**PAF-Karachi Institute of Economics & Technology**

**(The Center of Excellence)**

**College of Computing and Information Sciences**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Sno.*** | ***Stud. ID*** | ***Names*** | ***Course Name/CID*** |
| 1 | 9670 | Zareen Gul | Software Engineering/106269 |
| 2 | 9825 | Kehkashan Akram |
| 3 | 9876 | Fabeha Naqvi |
| 4 | 9877 | Tania Ayaz(Group leader) |

**Table of Contents**

1. Introduction

2. The General Description

3. Specific Requirements

4. Supporting Information

1. Introduction
   1. Purpose

This document describes the software requirements for the Hotel Management System.

* 1. Scope In

This system will be designed to provide the electronic version of hotel reservations system. The system will be user-friendly graphical interface and will be more cost effective compared to the current non-electronic systems.

The objectives of this development are:

* + To provide existing clerks with a new environment in which to make reservations for hotel when they are travelling.
  + To provide an avenue for customers to get their rooms in a more convenient way
  + To regain control of the hotel room sales to avoid scalping and overselling of rooms.
  + To implement a prototype of a scaled down version of the final system to test the solution and further develop requirements.
  + To collect statistics in a more efficient manner for future hotel development and construction.

1.3 Scope Out

PP - Project Plan

SDD - Software Design Description

SRS - Software Requirement Specification

SDS – Software Design Specification

SPMP - Software Project Management Plan

GUI – Graphical User Interface

QAM – Quality Assurance Manager

PDM – Project Development Manager

PMP – Project Management Professional

TBD – To be determined

UML – Unified Modeling Language

1.4 References

Nil

1.5 Overview

Chapter 2 of the SRS is a brief description of the characteristics of the software to be built, its functions, its users, its constraints and its dependencies.

Chapter 3 is about specific requirements, such as functional requirements, external interface requirements, performance requirements, and also design constraints and quality characteristics.

Finally, chapter 4 includes all the supporting information, such as the Table of Contents, the Appendices, and the Index.

1. The General Description:
   1. Product Perspective

The Hotel Management System diagram showing the overview of the system’s module and the relationship of the systems to external interfaces is presented here

Figure 2.1 Overview/ Architecture diagram of the HMS



HMS

Customer

Operatore

Admine

Servere

Database

Interface

PC

Terminalinale

Cell Phonee

Function of System Component:

Database:

* Stores data
* Creates reports
* Provides access to data
* Update information
* Delete unnecessary information

Server:

* Provides access to database
* Authentication users
* Process reservation
* Perform backups
* Produce reports

External Interface:

Terminal:

Users use terminals to access the server

Customer and operator use terminals to reserve the rooms and to get information about the available rooms on particular days.

Hotel administration may use terminals to see the reports generated by the database software.

Personal Computers

Users (customer ,operator, and hotel administration) may use personal computers to obtain a remote access to the server and the reservation database via the Internet.

Cell Phones

Serve as a medium of accessing the server and the reservation database.

Customer may use cell phones and the latest telecommunication technologies to access the server and the reservation database via Internet, or they may use cell phones to call operator to inquire about room information.

Computer Hardware and Peripheral Equipment to be used:

workstations, which include CPUs, monitors, keyboards, and mice

Printers

Network

Terminals

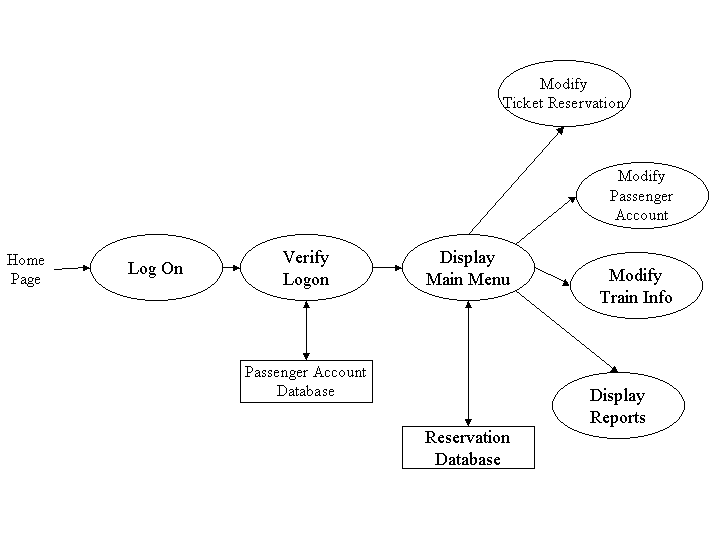
Cell phones to test connection to the server via remote access

* 1. Product Formation

This section provides summary of the functions that the software will perform

* + 1. Function Relationship

Figure 2.2 to 2.6 depict the relationships among the functions to be implemented by the system.  
  
Figure 2.2 HMS General Function Relationship/**Higher Level Usecase** Diagram



Module 1

Module 2

Module 3

Module 4

2.2.2 Function Description :

2.2.2.1 Log In function:

Description:

This function ensures that only authorized users gain access to the Reservation databases. An authorized user is a user who has an account on the system. Users include customer, hotel officials and operators. The user must type a valid username and password to gain access

2.2.2.1 Module 2

Description:

We create homepage for client .

Client can book the room on any page.

Client can contact us.

Client can easily pay bill by filling online form.

Client can review Gallery which has some details about the hotel.

Client can update his personal information Client can delete his personal information

Client can check the availability of room.

2.2.2.2 Module 3

Description:

Admin can login into the project and can enter new operators

Admin can review all the activities done by the operator and client and all the payment information and booked rooms

Admin can update all the details

Admin can delete all the details

Admin can search the customer record in the project.

2.2.2.1 Module 4

Description:

Operator can send email related to the booking.

Operator generate the bill.

Here logics are build

Here we will apply GUI

Operator can review the details of payment

GUI work reviewed here.

Operator can operate payment table once the payment done

Operator can delete the credit of the client after payment

Operator can search the unpaid bills.

2.2.2.1 Module 5

Description:

We will create a new database.

We will create tables.

We build relation between tables.

All the data will be reviewed

Data can be updated using queries

Data can be deleted using queries

Data can be searched by using queries

**2.3 User Characteristics**

The main users of the system will be the customer reserving rooms , the operator that process reservations for passengers, and the HMS administration that access the reports generated by the system. The users are not required to have knowledge in the computer field. The graphical interface provides an easy way of using the HMS system.

**2.4 User Characteristics**

The constraints for the project are:

* Only those users which enter correct username and password can only be entered.
* We will add more and more constraint during the development of project

**2.5 Assumptions and Dependencies** **or Business Logic**

The assumptions for the project are:

* Reservation can be made up to one month before a particular trip.
* Rooms are assigned during reservation.
* Phone reservation involves rooms being reserved within 24 hours after making the reservation. Otherwise, the reservation will be cancelled.
* No reservations can be made 48 hours prior to the trip. Rather, it will be done on a first come first serve basis from that point on.
* Network connection will always remain established

**3. Specific Requirements**

This section of the SRS contains design requirements for the Hotel Management System.

**3.1 Functional Requirements**

**3.1.1 Module 1 complete Log In Function**

1. ***Description:*** This function ensures that only authorized users gain access to the Reservation databases. An authorized user is a user who has an account on the system. Users include customer, hotel officials .The user must type a valid username and password to gain access.
2. ***Usage Scenario/ Use case Description/******Specification:***

|  |  |
| --- | --- |
| Description | Allows access to online HMS |
| Inputs | Username, password |
| Source | 1. User inputs username and password 2. Press Login Button |
| Alternate case |  |
| Outputs | Successful login; unsuccessful login |
| Destination | Dashboard |
| Pre-condition | Authorized User |
| Post Condition | No change to Customer Accounts Database |
| Side Effects | Failures and successful logins are sent to Reservation Database |

* + 1. **Module 2 complete CRUD Customer Side**

1. ***Description:***

Customer is also one of our actor who will visit our website to book room.

To book room

* First customer will visit all the pages
* Choose room according to his or her desire
* Once he has selected his room, he can book that room
* After clicking book a room option on our webpage
* Booking form will appear

There are following fields in booking

1.customer name

2.customer address

3. customer email

4.customer phone number

5.booking from

6.booking to

7.number of members

8. total amount

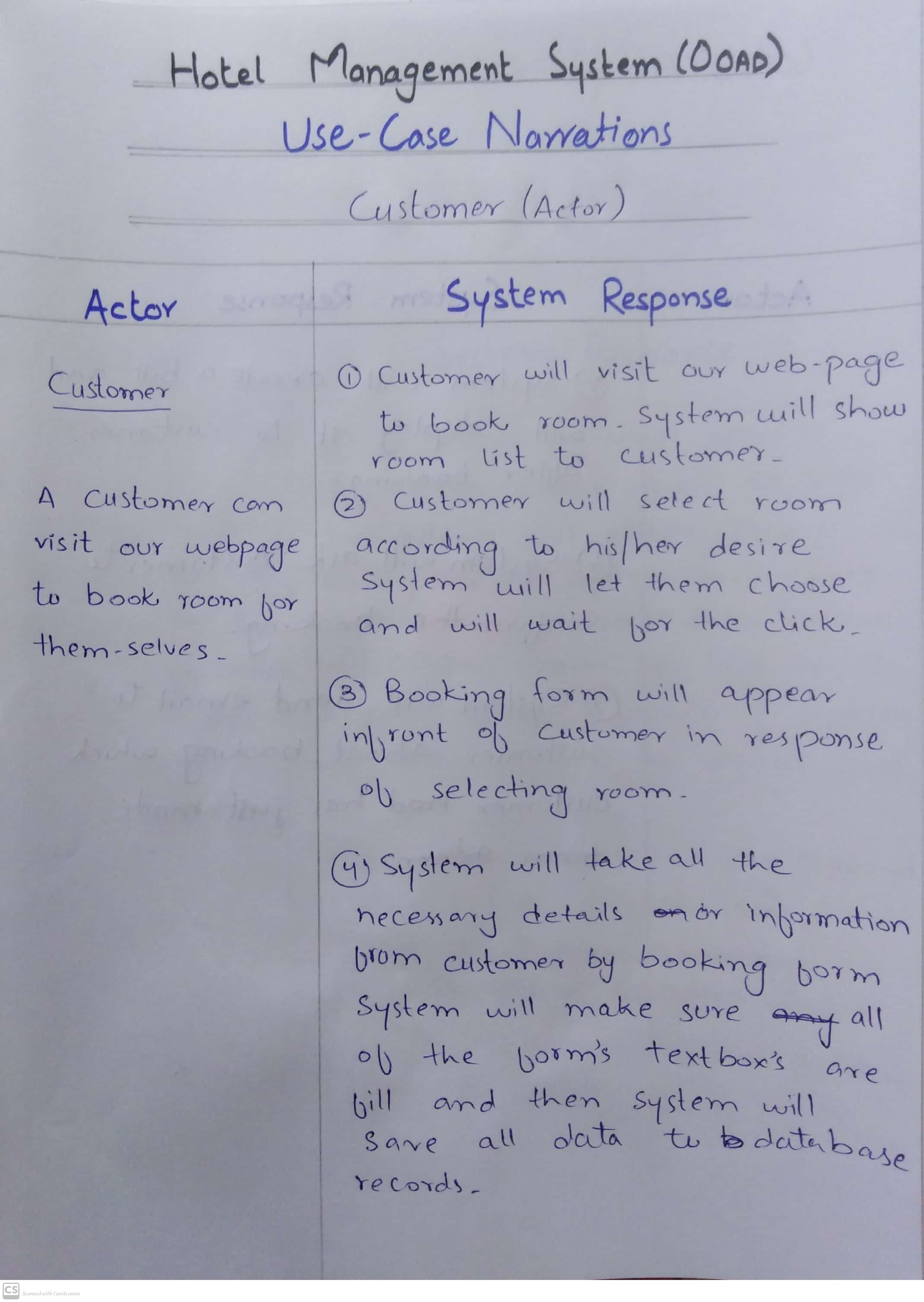
9. room number

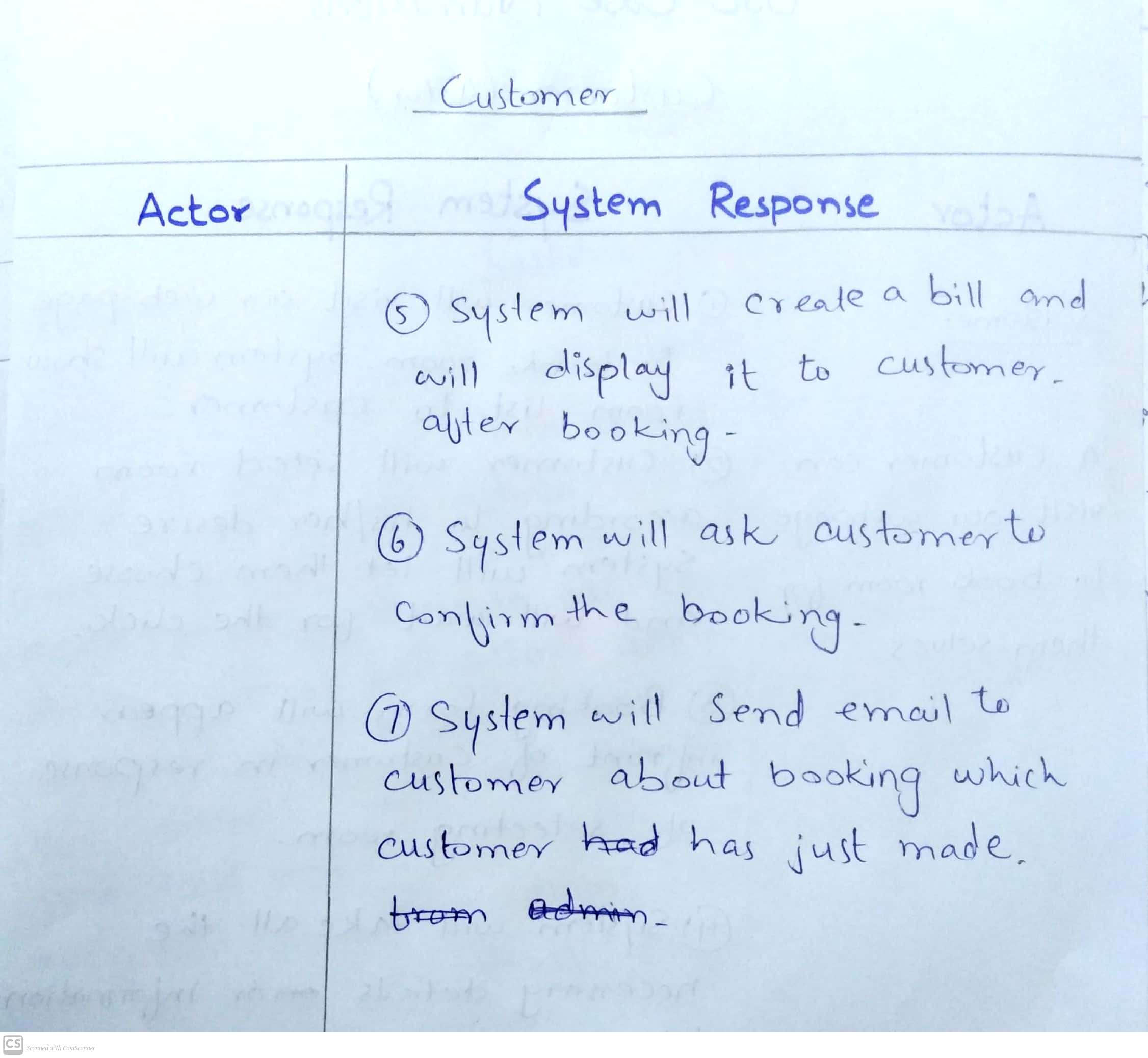
10. payment type

After booking customer will confirm booking

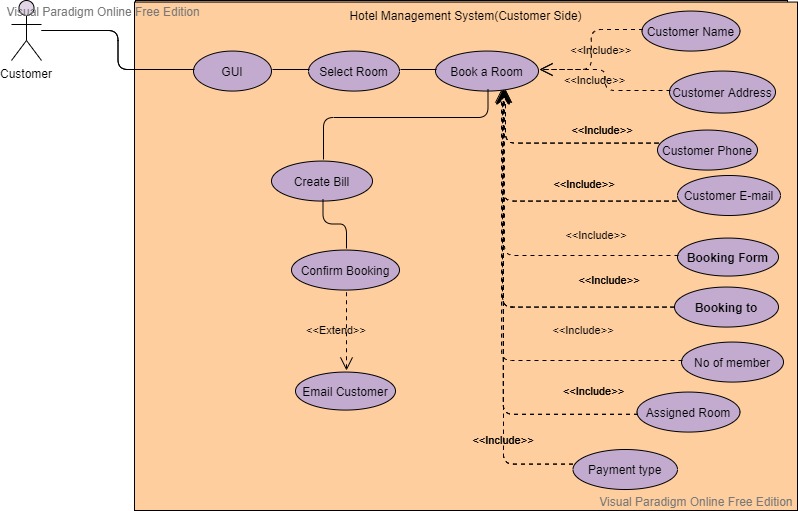
* Admin will show payment invoice to customer and customer will pay it by the payment type he has selected.
* And admin will email customer about his booking

1. ***Usage Scenario/ Use case Description/******Specification:***

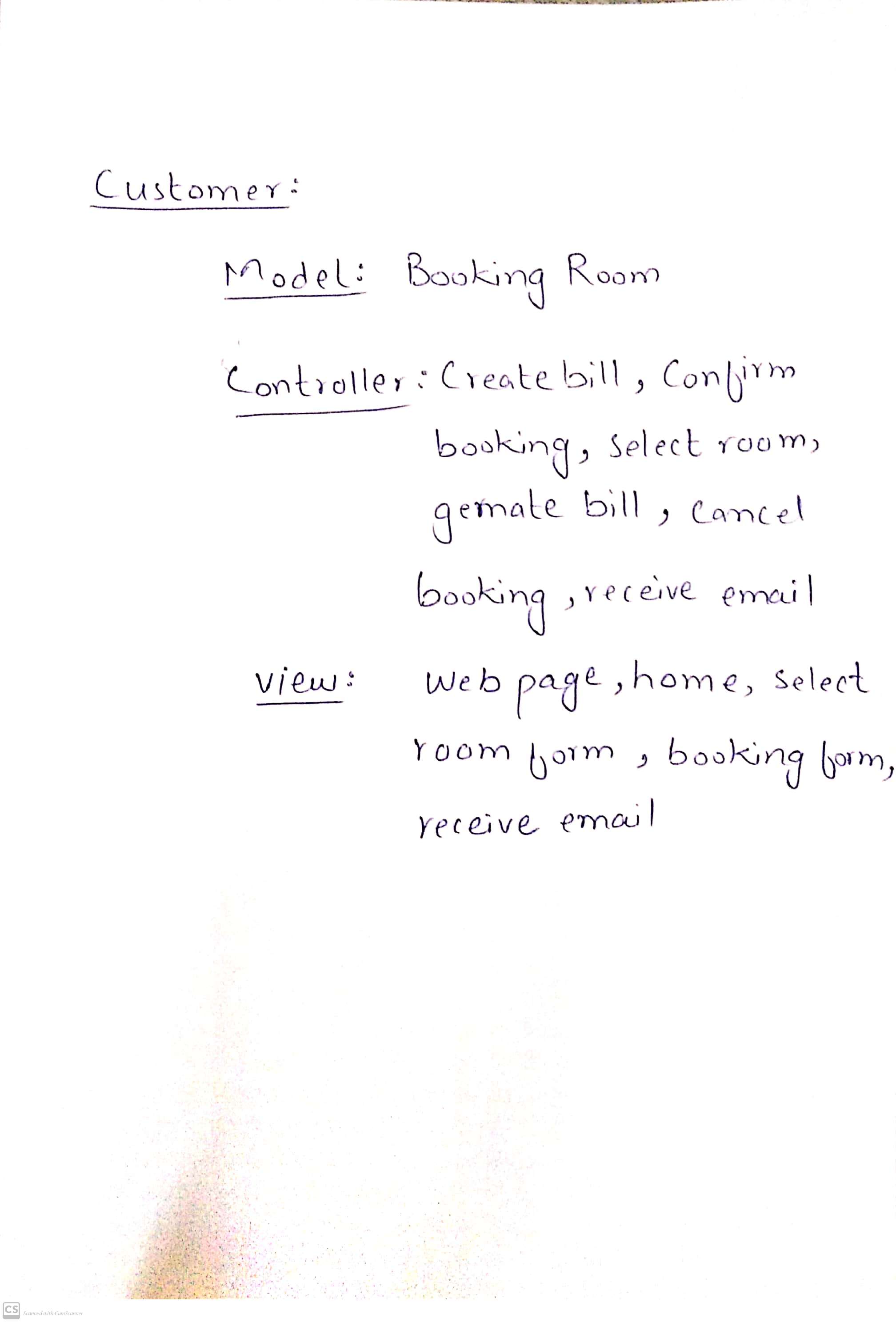
****

****

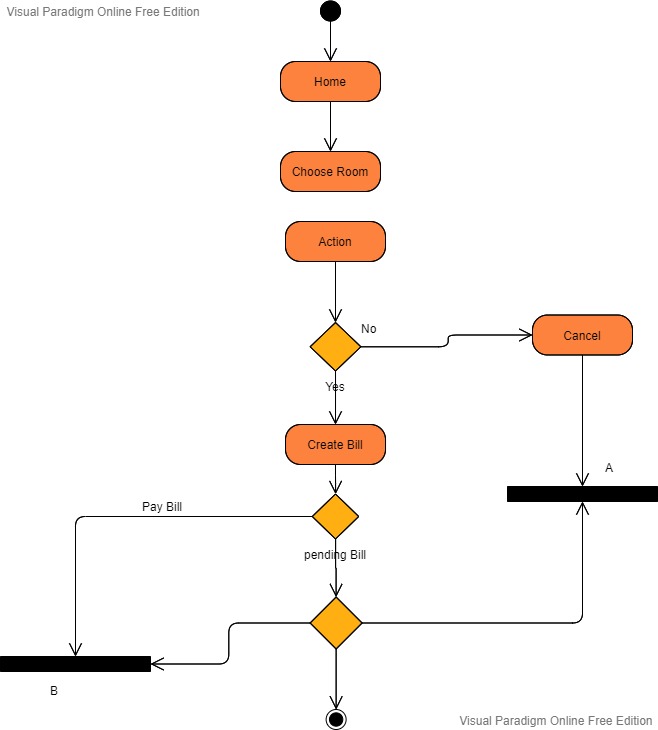
1. ***Use case Diagram:***

****

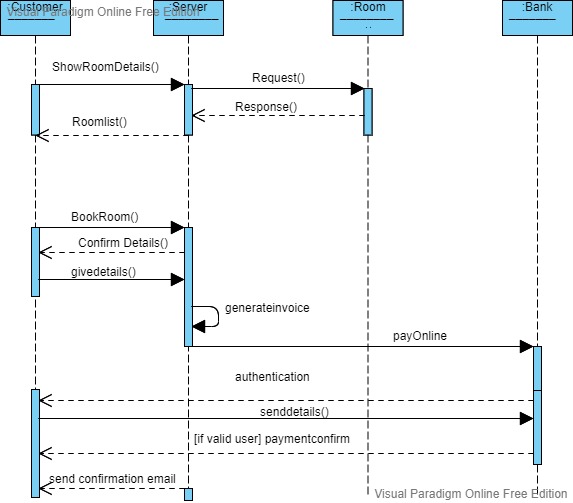
1. **Use case Realization**



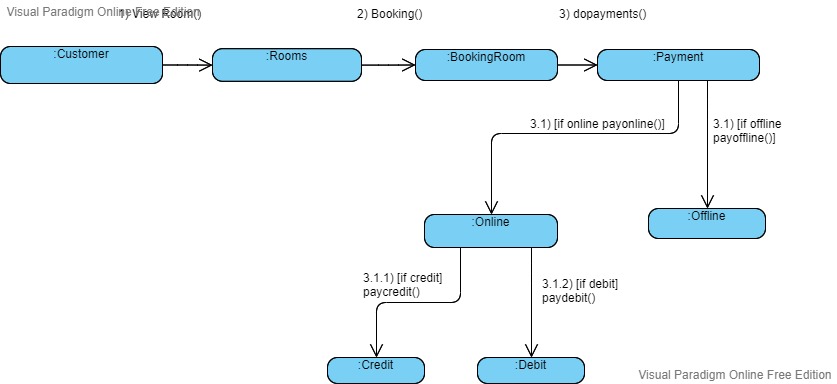
1. **Flow of event or Data Flow Diagram**

