

Hotel Management System

By

ABBAS KAZMI

2017-GCUF-059754

AMNA MALIK

2017-GCUF-059720

MOHAMMAD FAIZAN

2017-GCUF-059730

**BACHELOR OF SCIENCE
IN
COMPUTER SCIENCE**



DEPARTMENT OF COMPUTER SCIENCE

Government College University Faisalabad

2021

CERTIFICATE

This is to certify that **Abbas kazmi (2017-GCUF-059754), Amna Malik (2017-GCUF-059720), Mohammad Faizan (2017-GCUF-059730)** have worked on and completed their Software Project at Software & Research Projects Section, **Department of Computer Science, Government College University, Faisalabad** in partial fulfillment of the requirement for the degree of **BS in Computer Science** under my guidance and supervision.

In our opinion, it is satisfactory and up to the mark and therefore fulfills the requirements of BS in Computer Sciences.

Supervisor / Internal Examiner

Supervisor Name

Supervisor,

Software Projects & Research Section,

Department of Computer Science

Government College University, Faisalabad

(Signature)

External Examiner / Subject Specialist

Supervisor Name

(Signature)

Accepted By: _____

DECLARATION

I hereby declare that the contents of the project report Royal Sheikhpura Hotel is application of our own research and no part has been copied from any published source. I further declare that this project has not been submitted for award or any diploma. The university may take action if the information provided is found at any stage. In case of any default the scholar will be proceed against as per Govt. College and University of Faisalabad, University policy.

STUDENT SIGNATURE

Abbas kazmi (2017-GCUF-059754)

Amna Malik (2017-GCUF-059720)

Mohammad Faizan (2017-GCUF-059730)

Acknowledgment Page

In this world of competition, there is a race of existence in which the ones having will and ability to come forward certainly succeeds. Working on a particular project is like a bridge between theoretical and practical working. With the will power of succeeding in our work we chose this **Hotel Management System** project. First of all, we thank **Allah Almighty** for showering his blessings upon us. Without **Allah's grace** we would not have come this far in our project.

We would like to express our special thank of gratitude to our supervisors especially Professor **Abdul Salam** as well as our institute who gave us the golden opportunity to work on the topic "**Hotel Management System**". By the guidance and advice of our supervisor we performed all the stages of completing our project as he helped us all over the project in researching the new things and so we learnt a lot in this stage of life. Without his help we would not have been able to complete our project. We are so very thankful for his precious time and guidance.

Finally we would like to thank our parents and friends who supported us, showed faith in us and prayed for us all throughout our hard work. This project certainly helped us increase our knowledge and skills which will help us in our bright future for sure.

Dedication

We dedicate this project to Allah Almighty who had seen us throughout our hard work to success and to our hard working supervisors for their support and guidance. We also dedicate this to our parents for their support and love for us and all our friends and loved ones for their faith and priceless prayers.

Table of Contents

Contents

Acknowledgment Page	4
Dedication	5
Table of Contents	6
List of Tables	8
List of Figures	9
Chapter No 1: Introduction to the Problem.....	1
1.1 Introduction	1
1.2 Project Background	1
1.3 Project Purpose.....	1
1.4 Project Scope	2
1.5 Project Objective.....	2
1.5 Intended Audience and Reading Suggestions.....	3
1.6 Document Conventions	3
Chapter No 2: Software Requirement Specification	4
2.1 Overall Description.....	4
2.1.1 Product Perspective	4
2.1.2 Product Features.....	4
2.1.3 Design and Implementation Constraints	4
Design Constraints:	4
Implementation Constraints:	5
2.1.4 Assumptions and Dependencies.....	5
2.2 System Features	6
2.2.1 Login System	6
2.2.2 Make Reservations.....	6
2.2.3 Add Payments	7
2.2.4 View User Profile.....	7
2.2.5 Logout	7
2.3 External Interface Requirements	8
2.3.1 User Interfaces	8
2.3.2 Hardware Interfaces	8
2.3.3 Software Interfaces.....	9
2.3.4 Communication Interfaces.....	14

2.4 Other Nonfunctional Requirements	14
2.4.1 Performance Requirements	14
2.4.2 Safety Requirements	14
2.4.3 Security Requirements	14
Chapter No 3: Analysis (Use Case Model)	15
3.1 Identifying Actors and Use Cases using Textual Analysis	15
3.2 Forming Use Case Diagram with Candidate and Use Cases	16
3.3 Describe the Events Flow for Use Case	22
Chapter no 4: Design	26
4.1 Architecture Diagram	26
4.2 ERD with Data Dictionary	27
4.3 Data Flow Diagram (Level 0 and Level 1)	29
4.4 Class Diagram	30
4.5 Object Diagram	31
4.6 Sequence Diagram	32
4.7 Activity Diagram	52
4.8 Collaboration Diagram	75
4.9 State Transition Diagram	76
5. Implementation	77
5.1 Component Diagram	77
5.2 Deployment Diagram	78
5.3 Database Architecture	79
Chapter No 6: Testing (Software Quality Attributes)	84
6.1 Test Case Specification	84
6.2 Black Box Test Cases	89
6.3 White Box Testing	100
6.3.1 Statement Coverage	100
6.3.2 Branch Coverage	101
6.3.3 Path Coverage	101
Chapter No 7: Tools and Technologies	102
7.1 Programming Languages	102
7.2 Operating Environment	102
Appendix A: User Documentation	104
Appendix B: Source Code	105

List of Tables

Table 1: Project Objectives.....	3
Table 2: Use Case for Logins.....	16
Table 3: Use Case for Financial Reports.....	16
Table 4: Use Case for Room Checking.....	16
Table 5: Use Case for Add Room.....	17
Table 6: Use Case for Delete Room.....	17
Table 7: Use Case for Edit Room.....	17
Table 8: Use Case for Search Room.....	18
Table 9: Use Case for Make Booking.....	18
Table 10: Use Case for Make Payment.....	18
Table 11: Use Case for Delete Booking.....	19
Table 12: Use Case for Add Customer.....	19
Table 13: Use Case for Make Reservation.....	19
Table 14: Use Case for Delete Customer.....	20
Table 15: Use Case for Change Info.....	20
Table 16: Use Case for Logout.....	20
Table 17: Test Case for Main Screen.....	78
Table 18: Test Case for Registration.....	79
Table 19: Test Case for Room Booking.....	80
Table 20: Test Case for Login Success.....	81
Table 21: Test Case for Login Fail.....	82
Table 22: Black Box Test Case for Registration.....	83
Table 23: Black Box Test Case for Login.....	84
Table 24: Black Box Test Case for Logout.....	85
Table 25: Black Box Test Case for Change Password.....	86
Table 26: Black Box Test Case for Update Profile.....	87
Table 27: Black Box Test Case for Manage Users.....	88
Table 28: Black Box Test Case for Add Booking.....	89
Table 29: Black Box Test Case for Manage Booking.....	90
Table 30: Black Box Test Case for Find Rooms.....	91
Table 31: Black Box Test Case for Book Room.....	92
Table 32: Black Box Test Case for Cancel Booking.....	93

List of Figures

Figure 1: Main Screen.....	9
Figure 2: Login Screen.....	9
Figure 3: Profile Web-info.....	10
Figure 4: Profile Posts.....	10
Figure 5: Rooms page.....	11
Figure 6: Bookings page.....	11
Figure 7: Users page.....	12
Figure 8: Use Case Actors I.....	15
Figure 9: Use Case Actors II.....	15
Figure 10: Use Case Diagram.....	16
Figure 11: Architecture Diagram.....	26
Figure 12: ERD.....	27
Figure 13: Data Dictionary (ERD).....	28
Figure 14: Data Flow Diagram Level 0.....	29
Figure 15: Data Flow Diagram Level 1.....	29
Figure 16: Class Diagram.....	30
Figure 17: Object Diagram.....	31
Figure 18: Sequence Diagram for Admin Login.....	32
Figure 19: Sequence Diagram for Admin Register.....	33
Figure 20: Sequence Diagram for Receptionist Login.....	34
Figure 21: Sequence Diagram for Receptionist Registration.....	35
Figure 22: Sequence Diagram for Customer Registration.....	36
Figure 23: Sequence Diagram for Customer Login.....	37
Figure 24: Sequence Diagram for Customer Add.....	38
Figure 25: Sequence Diagram for Customer Info.....	39
Figure 26: Sequence Diagram for Customer Update.....	40
Figure 27: Sequence Diagram for Available Room Checking.....	41
Figure 28: Sequence Diagram for Room Booking.....	42
Figure 29: Sequence Diagram for Payment.....	43
Figure 30: Sequence Diagram for Generate Payment	44
Figure 31: Sequence Diagram for Room Add.....	45
Figure 32: Sequence Diagram for Room Update.....	46
Figure 33: Sequence Diagram for Reservation Update.....	47
Figure 34: Sequence Diagram for Check In.....	48
Figure 35: Sequence Diagram for Customer Feedback.....	49
Figure 36: Sequence Diagram for Check Out.....	50
Figure 37: Sequence Diagram for Reservation Delete.....	51
Figure 38: Activity Diagram for User Login.....	52
Figure 39: Activity Diagram for Admin Login.....	53
Figure 40: Activity Diagram for Manager Login.....	54
Figure 41: Activity Diagram for Receptionist Login.....	55
Figure 42: Activity Diagram for Customer Login.....	56
Figure 43: Activity Diagram for Manage Activity.....	57
Figure 44: Activity Diagram for Customer Info.....	58
Figure 45: Activity Diagram for Room Properties.....	59

