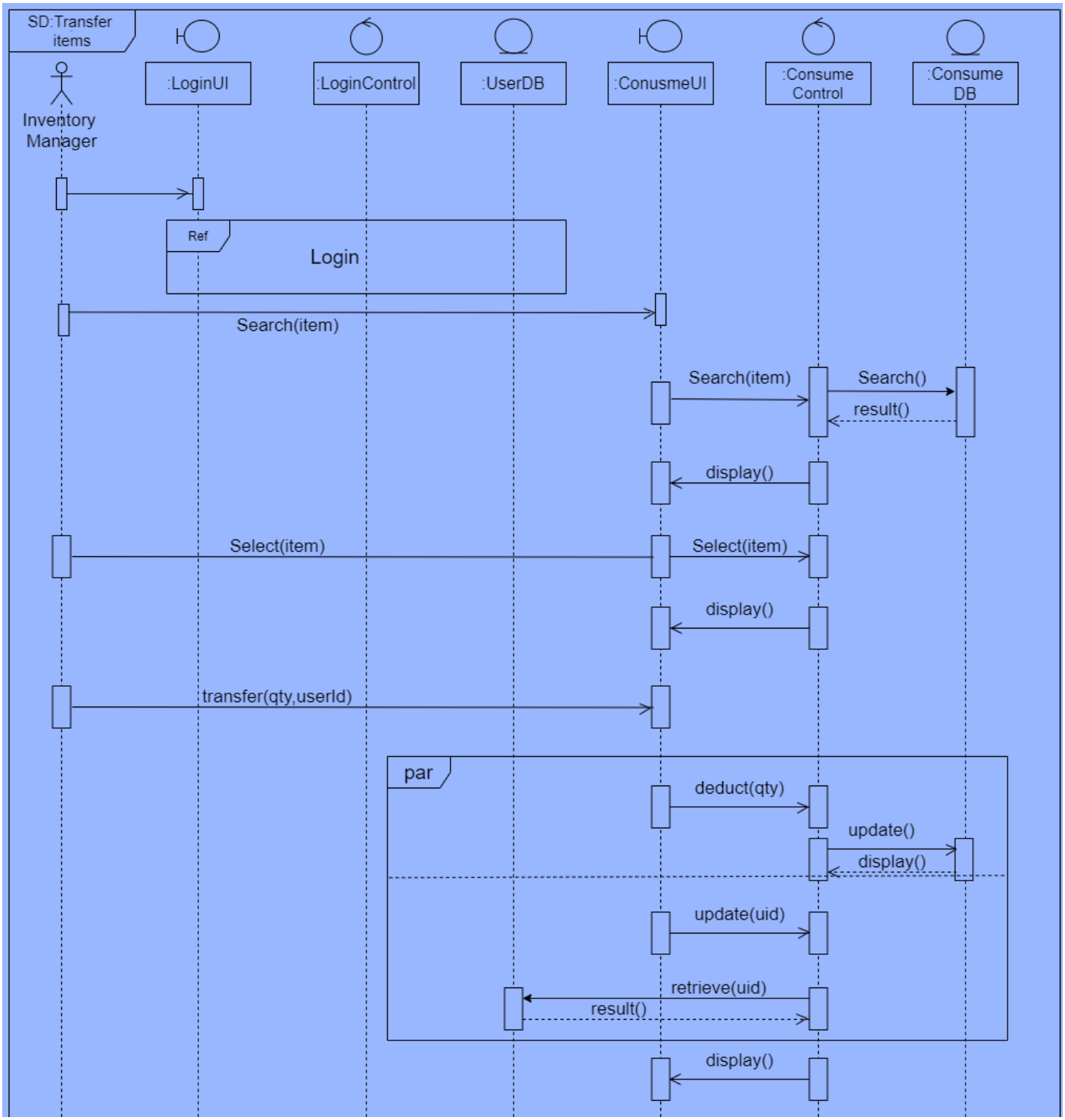


*Figure 19:consumable inventory management*

### 2.2.4 Communication Diagram

Figure 20 Illustrates the process of searching, adding, updating and deletion of the users in a form of a communication diagram.

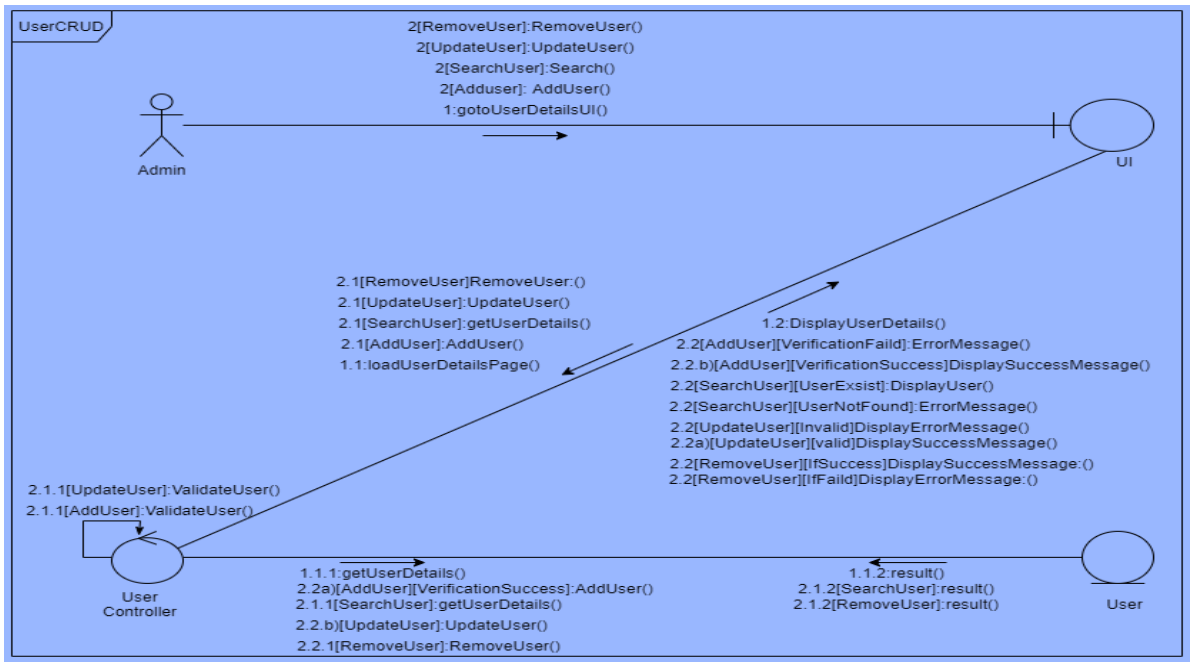


*Figure 20:manage users*

Figure 21 Illustrates an activity diagram to depict the process of transfer consumable inventory from inventory in a form of communication diagram

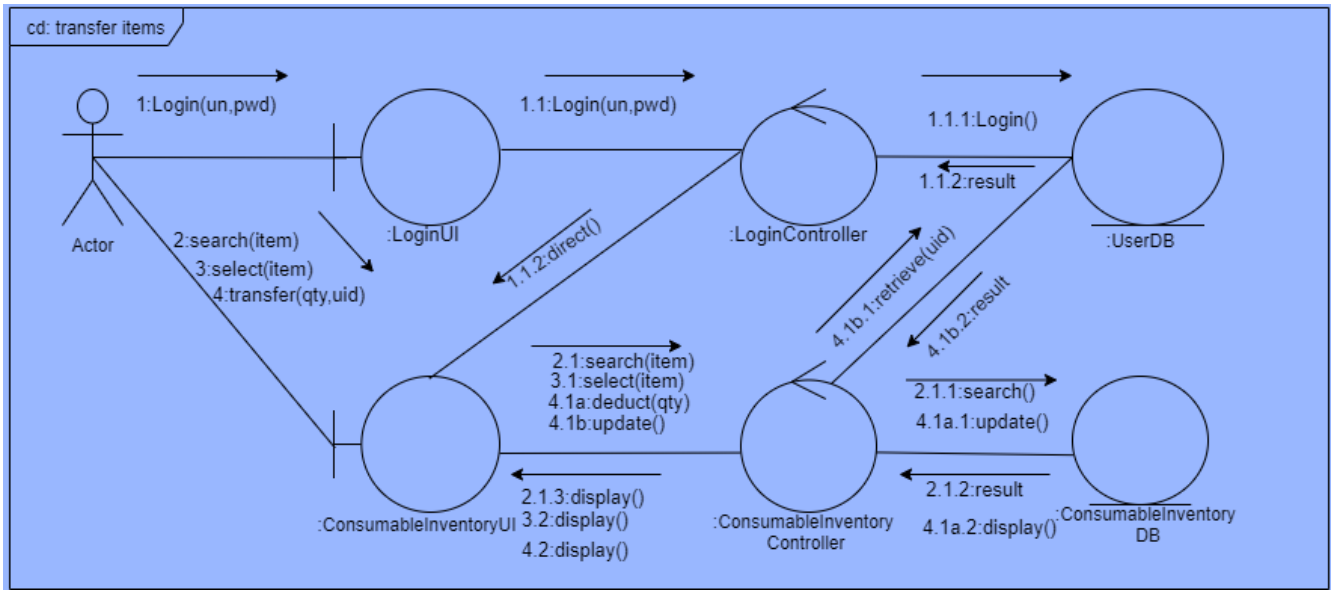


*Figure 21:manage consumable inventory*

Figure 22 Illustrates a communication diagram to depict the Room reservation function

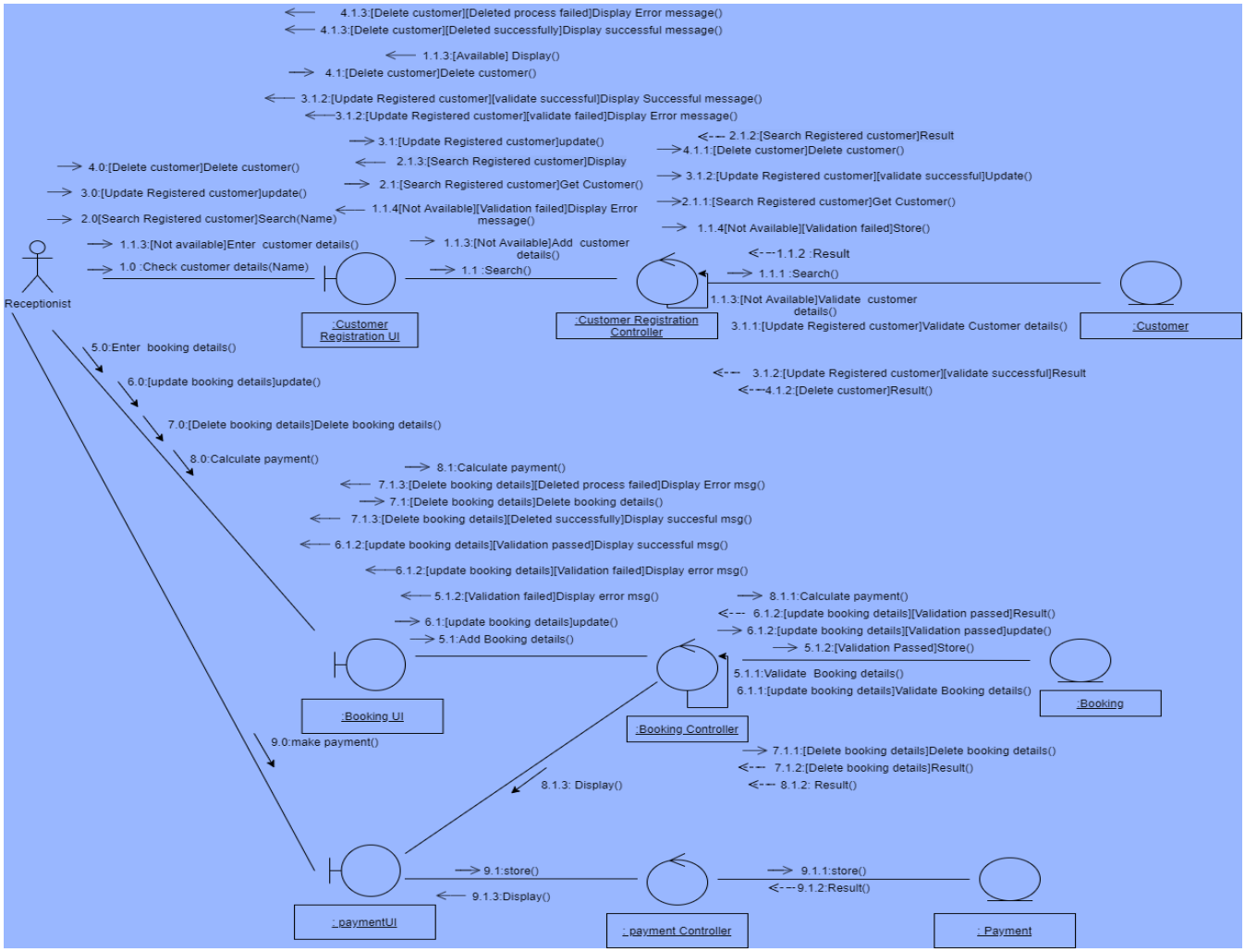


*Figure 22: reserve rooms*

Figure 23 Illustrates the process of adding, updating and deleting employees in the form of a communication diagram.

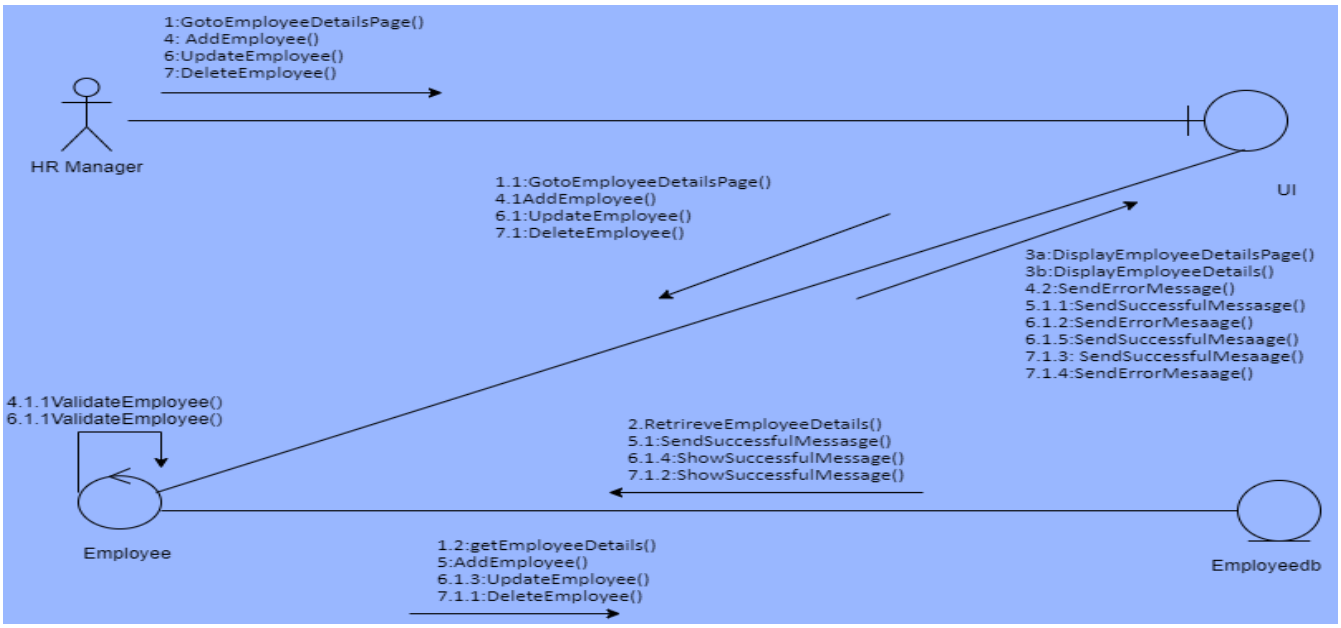


*Figure 23:manage employees*

Figure 24 Illustrates A communication diagram of Add, Update and Delete in restaurant.