****

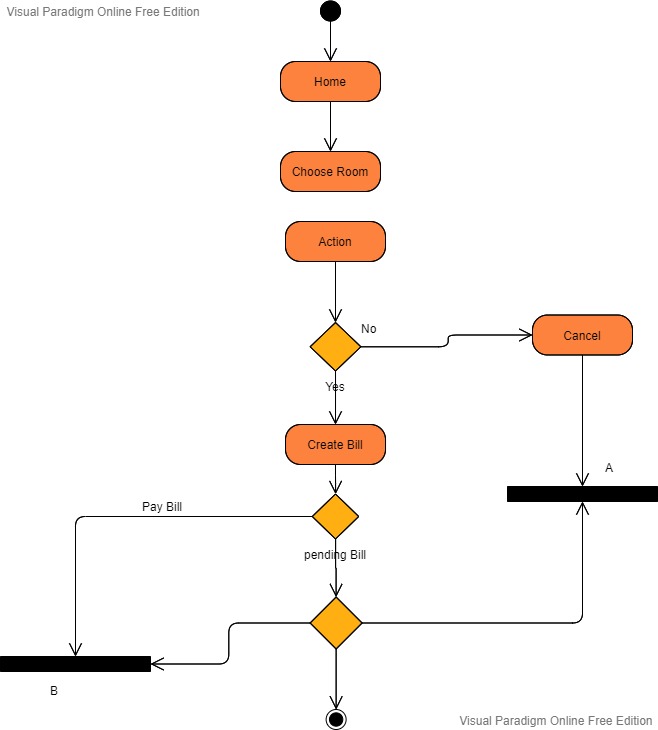
1. **Sequence Diagram**

****

1. Collaboration Diagram

****

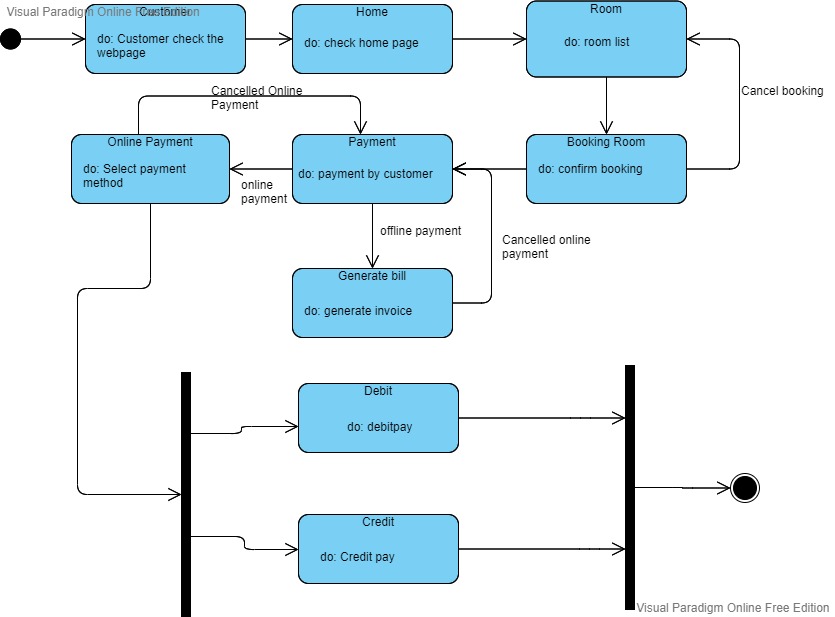
1. Activity Diagram

****

1. Class Diagram

****

1. State Chart Diagram

****

* + 1. **Module 3 complete CRUD Admin Side**

1. ***Description:***

* Admin is the actor who can access all the database and can do all the tasks. we have 2 type of admin here
* Old admin:

Old admin is the one who is already sign in to the dashboard. Admin can sign in by providing correct email and password to system. If there are more than one admin then all records of admins are saved in database.

* New admin:

New admin is the one who first have to sign up

then system will allow him to enter the database as he will sign in to the dashboard.

* After login admin will find him-self on dashboard where he will have 10 categories
* Admin can click any option there
* If admin choose Room-Info:

System will display a list of room info that includes

1. Room number
2. Is active
3. Booking status
4. Room name

Admin can do 4 functions

1. Create
2. Delete
3. Edit
4. Details

* If admin choose Room-Type info:

System will display a list of room type info that includes

1. Room name
2. Description
3. Room price
4. Room capacity

Admin can do 4 functions

1. Create
2. Delete
3. Edit
4. Details

* If admin choose Payment-Type info:

System will display a list of Payment type info that includes

1. Payment type

Admin can do 4 functions

1. Create
2. Delete
3. Edit
4. Details

* If admin choose Payment info:

System will display a list of Payment info that includes

1. Booking id
2. Payment amount
3. Is active
4. Payment type

Admin can do 4 functions

1. Create
2. Delete
3. Edit
4. Details

* If admin choose booking info:

System will display a list of booking info that includes

1.customer name

2.customer address

3. customer email

4.customer phone number

5.booking from

6.booking to

7.number of members

8. total amount

9. room number

10. payment type

Admin can do 4 functions

1.Create

2.Delete

3.Edit

4.Details

* If admin choose booking status info:

System will display a list of booking status info that includes

1. Booking status

Admin can do 4 functions

1.Create

2.Delete

3.Edit

4.Details

* If admin choose user level info:

System will display a list of user level info that includes

1.user type

Admin can do 4 functions

* + 1. Create
    2. Delete
    3. Edit
    4. Details
* If admin choose user info:

System will display a list of user info that includes

1. User name
2. email
3. password
4. user type

Admin can do 4 functions

1. Create
2. Delete
3. Edit
4. Details

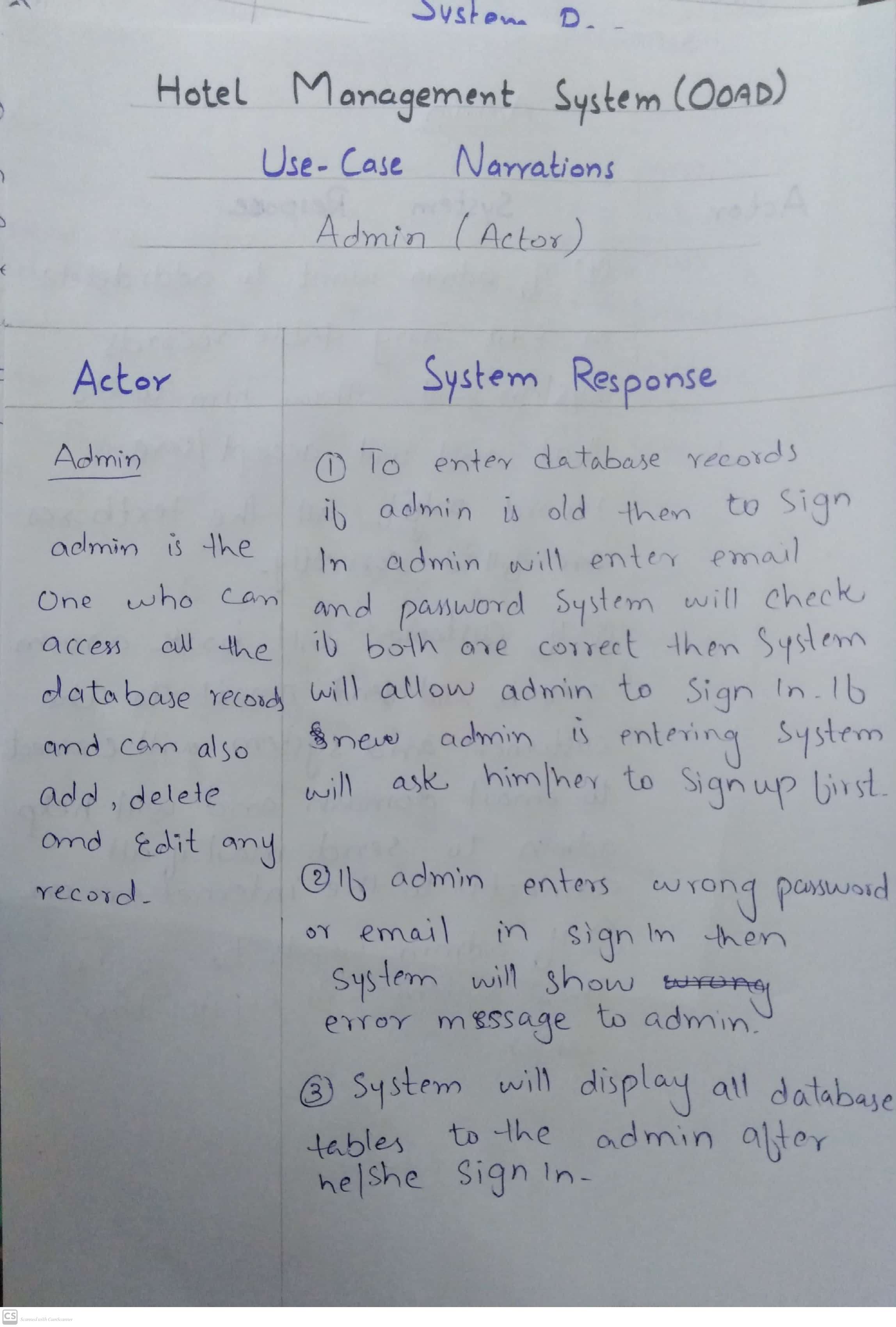
* If admin choose Email:

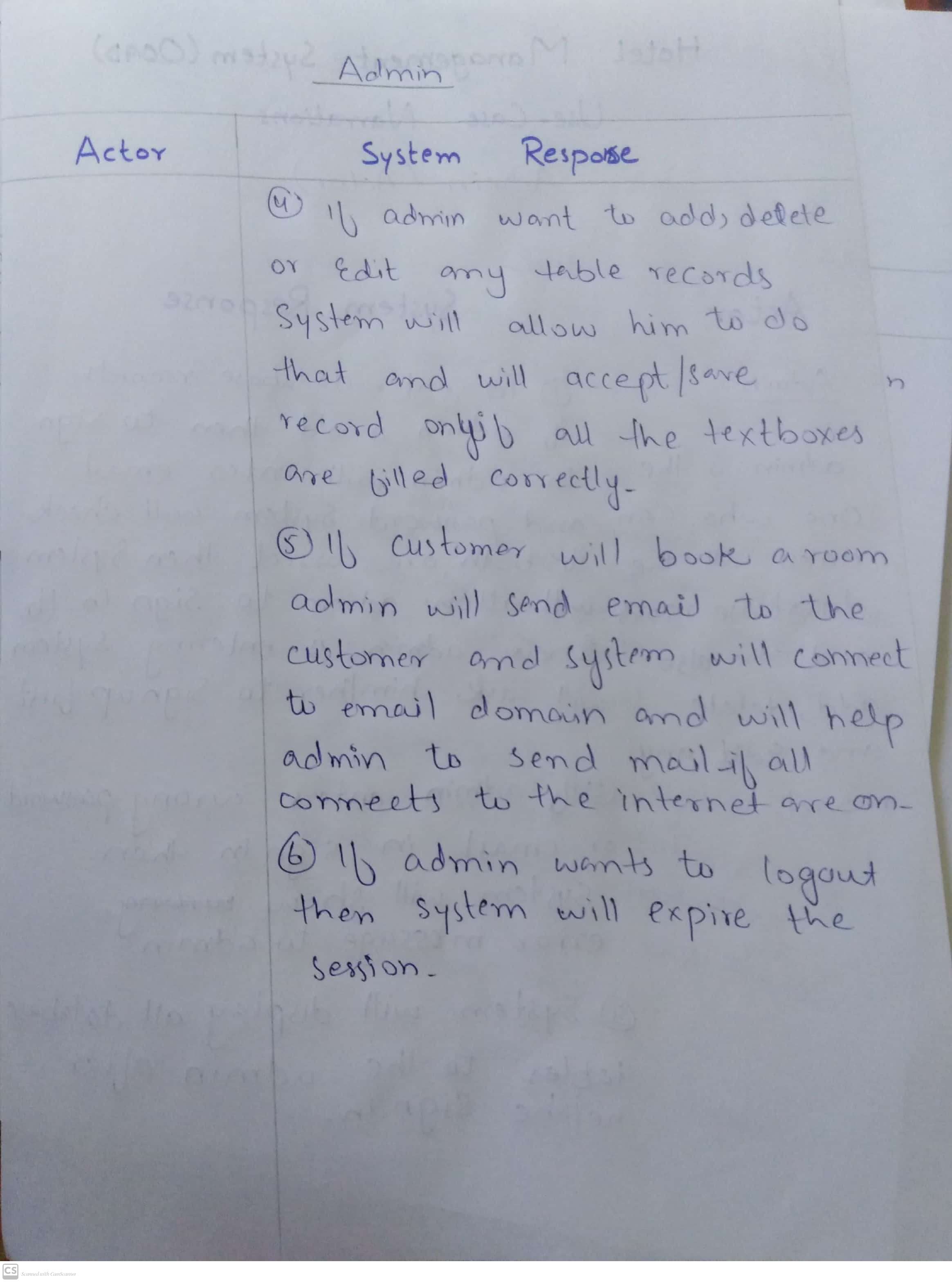
Admin is able to send email to the customer when booking is just made. he can inform customer by emailing him about the booking he has just made.

* If admin choose logout:

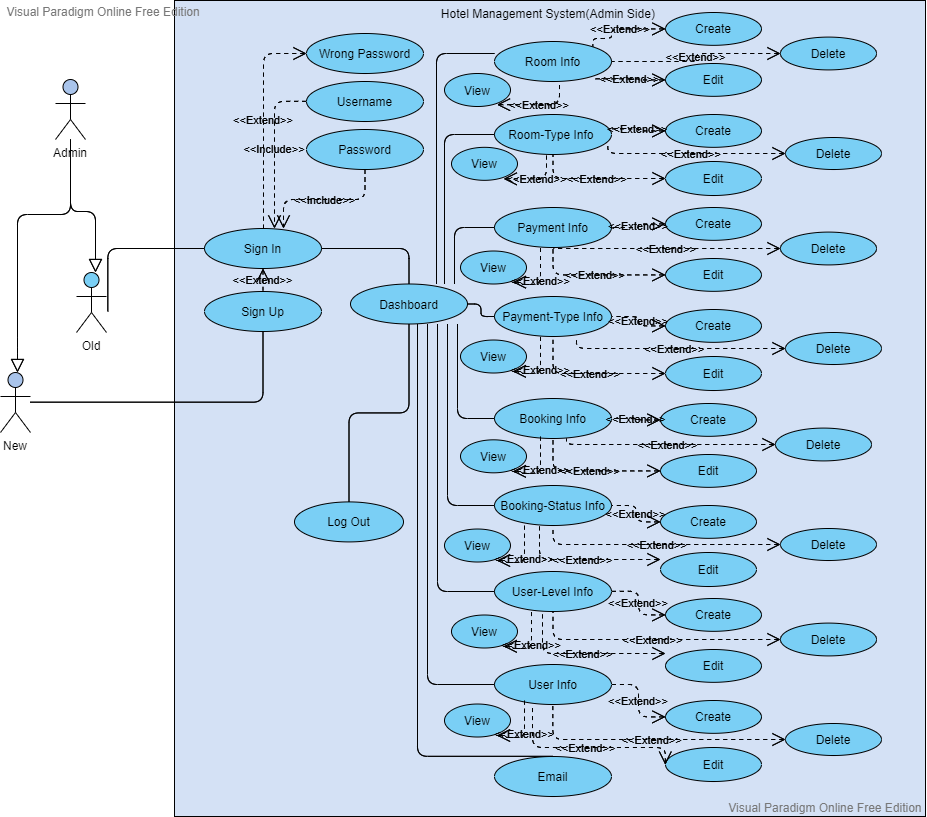
Admin can log out whenever he want.