Figure 46: Activity Diagram for Customer Add.....	60
Figure 47: Activity Diagram for Room Search.....	61
Figure 48: Activity Diagram for Rooms Available	62
Figure 49: Activity Diagram for Add Reservation.....	63
Figure 50: Activity Diagram for Make Booking.....	64
Figure 51: Activity Diagram for Make Reservation.....	65
Figure 52: Activity Diagram for Make Payment.....	66
Figure 53: Activity Diagram for Reservation Update.....	67
Figure 54: Activity Diagram for Delete Booking.....	68
Figure 55: Activity Diagram for Delete Reservation.....	69
Figure 56: Activity Diagram for Delete Room.....	70
Figure 57: Activity Diagram for Customer Feedback.....	71
Figure 58: Activity Diagram for Delete Customer.....	72
Figure 59: Activity Diagram for Update Room.....	73
Figure 60: Activity Diagram for Logout.....	74
Figure 61: Collaboration Diagram.....	75
Figure 62: State Transition Diagram.....	76
Figure 63: Component Diagram.....	77
Figure 64: Deployment Diagram.....	78
Figure 65: Database Architecture.....	79
Figure 66: 3-tier Architecture.....	80
Figure 67: Database Table for Users.....	80
Figure 68: Database Table for Webinfo.....	81
Figure 69: Database Table for Rooms.....	81
Figure 70: Database Table for Posts.....	82
Figure 71: Database Table for Food.....	82
Figure 72: Database Table for Customer.....	83
Figure 73: Database Table for Booking.....	83
Figure 74: Statement Coverage I.....	100
Figure 75: Statement Coverage II.....	100
Figure 76: Branch Coverage.....	101
Figure 77: Path Coverage.....	101
Figure 78: Code for index page.....	105
Figure 79: Code for Navbar.....	105
Figure 80: Code for Services.....	106
Figure 81: Code for Pricing.....	106
Figure 82: Code for Booking.....	107
Figure 83: Code for Payment.....	107
Figure 84: Code for Payment Save.....	108
Figure 85: Code for Add Comment.....	108

Table

Name	Date	Reason For Changes	Version

Chapter No 1: Introduction to the Problem

1.1 Introduction

The project Hotel Management System is a website that allows the Hotel Manager and Owner to handle all hotel activities easily and safely. The HMS project provides room booking, staff management and other necessary hotel management features. This System allows the manager to post available rooms in the system. Customers can view and book rooms online. Admin has the power of either approving or disapproving the customer's booking request. This Software Requirement Specification document provides a complete description of all the functionalities and the specifications of the Online Hotel Management System. This will allow a complete understanding of what is to be expected from the newly introduced system which is to be constructed. The clear understanding of the system and its functionality will allow correct software to be developed for the end user and will be used for the development of the future stages of the project. The developers and the testers can use this document as a reference for developing the design and test plan documents. Using interactive GUI anyone can easily access to our website.

Project Name:	<u>Hotel Management System</u>	Starting Date:	<u>12/06/2021</u>
Controlling Agency:	<u>Superior College SKP</u>	Final Date:	<u>25/09/2021</u>
Prepared By:	<u>ABBAS KAZMI, AMNA MALIK, M.FAIZAN</u>		Authorized by: <u>GCUF</u>

1.2 Project Background

Since the emergence of electronic computer and systems, decision making and processing of information has now become so very easy. Data can be now be stored in computer in a considerable low space and retrieved within a short period of time compared with the manual method which is tedious and time consuming as the size of the file increases. Even though, the ability of computer to store, retrieve and process data can also be done by human but the major difference is that computer can reliably execute millions of instructions within a nanosecond and store the result while it takes longer period of time for human being to execute. Our Hotel Management System is a software system where the management of entire hotel is computerized. The system stores customer record and daily activities performed in the hotel such as customer details, reservation details, creating a new room, vacating the rooms etc, all these are computerized and is done without any difficulty. It also generate report for authorized user on a daily basis and a weekly report. Therefore, All of these functions reduce time and stress undergone using the manual method.

1.3 Project Purpose

The purpose of this SRS is to present a detailed Online Hotel Management System. It will be explaining the different functional as well as non functional requirements of the system, the interfaces of the system, what the system will do or how the system will interact with the external users, the constraints under which it will operate. This Software Requirement Specification document will provide a clear understanding of what is expected by the client in the proposed Online Hotel Management System. This will give a clear idea on how the software should be developed by the development team for the end users. This SRS will provide a solid base or foundation for the project. From this SRS, the designers can design low level design documents and the testers can create test plans and various test case documents. The system would be very easy to use and user friendly. Using this HMS Manger does not have to sit and manage the entire activities on paper, and at the same time Owner of the Hotel will feel comfortable to keep check on Hotel easily from anywhere around the world, This system will give them power and flexibility to manage the entire system from a single portal.

This SRS will provide the foundation for our project. From this SRS, the Hotel Management System can be designed, constructed and finally tested. The main goal of the project is to design a scalable and extensible system for managing the hotel activities. The system will be designed with the user-centric approach that will ensure that the user requirements mentioned in the documents must be full-filled and must confirms to the required standards. The new proposed system will operate efficiently by eliminating all the time consuming issues and provides a better and much enhanced services to the customers as well as the management and staff. This SRS will then be used by the system development team which is constructing the HMS and the hotel end users. The project team will use the SRS to fully understand the expectations of this HMS to construct the appropriate software. The hotel end users can use this SRS as a test to see if the constructing team is constructing the system upto their expectations. If not then the end users can specify how it is not to their liking and the team will change the SRS to fit for the end users.

1.4 Project Scope

The objectives of our system 'Hotel Management System' will be to help the customers of the hotel to reserve rooms and other facilities of the hotel from anywhere around the world. The core part of the project is the reservation and the booking system to keep track of the reservations and room availability. There are four types of the end users for this Hotel Management System. The first ones are the customer who uses the system for the reservation purposes. The other end users are the admin user and the receptionist users and management users who are given separate authentication to the Hotel Management System. The booking module is used to reserve the hotel rooms. The customer can book the rooms through online or through phone or in person. The customer needs to enter their personal details and have to pay 10 percentage of the total payment to confirm the booking. An email containing the confirmation details will be sent to the customer's email address. When the customer books the room through telephone or in person, the staff members of the hotel need to enter the customer details by logging to employee modules with corresponding authorization.

The world is changing so the scope of hotel management system would increase as well. The software would be used in multi ways and system will automate the major hotel operations such as generating billing and keeping record of daily transaction. The reservation system is to keep track I room and hall reservation and check availability. The system also set rates of rooms. The system will have multiple room details according to their class and categories.

1.5 Project Objective

- The objective of HMS is to develop a project which maintains booking of rooms and handles accounts of a hotel.
- The system will be so simple and attractive which will make the customer comfortable to use and choose their ideal room.
- The system will induce simplicity to ensure that people are attracted and thus understand the system easily.
- The System will support multi-user environment.
- It will be a web based website which is to be very easy to use and accessible anytime worldwide.
- Customers can view and book an available room.
- The system allows the Owner to check the progress of the hotel from interactive Graphs and will notified to each new change made in system.
- The system allows the manager to keep track on available rooms in system.
- The system should be capable to keep track of all the detailed descriptions of the clients.
- The Manager can easily check all types of reports when required.
- The system keeps real time records of all activities including accounting.

Website Goals	Project Objectives
Fully helping and supportive system for customers.	Enable smooth transition to centralized web based hotel information management system
Documented the operations to improve quality.	Demonstrate how to improve operational process through centralized web based hotel information management system features and functions
Direct and easy booking to multiple hotel revenues.	Producing a very simple, basic, user friendly, and closest to the standard Pakistan hotel system kind of a system.
Easy helping guide for any query in the system.	Facilitate the users to book the room in advance and then fulfill their required demands and standards in system.

Table 1: Project Objectives

1.5 Intended Audience and Reading Suggestions

The intended audiences are:

- **Administrator:**

Admins are the one who manage the overall functions of the system.

- **Users:**

This document is intended for the help of users who are customers or hotel staff. It gives an overall description of the system which may be helpful for the users. This is helpful for users to know much more clear about the system.

1.6 Document Conventions

This document is prepared using Microsoft Word 2007 and has used the font type 'Times New Roman'. It has used the bold property to set the headings or sub headings but has used the normal property for body of the document. The diagrams have been created according Visio.

- **Convention for Main Title:**
 - Font Face : Times New Roman
 - Font Style: Bold
 - Font Size: 16
- **Convention for Subtitle:**
 - Font Face: Times New Roman
 - Font Style: Bold
 - Font Size: 14
- **Convention for Body:**
 - Font Face: Times New Roman
 - Font Style: Normal
 - Font Size: 12

Chapter No 2: Software Requirement Specification

2.1 Overall Description

2.1.1 Product Perspective

The Main Perspective of this project is to provide an online platform for booking of hotel's room with secure payment option and easily done through Cash/Easypaisa/Jazzcash. This HMS will provide an easy access to the system and will contain user friendly functions with interactive interface. The system will give better options for the problem handling large scale of physical file system, for the errors occurring in calculations and all other required tasks that has been specified by the client. The perspective of HMS is to develop software which maintains booking of room, and handles accounts of hotel. The final outcome of this document is to develop a system that increase the efficiency and manage the tasks of Hotel in a much convenient manner. This is basically to book before checking in to hotel, with the use of computers and fast growing digital networking payment will be done online and it will be no doubt very easy for people to book hotel anytime from anywhere.