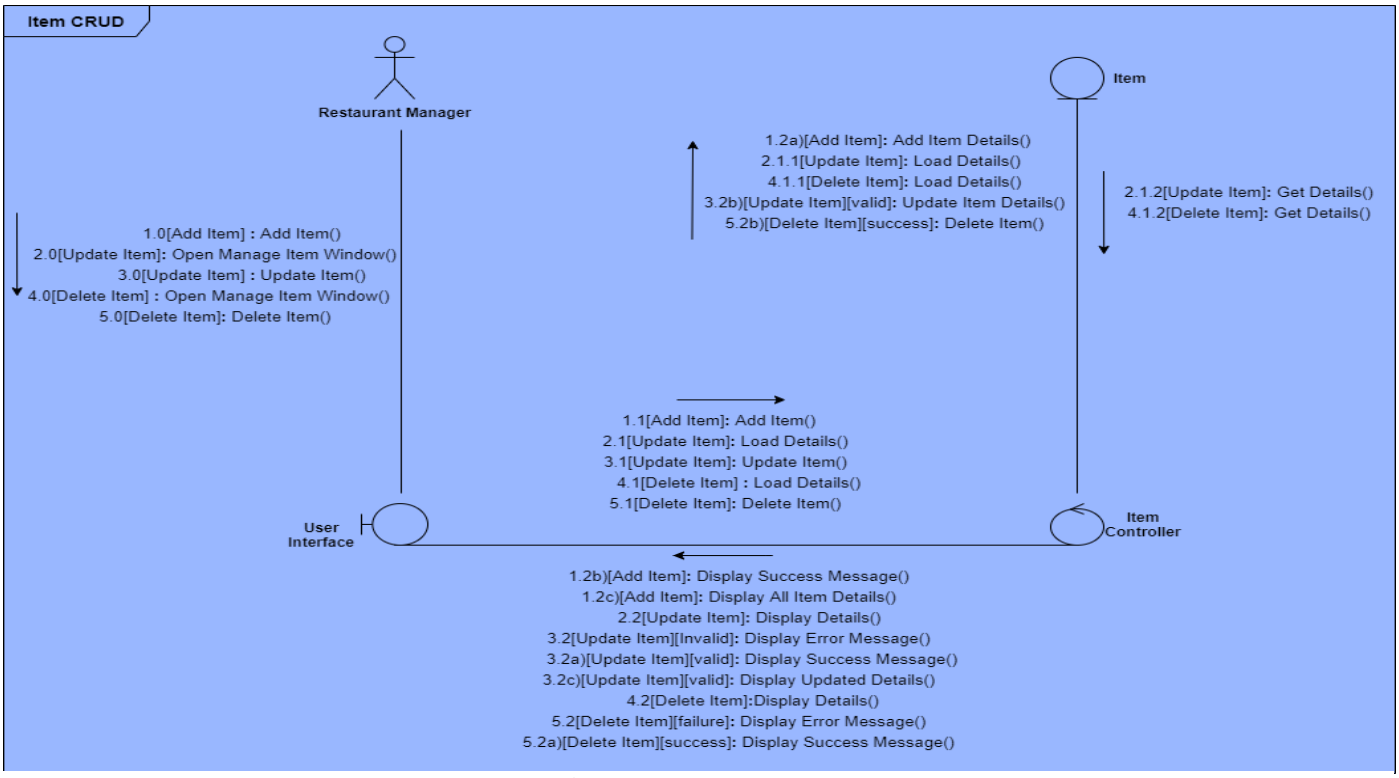


*Figure 24:manage restaurant*

Figure 25 Illustrates the process of adding, updating and deletion of the liquor details in a form of a communication diagram

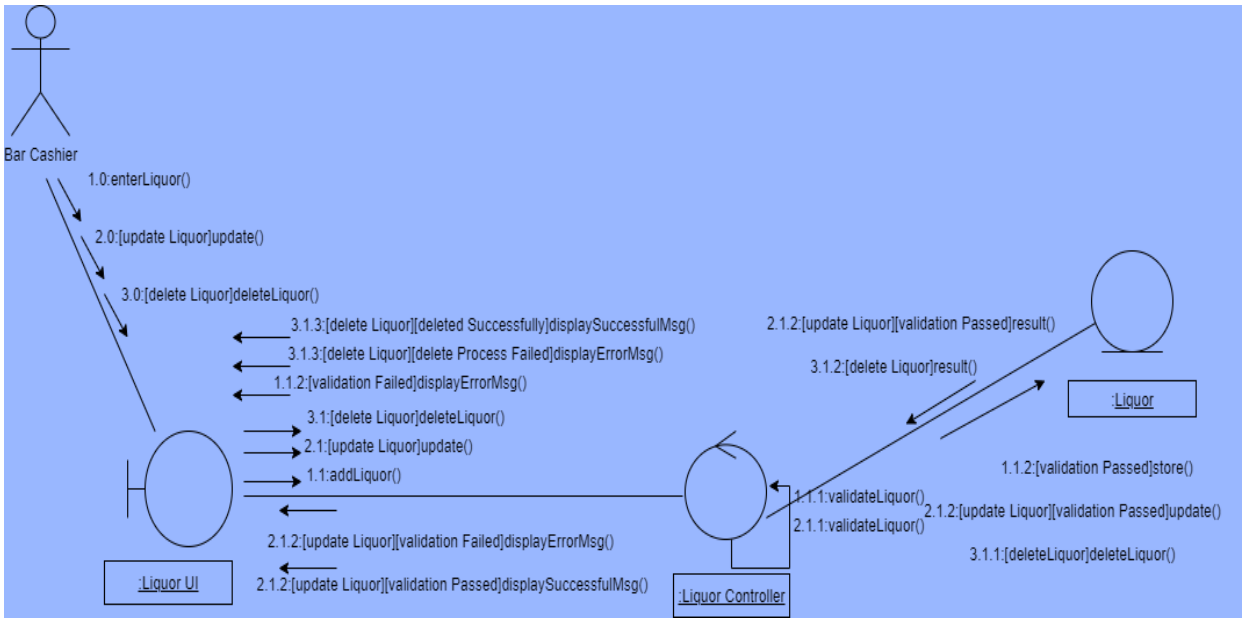


*Figure 25:manage liquor*

Figure 26 Illustrates Diagram of Hall Booking function which shows the Add, Update and Delete operations of Event Handler