1. ***Usage Scenario/ Use case Description/******Specification:***

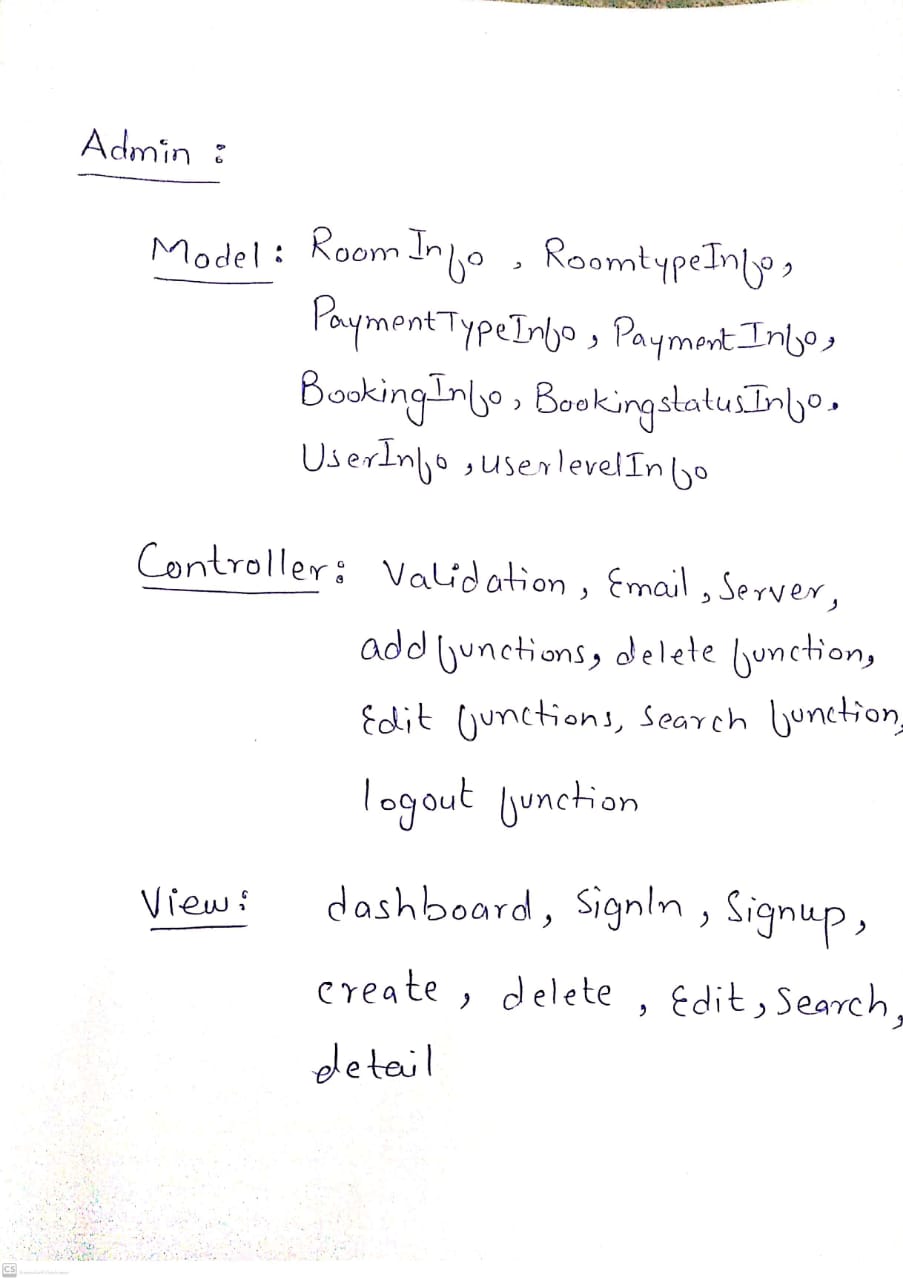
****

****

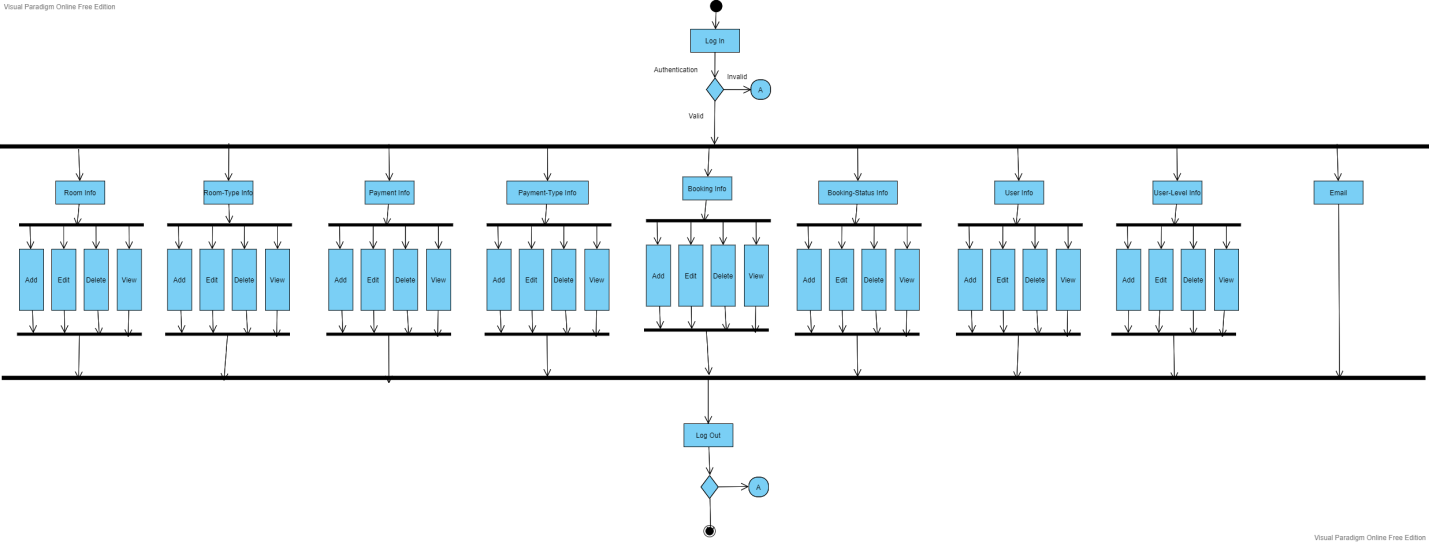
1. ***Detailed Use case Diagram***

******

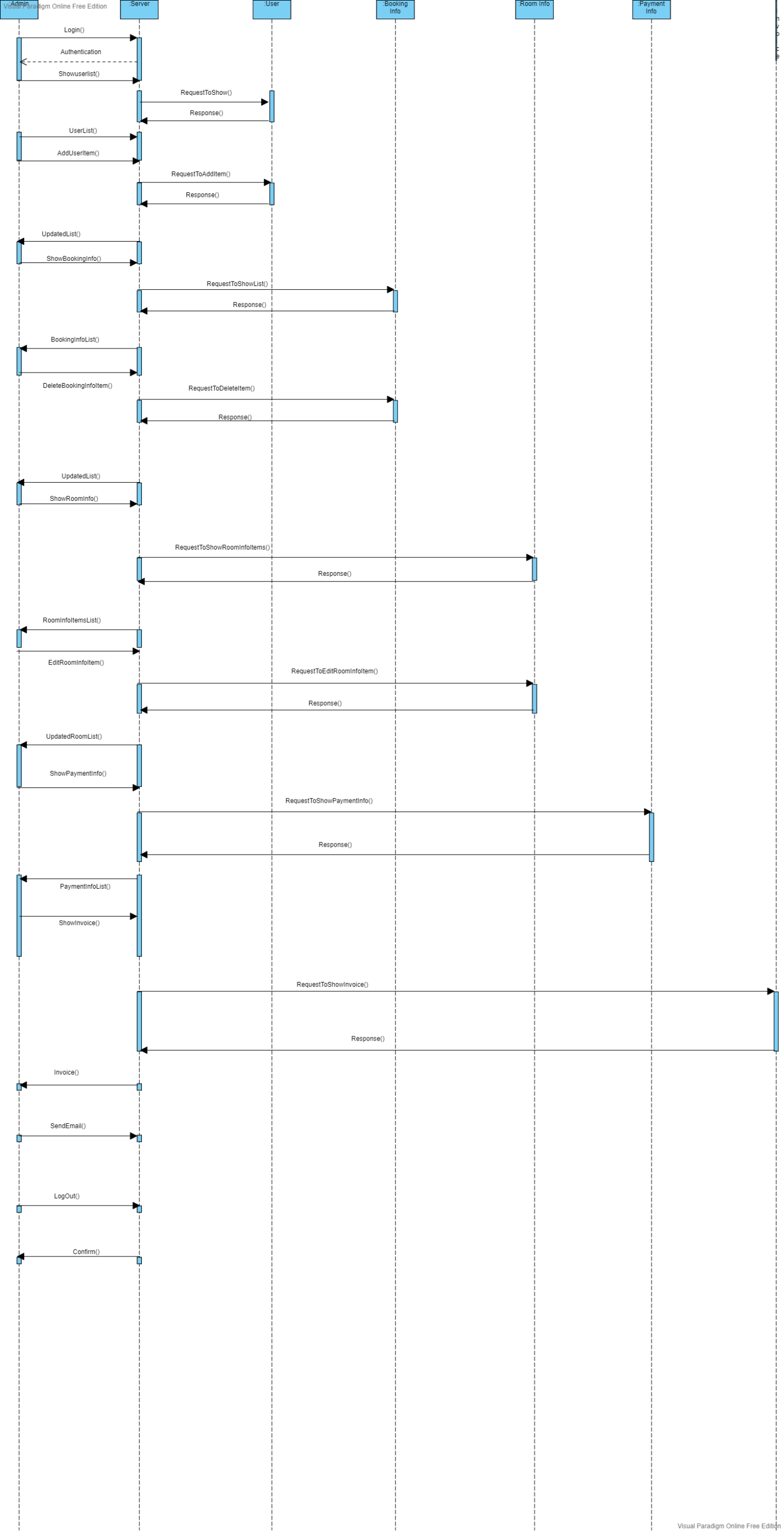
1. ***Use case Realization***

******

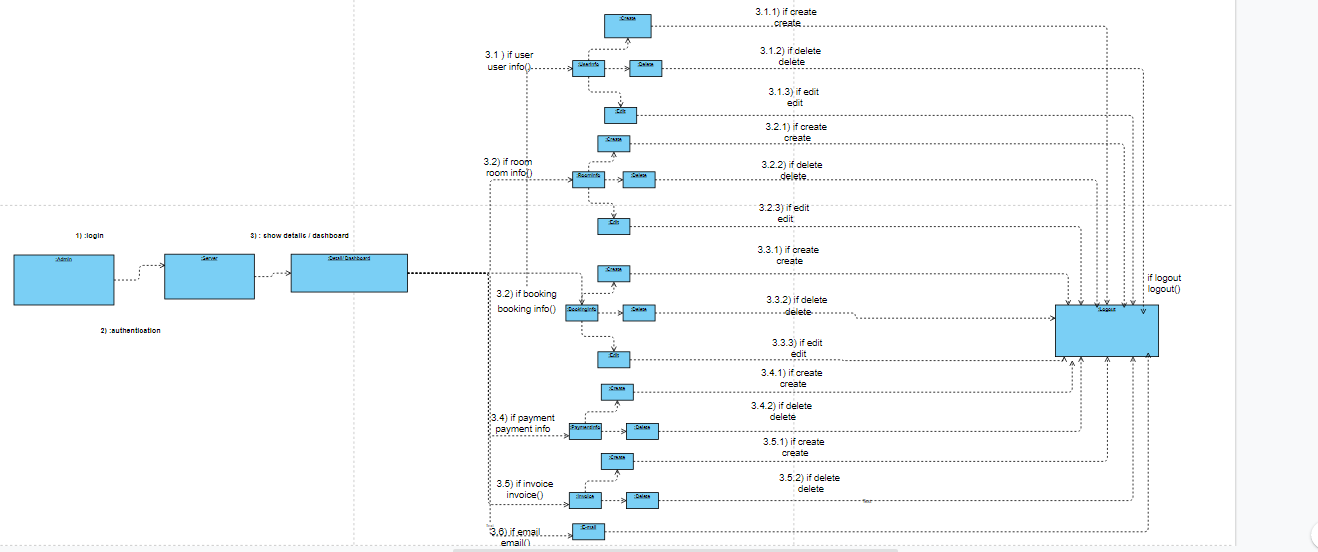
1. ***Flow of Event or Data Flow Diagram***

******

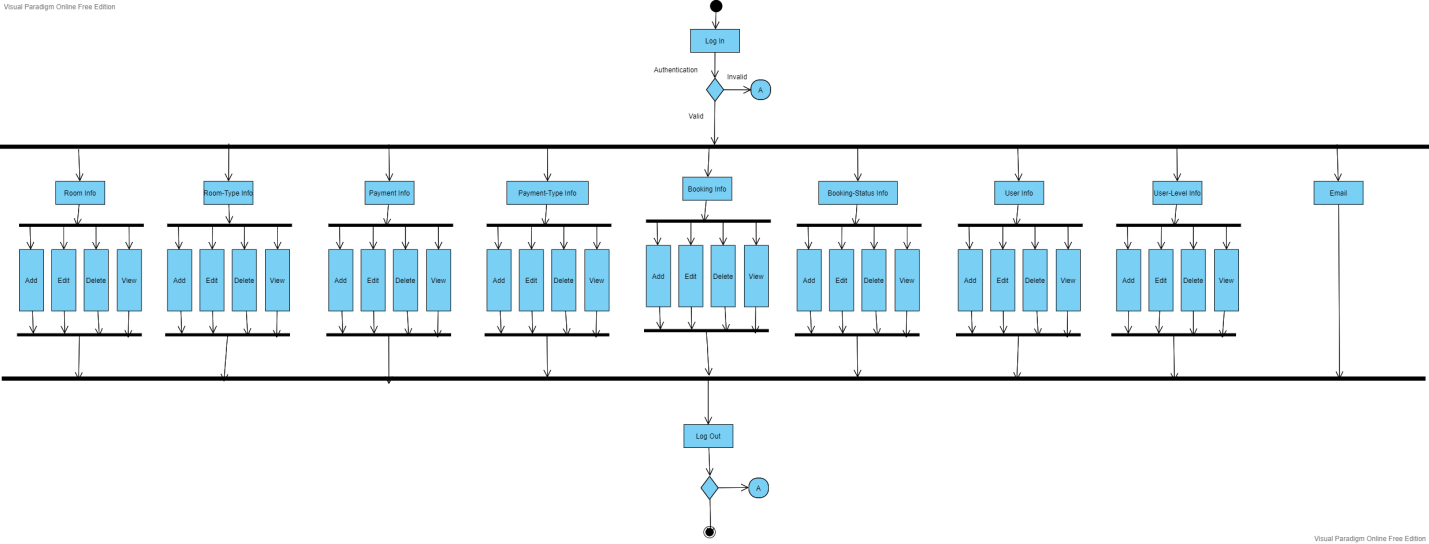
1. ***Sequence Diagram***



1. ***Collaboration Diagram***

******

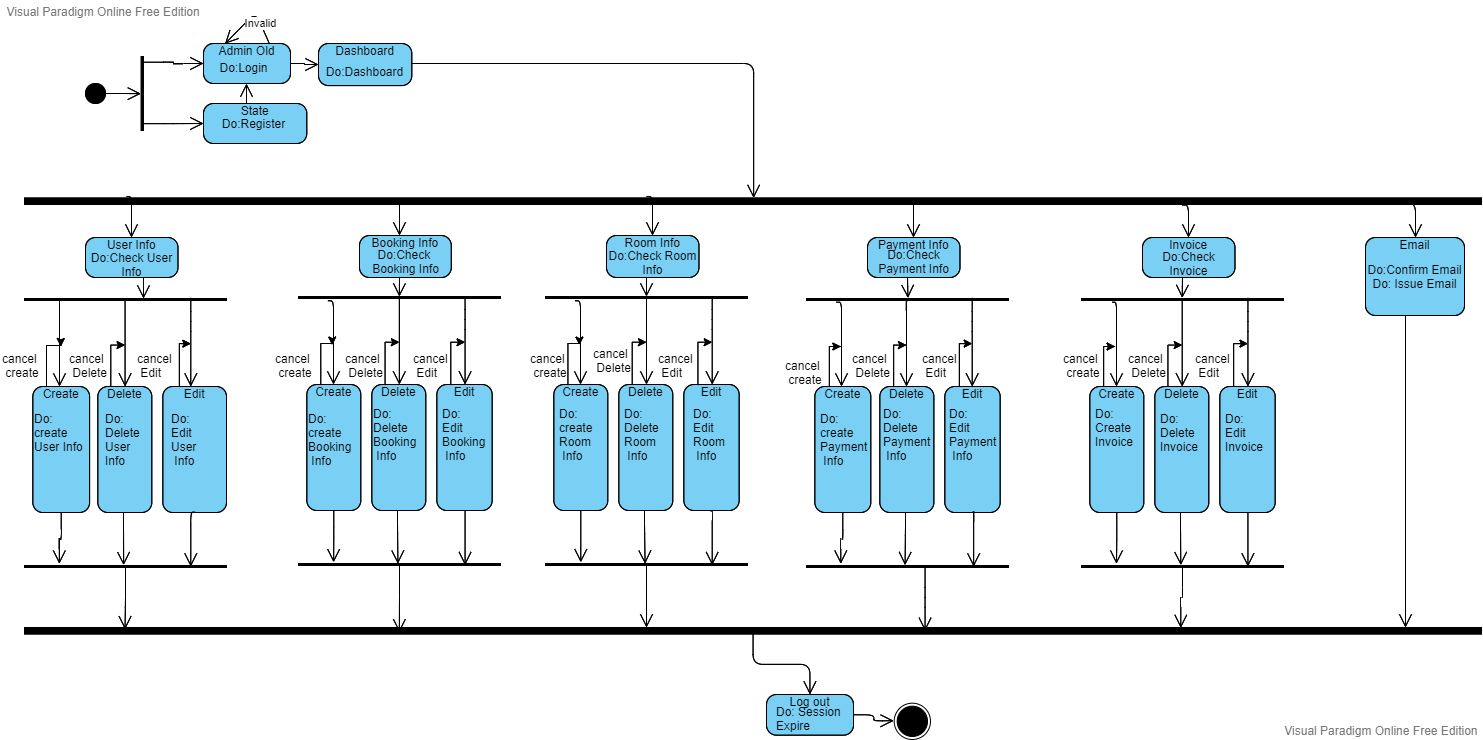
1. ***Activity Diagram***

******

1. ***Class Diagram***

******

1. ***State Chart Diagram***



* + 1. **Module 4 complete CRUD Operator Side**

1. ***Description:***

Operator is an actor who can excess limited database records. He can only view all the records he can’t add, delete and edit any records in any table. It has 2 categories

* Old operator:

Old operator is the one who is already sign in to the dashboard. operator can sign in by providing correct email and password to system. If there are more than one operator then all records of operator are saved in database.

* New operator:

New operator is the one who first have to sign up

then system will allow him to enter the database as he will sign in to the dashboard.

* After login operator will find him-self on dashboard where he will have 3 categories
* operator can click any option there
* If operator choose Room-Info:

System will display a list of room info that includes

1. Room number
2. Is active
3. Booking status
4. Room name

operator can do only one functions

1. View

* If operator choose booking info:

System will display a list of booking info that includes

1.customer name

2.customer address

3. customer email

4.customer phone number

5.booking from

6.booking to

7.number of members

8. total amount

9. room number

10. payment type

operator can do only one functions

1.view

* If operator choose Payment info:

System will display a list of Payment info that includes

1. Booking id
2. Payment amount
3. Is active
4. Payment type

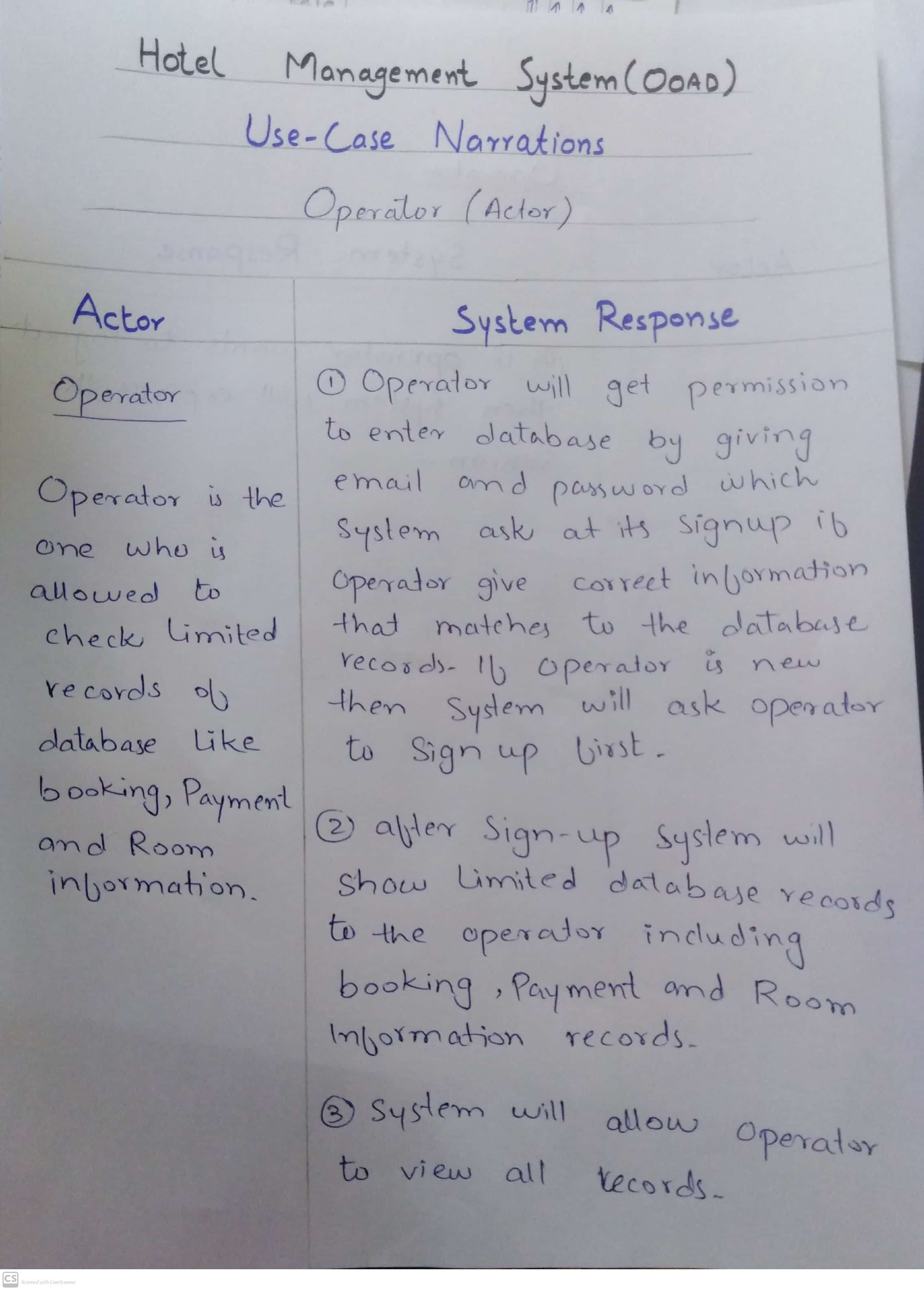
operator can do only one functions

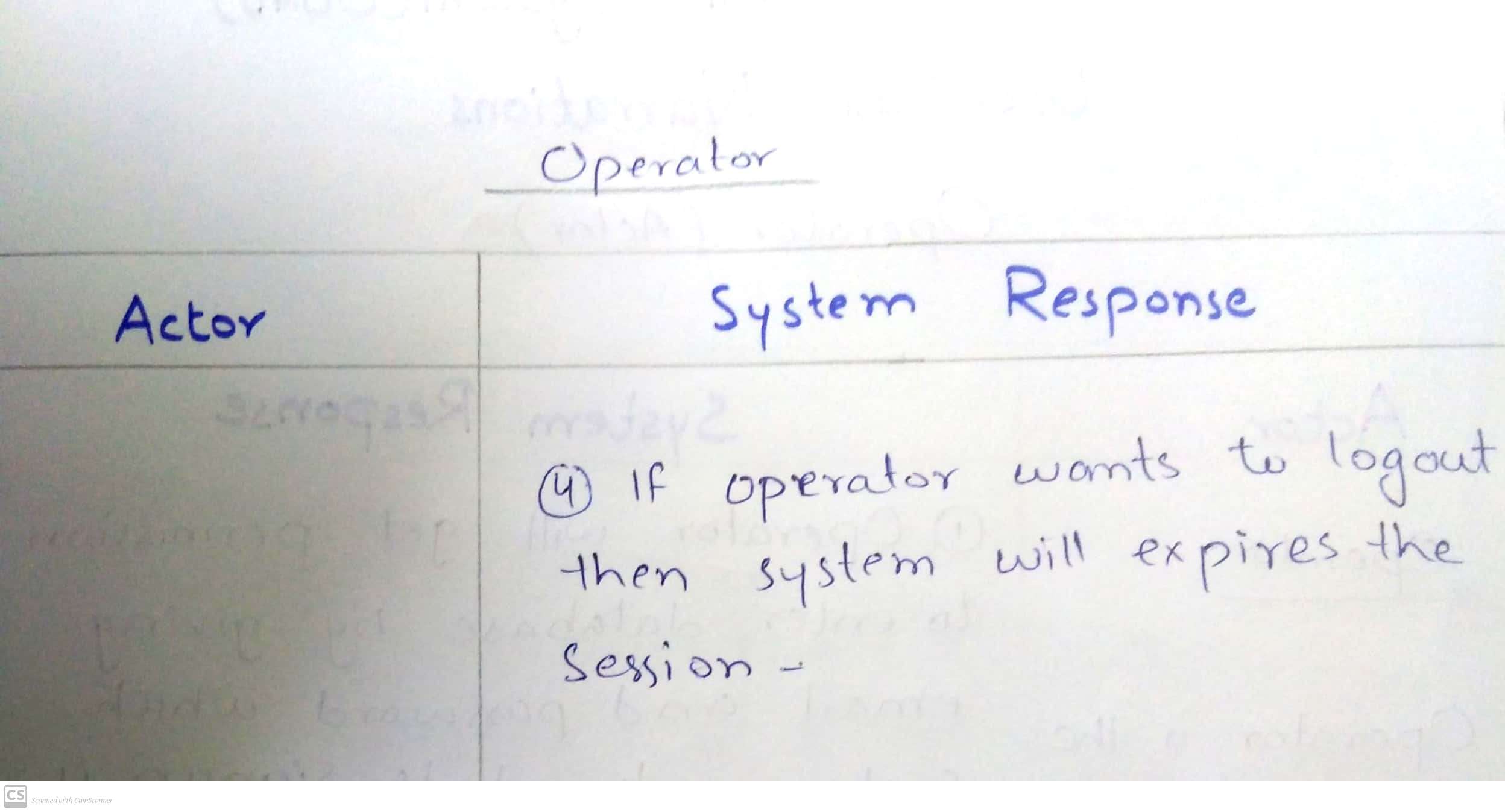
1.view

* If operator choose logout:

operator can log out whenever he wants.