2.1.2 Product Features

- **Admin:**
Admin is the one who checks the login info and views all the customer info along with the room records.
- **Customer:**
In this system, Customer registers and logs in to book the reservation and confirms the payment and the customer's check in/ check-out history is displayed respectively.
- **Receptionist:**
In this system, Receptionist can directly add or remove any customer if required. Receptionist has all the track of room availability, bill records and can ask the customers for feedback.
- **Room:**
In this system, room details are all visible on website along with customer info who is willing to check in. The updation of room occurs automatically in this system.
- **Room type:**
In this system, the info of room types are available and customer selects according to their budget or need.
- **Reservation:**
In this system, reservation info is displayed while booking the room and all the available rooms and reservations displayed respectively.
- **Transaction:**
The transaction for rooms and services is done after verifying the customer completely and the confirmation button confirms transaction in this system.
- **Payment:**
Payments can be done after reservation Jazzcash or Easypaisa or by hand.

2.1.3 Design and Implementation Constraints

Design Constraints:

The color schemes we are using in this system are:

- **Red**
- **Powder Blue**
- **Green**
- **Black**
- **White**

1. Red

In this System, Red color shows something which is highlighted as it shows some action or type of power. This color moreover is an appetite stimulant for a system.

2. Powder Blue

In this System, Powder Blue color shows trustworthiness and reliability. It means visitor trust on our website system and wants to get registered.

3. Green

In this System, Green color shows Growth. It is used for room booking page so users can login and book room and make the website grow.

4. Black

In this System, Black color represents strength and is used mostly for the font.

5. White

In this System, white color shows elegance minimalism and style. It is used to create space and to space and to move the attention to the important elements of the website and to maintain the site clean.

Implementation Constraints:

- **Username:** is a “String” which must contain small alphabets/letters.
- **Email:** must be a predefined format for e-mail which must include “@gmail.com” like email.
- **Mobile Number:** Only enter Integers with country code and then number, respectively.
- **Password:** must always use alphabetical characters mixed with numbers or letters or integers.

2.1.4 Assumptions and Dependencies

It is assumed that system developed will work perfectly that's going to be developed under the Windows OS, and Node JS server with React JS. We assume that users must be using windows 7 or windows 8. Otherwise, if users use an open source OS, there must be need to change SRS accordingly. Let's talk about Assumptions and dependencies of our system in detail as follows:

- The project should be completed within specified time period including Planning, Designing, Development, Testing and Deployment.
- The Requirement Traceability Matrix (RTM) should be correlated and completed.
- All the Entry and Exit criteria of all the stages should be met.
- The product should be user-friendly, reliable and should maintain the industry standards without compromising the quality.
- Each user must have a valid id and password.
- Users must login to the system to access their status.
- Only the Manager/Receptionist can update records.
- Graphical User Interface is dependable on English.
- Login and Password is used for identification of user and there is no facility for guest id.
- The system architecture and design should be open and in a standard way such that additional functionalities can be added later without much effort.
- We as a team will only make a website system, it is the responsibility of the user to set up their own hardware system to get access to the website system.

2.2 System Features

The system has capability of sorting basic customer contact information such as name, phone, etc. The system can modify the reservation likewise When a customer checks in the room number will be updated automatically in the database. The record for vacant room will be upfront. Most importantly the system will be able to check details in regards to the customer room, pricing room status etc. The system is controlled/managed by the Owner, Manager & Receptionist profiles. The system allows addition, deletion and modification of information on room & rates, menu item and prices, user profiles by Receptionist/Manager.

2.2.1 Login System

Description and Priority

Provides a user with a page to login the system or see the main panel.

Priority = 9

Stimulus/Response Sequences

Stimulus: User clicks on Login Link.

Response: Login Page is displayed.

Stimulus: User Enters Username and Password.

Response: Username and Password are validated from SQL Database.

Functional Requirements

REQ-1: The user shall be able to click the login link.

REQ-2: The user shall be able to enter his username and password.

REQ-3: The database shall be able to validate username and password and move the user to next page.

2.2.2 Make Reservations

Description and Priority

Provides a user with a page to make reservations and make activities associated with the system.

Priority = 8

Stimulus/Response Sequences

Stimulus: User clicks on booking link.

Response: Booking Page is displayed.

Stimulus: User confirms booking.

Response: Booking is confirmed.

Functional Requirements

REQ-1: The user shall be able to click the Booking Link.

REQ-2: The user shall be able to confirm the booking.

2.2.3 Add Payments

Description and Priority

Provides a user with a page to add payment methods

Priority = 7

Stimulus/Response Sequences

Stimulus: User clicks on Add Payments.

Response: Add Payments are displayed.

Stimulus: User confirms by selecting debit card option.

Response: Payment is done by debit card.

Functional Requirements

REQ-1: The user shall be able to do payment while booking.

REQ-2: The user shall be able to do payment with debit card.

2.2.4 View User Profile

Description and Priority

Provides a user with a page to be able to see user profile

Priority = 6

Stimulus/Response Sequences

Stimulus: User clicks on My Profile.

Response: User Profile page is displayed.

Functional Requirements

REQ-1: The user shall be able view user profile.

2.2.5 Logout

Description and Priority

Provides a user with an option to logout

Priority = 5

Stimulus/Response Sequences

Stimulus: User clicks on Logout.

Response: User is logged out and Index page is displayed.

Functional Requirements

REQ-1: The user shall be able to logout from system.

2.3 External Interface Requirements

2.3.1 User Interfaces

This Hotel Management System is designed for our customer's ease, which will allow our valuable customers to stay in luxury/premium hotels. And it will provide an easy and user friendly interface.

User Interface includes:

Login and sign up page, registration page, room services, homepage includes the rooms (allotted ones and available rooms). The feedback on bottom and many other useful options regarding our system.

1. Login
 - Username
 - Password
2. Registration
 - Username
 - Email
 - Phone Number
 - Password
3. Booking Form
 - Room Type
 - Room Number
 - Room Price
4. Payment Details
 - Cash payment
 - Easypaisa
 - Jazzcash
5. Redirect page
 - Takes back to homepage after payment success.
6. Feedback/Review page
 - Customers give their feedback after checking in.
 - Feedbacks visible on homepage for new coming people.

2.3.2 Hardware Interfaces

Operating System:

Supports all known Operating system like Windows.

Computer:

1GB+RAM with i3 core processor.

2.3.3 Software Interfaces

The screenshot displays the main interface of the Royal Sheikhpura Hotel website. The header includes navigation links: Home, Services, Gallery, Blog, Pricing, About us, and Contact us, along with a 'Call now' button. The main content area features a large background image of a hotel building with a thatched roof. On the left, a 'Welcome to Hotel' message is followed by the hotel's name 'Royal Sheikhpura Hotel' in large green letters, and the tagline 'Your Perfect Accommodation'. Below this, a paragraph states: 'Visit our Hotel to experience true relaxation from our services. We can meet even the most demanding expectations.' Two green checkmarks highlight 'Lots of great accommodation options' and '100% satisfaction guarantee'. On the right, a 'Make reservation' section contains a form with fields for Check-in, Check-out, No of Adults, No of Children, Type of room (Basic > Queen room), CNIC number, Contact number, and Food plans (None). Buttons for 'Terms & policy', 'View rooms', and 'Book now' are located below the form. A 'We Are Here!' badge is visible in the bottom right corner.

Figure 1: Main Screen

The screenshot shows the login interface of the Royal Sheikhpura Hotel. The header includes navigation links: Home, Services, Gallery, Blog, Pricing, About us, and Contact us, along with a 'Login' button. The main content area features a large background image of a hotel building with a thatched roof. On the left, the hotel's logo 'THE ROYAL LUXURY APARTMENTS' is displayed above the text 'Royal Sheikhpura Hotel' and 'Login'. Below this, a tagline 'We deliver trust' is visible. On the right, a login form contains fields for Email and Password, and a green 'Login' button.