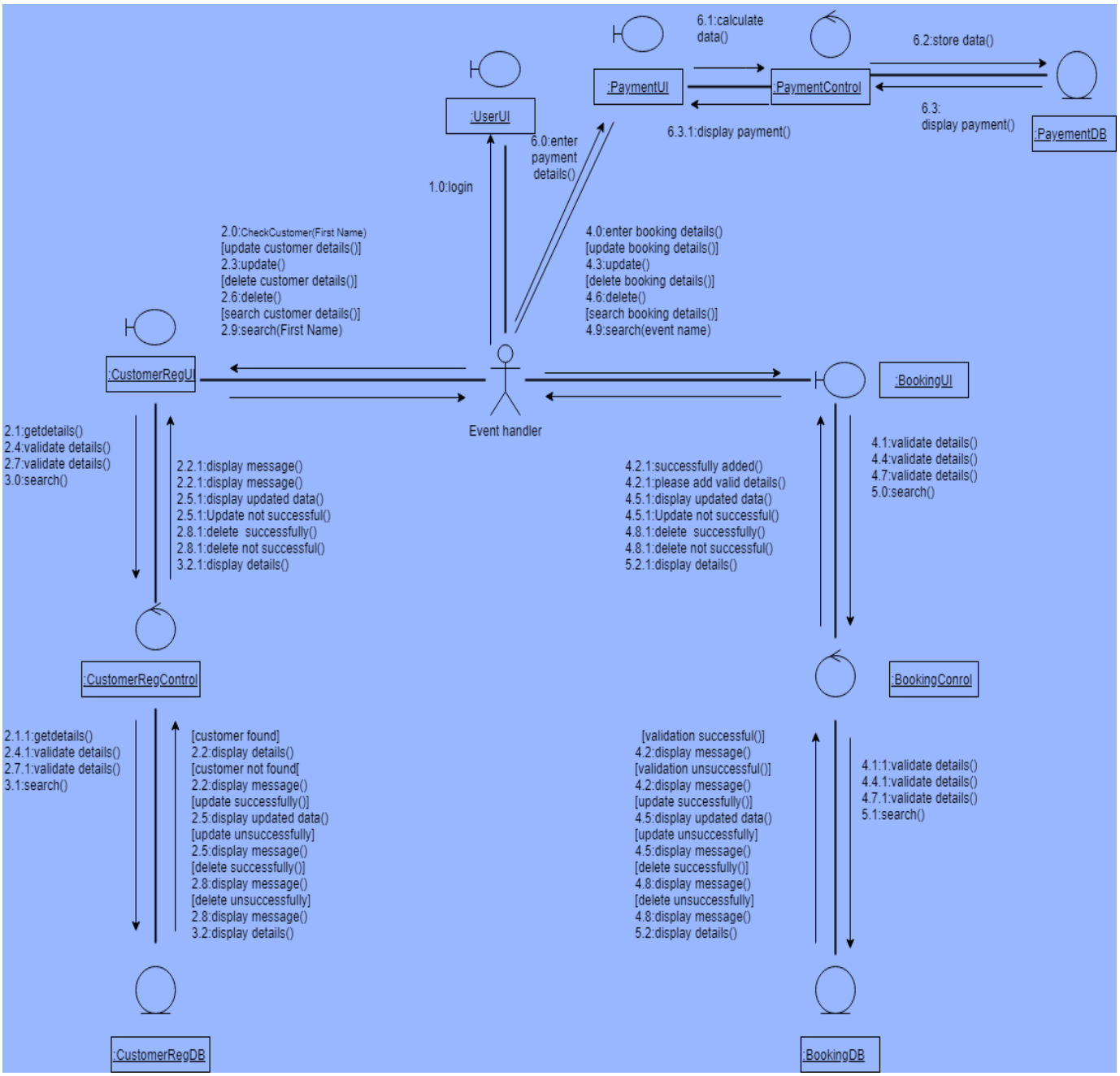


*Figure 26:manage hall bookings*

### 2.2.5 User Interfaces

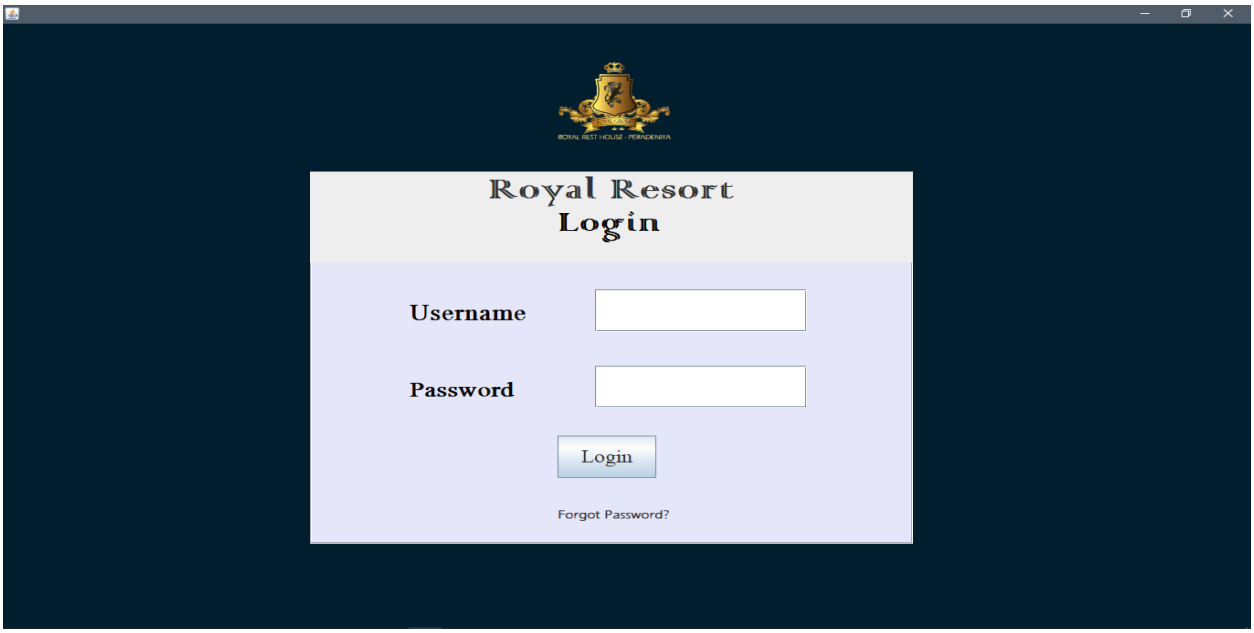


*Figure 27:login interface*

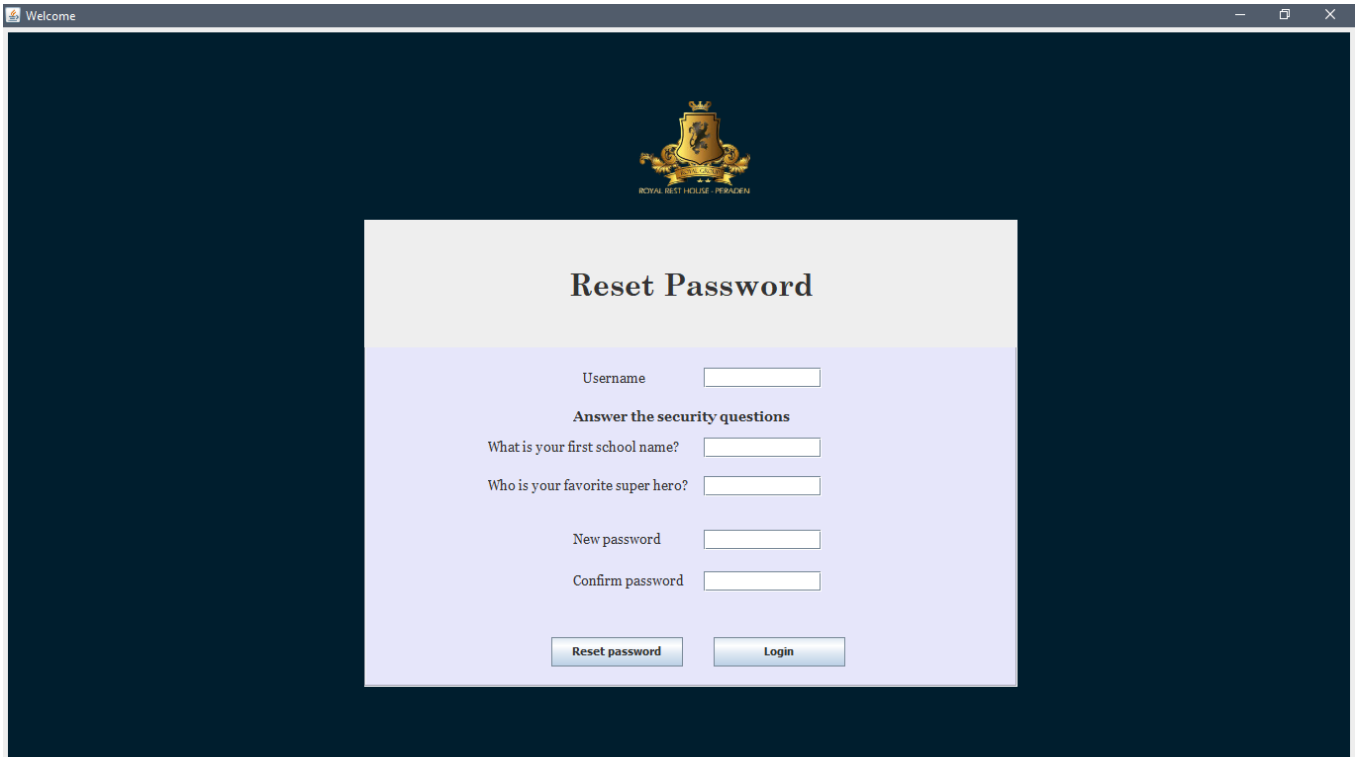


*Figure 28:reset password interface*

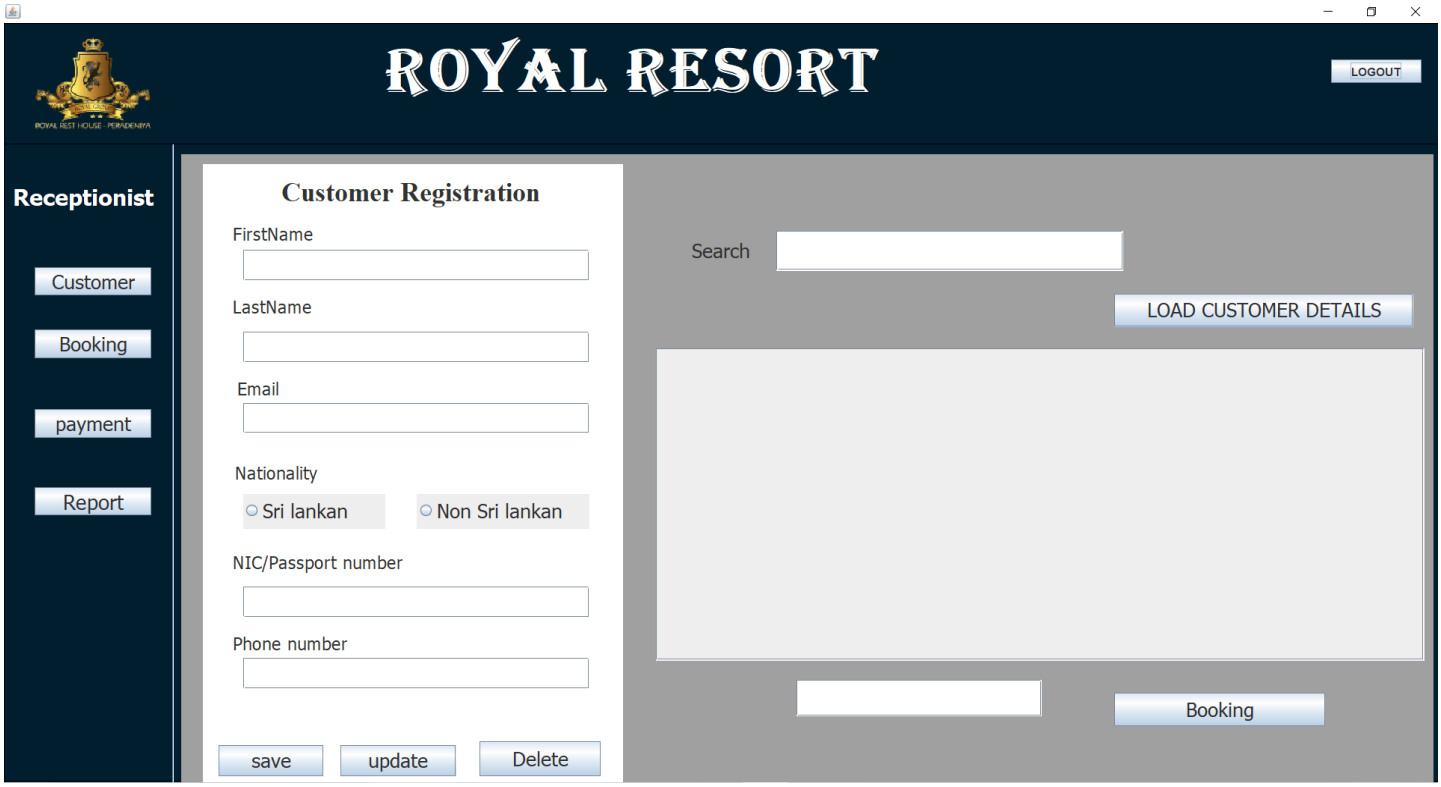


*Figure 29:receptionist interface*


*Figure 30:admin panel*

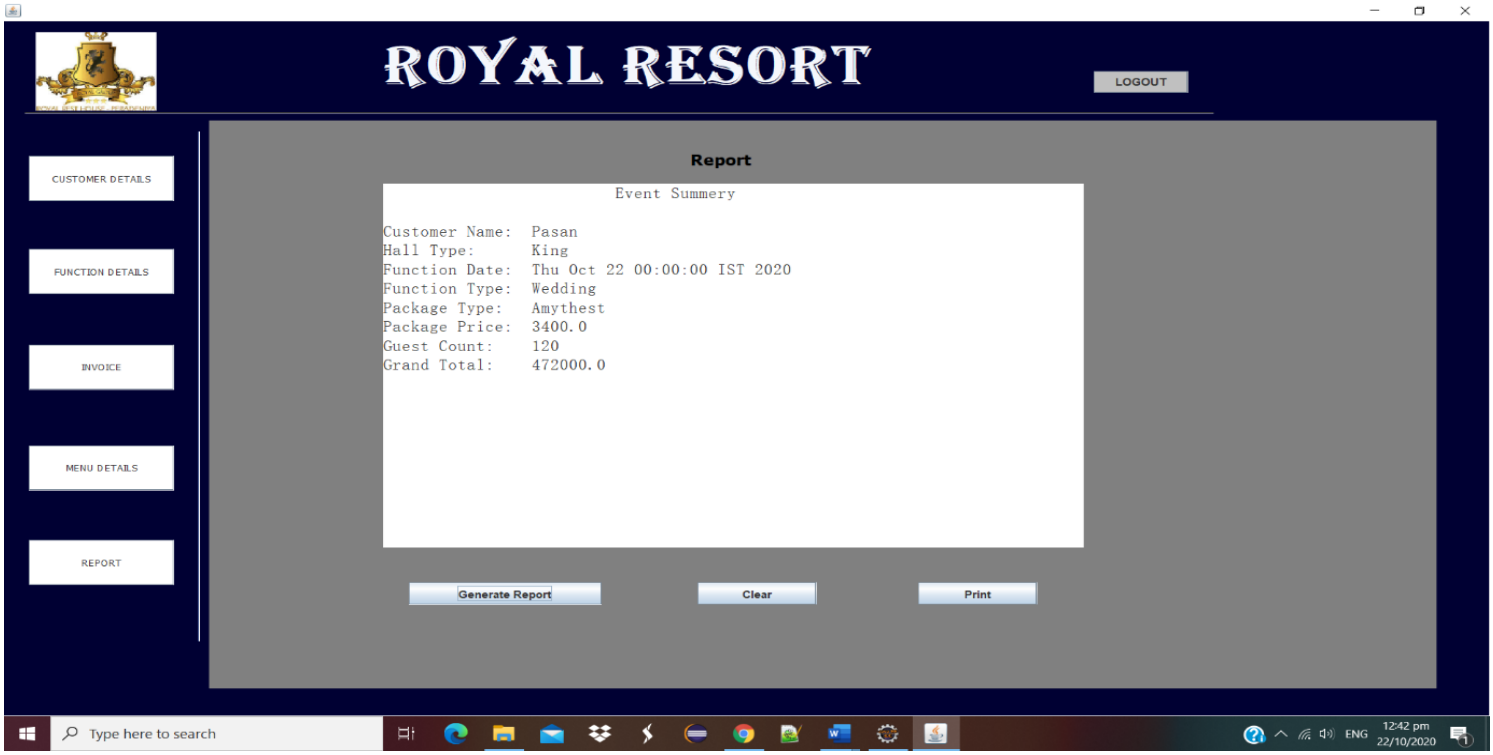


*Figure 32:event report interface*


*Figure 31:hall booking invoice interface*

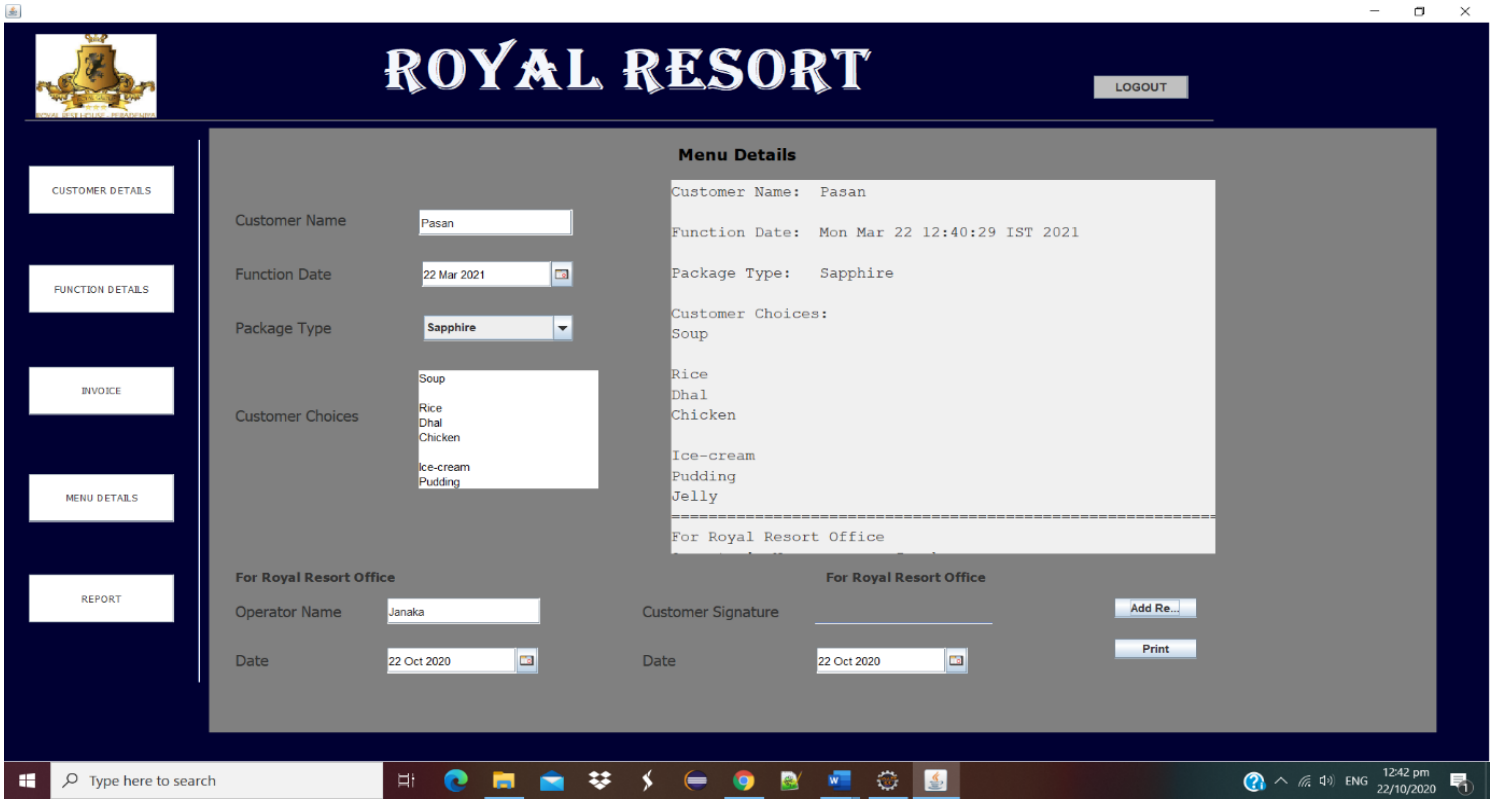


*Figure 31:hall booking interface*

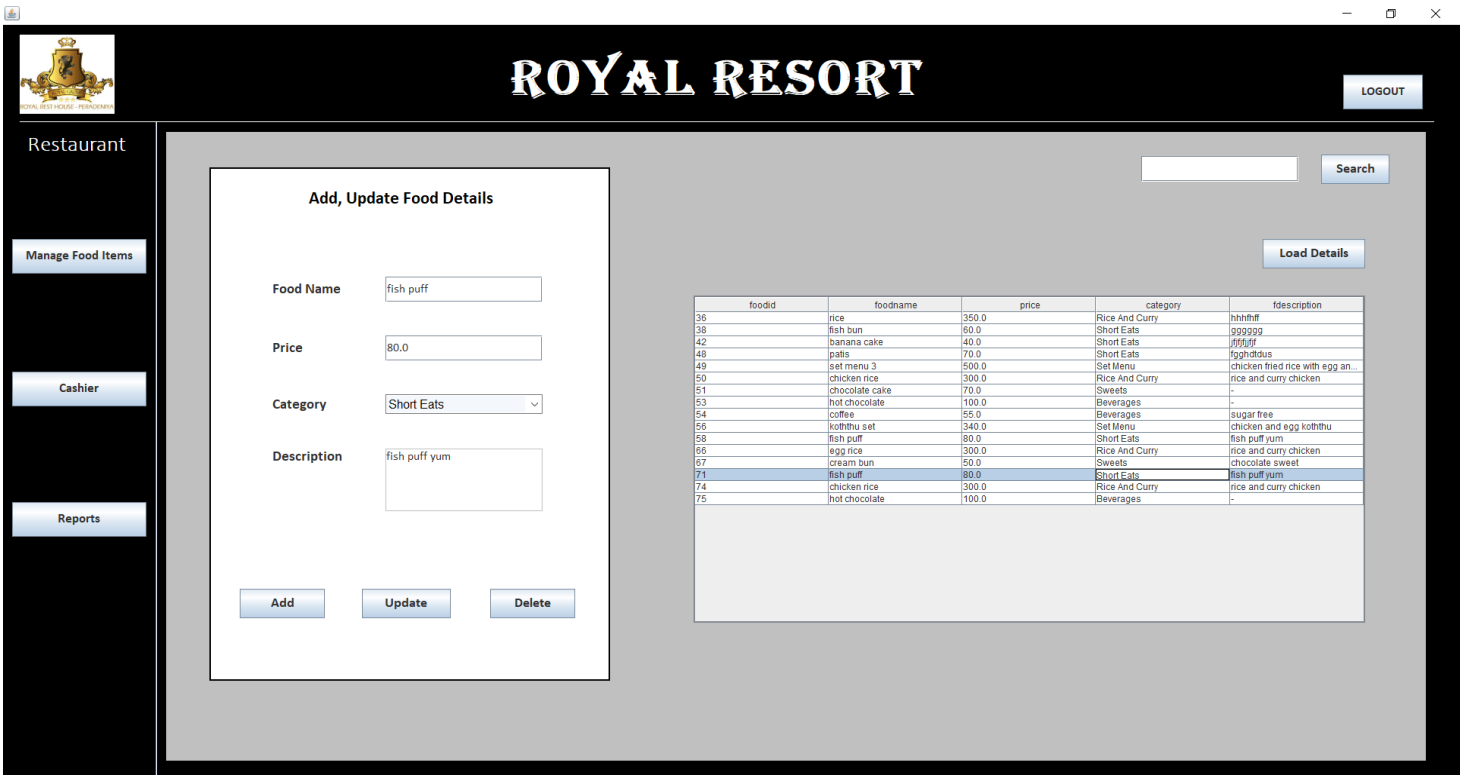


*Figure 32: food details interface*

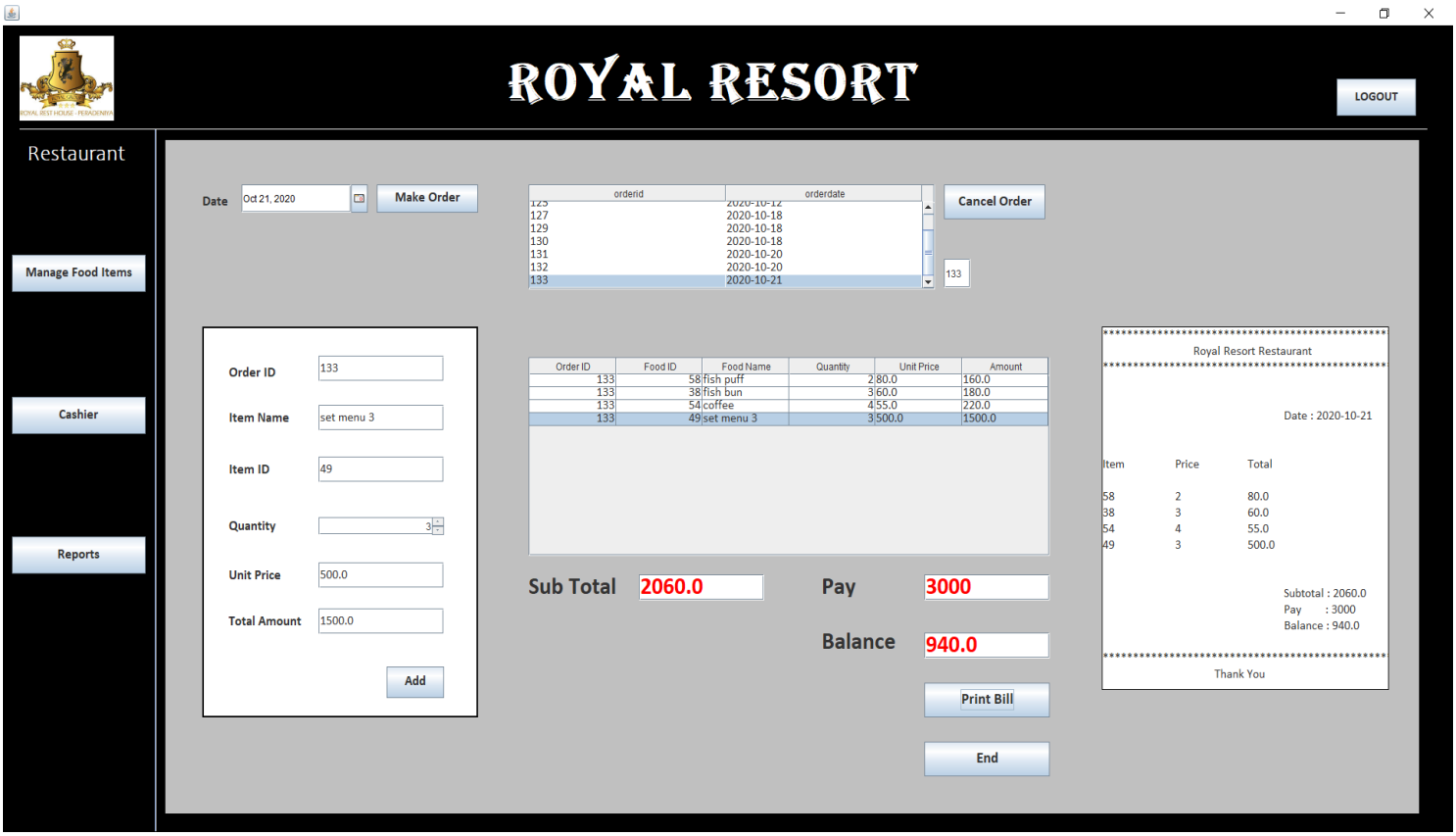


*Figure 35:restaurant report interface*

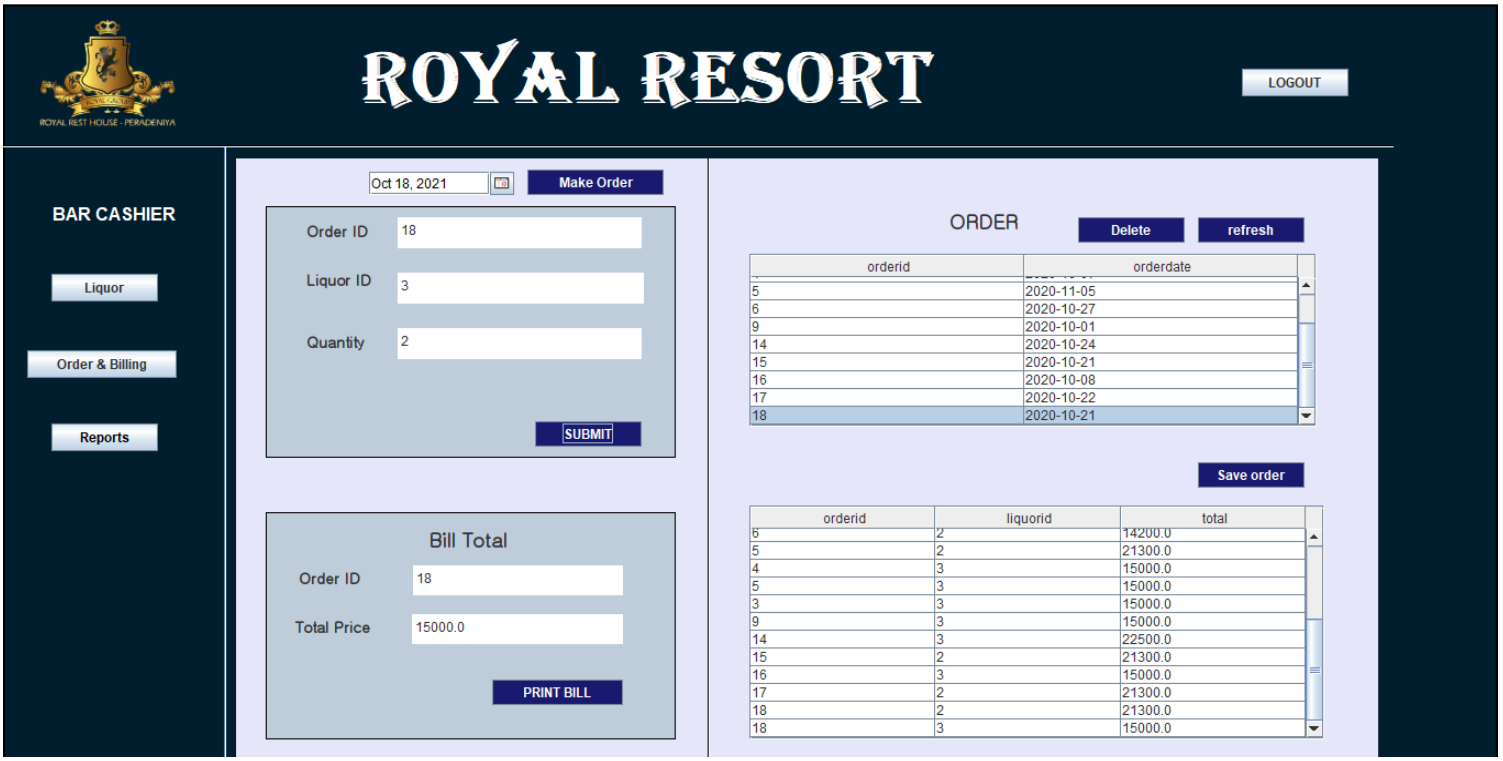


*Figure 34:bar cashier interface*

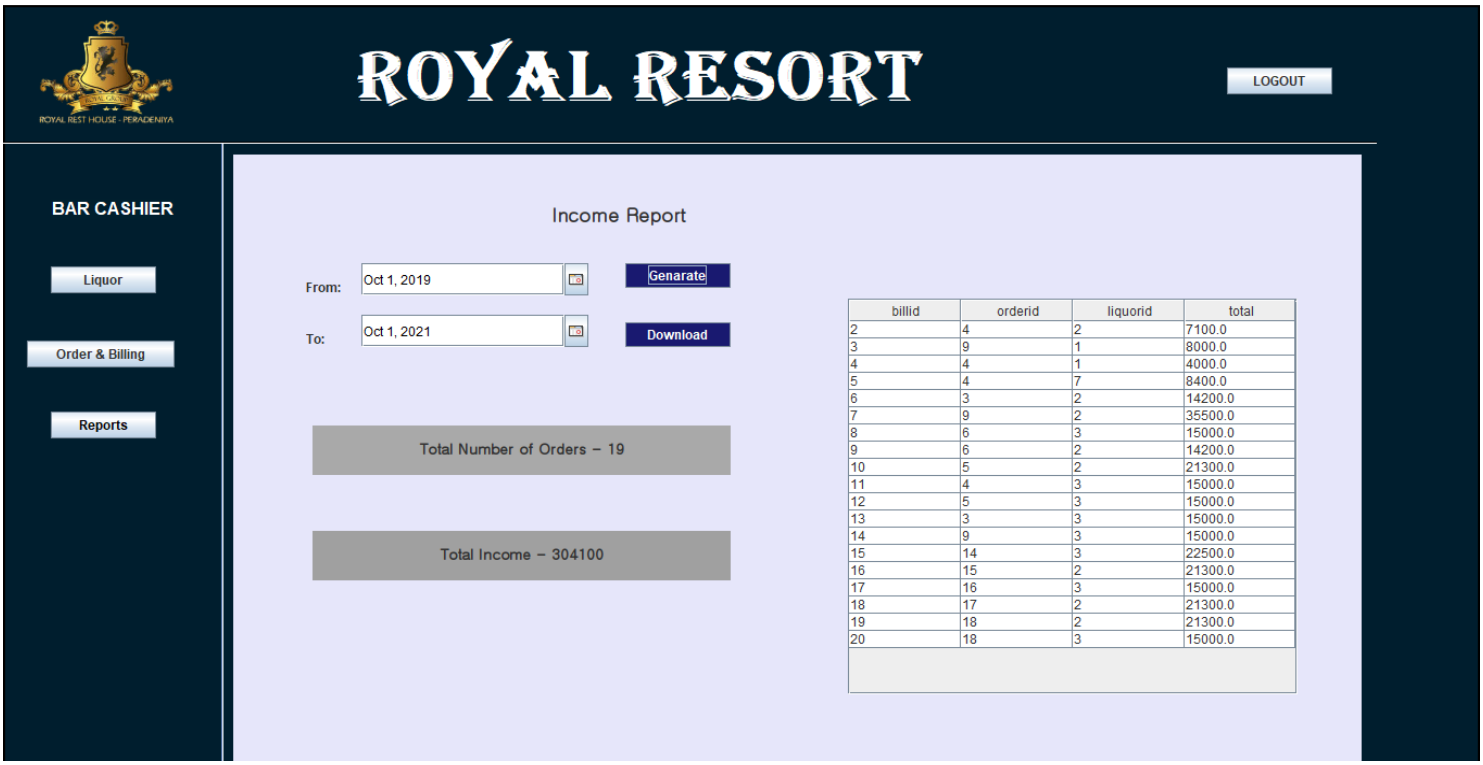


*Figure 37:bar report interface*

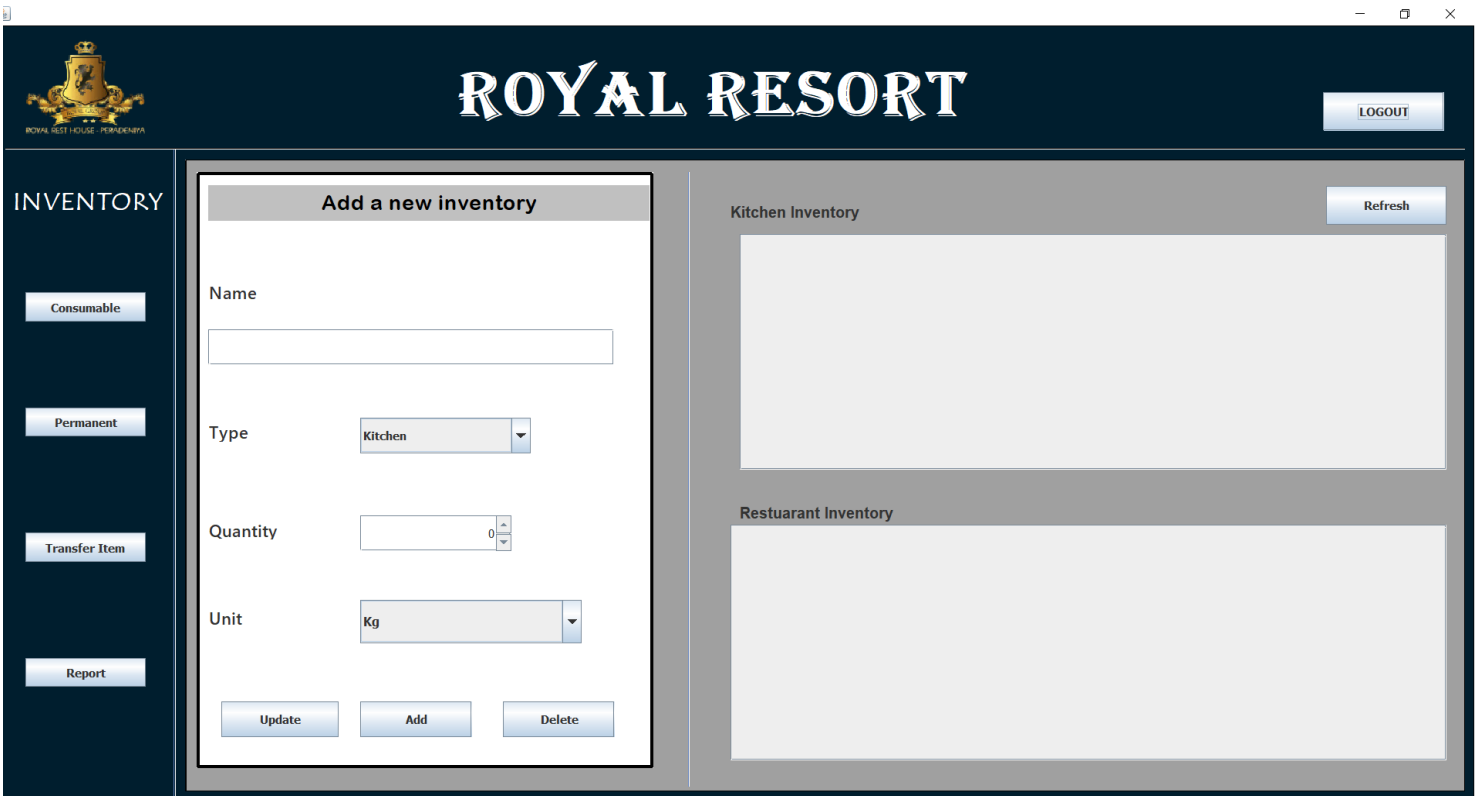


*Figure 36:consumable inventory interface*

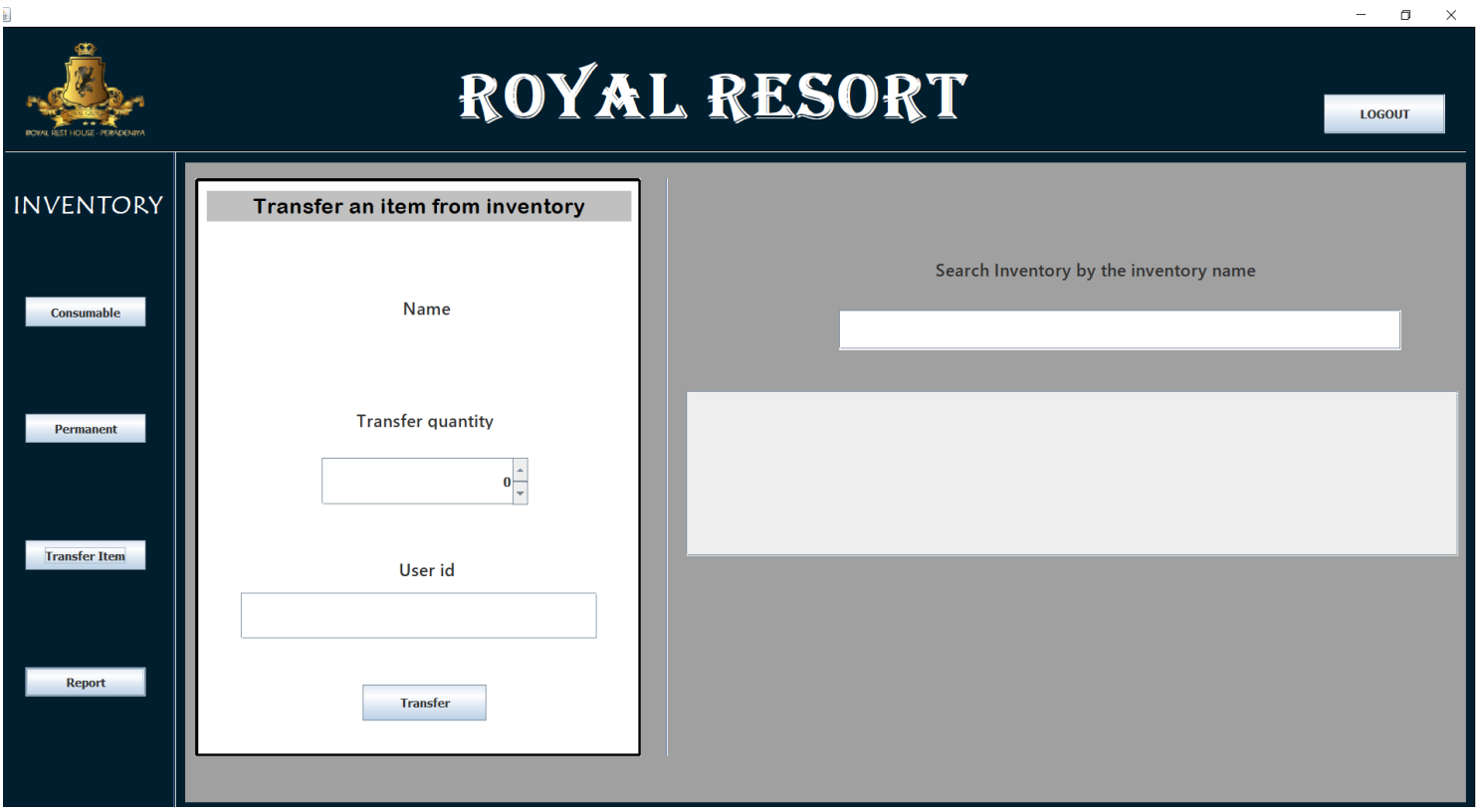


*Figure 38:transfer inventory interface*

## 2.3 Implementation

### 2.3.1 Database Management System

The DBMS we decided to use is the MySQL Workbench. It is a visual database design tool where it allows the user to design visual models. It is designed as an advanced database administrator tool where all the necessary database functions are available.
The reasons why we as a team decided to use MySQL workbench as our DBMS,

- Since we are all familiar with SQL it was unanimously decided to select a DBMS which supports SQL.