1. ***Usage Scenario/ Use case Description/******Specification:***

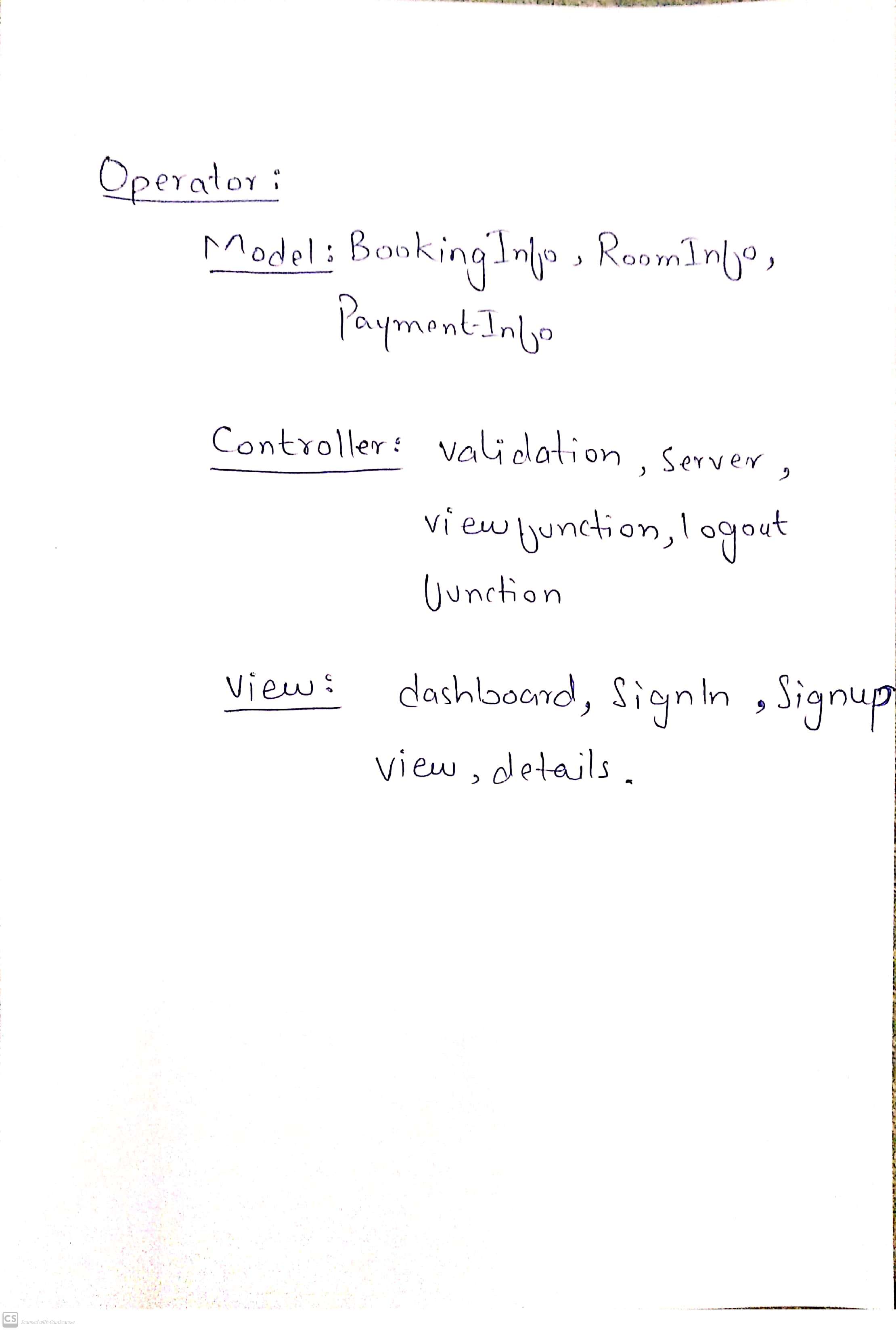
****

****

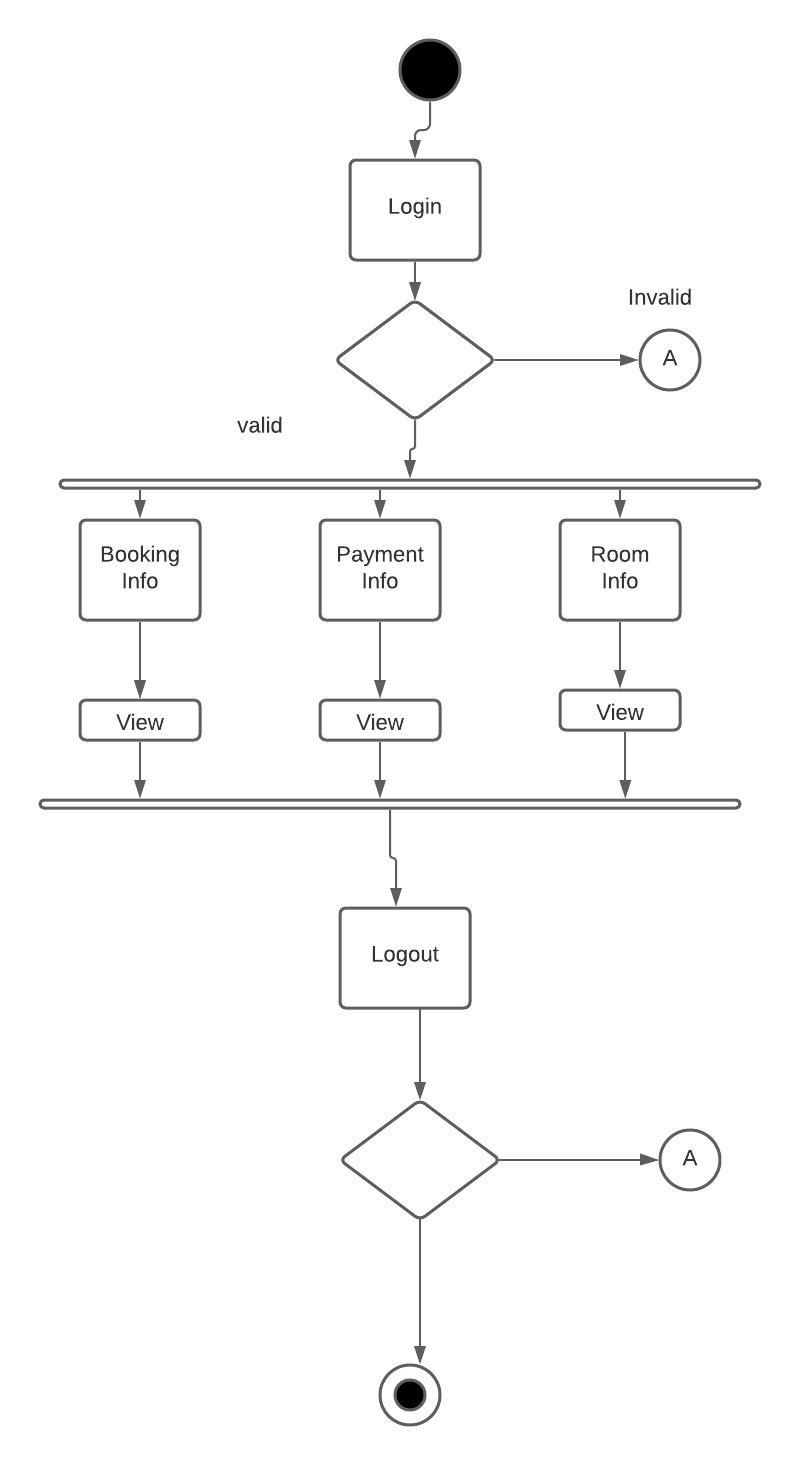
1. ***Detailed Use case Diagram***

**

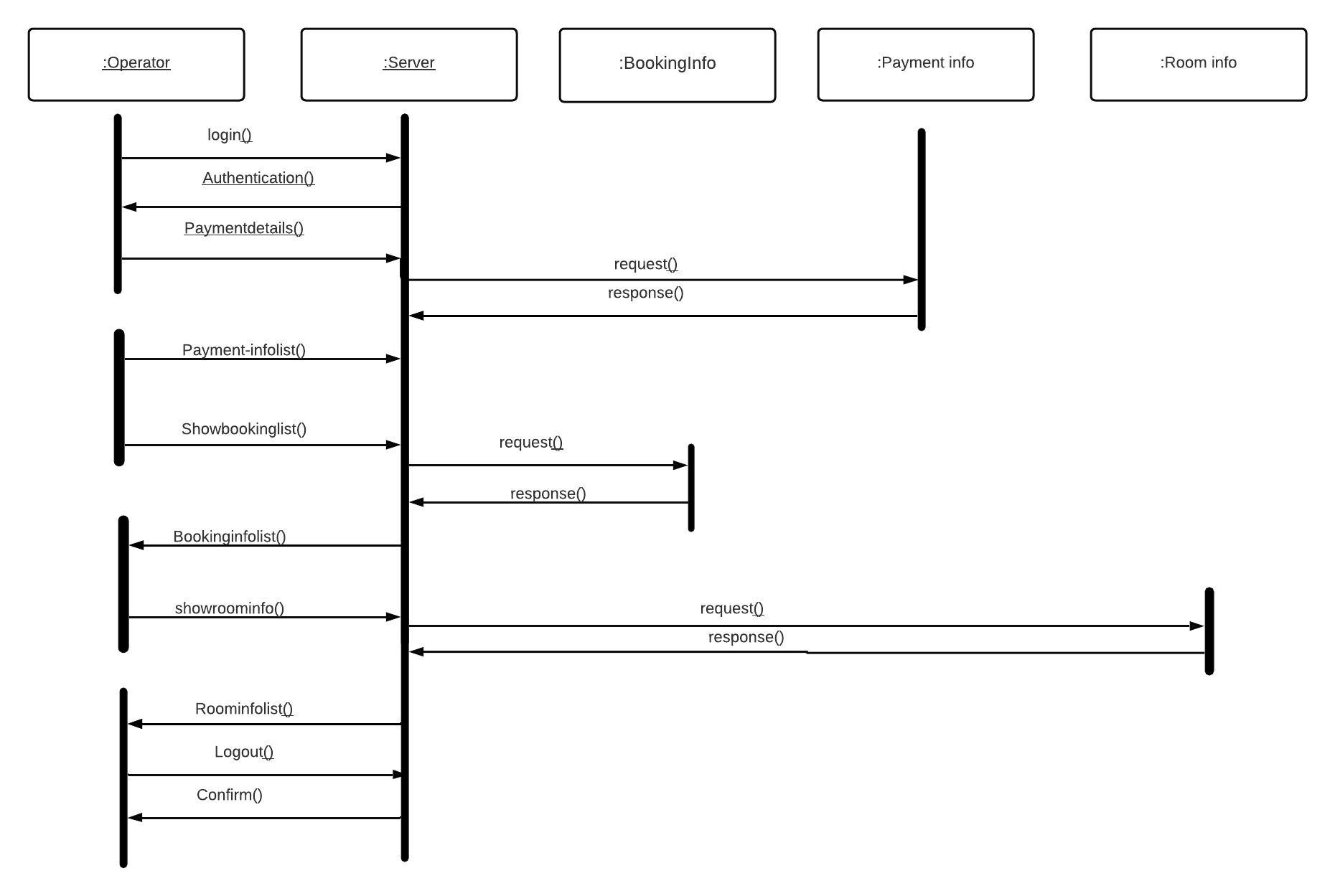
1. ***Use case Realization***

******

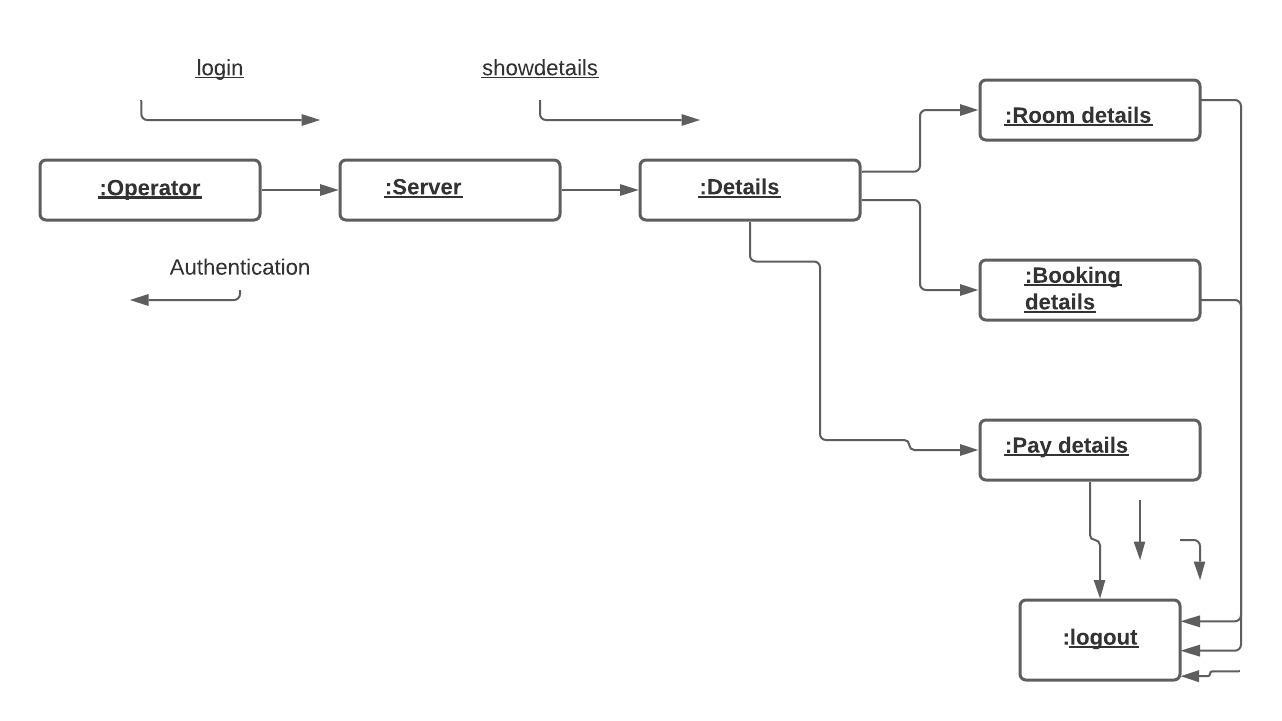
1. ***Flow of Event or Data Flow Diagram***

******

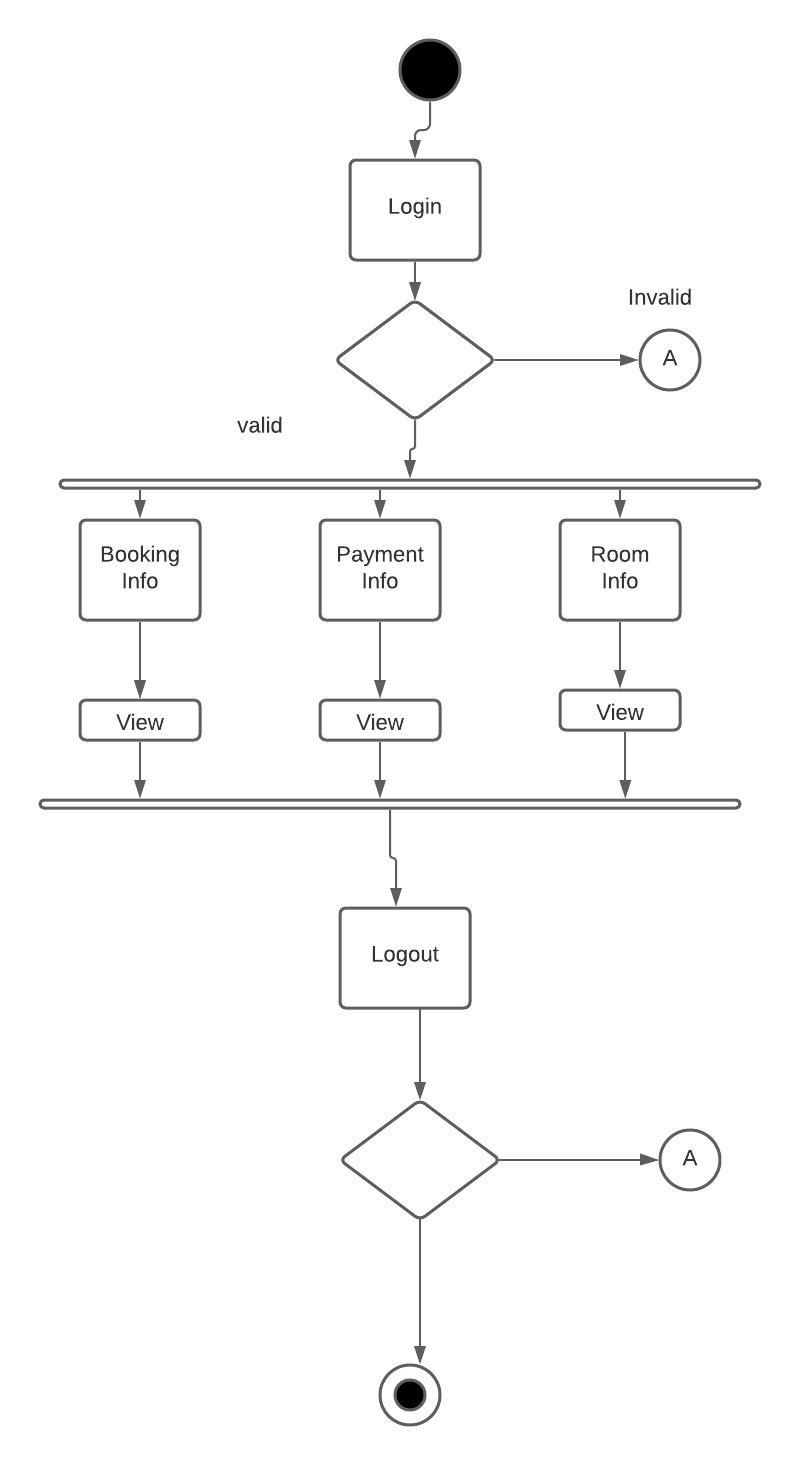
1. ***Sequence Diagram***



1. ***Collaboration Diagram for Login: optional***

******

1. ***Activity Diagram***

******

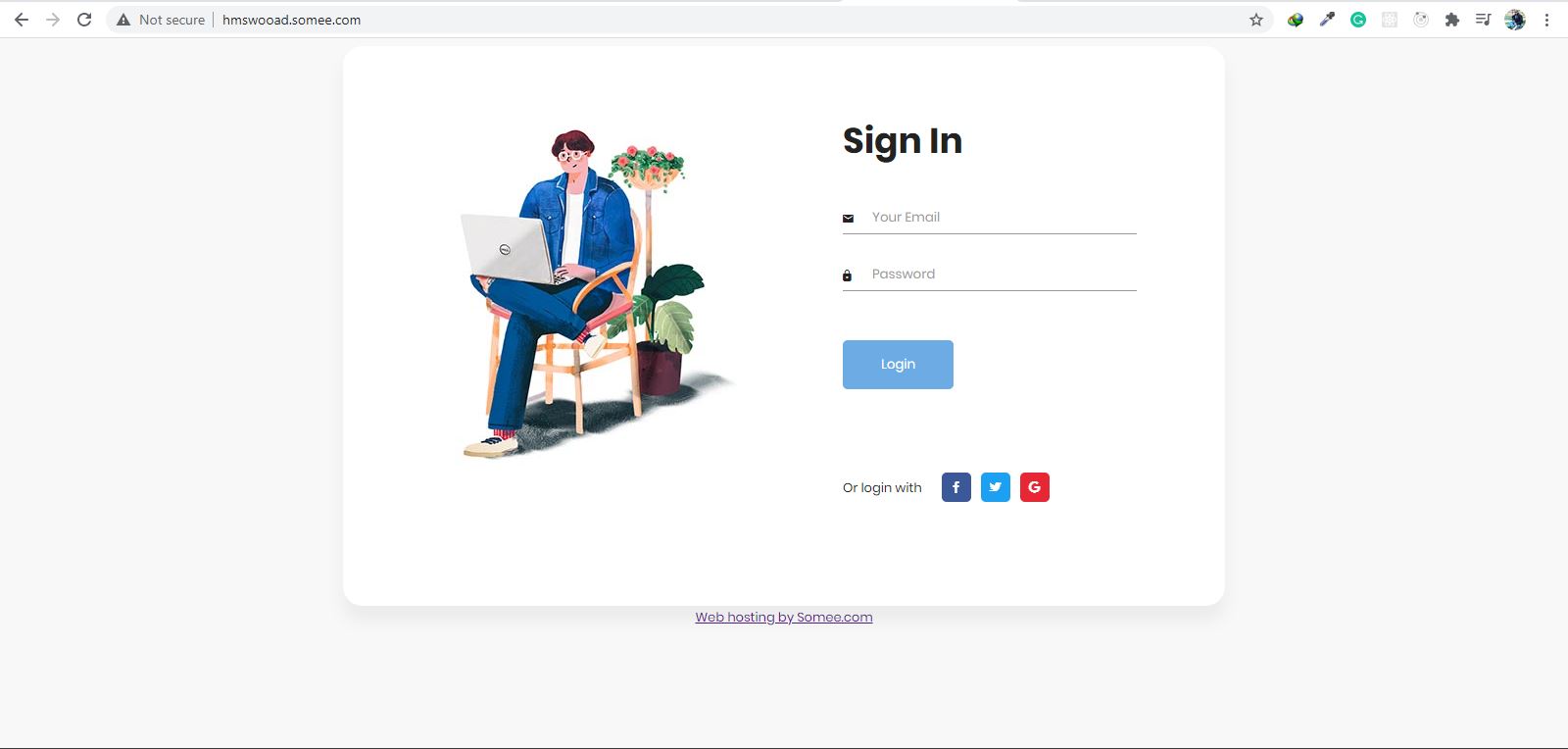
1. ***Class Diagram***

******

1. ***State Chart Diagram***



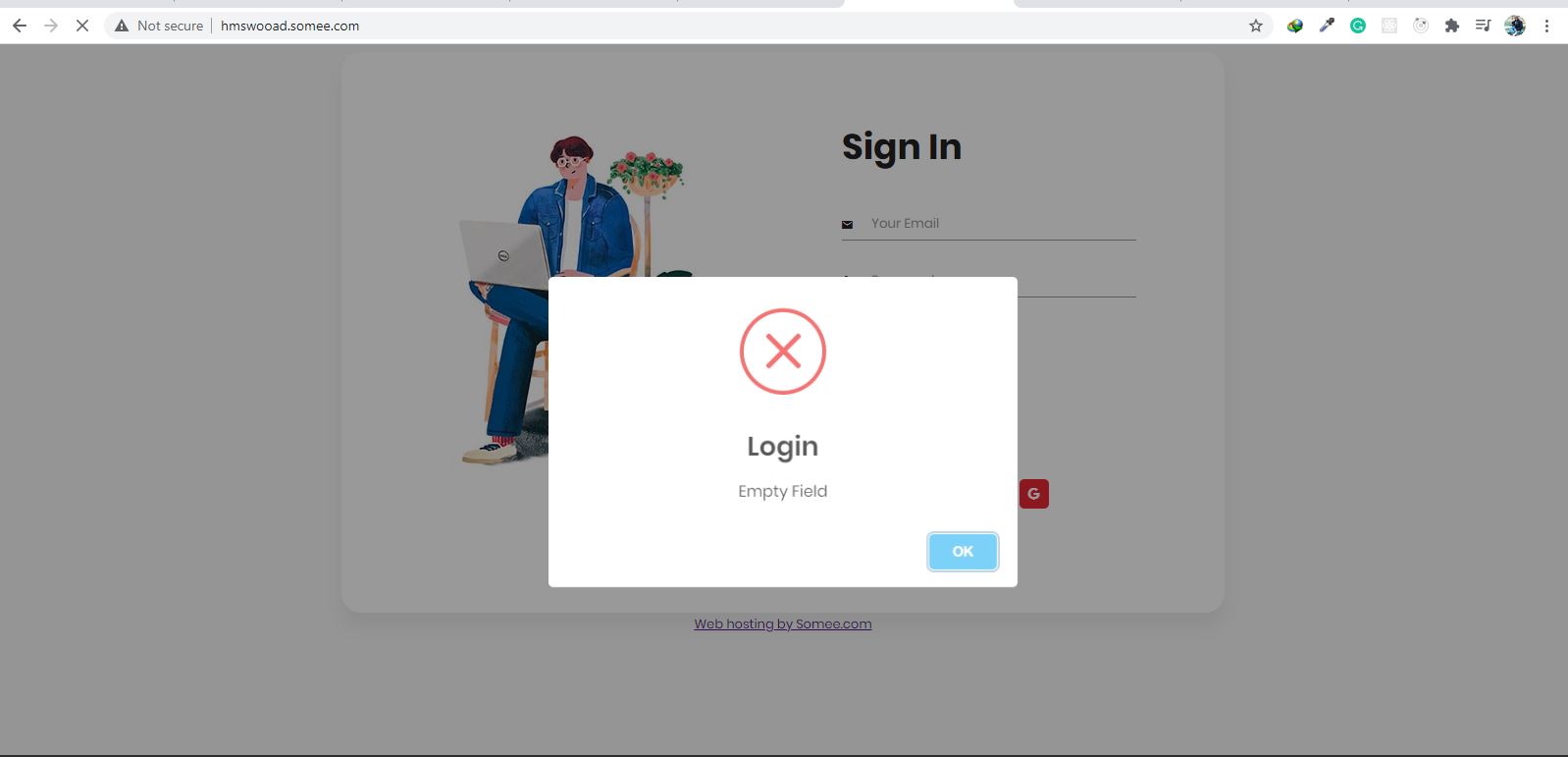
**Sign In**

****

Admin has to enter email and password to sign in. He can only enter if

email=”[admin@admi.com](mailto:admin@admi.com)” & password=”admin”