Figure 2: Login Screen

localhost / 127.0.0.1 | phpMyAdmin x | .:Home:. x | .:Web Info:. x +

localhost/hotel/admin/ProfileWebInfo.php

Profile

Post

People

Gallery

Rooms

Service

Booking

Mails

Comments

Logout

Batti Chowk Sheikhpura

Royal Sheikhpura Hotel

We deliver trust

+923011234567

info@royal.com

Choose File No file chosen

Save

Figure 3: Profile Web-info

localhost / 127.0.0.1 | phpMyAdmin x | .:Home:. x | .:posts:. x +

localhost/hotel/admin/posts.php

Profile

Post

People

Gallery

Rooms

Service

Booking

Mails

Comments

Logout

All posts +Post

1 About our Royal Sheikhpura Hotel

2 Our Vision & Mission

Figure 4: Profile Posts

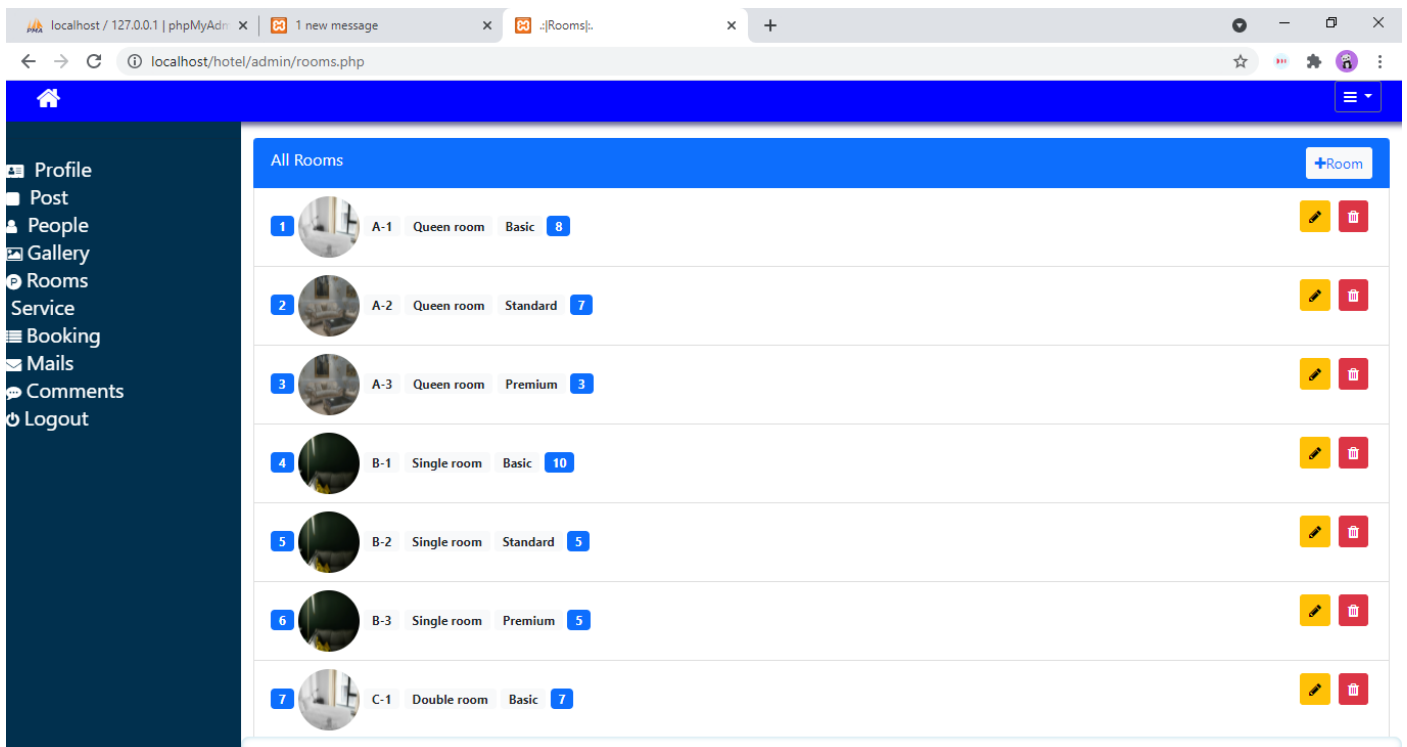


Figure 5: Rooms

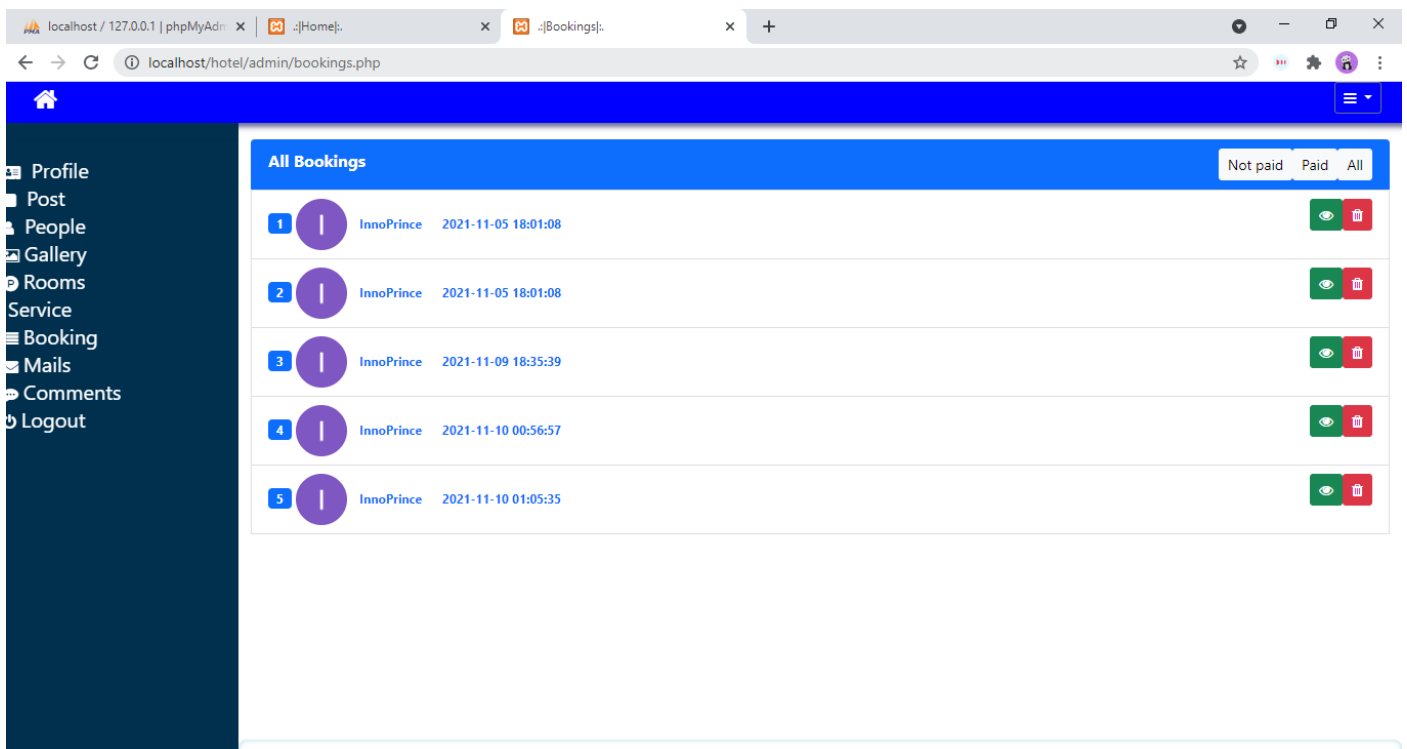


Figure 6: Bookings

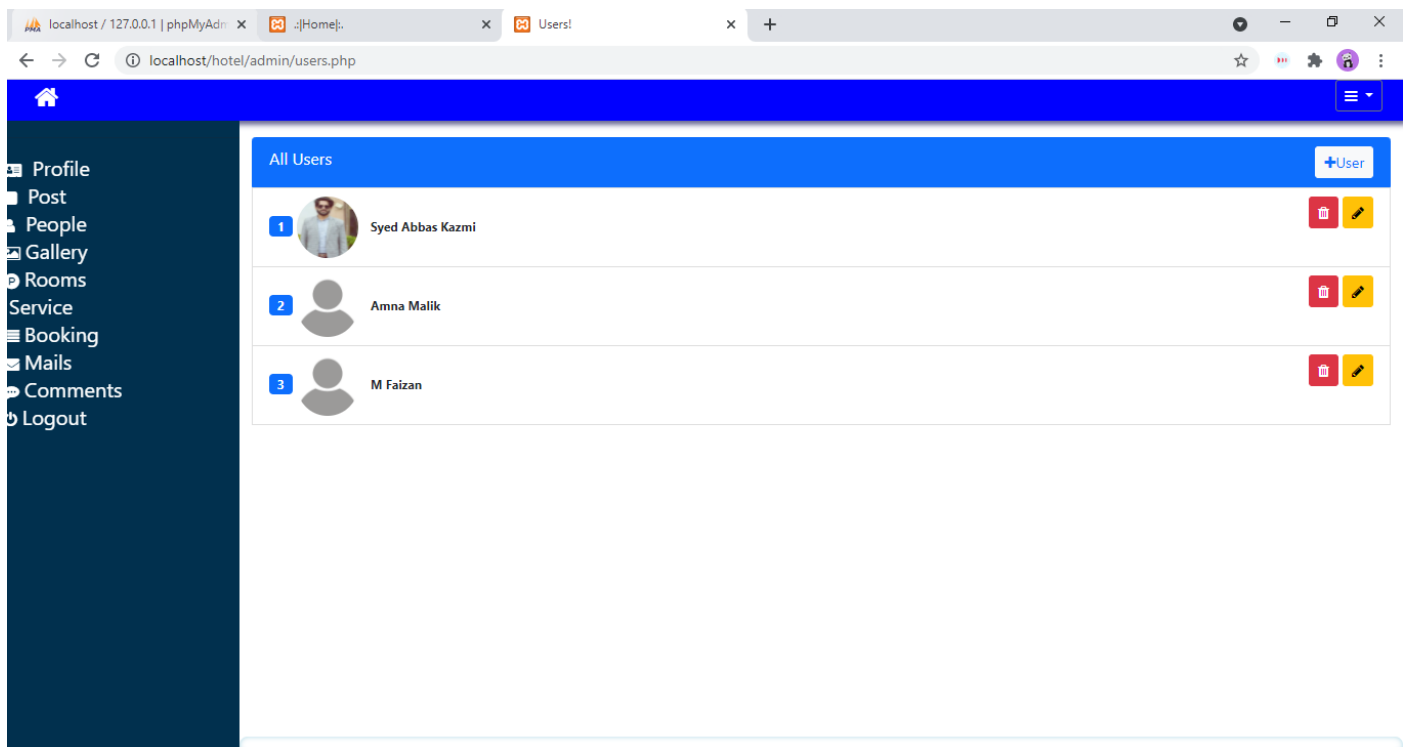


Figure 7: Users

HTML:

The **HyperText Markup Language**, or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript (JS).



CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

**jQuery:**

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. As of May 2019 jQuery is used by 73% of the 10 million most popular websites. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having at least 3 to 4 times more usage than any other JavaScript library.

**Sublime Text:**

Sublime Text is a commercial source code editor. It natively supports many programming languages and markup languages. Users can expand its functionality with plug-ins, typically community-built and maintained under free-software licenses. To facilitate plug-ins, Sublime Text features a Python API.



XAMPP Server:

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria-DB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

**2.3.4 Communication Interfaces**

To achieve that functionality, it requires having a stable internet connection. Mostly a broadband connection with the client computer will provide the efficient service. The system shall be using HTTP/HTTPS for communication over Internet.