- It provides the ability to utilize most of the SQL commands that we might need in a GUI design.
- While deciding on the pros and cons with regards to another DBMS, MySQL workbench was easy to handle.
- The User interface was a very user-friendly design.
- A major advantage is the MySQL workbench provides a functionality to visualize the Database structure.
- Since our system is a standalone system without internet connection, MySQL Workbench provides us a way to access an offline database.

### 2.3.2 Implementation Language

The implementation language we used was JAVA and we used JAVA SWING as the graphical user interface widget toolkit.

The reasons why we used JAVA as our implementation language,

- Since we decided to develop a standalone desktop system, it was decided to use Java as everyone was familiar with the language and was thorough in object-oriented programming.
- Java is Rich in APIs, and it allowed us the ability to communicate with all the necessary functions outside our system.
- Since Java is multithreaded, we were able to develop a smooth user experience.
- Since Java Swing follows the MVC architecture, we were able to provide a much flexible User Interface.
- Java Swing components are lightweight and therefore it used less resources.
- The Java Swing library is a rich library with numerous components suited for user interface design.

## 2.4 Testing

Admin is capable of registering only one user under one unique username.

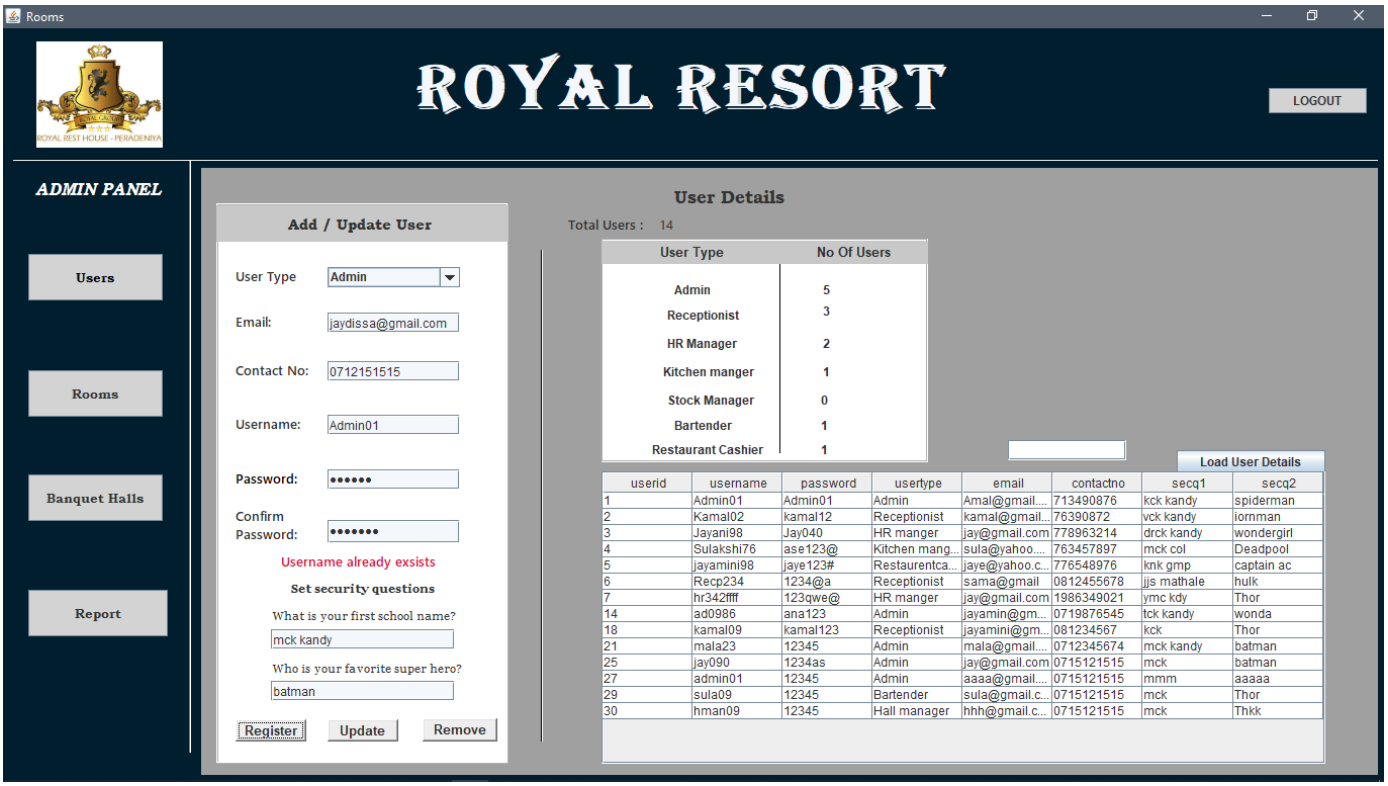


*Figure 39: register unique user test case*