**Empty Field Error**

****

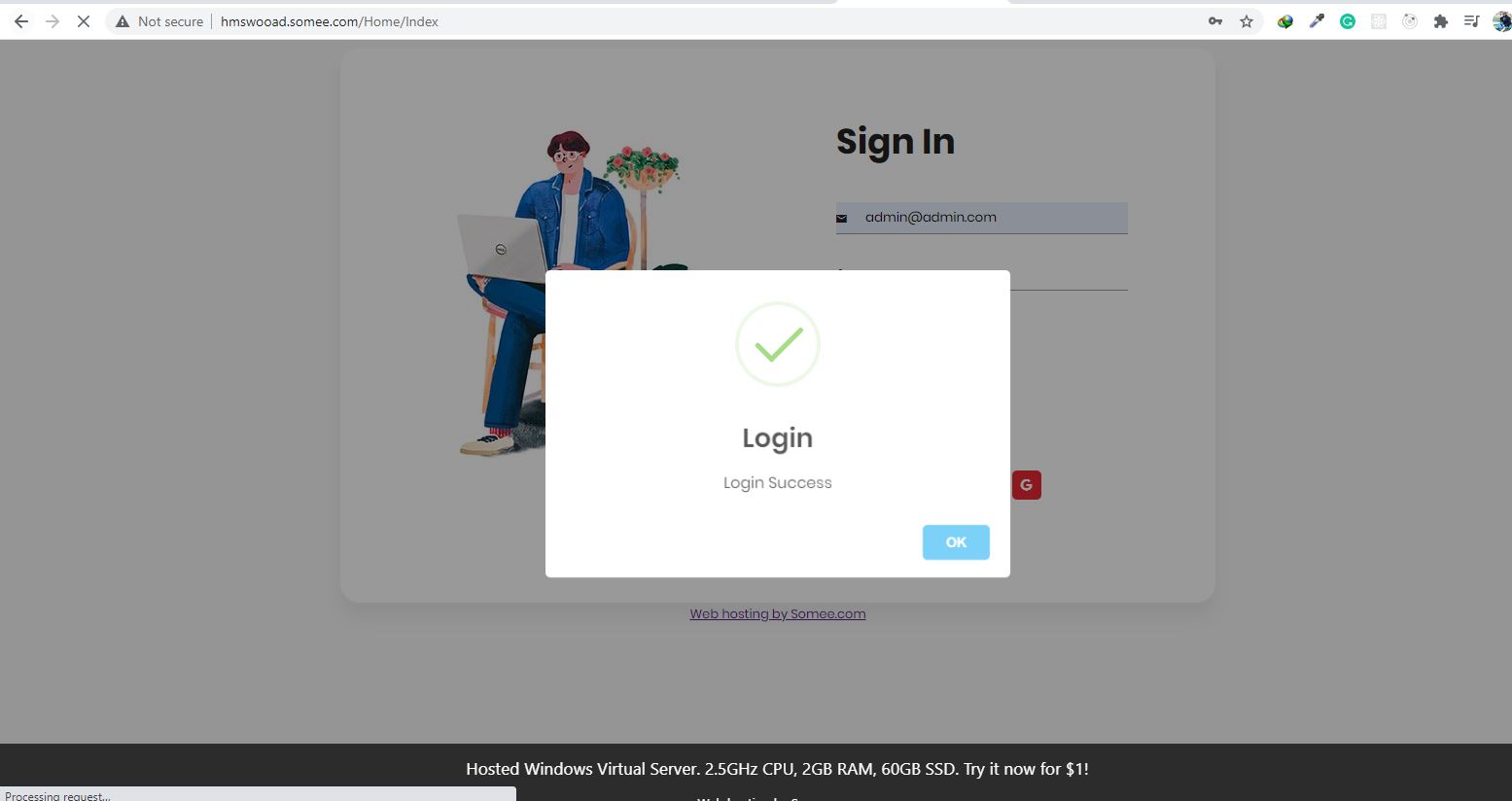
If admin try to sign in with empty fields. System will show error message as

email=”” & password=”” or

email=”admin@admin.com” & password=”” or

email=”” & password=”admin”

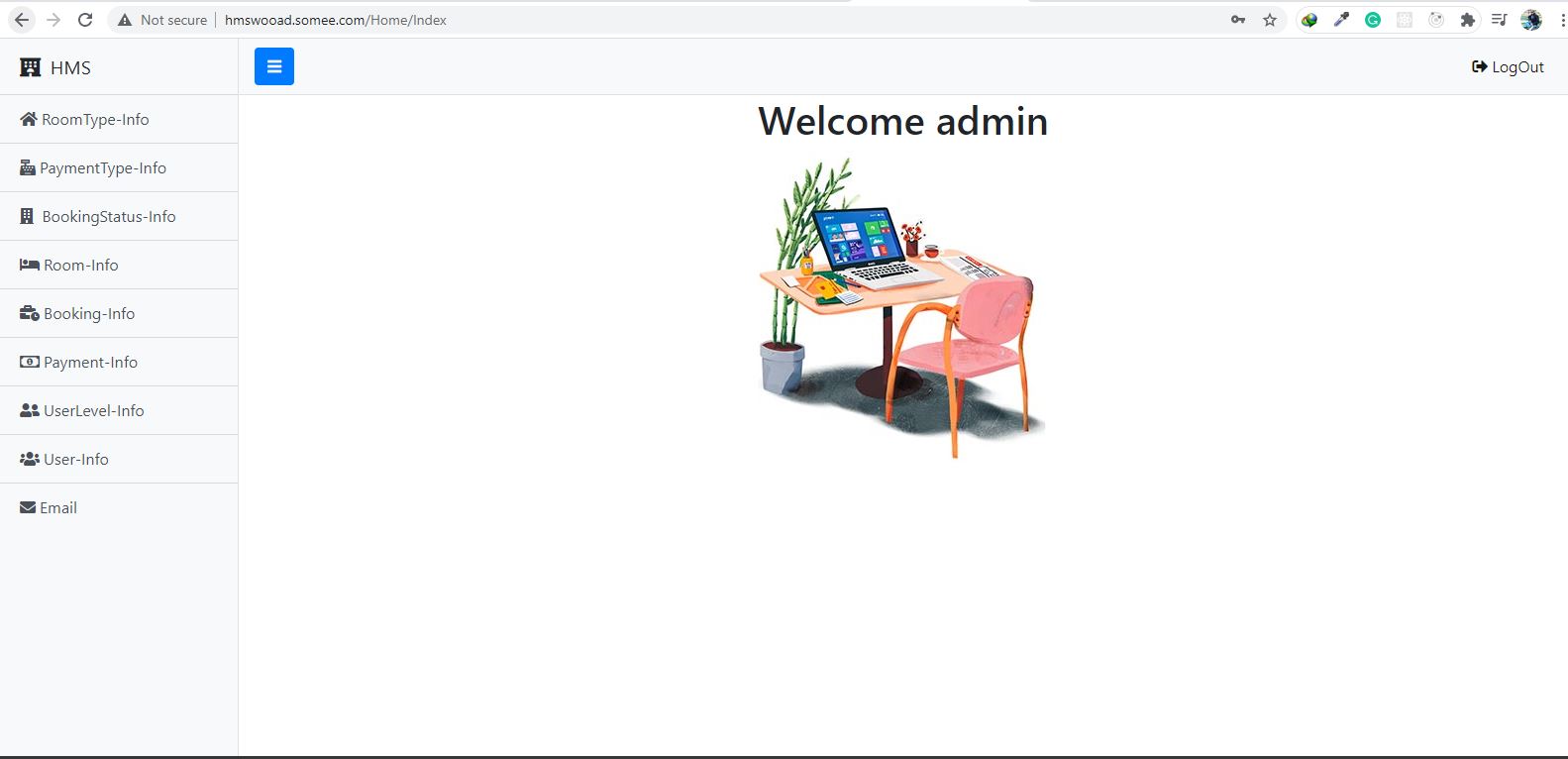
**Log In Successfully**

****

Admin can only log in successfully if he enter correct data in both fields.

email=”admin@admin.com” & password=”admin”

**Admin Dashboard**

****

* + 1. **Module 5 complete CRUD Database**

Dashboard has all the database records about all the tables

There are 8 tables

Room-Type:

It contains all the information about what type of room it is price of the room and description of the room and also the capacity of the room how many people can stay in single room.

Payments-Type:

It contains only what type of payment customer can do like credit, debit, online etc

Booking-Status:

It contains data about different status of booking like reserved, free, occupied etc

Room-Info:

It has all the information about rooms like room no and room name plus is it free to be booked and its booking status.

Booking-Info:

Booking info has all the information about the customer and room which has been booked

Payment-Info:

It contains all the payment information about booking or booked room.

User-level-Info:

All the user level information is recorded in this table admi can check all the levels of user. User levels can be operator or admin.

User-Info:

Data about all the sign-ups are saved here included email, password and name.

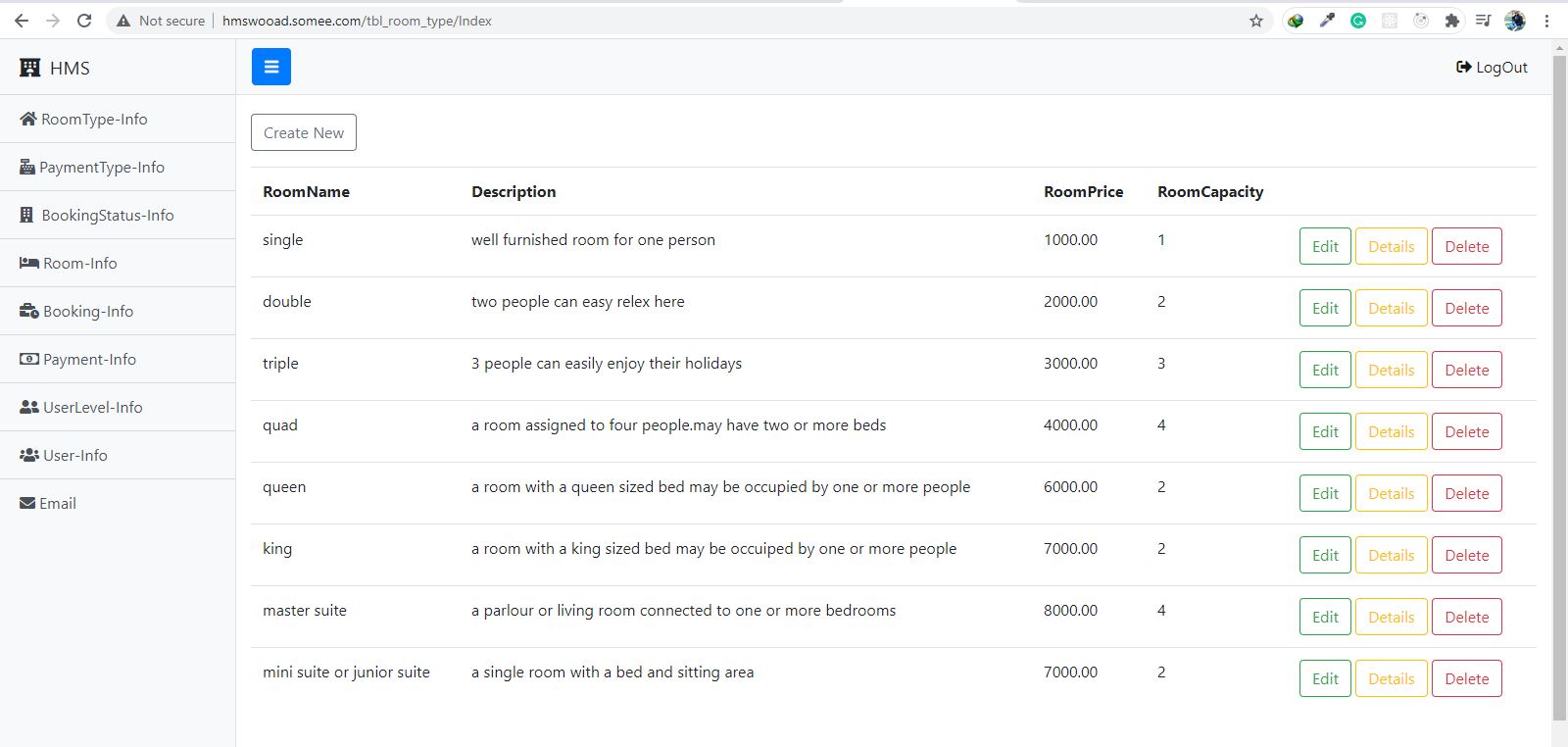
Email:

Admin can mail customer about their booking.

Logout:

Admin can also log out if he wants to.

**Room Type**

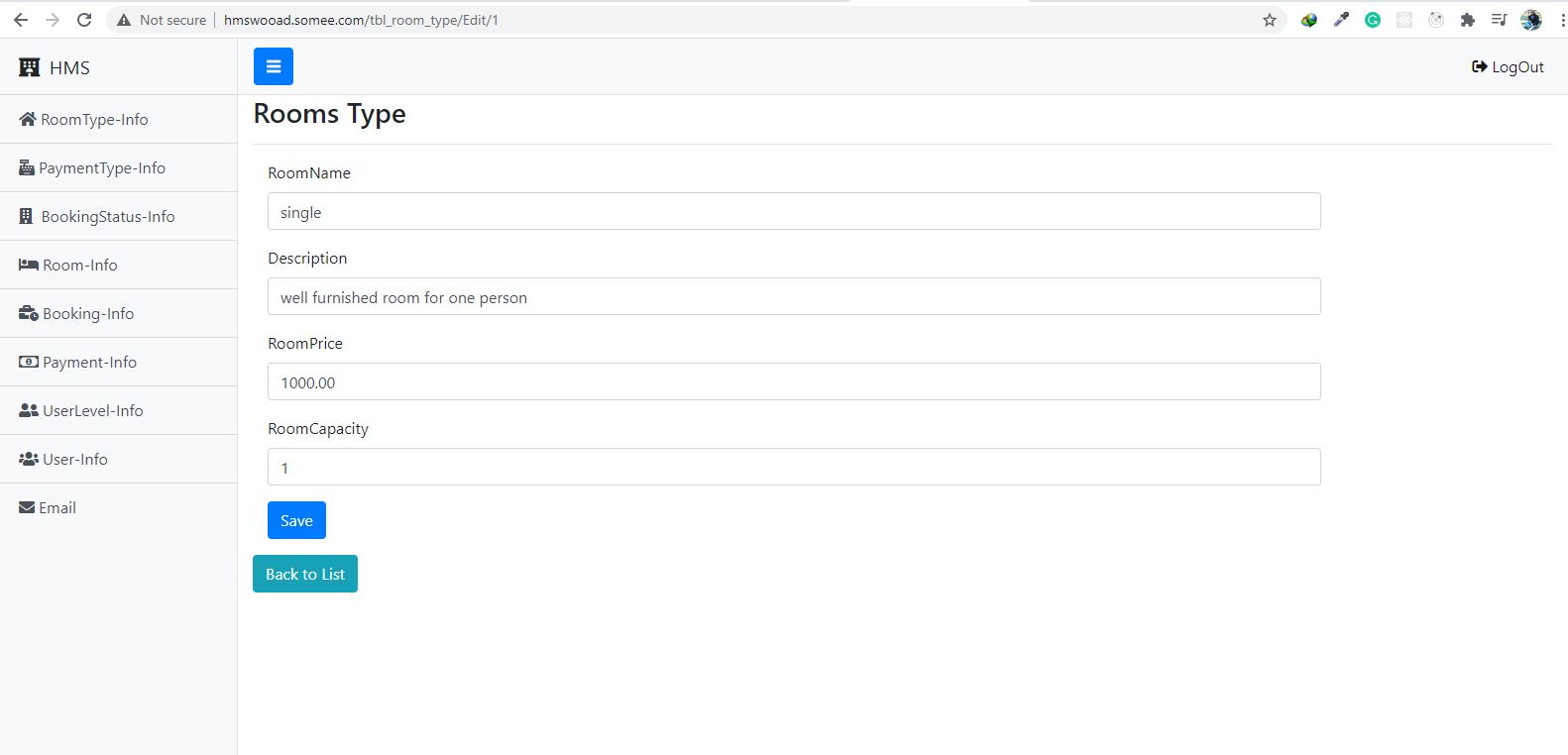
****

Room Type:

Admin can add new record to this table can edit, see details and delete too. It contains room name, description, room price and capacity.

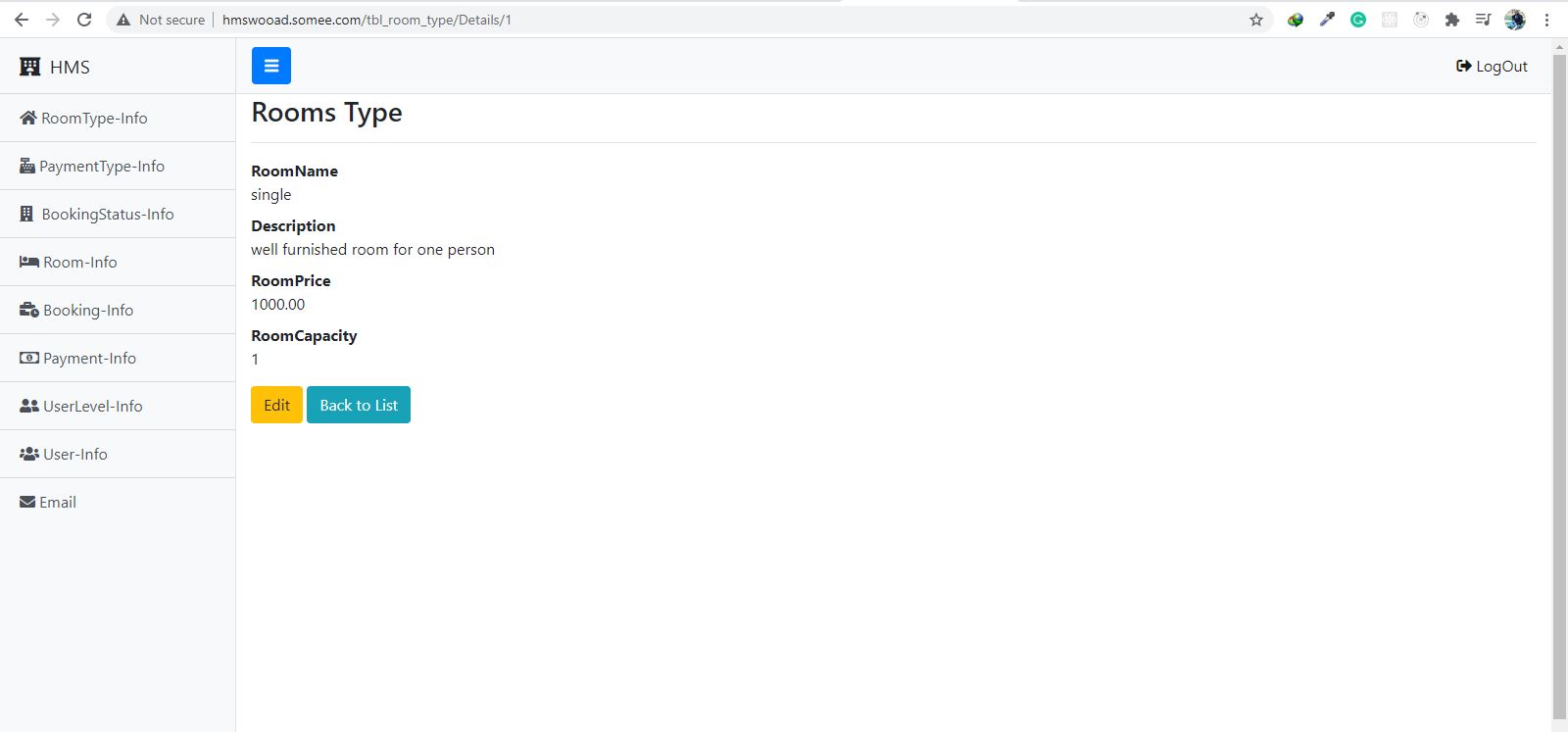
Attributes of room-type: room type id, room name, description, room price, room capacity.