- Using LAN (local area network)
 - With server computer and internet

2.4 Other Nonfunctional Requirements**2.4.1 Performance Requirements**

Performance requirements define acceptable response time for system functionality. Although the system is developed suiting for the least system performances, the performance of the system will highly depend on the performance of the hardware and software components of the system will highly depend on the performance of the hardware and software components of the installing computer.

2.4.2 Safety Requirements

There are several user levels in HMS. Access to the various subsystems will be protected by a user log in screen that requires a user name and password. This gives different views and accessible functions of user levels through the system. Maintaining backups ensure the system security. System can be restoring in any case of emergency.

2.4.3 Security Requirements

Owner will be able to log in to the HMS. Owner has the maximum privilege to all subsystems. Access to the various subsystems will be protected by a user log in screen that requires a user name and password. Payment Process will use HTTP over Secure protocols to secure the payment transactions.

Chapter No 3: Analysis (Use Case Model)

3.1 Identifying Actors and Use Cases using Textual Analysis

Every user it may be Admin, Manager, Customer, Receptionist. They must have at least login. Admin can view-financial report. Admin check room availability Manager can add room delete room. Manager can also Edit room property. Receptionist can check Check-in, Check-out of customer. Receptionist can add reservation also delete reservation. Receptionist can add customer one or more at a time and also delete customer. Customer can search room one or more at a time. Customer can make booking also can delete booking. Customer can check in check out from hotel. Admin, Manager, Receptionist, Customer logout from the system after performing task.

No.	Candidate Class	Extracted Text	Type	Description	Occurrence	Highlight
1	Admin	Admin	Actor	Admin has all access on the system.	4	
2	Manager	Manager	Actor	Manager has limited access on the system.	4	
3	Receptionist	Receptionist	Actor	Receptionist has limited access on the system.	5	
4	Customer	Customer	Actor	Customer has limited access on the system.	5	
5	login	login	Use Case	Admin, Manager, Receptionist, Customer login into the system.	1	
6	view-financial	view-financial	Use Case	Admin can view financial report.	1	
7	check room availab	check room availability	Use Case	Admin can check room availability.	1	
8	add room	add room	Use Case	Manager can add room.	1	
9	delete room	delete room	Use Case	Manager can delete room.	1	
10	Edit room property	Edit room property	Use Case	Manager can edit room property.	1	
11	Check-in	Check-in	Use Case	Receptionist can check check-in.	1	
12	Check-out	Check-out	Use Case	Receptionist can check check-out.	1	

Figure 8: Use Case Actors I

Every user it may be Admin, Manager, Customer, Receptionist. They must have at least login. Admin can view-financial report. Admin check room availability Manager can add room delete room. Manager can also Edit room property. Receptionist can check Check-in, Check-out of customer. Receptionist can add reservation also delete reservation. Receptionist can add customer one or more at a time and also delete customer. Customer can search room one or more at a time. Customer can make booking also can delete booking. Customer can check in check out from hotel. Admin, Manager, Receptionist, Customer logout from the system after performing task.

No.	Candidate Class	Extracted Text	Type	Description	Occurrence	Highlight
9	add room	add room	Use Case	Manager can add room.	1	
10	delete room	delete room	Use Case	Manager can delete room.	1	
11	Edit room property	Edit room property	Use Case	Manager can edit room property.	1	
12	Check-in	Check-in	Use Case	Receptionist can check check-in.	1	
13	Check-out	Check-out	Use Case	Receptionist can check check-out.	1	
14	add reservation	add reservation	Use Case	Receptionist can add reservation.	1	
15	delete reservation	delete reservation	Use Case	Receptionist can delete reservation.	1	
16	add customer	add customer	Use Case	Receptionist can add customer.	1	
17	search room	search room	Use Case	Customer can search room.	1	
18	make booking	make booking	Use Case	Customer can make booking.	1	
19	check in	check in	Use Case	Customer can check in from hotel.	1	
20	check out	check out	Use Case	Customer can check our from hotel.	1	
21	logout	logout	Use Case	Admin, Manager, Receptionist, Customer logout from system.	1	

Figure 9: Use Case Actors II

3.2 Forming Use Case Diagram with Candidate and Use Cases

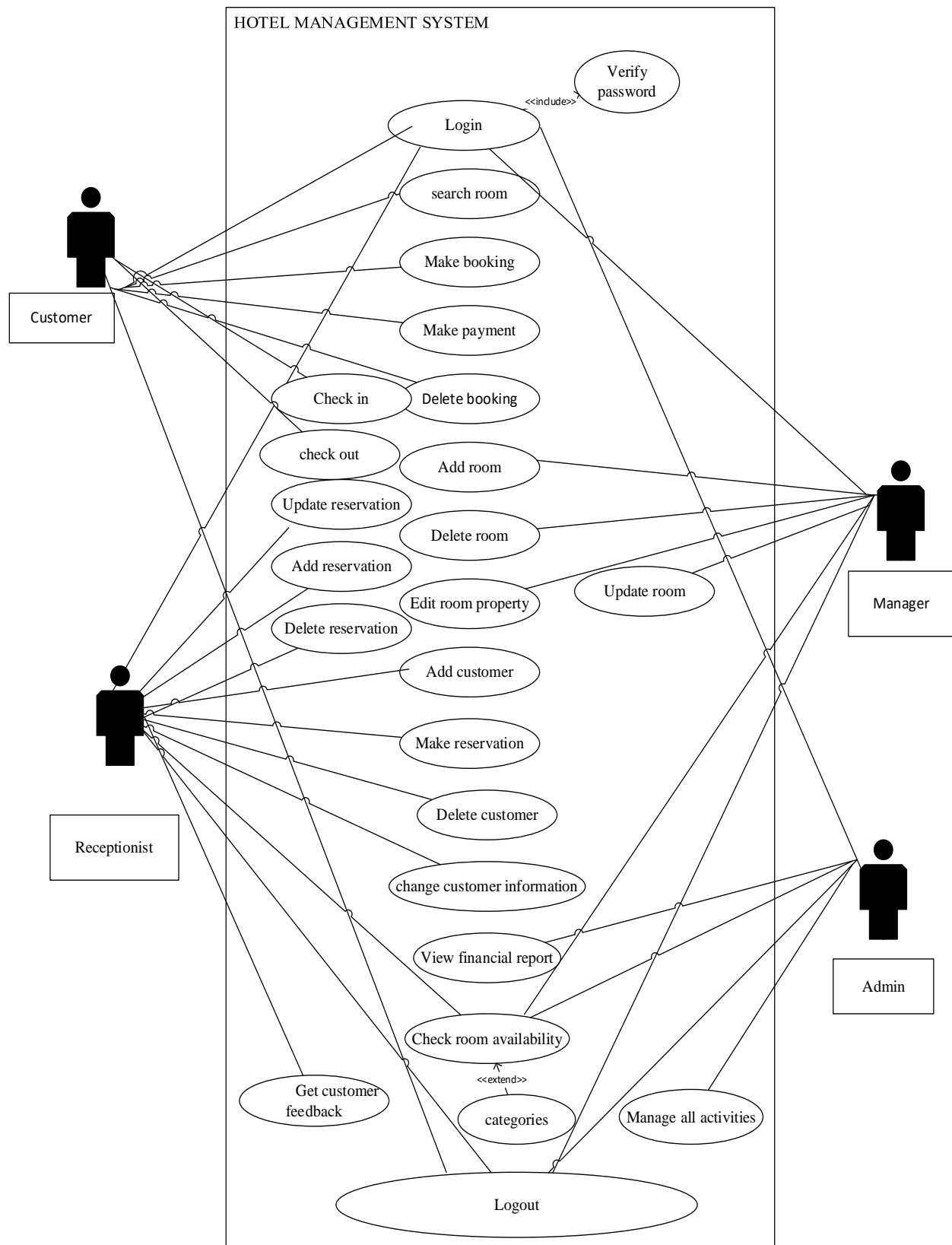


Figure 10: Use Case Diagram

Use Case Description

Use case 1	
Use case Title:	Login
Actor:	Admin, Manager, Customer, Receptionist.
Pre-Condition:	User is registered in a system.
Post-Condition:	User is registered and logged in.
Exception:	User enters Invalid Password.
Alternative:	<ol style="list-style-type: none"> 1. User and service provider should re-enter correct password. 2. User should have good internet connection.

Table 2: Use Case for Login

Use case 2	
Use case Title:	View Financial Report
Actor:	Admin
Pre-Condition:	Already Login to the system.
Post-Condition:	View financial report for specific time.
Exception:	If invalid details entered. Display “Unsuccessful” message.
Alternative:	<ol style="list-style-type: none"> 1. Use should have good internet connection. 2. Enter correct date for viewing report.

Table 3: Use Case for Financial Reports

Use case 3	
Use case Title:	Check room Availability
Actor:	Admin, Manager, Receptionist, Customer.
Pre-Condition:	Already Login to the system.
Post-Condition:	Check whether a room available or not.
Exception:	No room available for searched room.
Alternative:	<ol style="list-style-type: none"> 1. Those rooms should be searched which are available in system.

Table 4: Use Case for Room Checking

Use case 4	
Use case Title:	Add room
Actor:	Manager
Pre-Condition:	Already Login to the system.
Post-Condition:	New room added to system.
Exception:	Room details incorrect. Display the message “Unsuccessful”
Alternative:	User should re-enter room details.

Table 5: Use Case for Add Room

Use case 5	
Use case Title:	Delete Room
Actor:	Manager
Pre-Condition:	Already Login to the system.
Post-Condition:	Room deleted from system.
Exception:	Selected wrong room no for deletion.
Alternative:	<ol style="list-style-type: none"> 1. Show confirmation message for deletion with room no. 2. Internet should be stable while processing deletion.