Illustrates the test case done when adding a unique user into the system.

*Table 2:adding unique user test case*

| Test ID | Test Inputs | Expected Output | Actual Output | Test Result Pass/Fail | Description |
|---|---|---|---|---|---|
| **User01** | Username: Admin01 | Show success message after inserting data | Show success message. | Pass | Admin can register one user under one unique username. |
| **User02** | Username: Admin01 | Show error message" Username already exist" | Show error message. | Fail | Admin cannot register another user under same username. |

Illustrates that consumable inventory quantity entered is 0 and it will be display error message

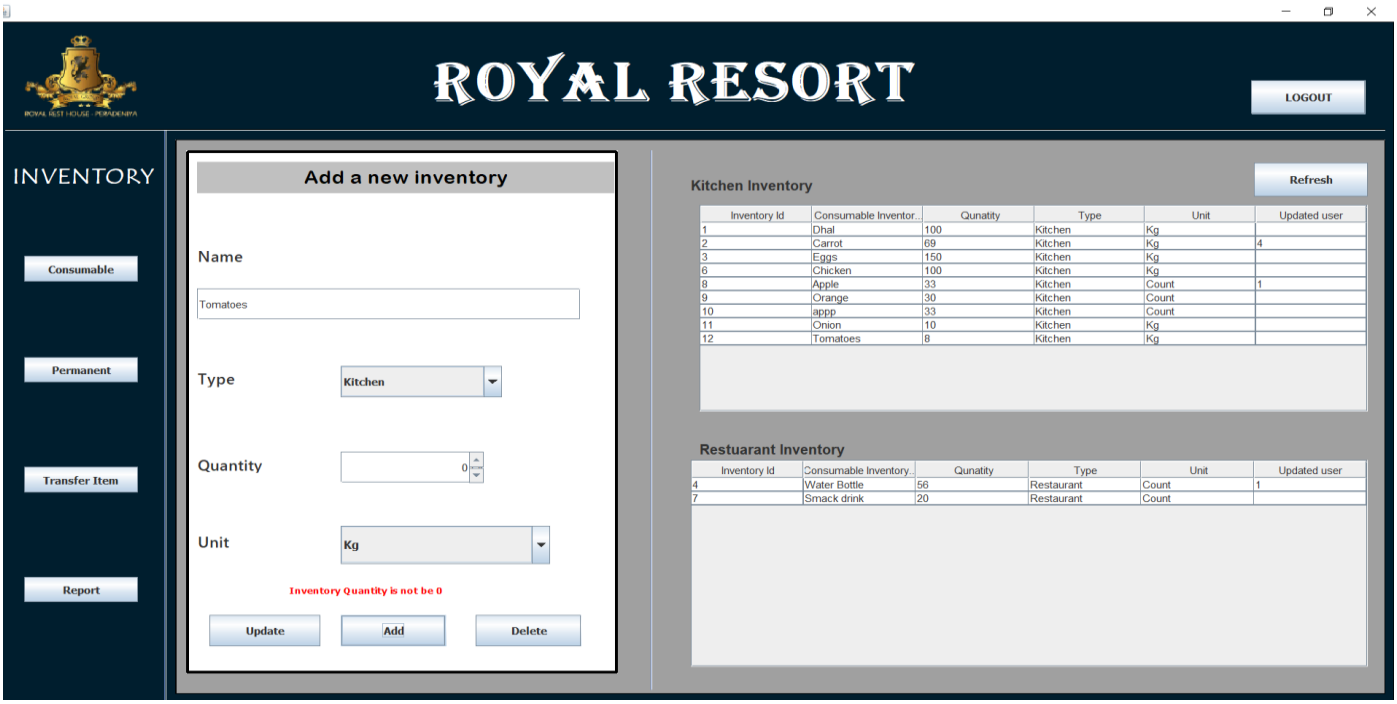


*Figure 40:consumable inventory test case*

Table 3 Illustrates the test case done when adding a unique consumable inventory into the system.

*Table 3:adding a unique consumable inventory test case*

| Test ID | Test Inputs | Expected Output | Actual Output | Test Result Pass/Fail | Description |
|---|---|---|---|---|---|
| **User01** | Name: Tomatoes<br>Type: Kitchen<br>Quantity: 8<br>Unit: Kg | Show success message after inserting data | Show success message. | Pass | Admin can register consumable inventory. |
| **User02** | Name: Tomatoes<br>Type: Kitchen<br>Quantity: 0<br>Unit: Kg | Show error message “Quantity cannot be 0“ | Show error message. | Fail | Admin cannot register consumable inventory. |

Illustrate that the form is validated, so customer details cannot be added with null values.