**Adding new record in Room-Type**

****

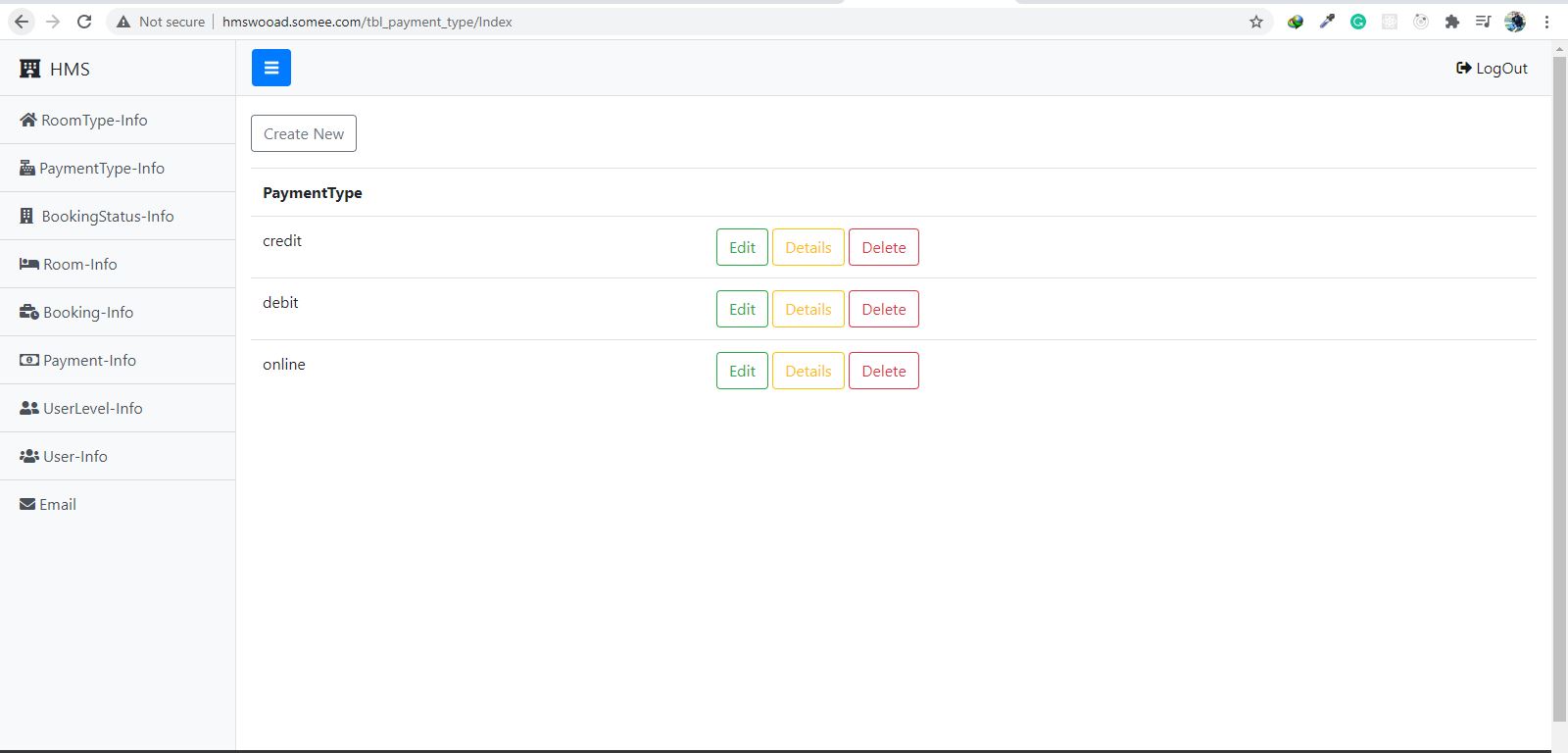
Adding new record in room type. After all fields are filled by admin then admin can save the record or can to back to list.

**Editing old record in Room-Type**

****

When admin want to edit the record, he can just click edit and can easily edit the record or can go back to the list.

**Payment-Type**

****

Payment-Type contains just the record about payment type admin can add, delete and edit any record.

Admin can add new record by create new. Database will auto generate payment id and admin just has to write only payment type. Here we have credit debit and online option for customers.

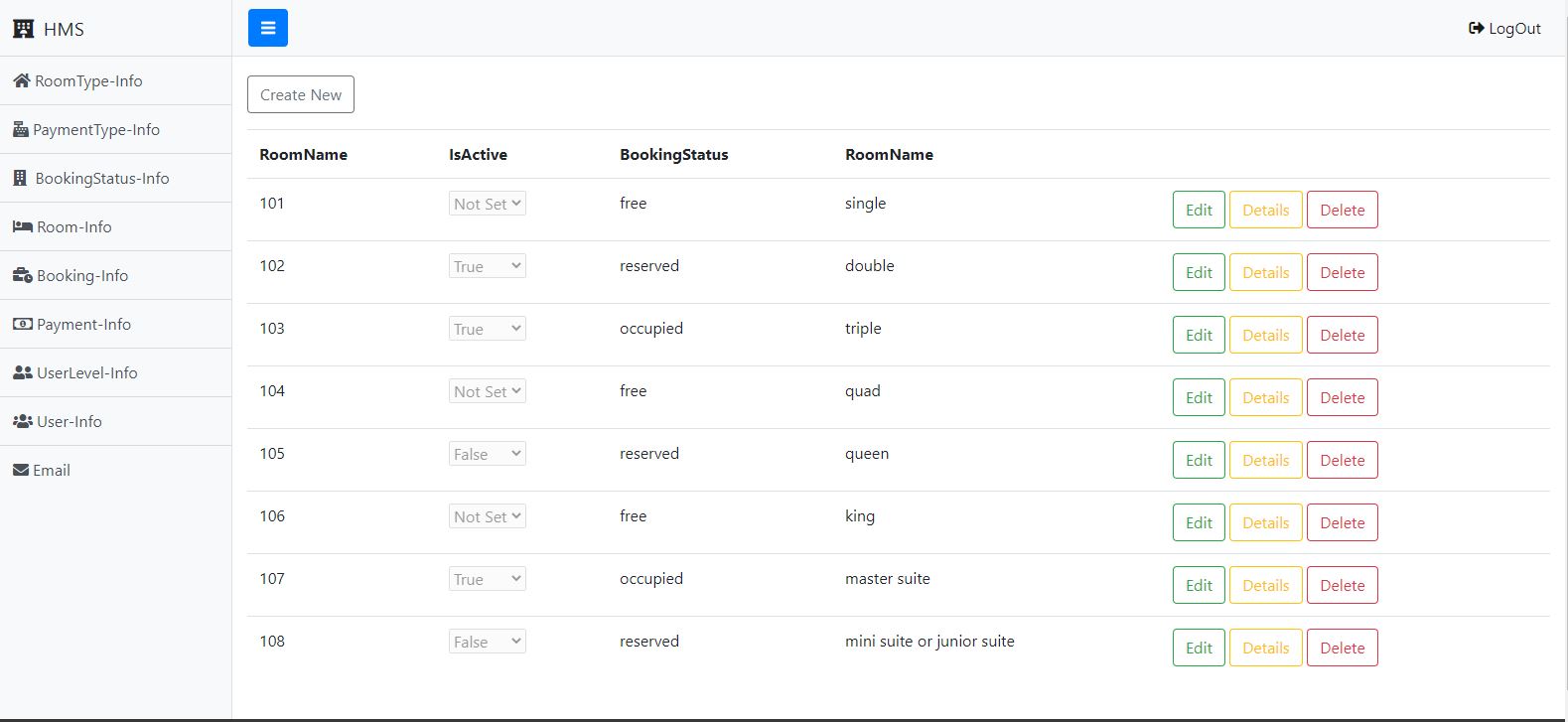
Attributes of payment type: Payment id and payment type

**Booking-Status**

****Booking status contains status related about booking like free reserved or occupied. Admin can here also can add new record, can delete and can also edit the pervious record too.

Attributes of Booking-Status: booking status id and booking status

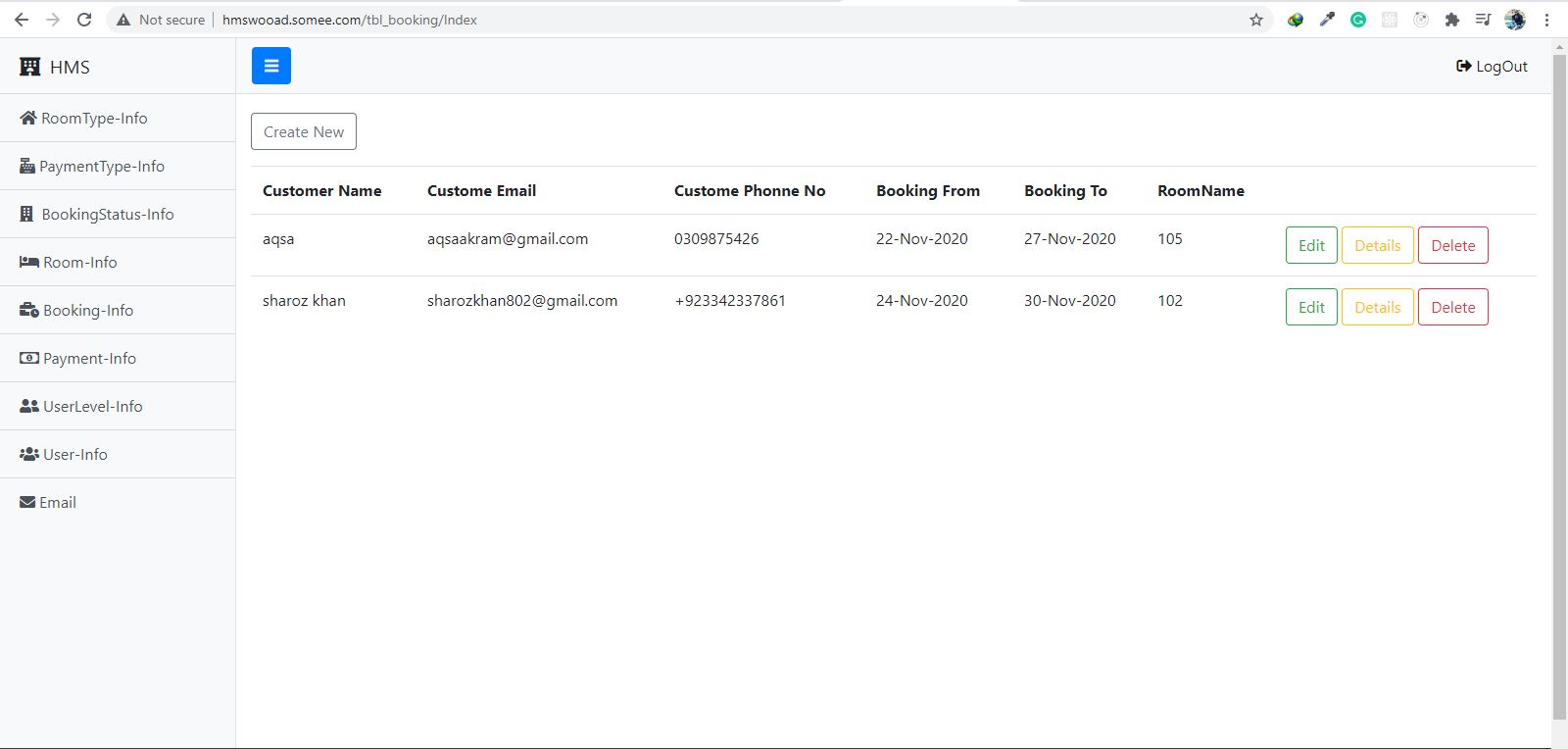
**Room-Info**

****

Admin can view all the database record of table room Info. He can create new record can do all other functions like edit and delete or view details also. This table contains information about room name, booking status, room number and also is active so see if the room is assigned to someone or is booked by someone or its free. This table also contains room-info-id that is auto generate by database.

Attributes of Room-Info: room info id, room name, is active, booking status, room number

**Booking-Info**

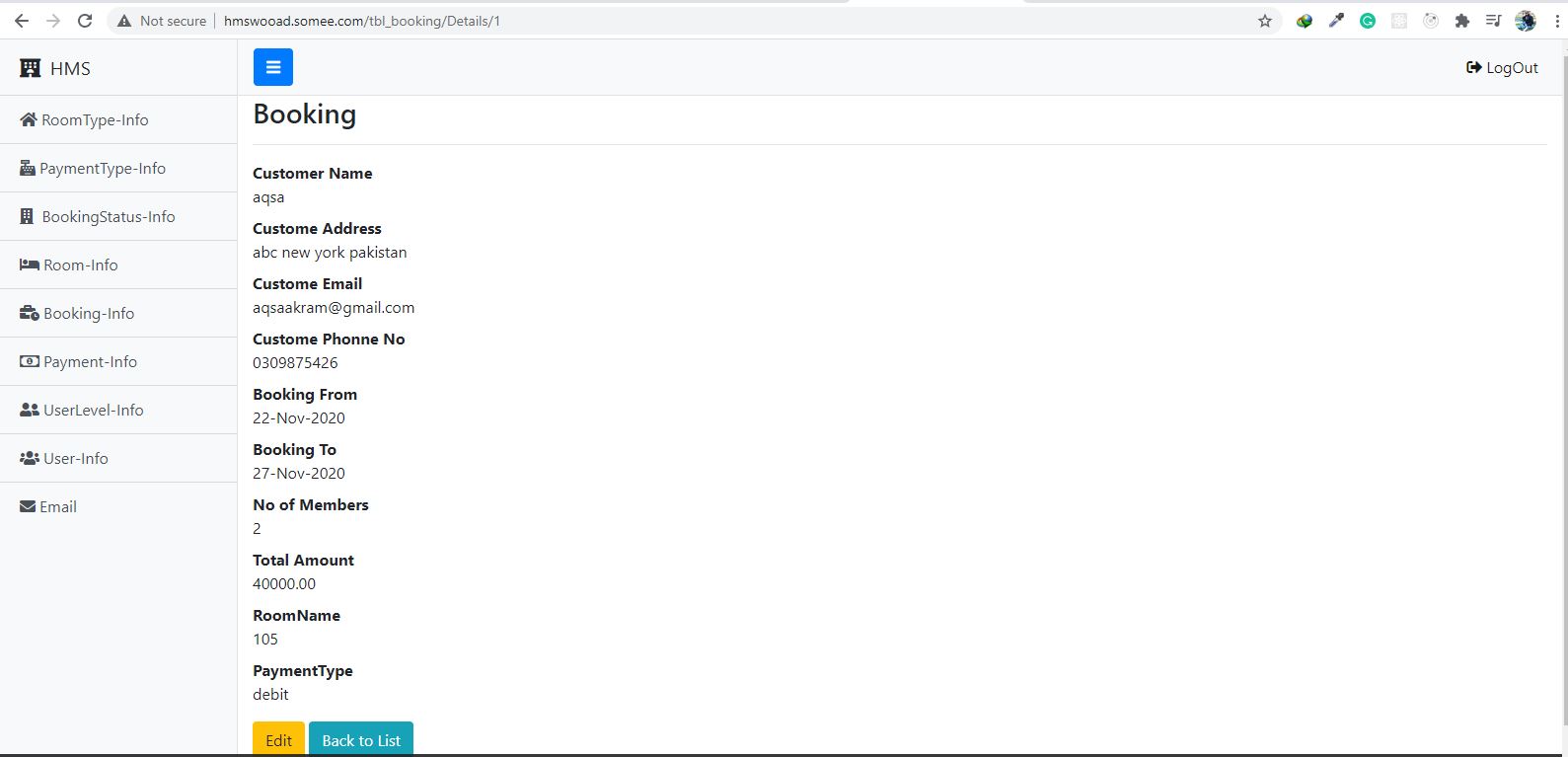
****

Booking info:

Booking info contains all the history of bookings done in our hotel. One booking record contains booking id, customer name, customer address, customer number his email and the dates of booking ‘From’ to ‘To’ as well as room name, room number, no of members, payment type and the total amount.

Here admin can also do all the functions include add new record by him-self, can delete any record what he wants to and can also edit and update any record.

**Details in Booking-Info**

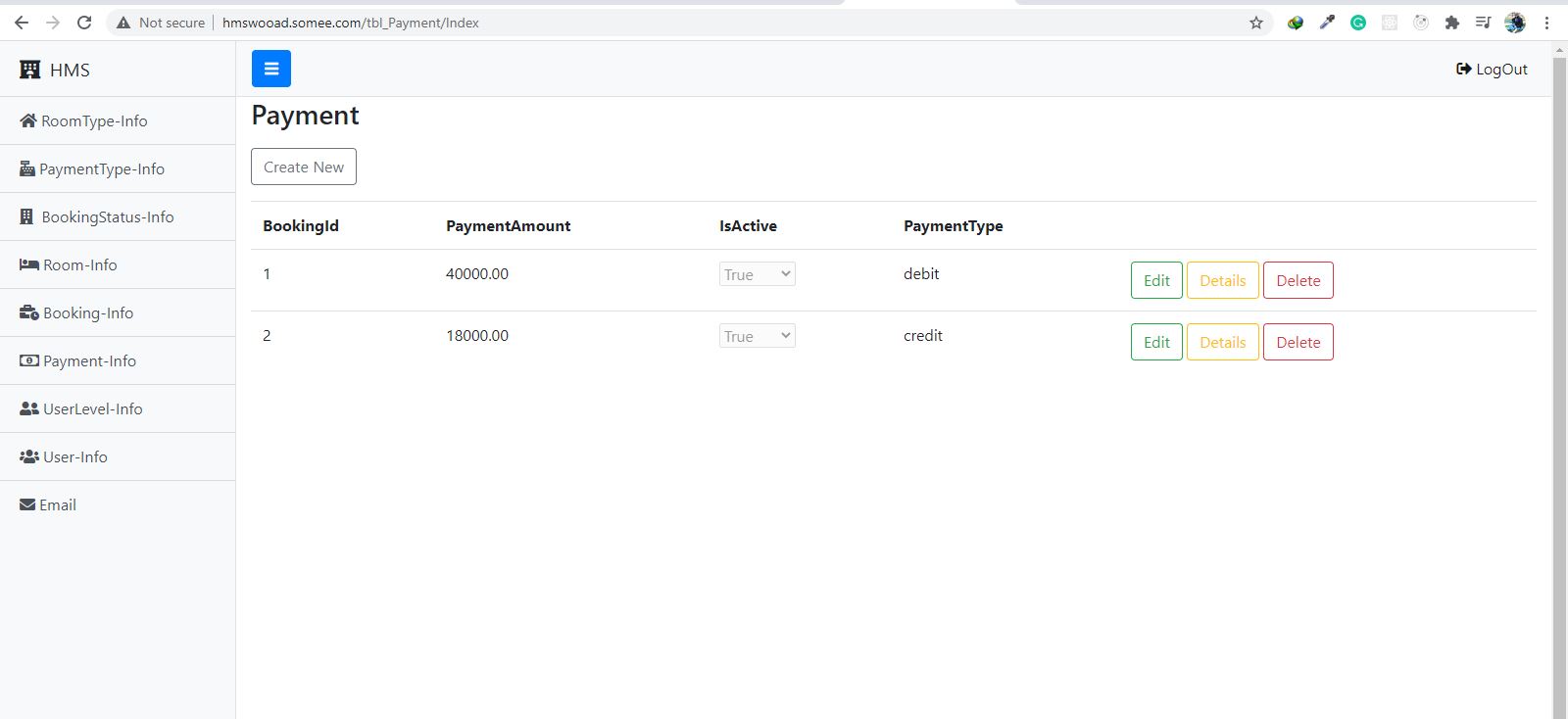
****

Details:

Details option in every table can show all the details about the record that is been selected as some information is not directly view to get that information admin has to click the details then the system will display all the details related to the record.

Just like here after clicking details system is showing no of members, total amount and payment type.

**Payment-Info**

****

**Payment info:**

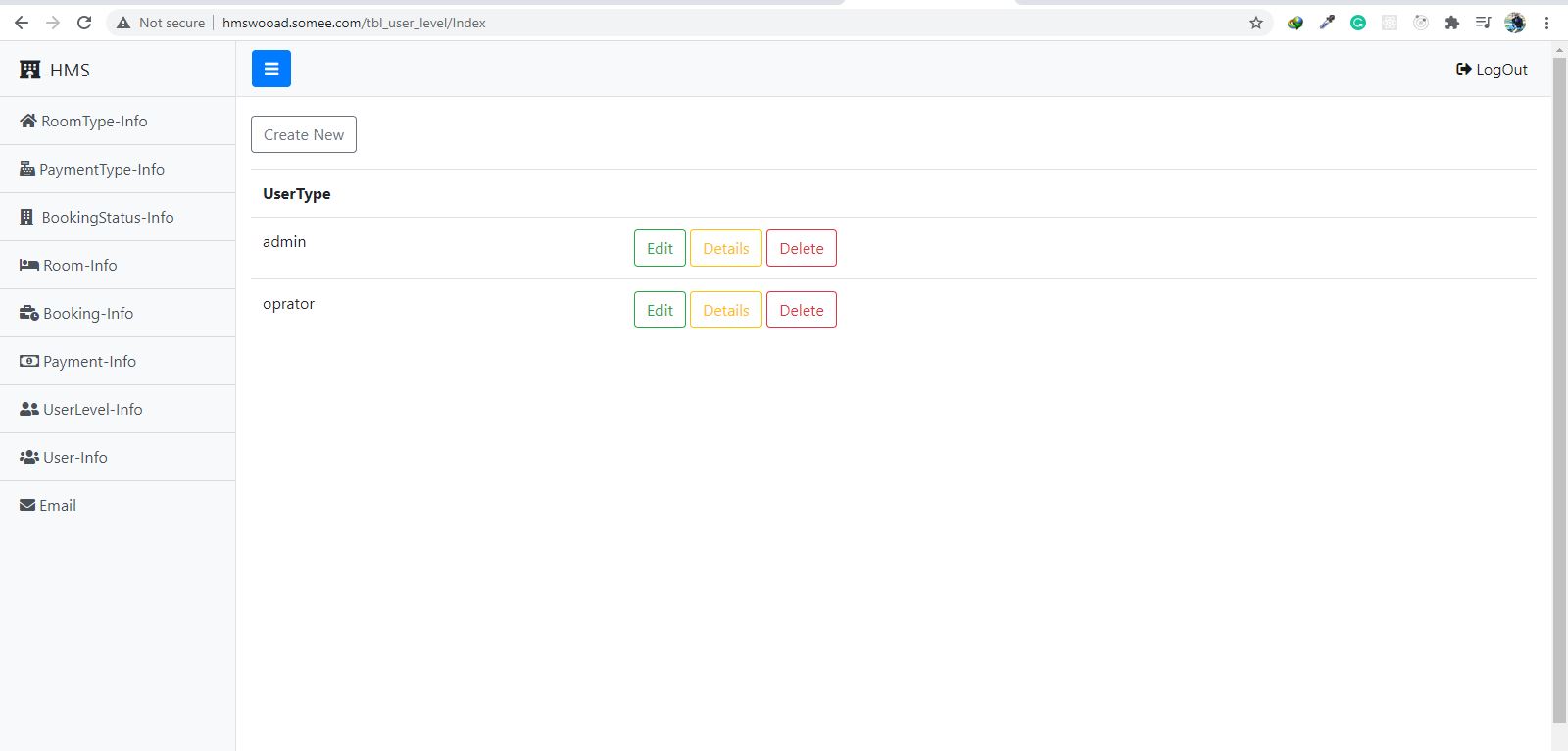
In this table admin can view all the booking related payment information. Here we can also do all add, delete, edit and details functions.