Table 6: Use Case for Delete Room

Use case 6	
Use case Title:	Edit room Properties
Actor:	Manager
Pre-Condition:	Already Login to the system.
Post-Condition:	Added properties such as view or type of room.
Exception:	“Edit room properties” option is not working due to bad internet connection.
Alternative:	Internet connection should be stable while editing room details.

Table 7: Use Case for Edit Room prop

Use case 7	
Use case Title:	Search Room
Actor:	Customer
Pre-Condition:	Already login to the system.
Post-Condition:	The required room displayed to customer.
Exception:	Rooms not being searched.
Alternative:	Customer should have good internet speed to search rooms.

Table 8: Use Case for Search Room

Use case 8	
Use case Title:	Make Booking
Actor:	Customer
Pre-Condition:	Already login to the system.
Post-Condition:	Room is booked by customer.
Exception:	Customer entered wrong room no for booking.
Alternative:	1. User re enter right room no.

Table 9: Use Case for Make Booking

Use case 9	
Use case Title:	Make Payment
Actor:	Customer
Pre-Condition:	Already login to the system.
Post-Condition:	Room is booked by Customer and Payment has been payed by customer.
Exception:	1. Customer has not enough money in debit card. 2. Customer entered incorrect card information.
Alternative:	1. Customer should recharge his card or change it. 2. Re-enter Card information.

Table 10: Use Case for Make Payment

Use case 10	
Use case Title:	Delete Booking
Actor:	Customer
Pre-Condition:	Already login to the system. Customer deleted the booking.
Post-Condition:	Booking deleted for Customer.
Exception:	Customer did not booked any room.
Alternative:	Customer should book a room first.

Table 11: Use Case for Delete Booking

Use case 11	
Use case Title:	Add Customer
Actor:	Receptionist
Pre-Condition:	Already Login to the system.
Post-Condition:	New customer added to system.
Exception:	Customer details are incorrect.
Alternative:	The details of customer should be taken correctly.

Table 12: Use Case for Add Customer

Use case 12	
Use case Title:	Make Reservation
Actor:	Receptionist
Pre-Condition:	Already login to the system. Click Make reservation.
Post-Condition:	Hotel guest details updated to include current guest.
Exception:	Rooms not available for reservation.
Alternative:	There should be a waiting room for customer.

Table 13: Use Case for Make Reservation

Use case 13	
Use case Title:	Delete Customer
Actor:	Receptionist
Pre-Condition:	Already Login to the system. Click Delete Customer.
Post-Condition:	Customer has been deleted.
Exception:	Customer's information is in the db since very long time.
Alternative:	Customer did not return to the hotel.

Table 14: Use Case for Delete Customer

Use case 14	
Use case Title:	Change Customer Information
Actor:	Receptionist
Pre-Condition:	Already Login to the system.
Post-Condition:	Customer details edited.
Exception:	Invalid details entered.
Alternative:	Use customer's valid information.

Table 15: Use Case for Change info

Use case 15	
Use case Title:	Logout
Actor:	Admin, Manager, Customer, Receptionist.
Pre-Condition:	Click on logout.
Post-Condition:	User logged out from system.
Exception:	User could not logout.
Alternative:	Internet connection should be stable for logging out.

Table 16: Logout

3.3 Describe the Events Flow for Use Case

Admin Login

1. Admin click on login button
 2. **SYSTEM** Login page displayed
 3. **for each** Admin login
 - 3.1. Enter password
 - 3.2. Enter name
 - 3.3. **SYSTEM** successfully login**end for each**
-

Manager Login

1. Manger click on login button
2. **SYSTEM** Login page displayed
3. **for each** manger login
 - 3.1. Enter password
 - 3.2. Enter name
 - 3.3. **SYSTEM** successfully login**end for each**

Receptionist Login

1. Receptionist click on login button
 2. **SYSTEM** Login page displayed
 3. **for each** Receptionist login
 - 3.1. Enter password
 - 3.2. Enter name
 - 3.3. **SYSTEM** successfully login**end for each**
-

Customer Login

1. Customer click on login button
 2. **SYSTEM** Login page displayed
 3. **for each** customer login
 - 3.1. Enter password
 - 3.2. Enter name
 - 3.3. **SYSTEM** successfully login**end for each**
-

Add Room

1. Manager click on rooms button
2. **SYSTEM** room list display
3. **for each** Add room
 - 3.1. User insert the room code
 - 3.2. Click save
 - 3.3. **SYSTEM** Successfully add room
- end for each**

Delete Room

1. Manager click on rooms button
2. **SYSTEM** room list display
3. **for each** Delete room
 - 3.1. Select room from list
 - 3.2. Click Delete
 - 3.3. **SYSTEM** Show Confirmation box "Are you sure"
 - 3.4. Click yes
 - 3.5. **SYSTEM** Successfully delete
- end for each**

Update Room

1. Receptionist click on booking button
2. **SYSTEM** Display room list
3. **for each** Do some changing
 - 3.1. click update button
 - 3.2. **SYSTEM** Successfully update
- end for each**

Search Room

1. Customer click on search room button
2. **SYSTEM** Search room list display
3. **for each** search room
 - 3.1. User insert search room
 - 3.2. Click save
 - 3.3. **SYSTEM** Successfully search room
- end for each**

Check Room Availability

1. Admin click on room availability button.
2. **System** check room availability display.
3. **for each** check room availability.
 - 3.1 User insert the room available
 - 3.2 Click **save**
 - 3.3 **System** successfully check room availability **end for each**

Add Reservation

1. Receptionist click on rooms button
2. **SYSTEM** room list display
3. **for each** Add reservation
 - 3.1. User insert the reservation
 - 3.2. Click save
 - 3.3. **SYSTEM** Successfully add reservation**end for each**

Make Payment

1. Customer click on room payment button
2. **SYSTEM** payment page displayed
3. **for each** make payment
 - 3.1. User insert the payment
 - 3.2. click save
 - 3.3. **SYSTEM** Successfully make payment**end for each**

Delete Reservation

1. Receptionist click on rooms button
 2. **SYSTEM** customers list display
 3. **for each** Delete reservation
 - 3.1. Select customer from list
 - 3.2. Click delete
 - 3.3. **SYSTEM** show confirmation box "Are you sure"
 - 3.4. Click yes
 - 3.5. **SYSTEM** successfully delete**end for each**
-

Delete Customer

1. Receptionist click on rooms button
2. **SYSTEM** customers list display
3. **for each** Delete customer
 - 3.1. Select customer from list
 - 3.2. Click delete
 - 3.3. **SYSTEM** show confirmation box "Are you sure"
 - 3.4. Click yes
 - 3.5. **SYSTEM** successfully delete
- end for each**

View Financial Reports

1. Admin click on view report
2. **SYSTEM** Check the Customer
3. **for each** Click report
 - 3.1. **SYSTEM** Display report
- end for each**

Chapter no 4: Design

4.1 Architecture Diagram

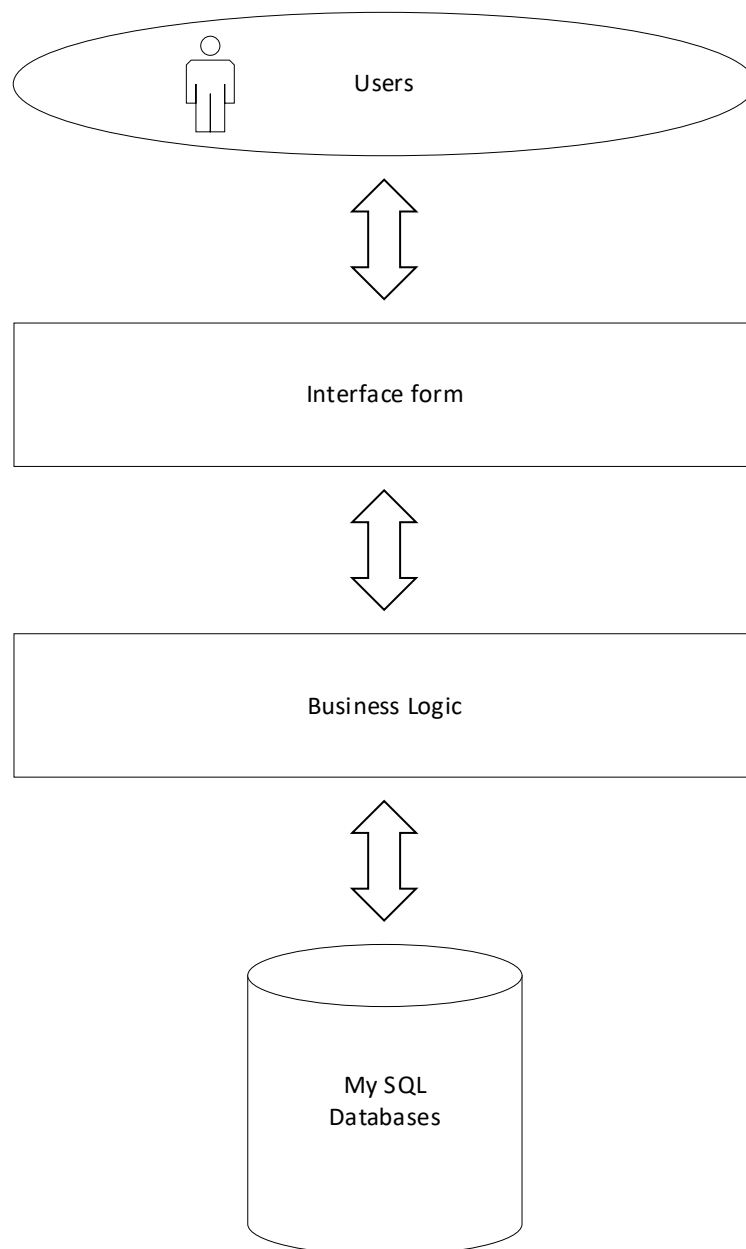


Figure 11: Architecture Diagram

Description:

In HMS, architecture diagram describe what we are building and how our stakes interacting. This also shows where constraints lie.