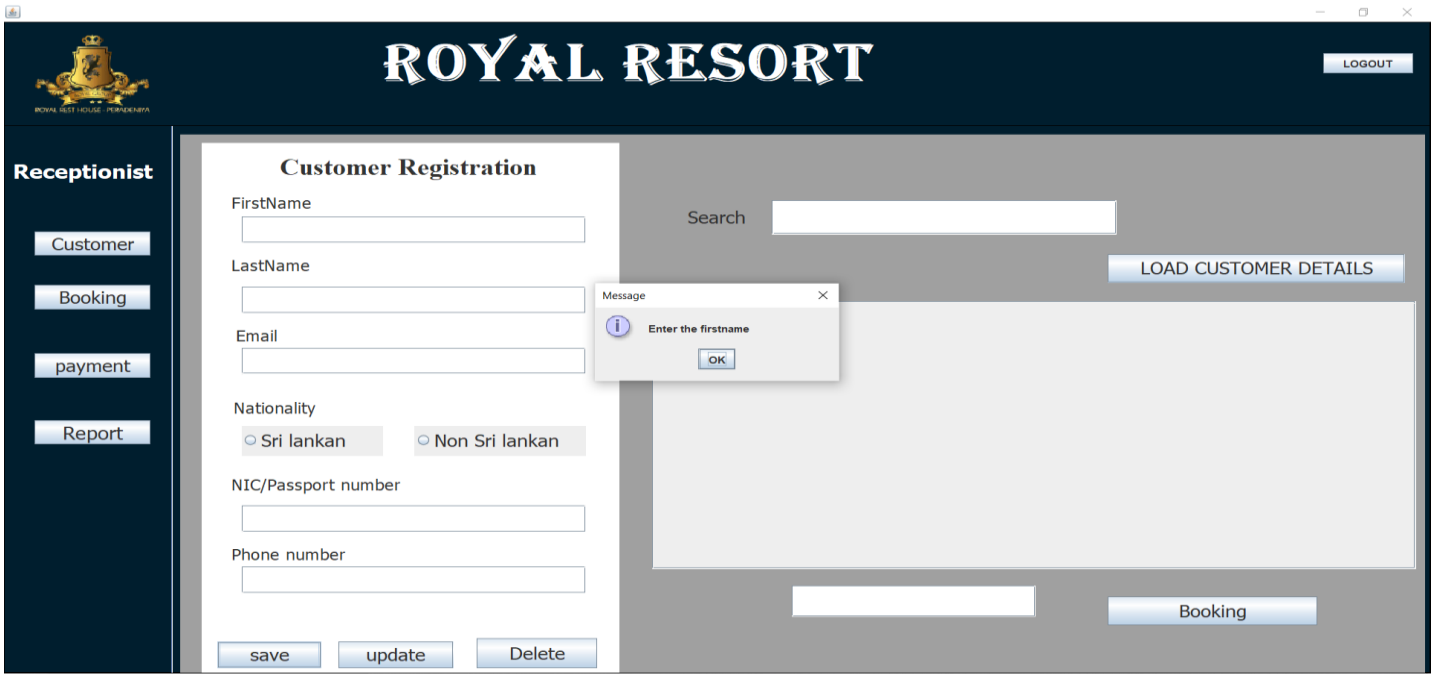


*Figure 41:customer details test case*

Table 4 Illustrates the test cases done when adding a customer details for a particular order.

*Table 4:adding customer details test case*

| Test ID | Test Inputs | Expected Output | Actual Output | Test Result Pass/Fail | Description |
|---|---|---|---|---|---|
| **Customer01** | First Name:- | Show error message "Enter the first name" | Show Error message. | Fail | As the form is validated, customer details with null values cannot be added. |
| **Customer02** | First Name:-Nimal<br>Last Name:-Perera<br>Email:-Nimal123@gmail.com<br>Nationality:-Srilankan<br>NIC/Passport number:-988520217V<br>Phone number 0714455773 | Show successful message "New customer has successfully added" | Show successful message. | Pass | If all the details required are provided according to the validations, then the details will be added successfully to the database. |

Figure 42 Illustrates that an employee cannot be added without the basic information.

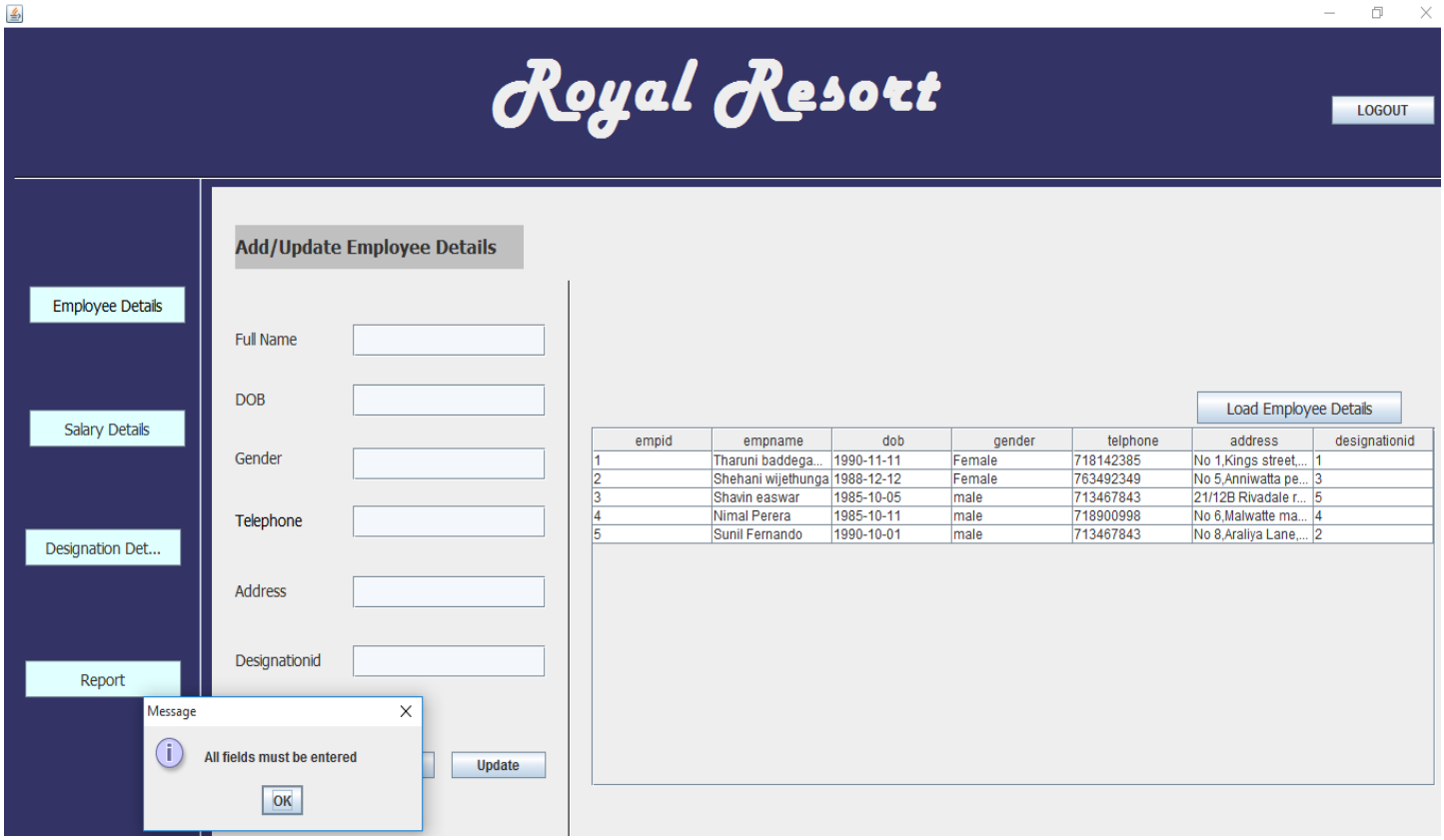


*Figure 43:add employee test case*

Table 6 Illustrates the test cases done when adding an Employee details for a particular order

*Table 5:adding employee details test case*

| Test ID | Test Inputs | Expected Output | Actual Output | Result(Pass/Fail) | Description |
|---|---|---|---|---|---|
| Employee01 | Full name : -<br>DOB: -<br>Gender : -<br>Telephone: -<br>Address: -<br>Designationid: - | Show error message ”All fields must be entered” | Show successful message | Fail | As the form is validated, an employee cannot be added to the system without the basic information. |
| Employee02 | Full name : Kelum Perera<br>DOB: 1990-11-26<br>Gender : Male<br>Telephone:0788887766<br>Address:- No.3,Nelum mawatha, Kandy<br>Designationid:-3 | Show successful message adding the employee to the system. | Show successful message | Pass | If all the details are correct then the employee will be added to the database. |

Illustrate that the form is validated, so liquor details cannot be added with null values.

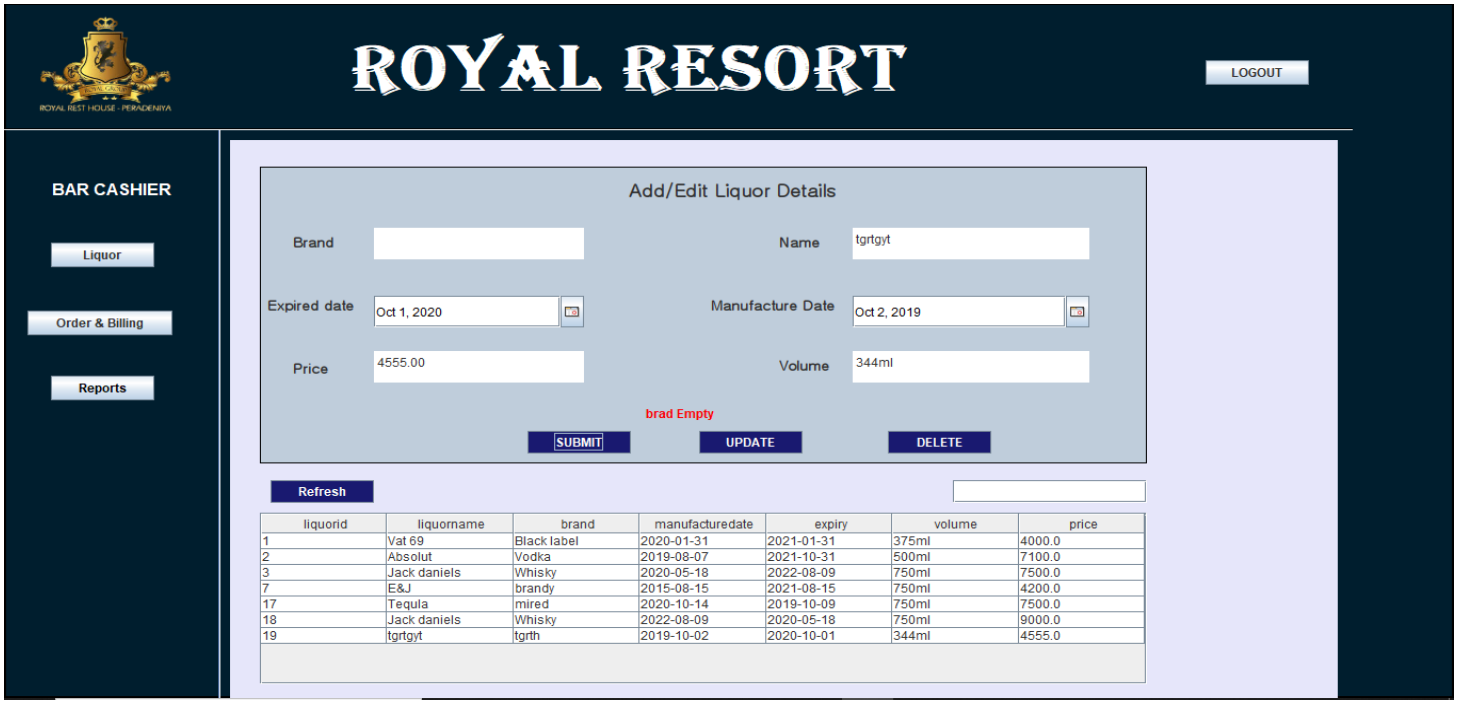


*Figure 44:add liquor test case*

Table 7 Illustrates the test case done when adding Liquor in to the system.

*Table 6:adding Liquor test case*

| Test ID | Test Inputs | Expected Output | Actual Output | Test Result Pass/Fail | Description |
|---|---|---|---|---|---|
| **Liquor01** | Brand:- | Show error message “brand Empty” | Show error message. . | Fail | Form is validated liquor details with null value can not be added. |
| **Liquor02** | Brand:- vodka | Show success message “New liquor is Successfully added” | Show success message | pass | Details will be added successfully to the database. |

Illustrates that the restaurant cashier is only capable of adding an item (for a particular order) which is already stored in the system.

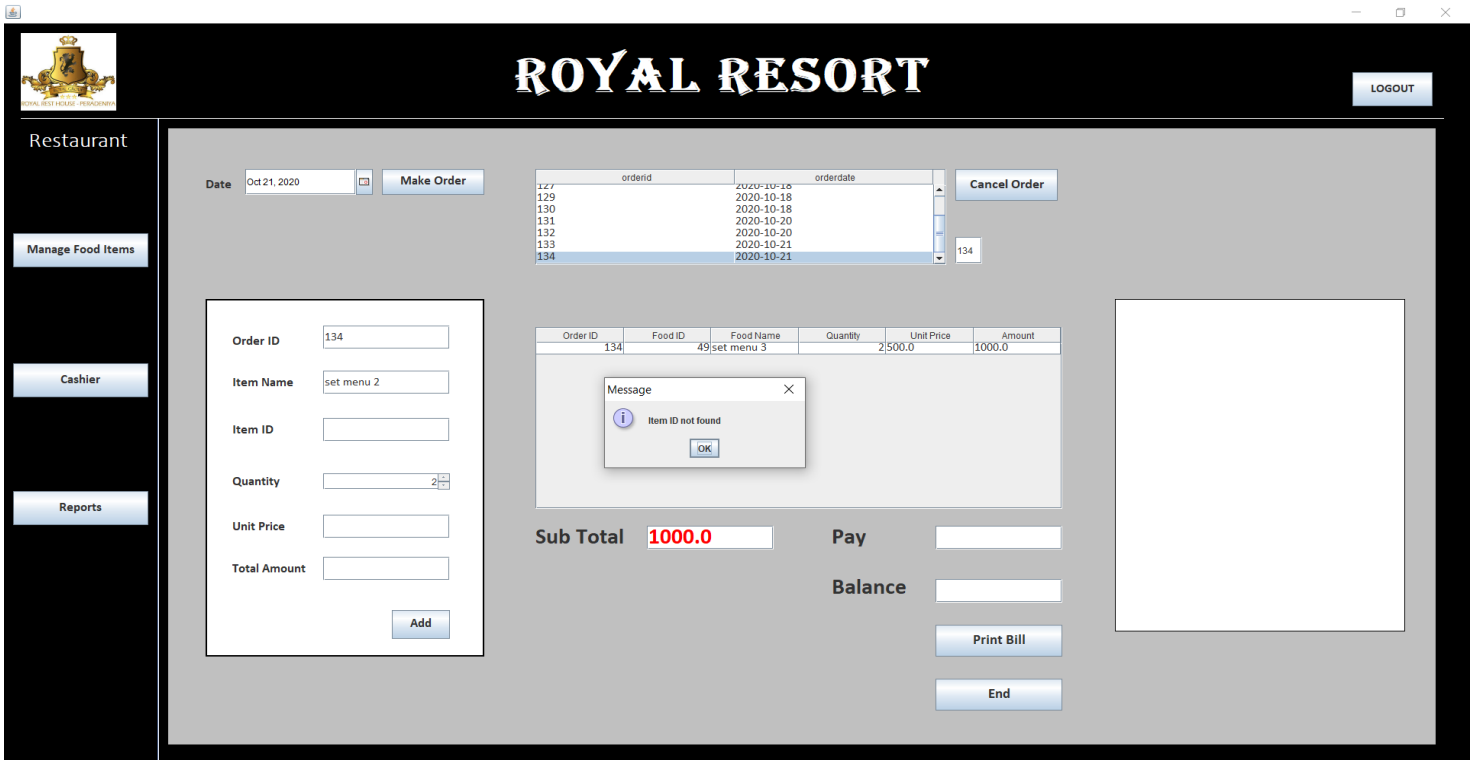


*Figure 45:restaurant order test case*

Table 9 Illustrates the test cases done when adding an Item for a particular order.