Attributes of payment-info:

Payment id, booking id, payment amount, inactive, payment type.

Here also payment id is auto generatable. Booking id refers to the customers booking id and payment amount is the for our services and inactive here refers to that is the payment is payed or it is still pending. The payment type can define the way of payment which customer has choose to pay.

**User-Type**

****

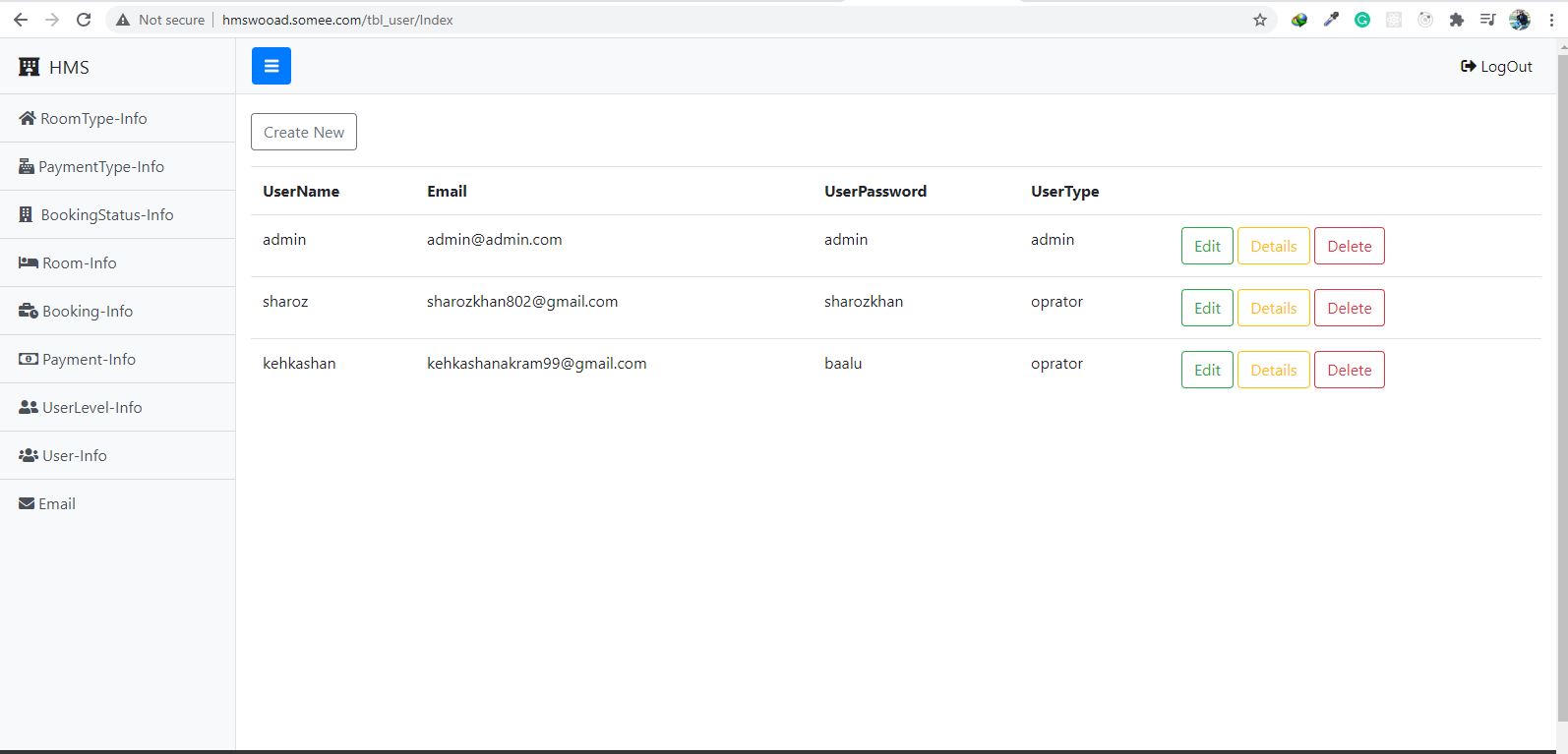
User-type:

In user level there is just auto generated user type id and user type which is given by admin. There are just two type of current users in our database admin and operator. Admin can access all database where some details of database are shown to operator only and unlike admin, he can’t do edit delete or add record in database.

Attributes of user-type:

User type id, user type

**User-Info**

****

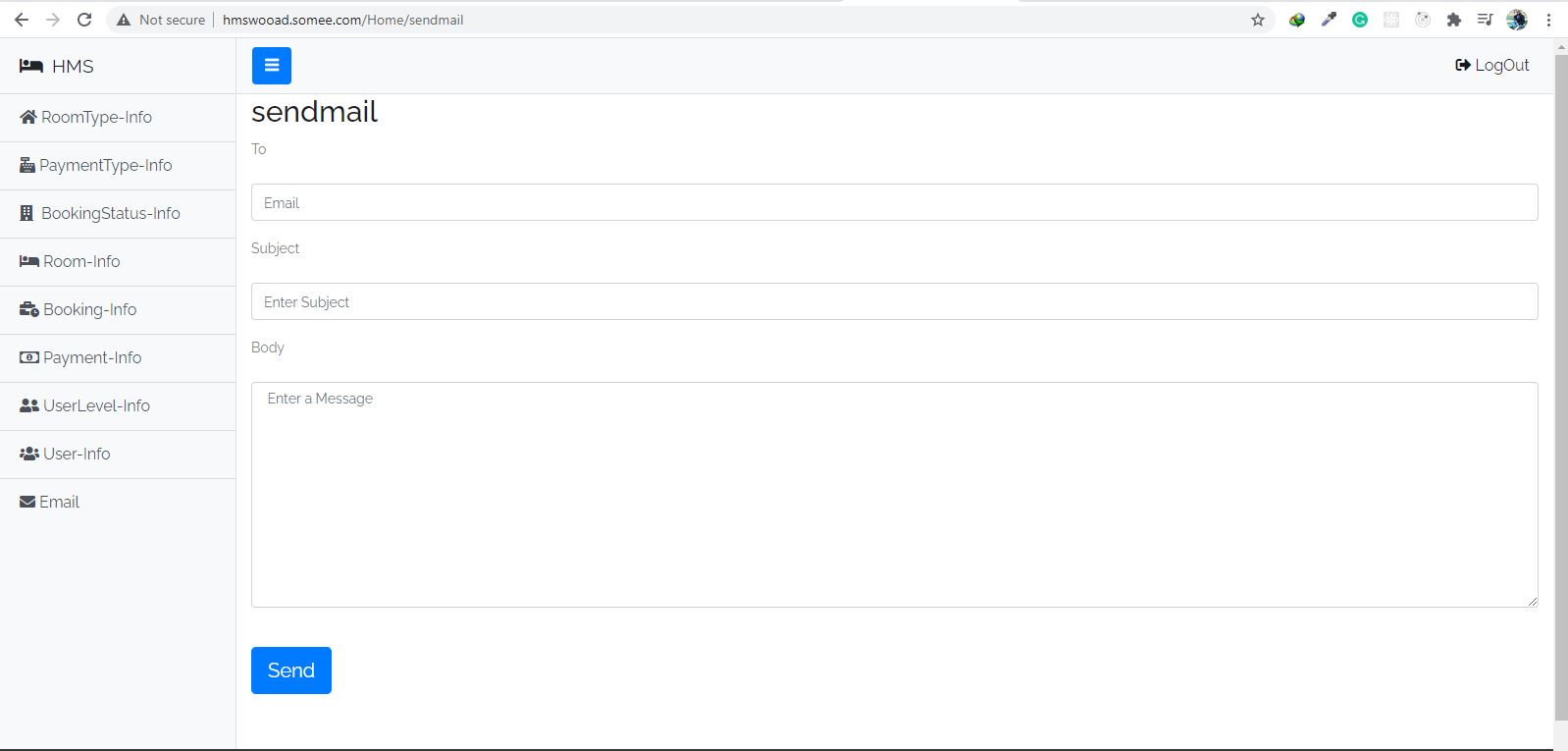
User-info:

In this table admin can record all the necessary information about users that are using database. Admin can add, edit, delete database record.

Attributes of user-info:

User id, username, user email, user password and use level

**E-mail**

****

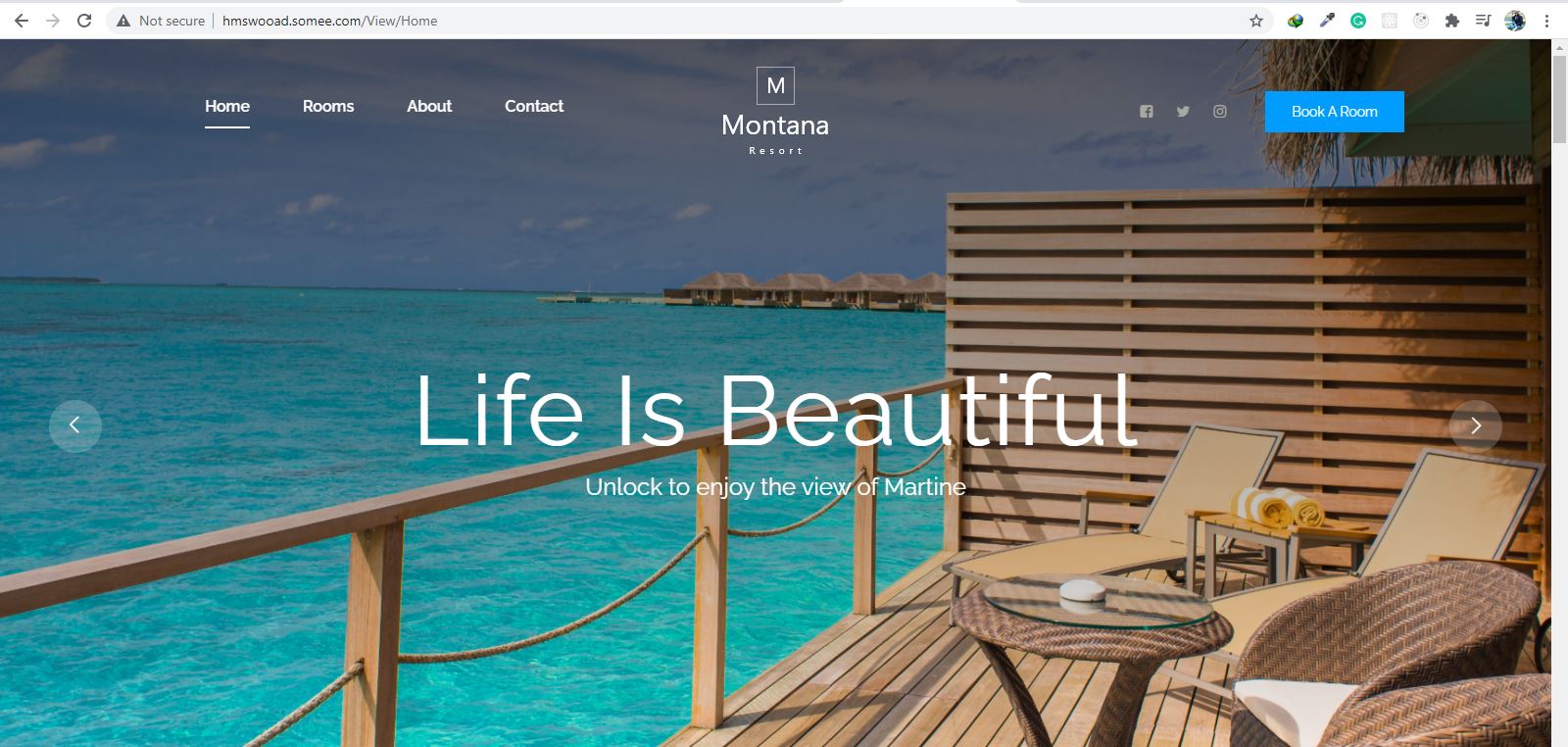
Email:

Admin can inform our customers about their bookings in our hotel by emailing them.

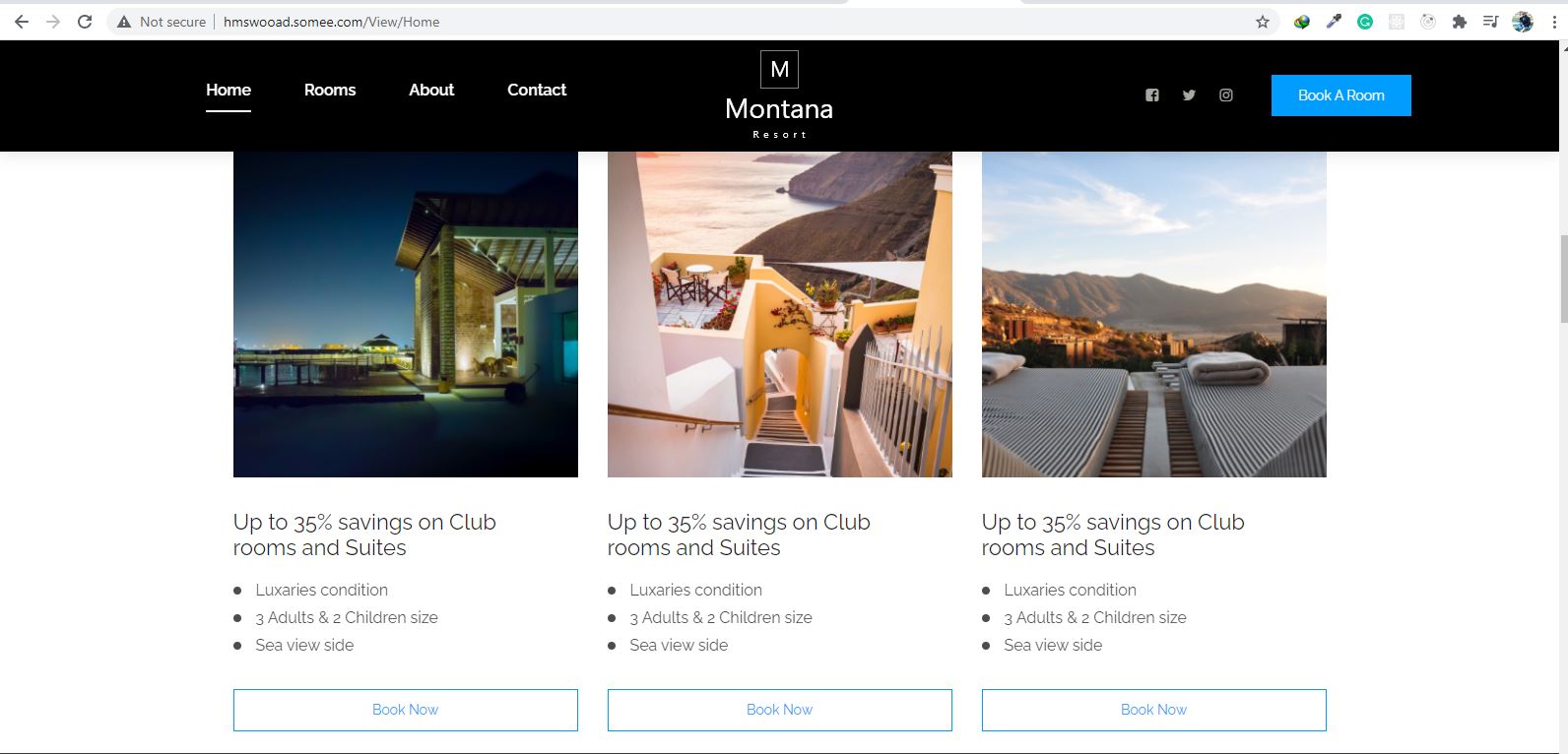
* 1. **External Interface Requirements**