4.2 ERD with Data Dictionary

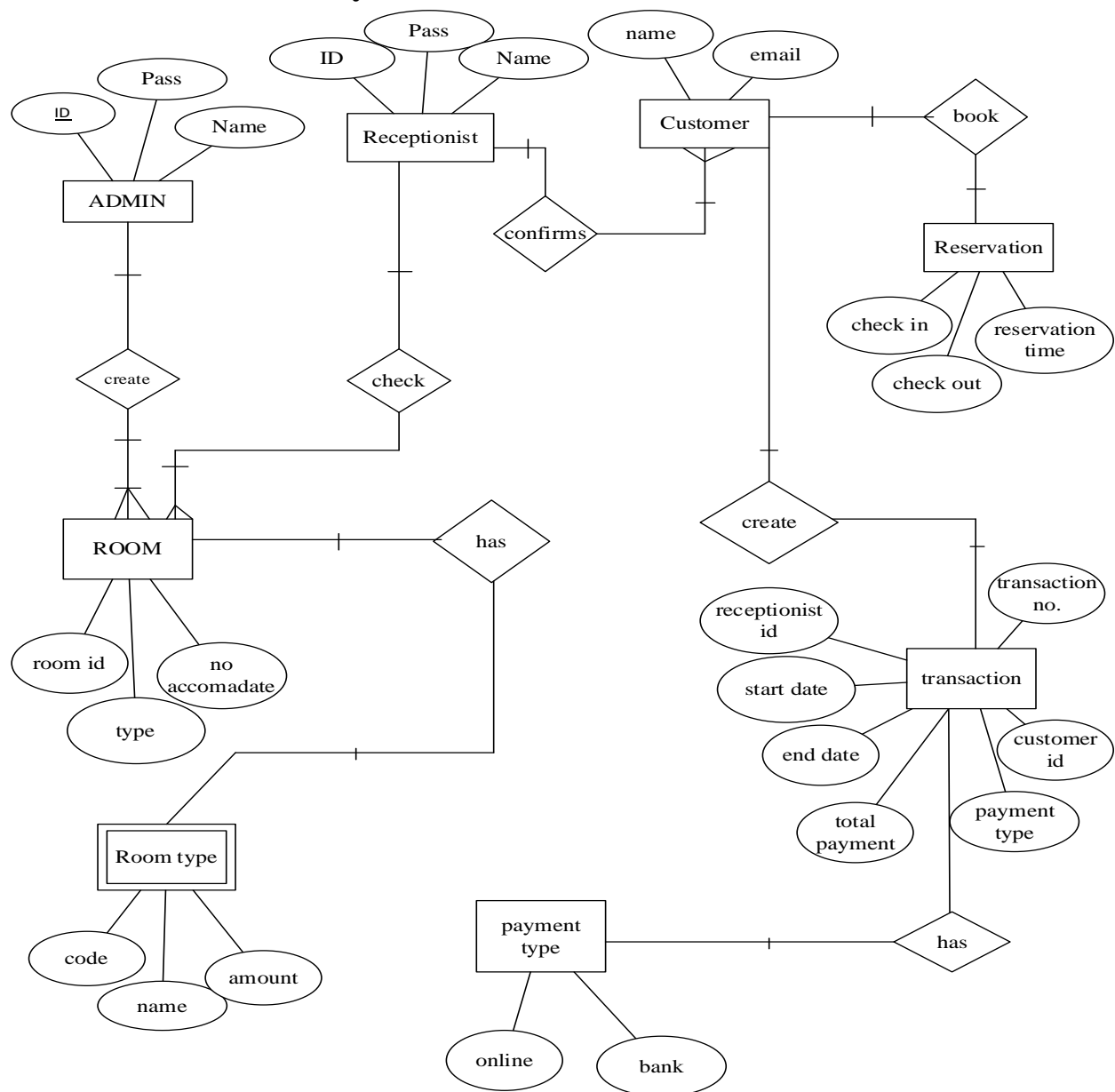


Figure 12: ERD

Relationship	
Entity	
Primary key	
Weak entity	
Strong entity	
Attribute	

Data Dictionary

The screenshot shows the phpMyAdmin interface for the 'webdata' database. The left sidebar lists several databases, with 'webdata' selected. The top navigation bar includes options like Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, and Triggers. The main area displays a table with 16 rows, each representing a table in the database. Each row includes a checkbox, the table name, a star icon, and several action icons (Browse, Structure, Search, Insert, Empty, Drop). The table also shows the number of rows, the storage engine (InnoDB), the character set (utf8mb4), the collation (general_ci), the size (16.0 KiB), and the overhead (0 B).

Table	Action	Rows	Type	Collation	Size	Overhead
booking	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
category	★ Browse Structure Search Insert Empty Drop	93	InnoDB	utf8mb4_general_ci	16.0 KiB	-
customer	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
erd	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
feature	★ Browse Structure Search Insert Empty Drop	12	InnoDB	utf8mb4_general_ci	16.0 KiB	-
food	★ Browse Structure Search Insert Empty Drop	39	InnoDB	utf8mb4_general_ci	16.0 KiB	-
gallery	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
newsletter	★ Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
posts	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
profiles	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
room	★ Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	16.0 KiB	-
tbl_comment	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
team	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
usermails	★ Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
users	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
webinfo	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
16 tables Sum		188	InnoDB	utf8mb4_general_ci	256.0 KiB	0 B

Figure 13: Data Dictionary (ERD)

4.3 Data Flow Diagram (Level 0 and Level 1)

Data Flow Diagram Level 0

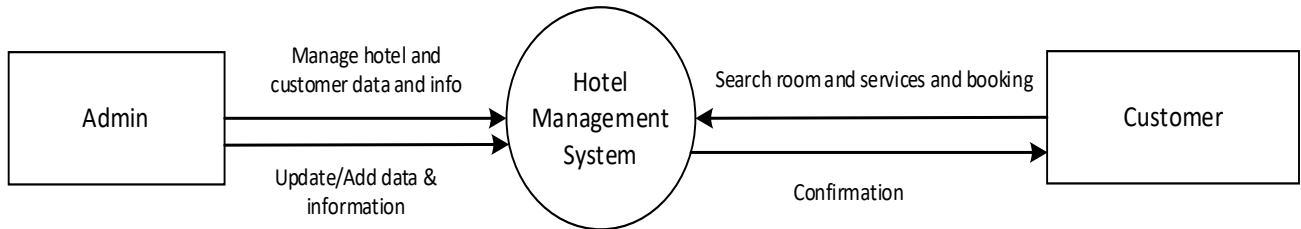


Figure 14: Data Flow Diagram level 0

Data Flow Diagram Level 1

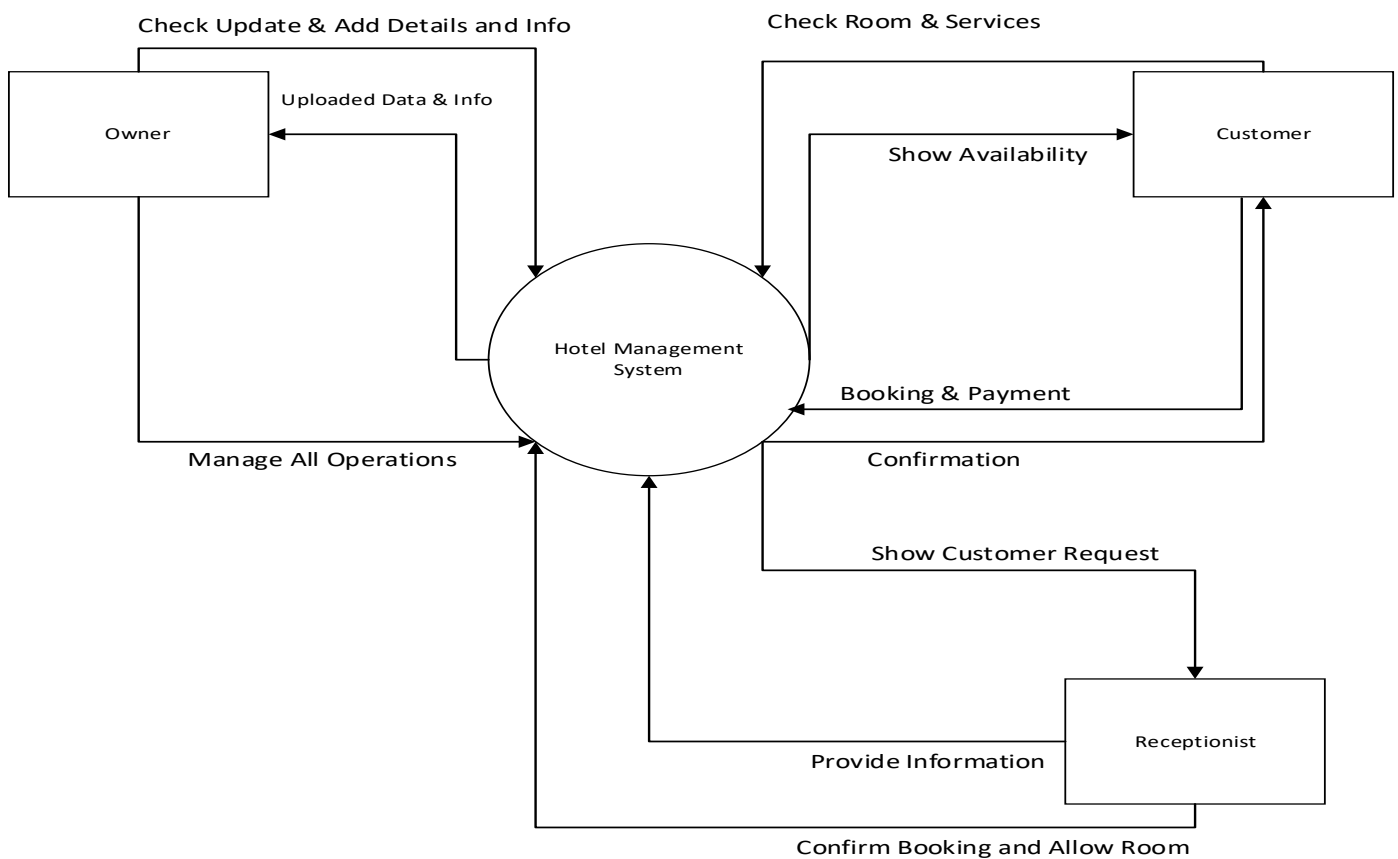


Figure 15: Data Flow Diagram level 1

Description:

In HMS, DFD shows flow of our data in a complete process. Our admin and customer are always to be linked in our system. Our Owner, customer and receptionist all links to each other shown in this diagram.