*Table 7:adding an Item for a particular order test case*

| Test ID | Test Inputs | Expected Output | Actual Output | Test Result Pass/Fail | Description |
|---|---|---|---|---|---|
| **Order 1** | Item Name : set menu 3 | Display relevant details (price and item ID) in the fields | Display relevant details (price and item ID) in the fields | Pass | If all the details of a particular item is entered correctly user can proceed the order. |
| **Order 2** | Item Name : set menu 2 | Display error message “Item ID not found” | Display error message | Fail | Since the item name field is validated user can’t enter a item that is not stored in the system, so that he can’t proceed the order. |

Illustrates that the customer cannot add an invalid phone number or email address to the system

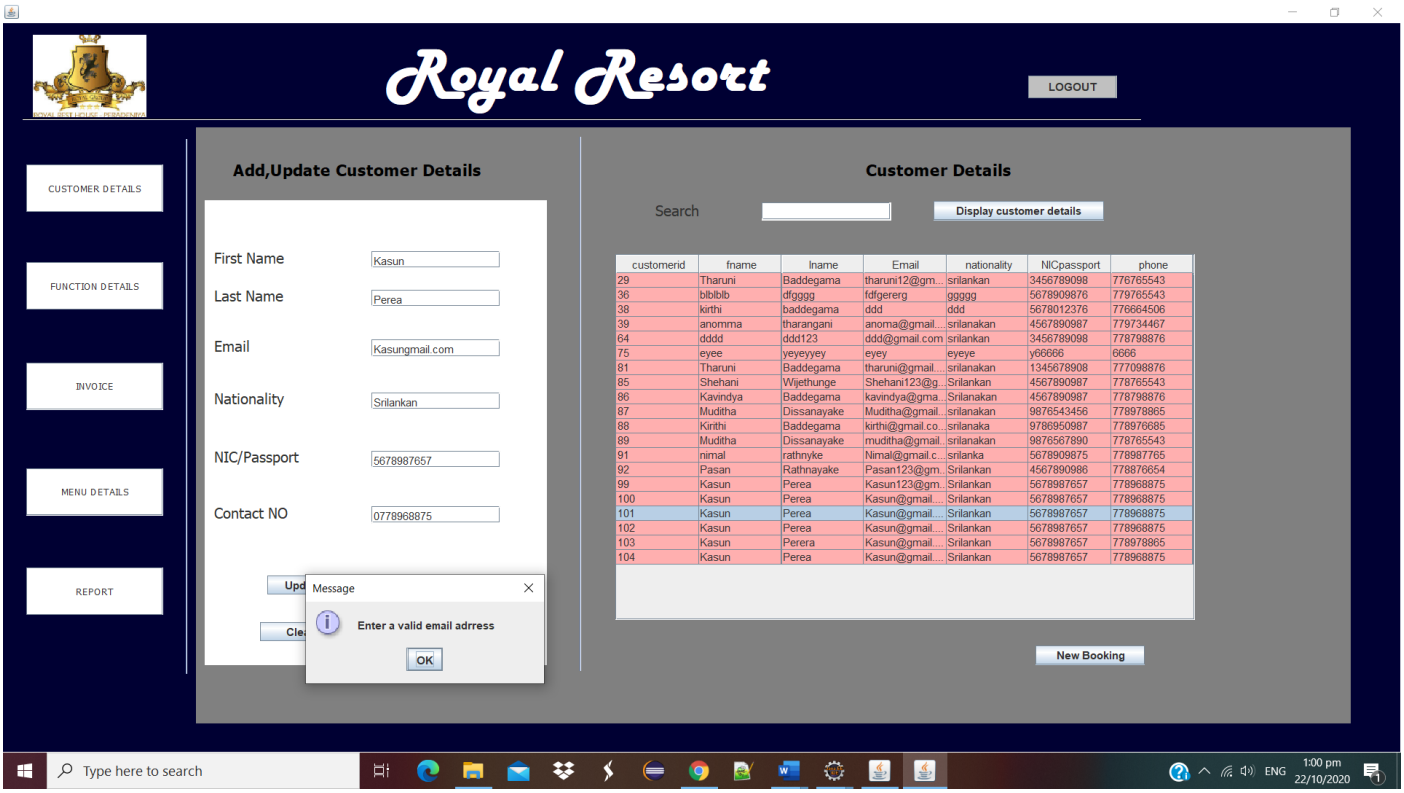


*Figure 46:add customer test case*

*Table 8:invalid email address or phone number test case*

| Test ID | Test Inputs | Expected Output | Actual Output | Test Result Pass/Fail | Description |
|---|---|---|---|---|---|
| **Customer01** | First Name: -Pasan<br>Last Name: -Perera<br>Email:-Pasan@gmail.com<br>Nationality:-Srilankan<br>NIC/Passport number:-9876545678<br>Phone number:0778596654 | Shows a successful message and the data is saved in the database | Successful message is displayed. | Pass | Since the data is entered is correct it will be successfully entered in to the database |
| **Customer02** | First Name: -Pasan<br>Last Name: -Perera<br>Email:-Pasangmail.com<br>Nationality:-Srilankan<br>NIC/Passport number:-9876545678<br>Phone number 077856654 | Shows an error message | Shows an error message | Fail | An error message will appear on the screen as the phone number and email address are entered incorrectly |

# 3. Conclusion

The project “Royal Resort” is a hotel and restaurant management system which was designed to automate the manual process of data handling in the hotel “Peradeniya Rest House”. From the requirements we gathered, we were able to identify eight key points in the manual process currently in use. The system was developed in order to resolve those which were identified. As soon as the final method was presented to the client, the advocates concluded that it was dynamic and fully operational. Furthermore, the system has been developed in line with the required automation company requirements.

The main objective behind developing an automated system was to eradicate the human made errors and to completely erase fraudulent activities, by implementing such a system we were able to achieve the objectives by making the system a fool proof and error free system. Apart from the main objective there were minor goals that we as a team wanted to implement in the system, by analyzing the needs of the company, we were able to execute different types of logic help the company become more efficient even if the client does not request it.

Although the system is automated, it still has some limitations.

- Technology is an ever-changing factor and there will always be better technologies, so with time the technology we used and methods we used will be outdated it is necessary to maintain and upgrade from time to time.
- Building a fool proof system is extremely hard, there will be unauthorized outside interference and it is important to keep the system up to date with security patches.
- There should always be a technician to fix bugs and errors with the system and it can be difficult to deal with people who do not know the system properly.

Even though the above-mentioned limitations may sound scary there will always be several advantages of using the automated system we built for the company.

- Since there will be a drastic decrease in the fraudulent activities, the company will be able to witness a huge financial increment, because the data will be accessed by one or few authorized people with access credentials to the system.
- The use of automation system will enable the company to become more efficient and organized.
- The productivity of the organization will see a gradual increase with time and will affect all the employees working in the organization to be more vigilant.

Throughout this project we had to face many difficulties. Some days we thought of giving up some parts, but the team spirit and unity somehow enabled us to develop the system as we wanted. Finally, able to see our hard work pay off and deliver a fully functional automated system on time that we are proud of.

# 4. Reference

**[1] https://opus.govst.edu/capstones/198/**

**[2] Nijith Jacob (2007A7PS076G)Karandikar Rajas Hemant (2007A7PS046G) Submitted in partial fulfilment of Object Oriented Analysis And Design course (SS G514) Under the guidance of J.V RAO**

# Appendix A: Design Diagrams

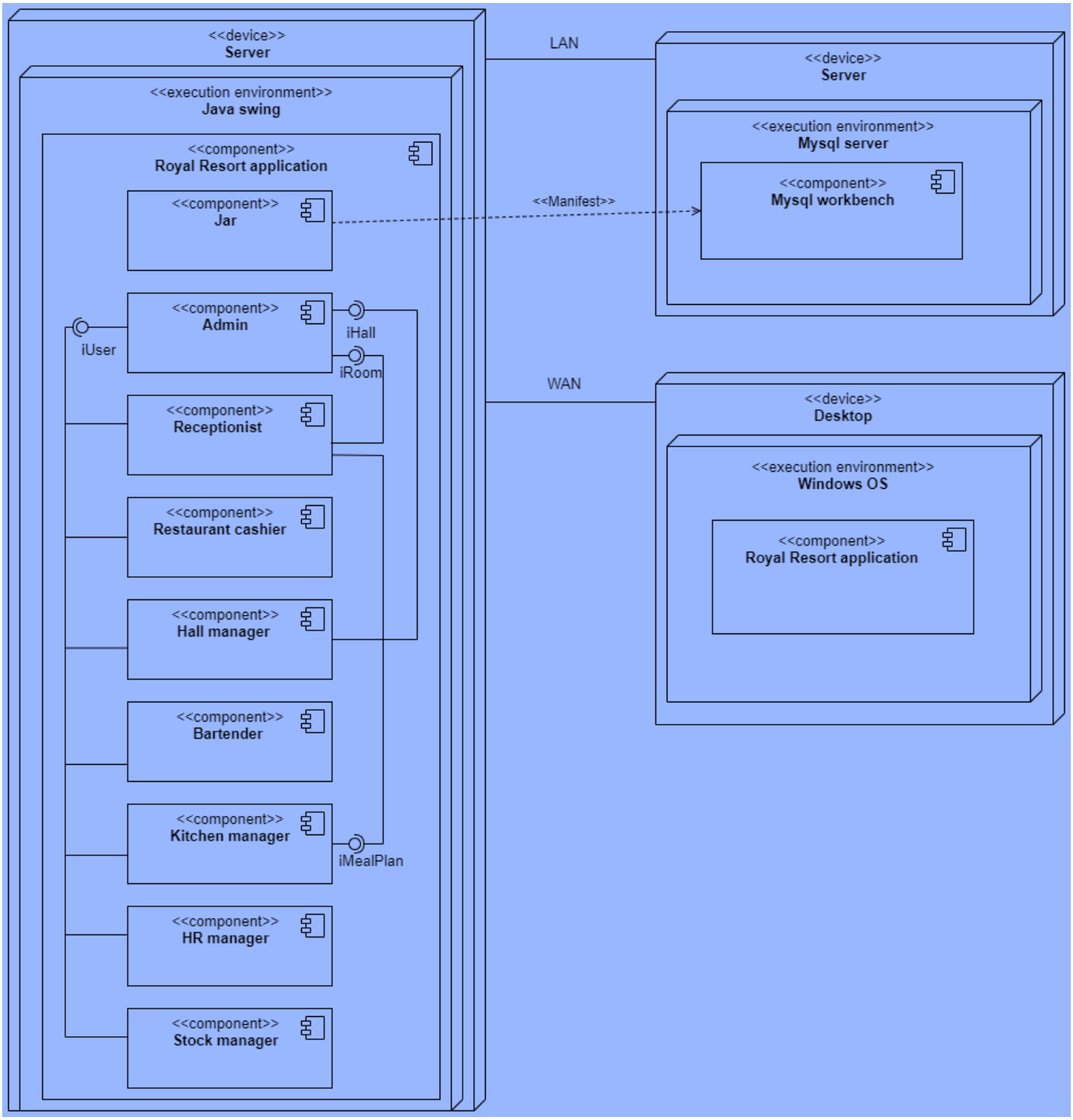


*Figure 47:deployment diagram*

# Appendix B: Test Results

Illustrates that consumable inventory data entered is correct and it will be successfully entered in to database