**3.2.1 User Interface**

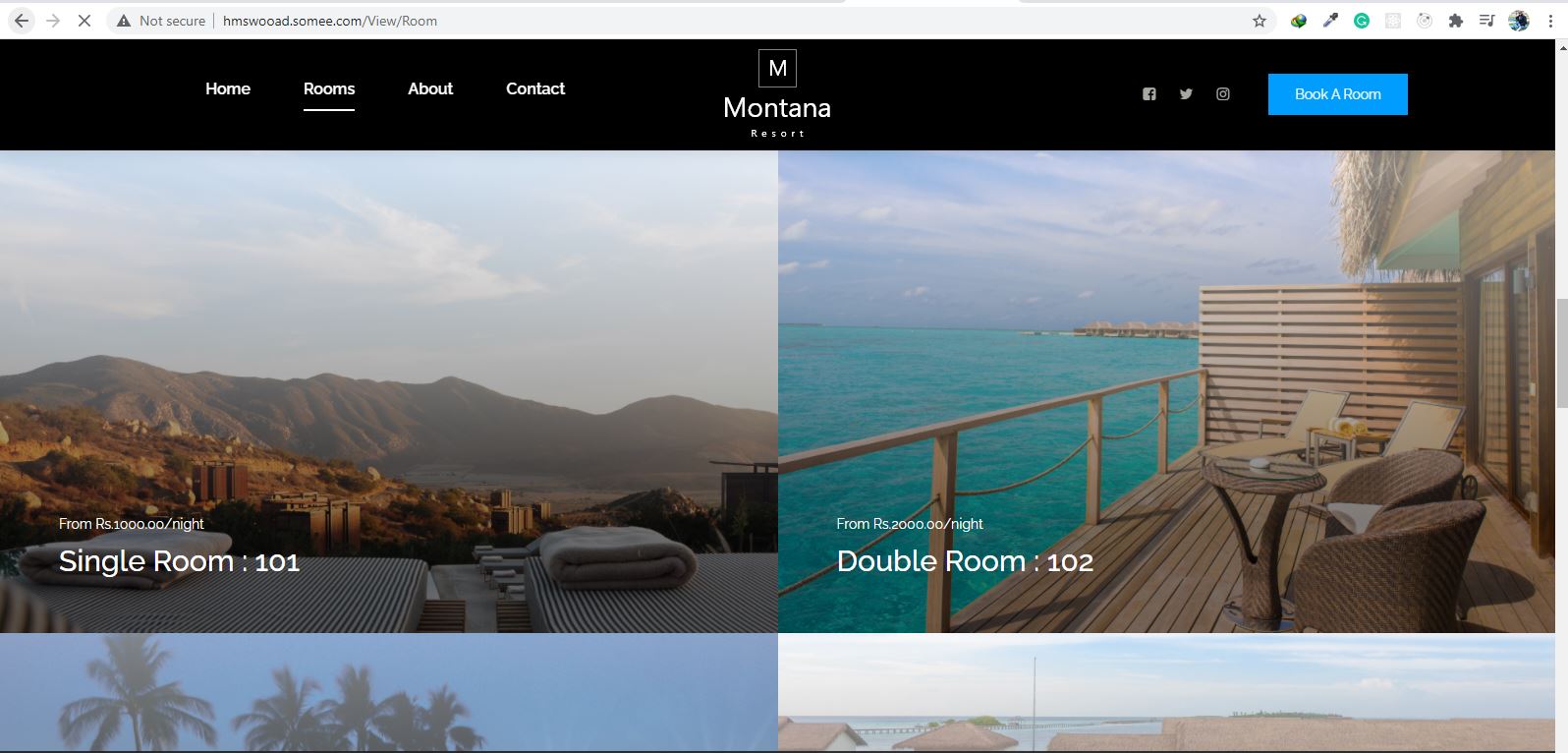
**Webpage(Home)**

****

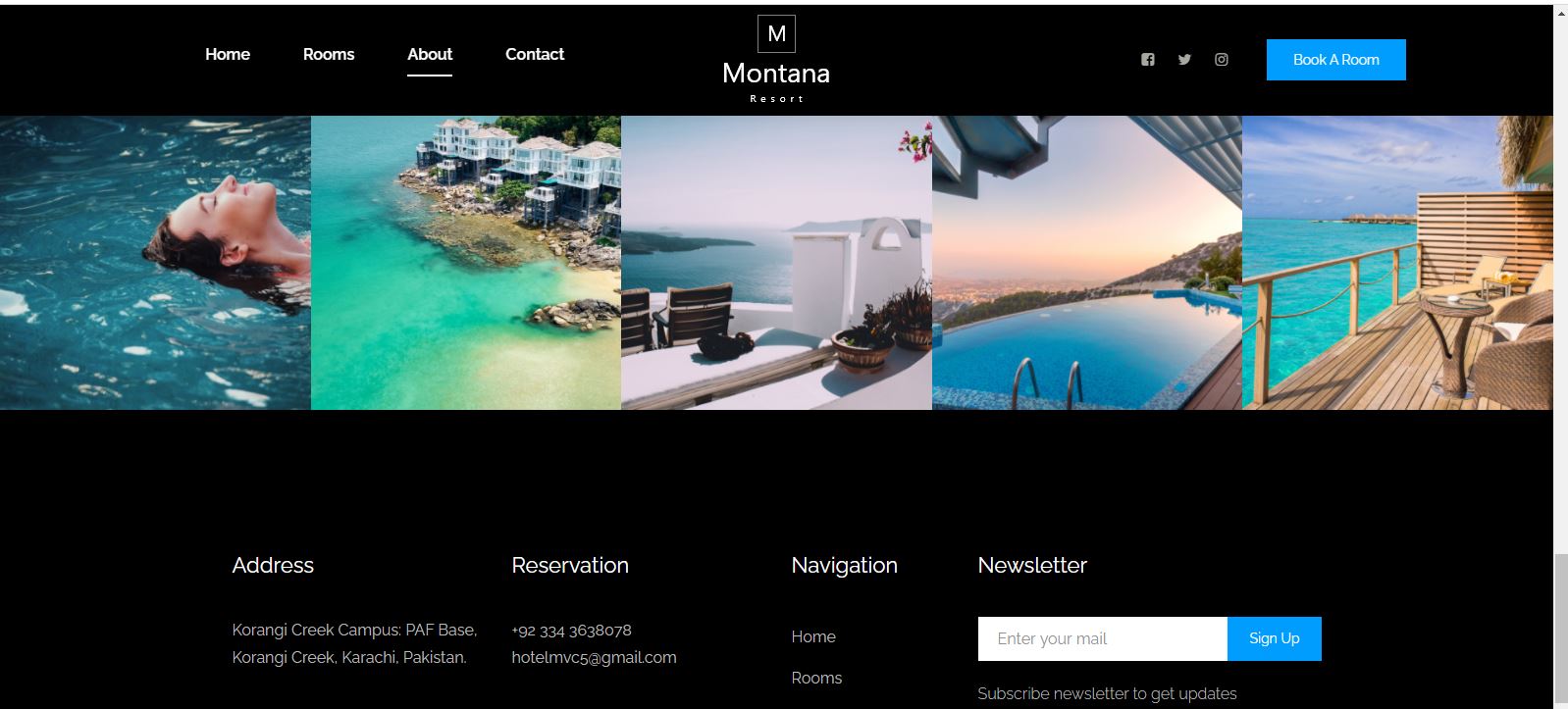
**Rooms**

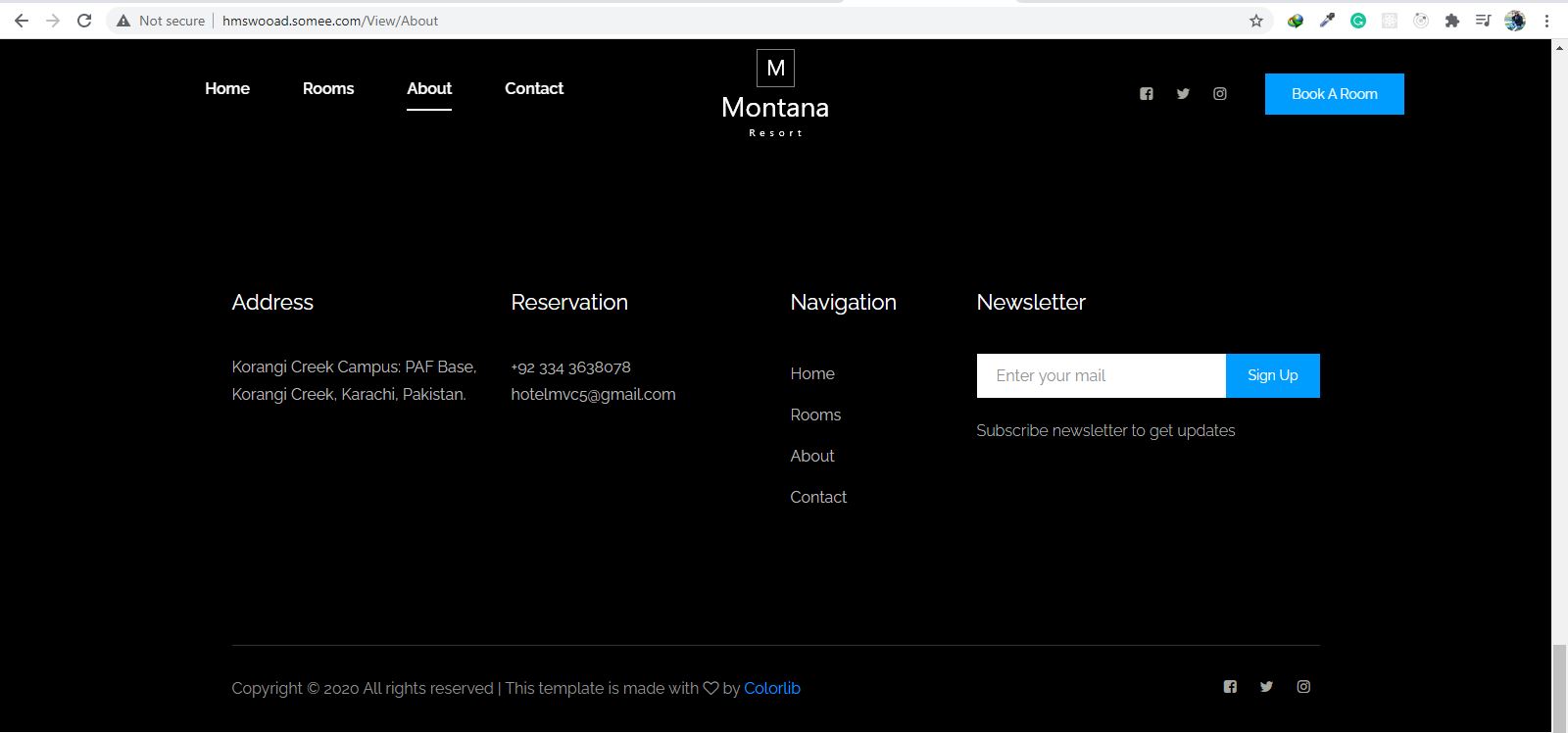
****

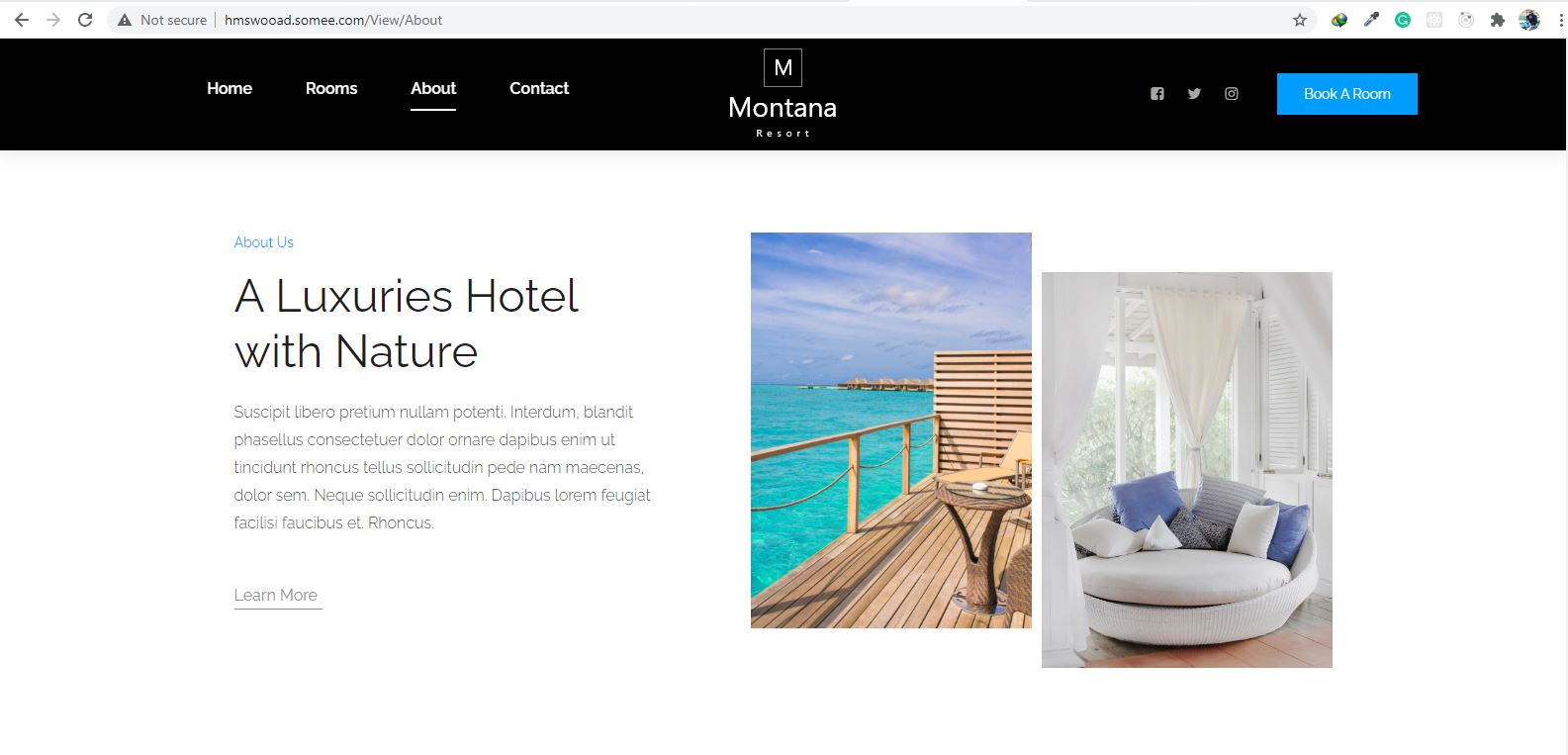
Home Page contains different offers and room booking form by which our customers can get discounts and get best offers.

****

Room has all the information about our all rooms present in our hotels we have clearly mentioned our room names and their prices with them so our customers can easily choose what they want.

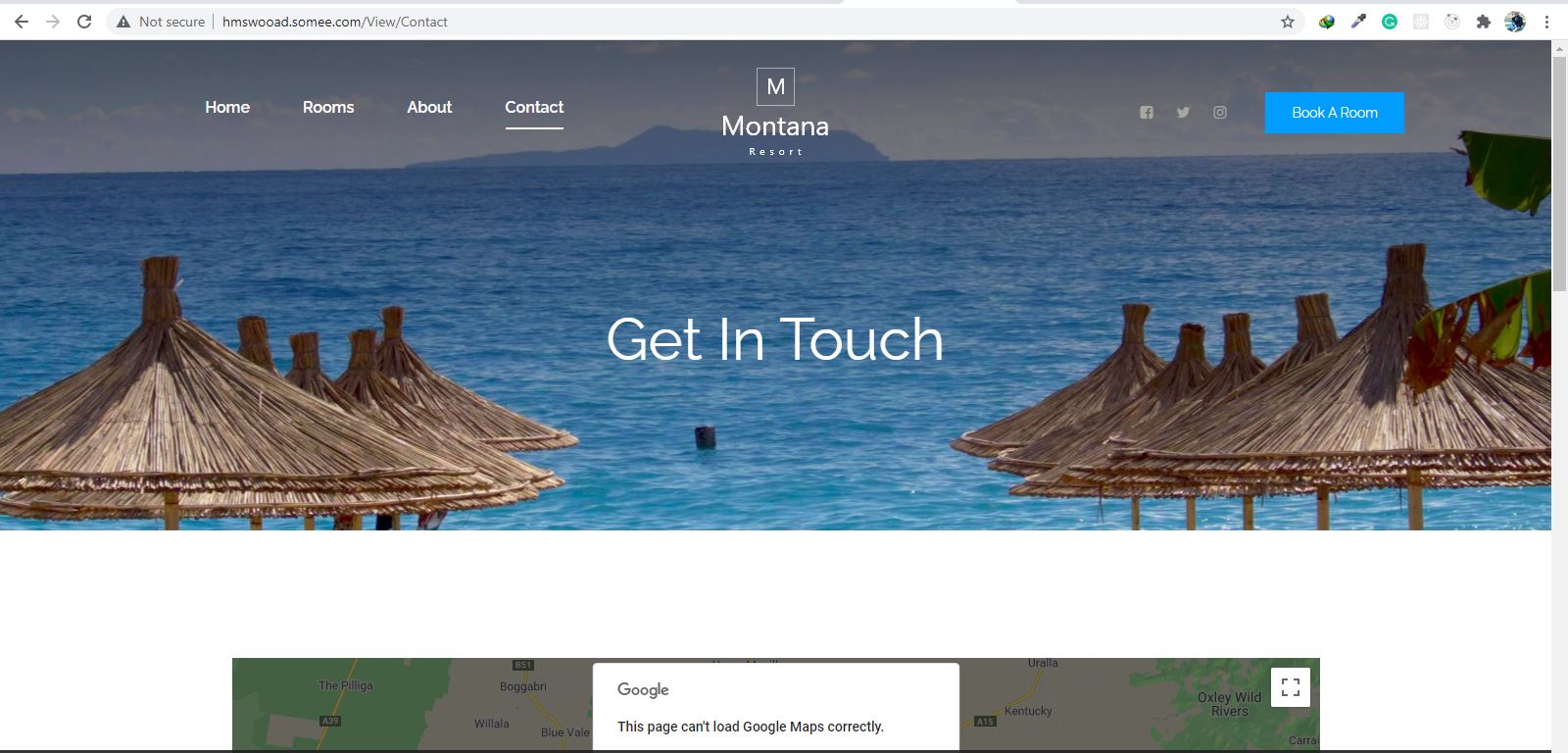
**About**

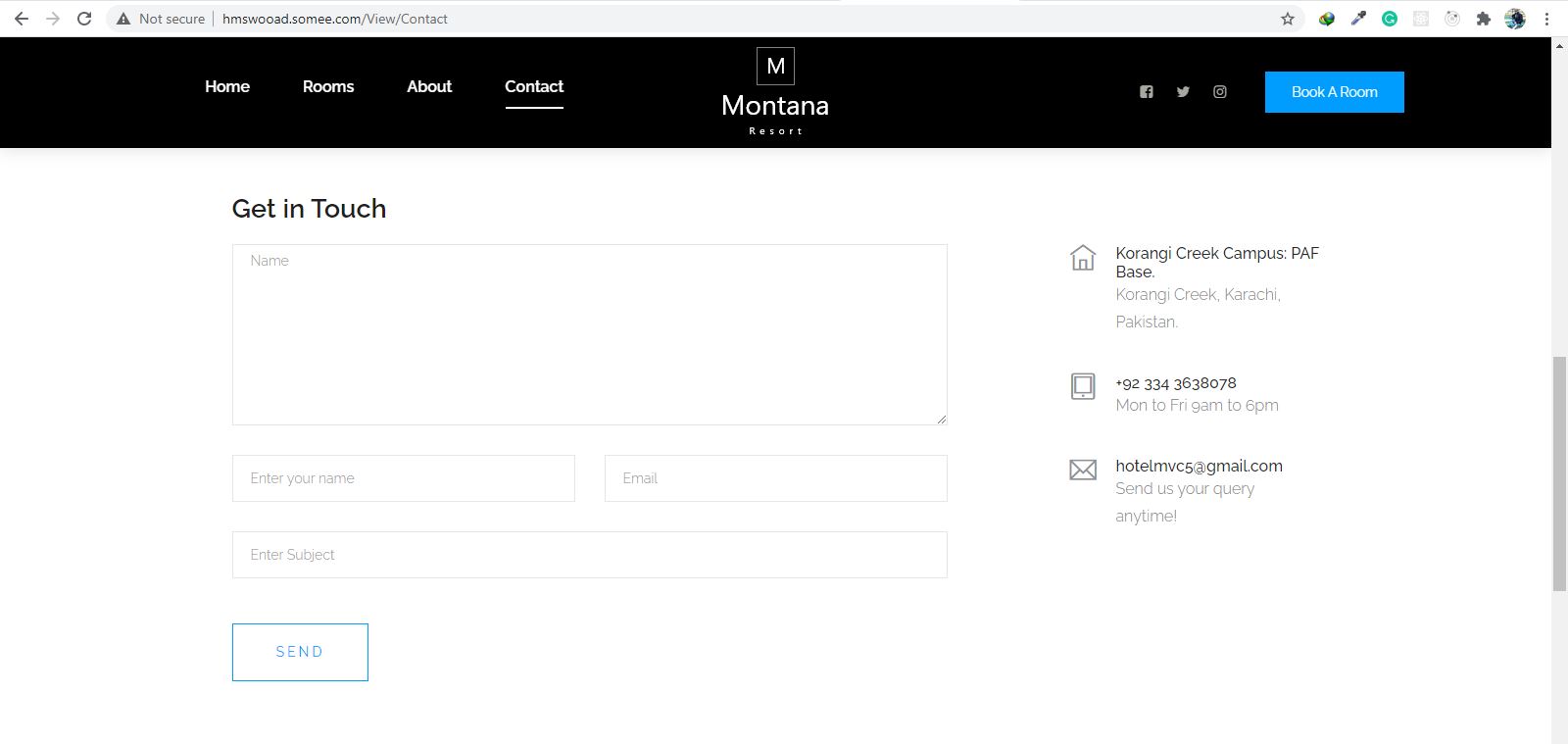
****

****

here we have given information about our hotel and we have attached pics of different rooms and places where our customers can enjoy their holidays. We have also mentioned about our facilities we will give to our customers.

**Contact Us**

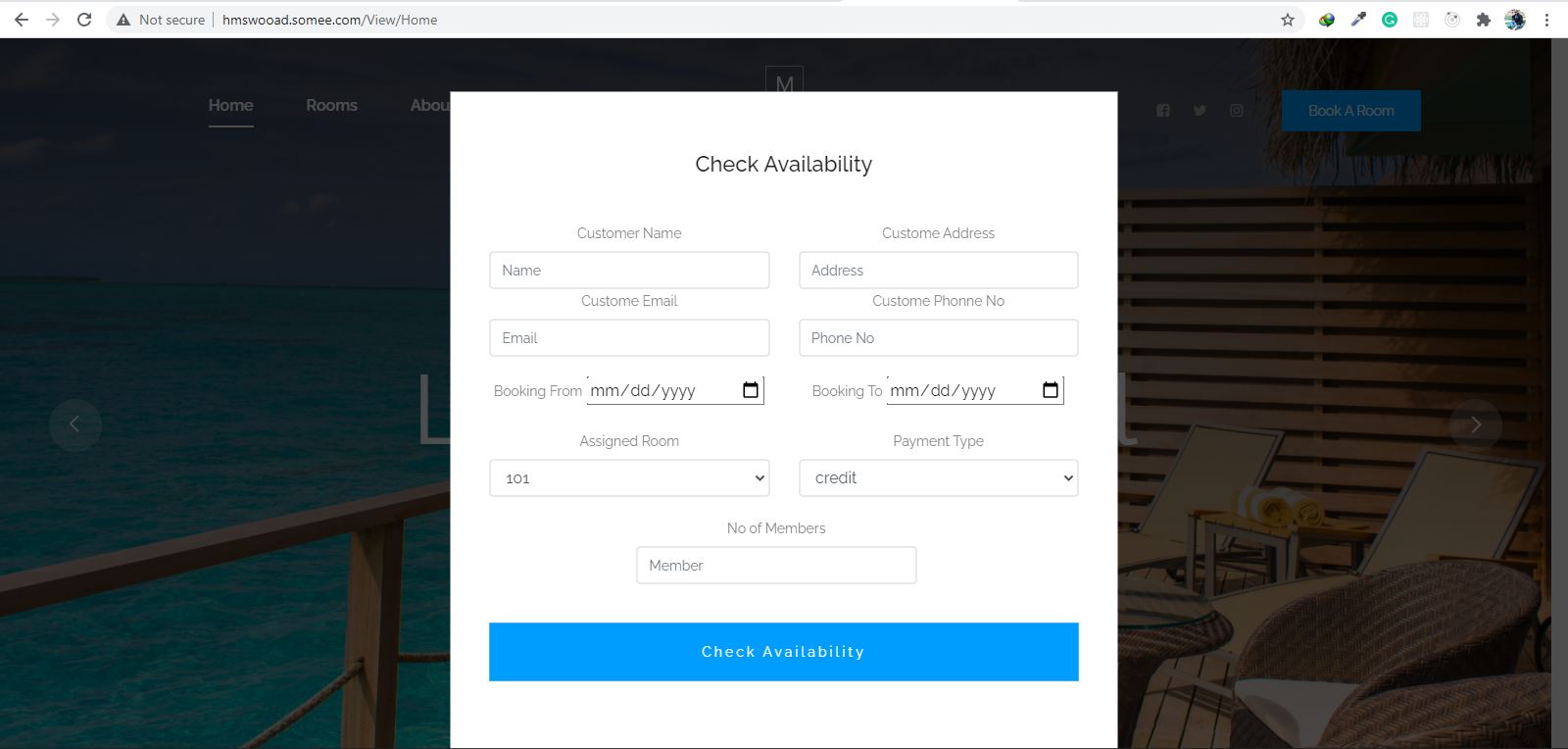
****

****

Contact us:

Customer can email us from our contact page, here we have given our map our contact number our address as well as our email. They can also visit our Instagram page, Facebook page and can also follow as at twitter.

**Booking Form**

****

Booking form: customer can book a room with this form what ever data he/she will write in this form after submitting it all data will automatically will be recorded into database. When customer will click a room which he/she saw at our page this form will appear. customer has to fill all this information.

Attributes of booking form: Customer name, customer address, customer phone number, customer email, booking from, booking to, room assigned, payment type, no of members and booking id will auto generate at the backend.

**3.2.2 Hardware Interface**

The HMS includes two major hardware components: cellular phones and regular PC's. The cell phones require WAP (wireless application protocol) network protocol, which is already programmed in the latest phones.

The second component involves the regular PC’s, which communicate with the server. The server then communicates with the database. The protocol involved between the PC's and the server is the HTTP protocol, which allows communication between the PC's and the Server. The remote PC's, such as someone accessing the HMS from home using the Internet, are able access the information through the CGI. The requests come in through the HTTP protocol, and using an ODBC the database results are returned and processed using Perl to give an HTML web page. The format of the output is displayed as web pages.

**3.2.3 Software Interface**

* + - Visual Studio 2015
    - Google Chrome
    - SQL Server Management
  1. **Performance Requirements**

**3.3.3 Host Requirements**

|  |  |
| --- | --- |
|  | Type of Host or  Equipment |
| Host A | PC |
| Host B | Database Server |
| Host C | Application Server |

* 1. **Standard Compliance**

There are no design constraints that can be imposed by other standards limitations.

**3.4.2 Software Limitations**

        must be able to run Internet Explorer or browsers to access the system.

        must have cell-phone web based capability to access the system from a mobile phone.

**3.4.3 Hardware Limitations**

        Input/Output: One or two-button mouse, keyboard, cell-phone, or touch screen required.

        Network card required at thin-client terminals to make communication with server possible.

* 1. **Quality Characteristics**

There are a number of quality characteristics that apply to the ARRS software system.

**3.5.1 Portability**

The HMS system will be developed using HTML and Java so that it can be accessed from any type of system using just a regular web browser. It will also be available to users that have web access on their cellular phones. The system will be tested on all types of hardware before being released to ensure that is it compliant with this requirement.

**3.5.2 Reliability**

The system should be capable of processing a given number of rooms booked within a give time frame with no errors and the system should be available and operational all the time.

**3.5.3 Usability**

The HMS system will be developed so that it is an easy to use system that requires the least amount of user input possible. Every input will be validated. The user should only have general computer use knowledge. Error messages will be displayed if the user enters an invalid value or tries to access a function without the required permissions. An easy and well-structured user manual will be provided to the HMS and the system will include descriptive help for all operations allowed.

**3.5.4 Flexibility**

The HMS system should be developed in such a way that it is easily customizable. If new functions are required by admin, there will be little effort required to update the system to support new cities or new transactions.

**3.5.6 Security**

The HMS system should not compromise the customer information at any time. The user information will never be sold to other parties and will be kept secure at all times. Users will be authenticated to ensure that no unauthorized users gain access to private information.

**3.5.7 Maintainability**

The HMS source code will be kept well structure and documented so that it is easier to maintain and extend the system. All changes to the system shall be documented.

**3.6.2 Operations**

The normal operations required by the user can be viewed as the following:

User-initiated Operations:

These operations include the login operation, which is initiated by the users. Also, the process of becoming a new user is in this category. Building, changing, and viewing itineraries, as well as paying for the itinerary are all initiated by the users. The user initiates the report generation activity.

Interactive Operations and Unattended Operations:

The users initiate all the operations mentioned above, and almost all of them are somehow interactive.. The report display is a non-interactive operation, although selecting the desired reports will require user input.

Data Processing Support Functions:

The user account data is used to create new accounts, as well as to validate user id's during login functions. For building itineraries, user input, user account data, processed. User data along with final results of user interaction (whether the user purchased a room, number of people, etc.) are collected, and used for report generation purposes. Administrative users' inputs are collected in order to modify and present schedules.

Backup and Recovery Operations:

Both databases used (customer account database and reservations database) are production databases.

**3.6.3 Site Adaptation Requirements**

There are no site adaptation requirements for this project.

**4. Supporting Information.**

There is no supporting information required for this project.