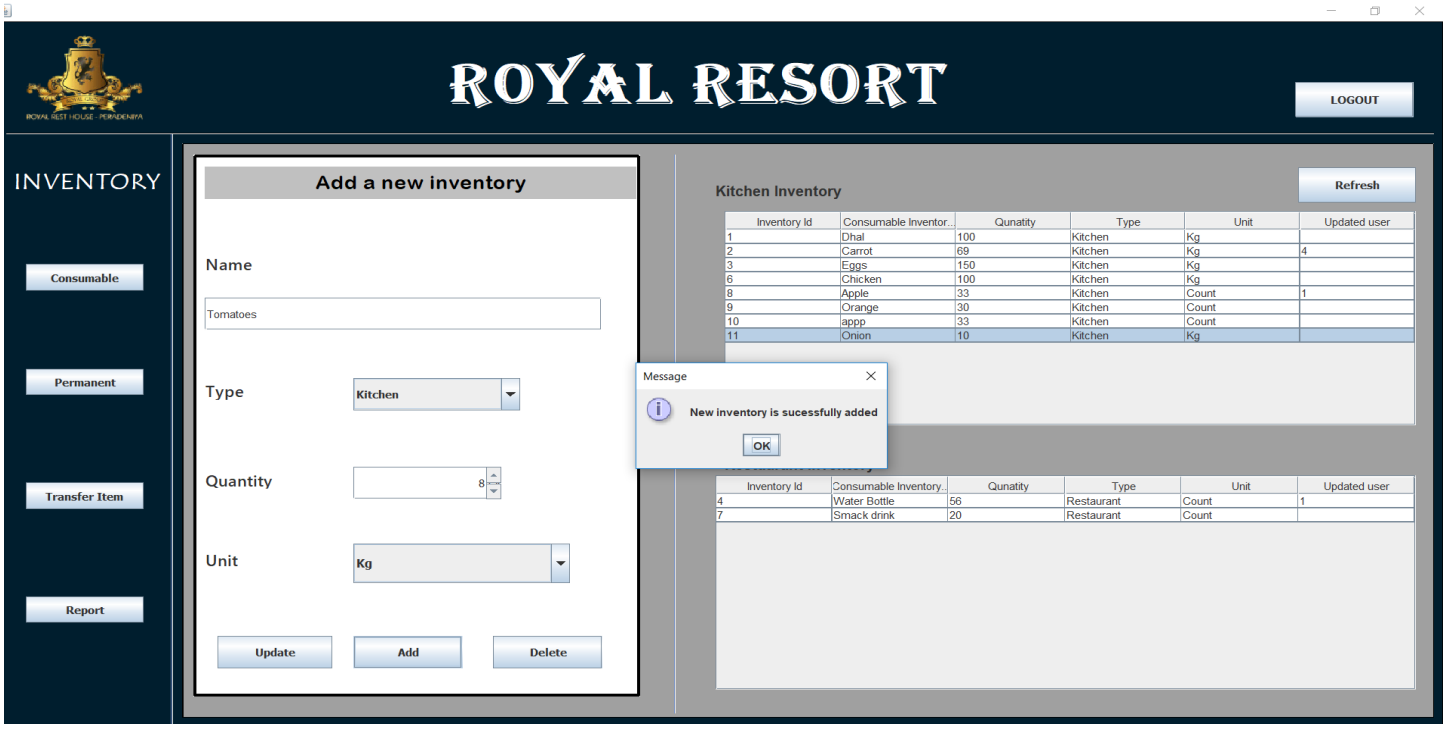


*Figure 48:add consumable inventory success test case*

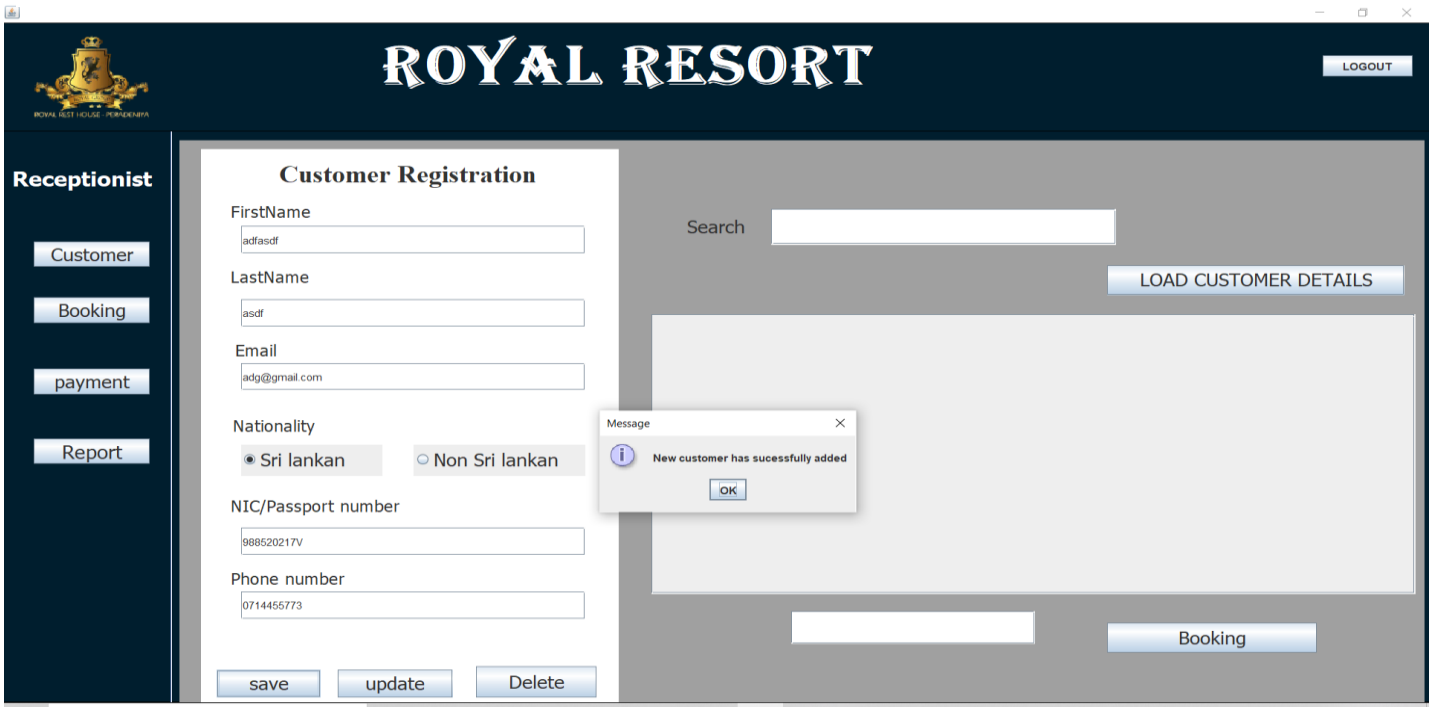


*Figure 49:add customers test case*

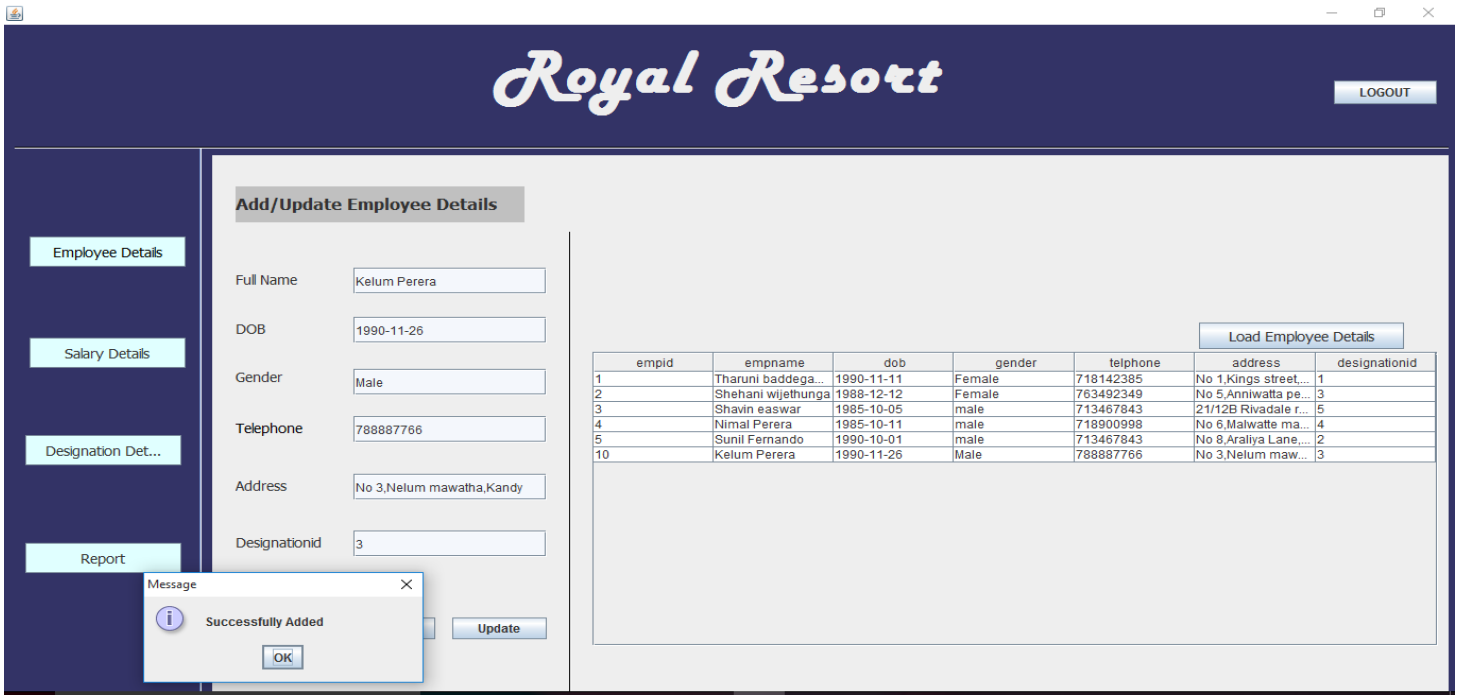


*Figure 50:add employees success test case*

# Appendix C: Selected code listing.

Figure 51 Illustrates the login function, where users are directed to their respective interfaces according to their user type.

```
public void actionPerformed(ActionEvent e) {
    String userName = textField.getText();
    String password = passwordField.getText();
    String type="Admin";
    try {
        Connection connection = (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/hotelmanagement","root", "1998Jayami
        PreparedStatement st = (PreparedStatement) connection
            .prepareStatement("Select username, password,usertype from user where username=? and password=?");

        st.setString(1, userName);
        st.setString(2, password);
        ResultSet rs = st.executeQuery();
        if (rs.next()) {
        String t =  rs.getString("usertype");

            if(t.equalsIgnoreCase("Admin") ) {
                  dispose();
            Rooms ah = new Rooms();
            ah.setTitle("Rooms");
            ah.setVisible(true);
            JOptionPane.showMessageDialog(btnNewButton, "You have successfully logged in");
            }
           else if(t.equalsIgnoreCase("Receptionist") ) {
                  dispose();
            hotel1 ah = new hotel1();
            ah.setTitle("Room Reservation");
            ah.setVisible(true);
            JOptionPane.showMessageDialog(btnNewButton, "You have successfully logged in");
            }
           else if(t.equalsIgnoreCase("Inventory manger") ) {
                  dispose();
                  Permanent ah = new Permanent();
            ah.setTitle("Inventory");
            ah.setVisible(true);
            JOptionPane.showMessageDialog(btnNewButton, "You have successfully logged in");
            }
           else if(t.equalsIgnoreCase("Kitchen manger") ) {
                  dispose();
                  MealJFrame  ah = new MealJFrame();
            ah.setTitle("Meal");
```

*Figure 51:special coding Login*

Figure 52 Illustrate the inventory function, where consumable inventory transfer from inventory

```
    JButton btnconupdate_1 = new JButton("Transfer");
    btnconupdate_1.setFont(new Font("Tahoma", Font.BOLD, 12));
    btnconupdate_1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {


            String userid= txtunit.getText();;
            int tqty = (int) spinnerTQ.getValue();
            String driverName = "com.mysql.cj.jdbc.Driver";

            int selectedRow = table_3.getSelectedRow();
            DefaultTableModel model = (DefaultTableModel) table_3.getModel();
            String id =(model.getValueAt(selectedRow, 0).toString());
            int qty = (int) (model.getValueAt(selectedRow, 2));



            try {

                Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/hotelmanagement?useSSL=false", "root", "root");

                int quantity= (qty - tqty);

                String query = "UPDATE consumable set quantity='"+quantity+"', userid='"+userid+"' WHERE inventoryid="+id;
                PreparedStatement pst1 = connection.prepareStatement(query);
                int x = pst1.executeUpdate(query);


                if (x == 0) {
                    JOptionPane.showMessageDialog(btnconupdate, "This is alredy exist");
                } else {
                    JOptionPane.showMessageDialog(btnconupdate,
                       "Consumable inventory successfully tranfered");
                }
                connection.close();
            } catch (Exception exception) {
                exception.printStackTrace();
```

*Figure 52:special coding transfer inventory*

Figure 53 Illustrate the function to calculate the total price of a particular booking. It will be calculated by multiplying meal price with number of days customer stayed and adding that value with room price multiplied with number of day's customer stayed in the hotel. Number of days the customer stayed will be calculated by subtracting check in date from the checkout date.

```
JButton load_2 = new JButton("LOAD PAYMENT DETAILS");
load_2.setFont(new Font("Tahoma", Font.PLAIN, 21));
load_2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
              Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/hotelmanagement", "root", "root");
            String query="select cid,bookingid,checkindate,checkoutdate,(checkoutdate-checkindate) as Noofdays,rno,roomprice,Mid,price,(checkoutdate-checkindate)*roomprice"
                    + "+(checkoutdate-checkindate)*price as subtotal "
                    + "from booking b,room r,mealplan m where r.roomno=b.rno and m.mealid=b.Mid";
            PreparedStatement pt=connection.prepareStatement(query);
            ResultSet rs=pt.executeQuery();

            table_2.setModel(DbUtils.resultSetToTableModel(rs));



        }
        catch(Exception ex)
        {
            ex.printStackTrace();
        }
    }

});
```

*Figure 53:special coding Room Booking*

Figure 54 Illustrates the implementation of search liquor by liquor name.

```
23
24      textFieldsearch = new JTextField();
25      textFieldsearch.addKeyListener(new KeyAdapter() {
26          @Override
27          public void keyReleased(KeyEvent e) {
28              try {
29
30                  Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/swing_bar?useSSL=false", "root", "123456");
31                  String query="select liquorid,liquorname,brand,manufacturedate,expiry,volume,price from liquor where liquorname=?";
32                  PreparedStatement pst=connection.prepareStatement(query);
33                  pst.setString(1, textFieldsearch.getText());
34                  ResultSet rs=pst.executeQuery();
35
36                  table.setModel(DbUtils.resultSetToTableModel(rs));
37
38
39
40              //  while(rs.next())
41              //  {
42
43              //}
44                  pst.close();
45              }
46              catch(Exception ex) {
47                  ex.printStackTrace();
48
49              }
50          }
51      });
52      textFieldsearch.setBounds(678, 345, 182, 23);
53      panel1.add(textFieldsearch);
54      textFieldsearch.setColumns(10);
55
Source  Design
```

*Figure 54:special coding Search Liquor*

Figure 55 Illustrate the calculate the total amount of a particular item by multiplying the unit price and quantity. Then that total will display in the field and as well as in the table. Then after adding some items it will calculate the sub total of all items in the order.

```
JSpinner txtqty = new JSpinner();
txtqty.addChangeListener(new ChangeListener() {
    public void stateChanged(ChangeEvent e) {

        int qty = Integer.parseInt(txtqty.getValue().toString());
        float price = Float.parseFloat(unitpriceField.getText());
        float total = qty*price;
        totalSub.setText(String.valueOf(total));
    }
});
txtqty.setBounds(154, 236, 166, 22);
panelOrder.add(txtqty);

JButton btnAddOrder = new JButton("Add");
btnAddOrder.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        DefaultTableModel model = new DefaultTableModel();
            model = (DefaultTableModel)amountTable.getModel();
            model.addRow(new Object[] {

                    textFieldOrder.getText(),
                    textFieldItemID.getText(),
                    textFieldItemName.getText(),
                    txtqty.getValue().toString(),
                    unitpriceField.getText(),
                    totalSub.getText(),
            });

            float sum = 0;
            for(int i =0; i<amountTable.getRowCount();i++) {

                sum = sum + Float.parseFloat(amountTable.getValueAt(i, 5).toString());
                totalField.setText(Float.toString(sum));
            }

            int foodid = Integer.parseInt(textFieldItemID.getText());
            int orderid = Integer.parseInt(textFieldOrder.getText());
            int qty = (int) txtqty.getValue();
            double fprice = Double.parseDouble(totalSub.getText());
```

*Figure 55:special coding calculate bill*

Figure 56 illustrates calculating the final amount of hall rent, service charges and guest count multiplied by the food package. All the above sections can be calculated separately and the user can get each sub-total before calculating the grand total.

```
1376            txt_invo_gcount = new JTextField();
1377            txt_invo_gcount.setColumns(10);
1378            txt_invo_gcount.setBounds(113, 50, 96, 19);
1379            panel.add(txt_invo_gcount);
1380
1381            JButton btn_sub_pcal = new JButton("Calculate");
1382            btn_sub_pcal.addActionListener(new ActionListener() {
1383                public void actionPerformed(ActionEvent e) {
1384
1385                double three = Double.parseDouble(txt_invo_pprice.getText());
1386
1387                    int four = Integer.parseInt(txt_invo_gcount.getText());
1388                    Double d = new Double(four);
1389
1390                    String answer2 = String.valueOf(three*four);
1391
1392                    txt_sub_pcal.setText(answer2);
1393                }
1394            });
1395            btn_sub_pcal.setBounds(111, 93, 98, 21);
1396            panel.add(btn_sub_pcal);
1397
1398            JLabel lblNewLabel_3_1_2_1 = new JLabel("Sub Total");
```

*Figure 56:special coding hall booking*

```
1433
1434        JButton btn_sub_hcal = new JButton("Calculate");
1435        btn_sub_hcal.addActionListener(new ActionListener() {
1436            public void actionPerformed(ActionEvent e) {
1437
1438                double one  = Double.parseDouble(txt_invo_rent.getText());
1439                double two =  Double.parseDouble(txt_invo_charge.getText());
1440
1441                String answer = String.valueOf(one+two);
1442
1443                txt_sub_hcal.setText(answer);
1444            }
1445        });
1446        btn_sub_hcal.setBounds(111, 93, 98, 21);
1447        panel_2.add(btn_sub_hcal);
1448
1449        JLabel lblNewLabel_3_1_2_1_1 = new JLabel("Sub Total");
1450        lblNewLabel_3_1_2_1_1.setFont(new Font("Tahoma", Font.PLAIN, 14));
1451        lblNewLabel_3_1_2_1_1.setBounds(20, 426, 70, 13);
1452        txt_fun_invoid.add(lblNewLabel_3_1_2_1_1);
```

*Figure 57:special coding hall booking*

```
JButton btn_gtot = new JButton("Grand Tot");
btn_gtot.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        double tot =  Double.parseDouble(txt_sub_hcal.getText());
        double tot2 =  Double.parseDouble(txt_sub_pcal.getText());

        String answerfinal = String.valueOf(tot+tot2);

        invo_grand.setText(answerfinal);

    }
});
btn_gtot.setBounds(289, 417, 98, 21);
txt_fun_invoid.add(btn_gtot);

JLabel lblNewLabel_3_1_2_1_1_1 = new JLabel("Grand Total");
lblNewLabel_3_1_2_1_1_1.setFont(new Font("Tahoma", Font.PLAIN, 14));
lblNewLabel_3_1_2_1_1_1.setBounds(20, 461, 85, 19);
txt_fun_invoid.add(lblNewLabel_3_1_2_1_1_1);
```

*Figure 58:special coding hall booking*