4.4 Class Diagram



Figure 16: Class Diagram

Description:

The class diagram in HMS describes the structure of HMS classes, their attributes, operations and the relationship among them. The classes are admin, receptionist, costumer, payment, reservation, transaction room and room types.

4.5 Object Diagram

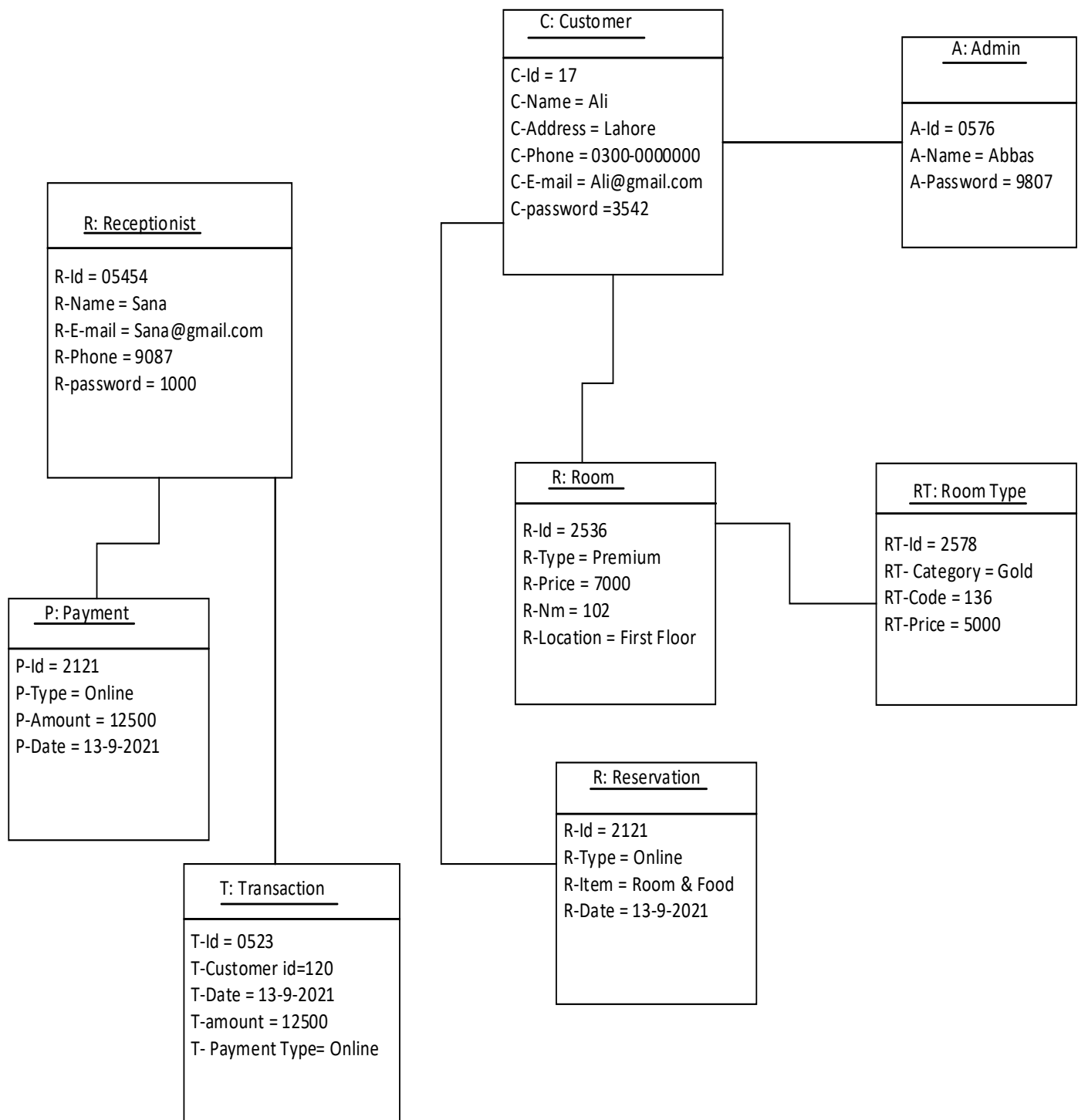


Figure 17: Object Diagram

Description:

In HMS, this diagram represents instances, including objects and data values.

4.6 Sequence Diagram

Sequence Diagram for Admin login

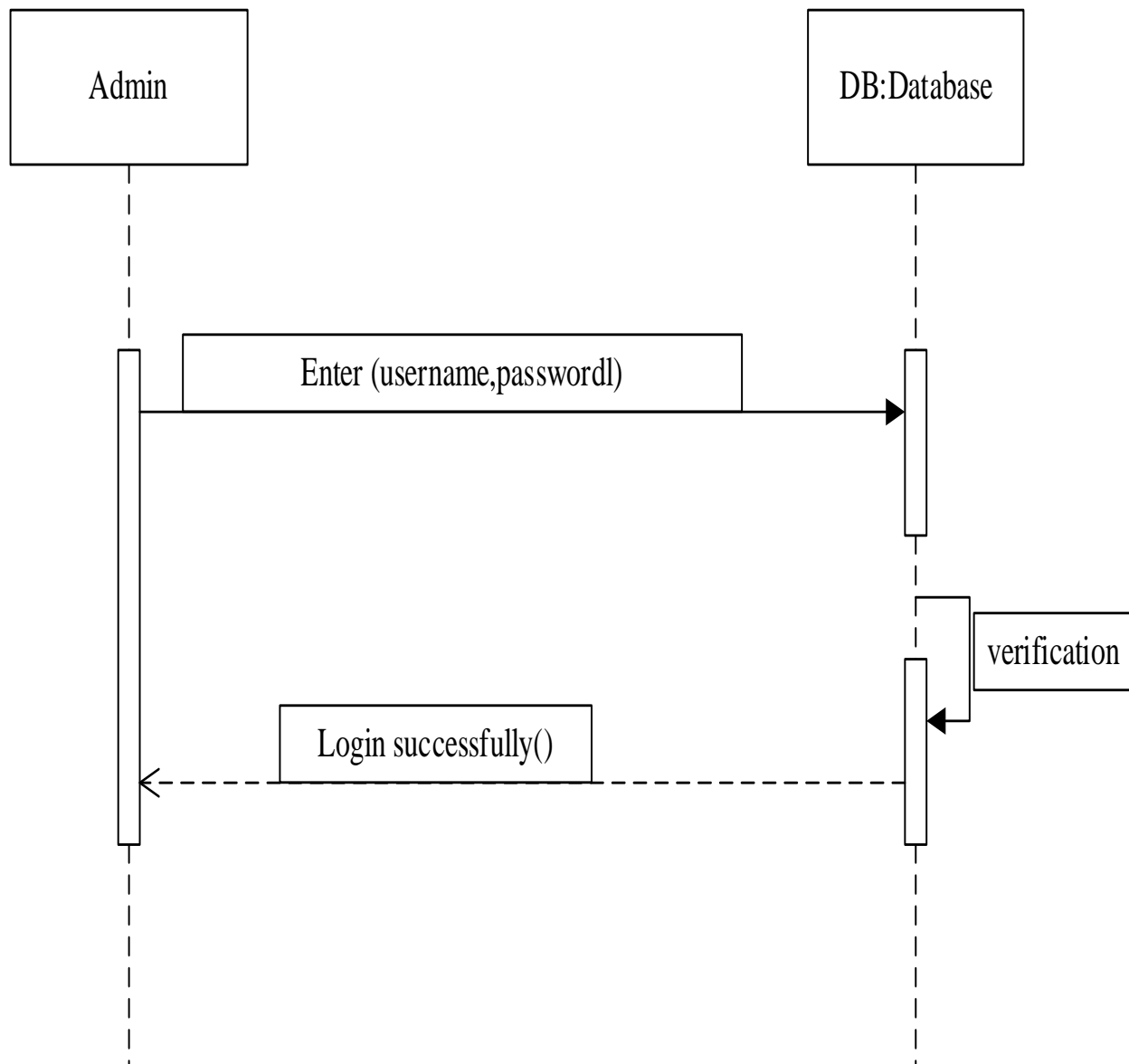


Figure 18: Admin Login

Description:

Users login to the system. System gives login detail to the database. Database does some operations and give the required login page. User enters the correct information. System gives authentication and pass message to database. Database verifies the user login to the system.

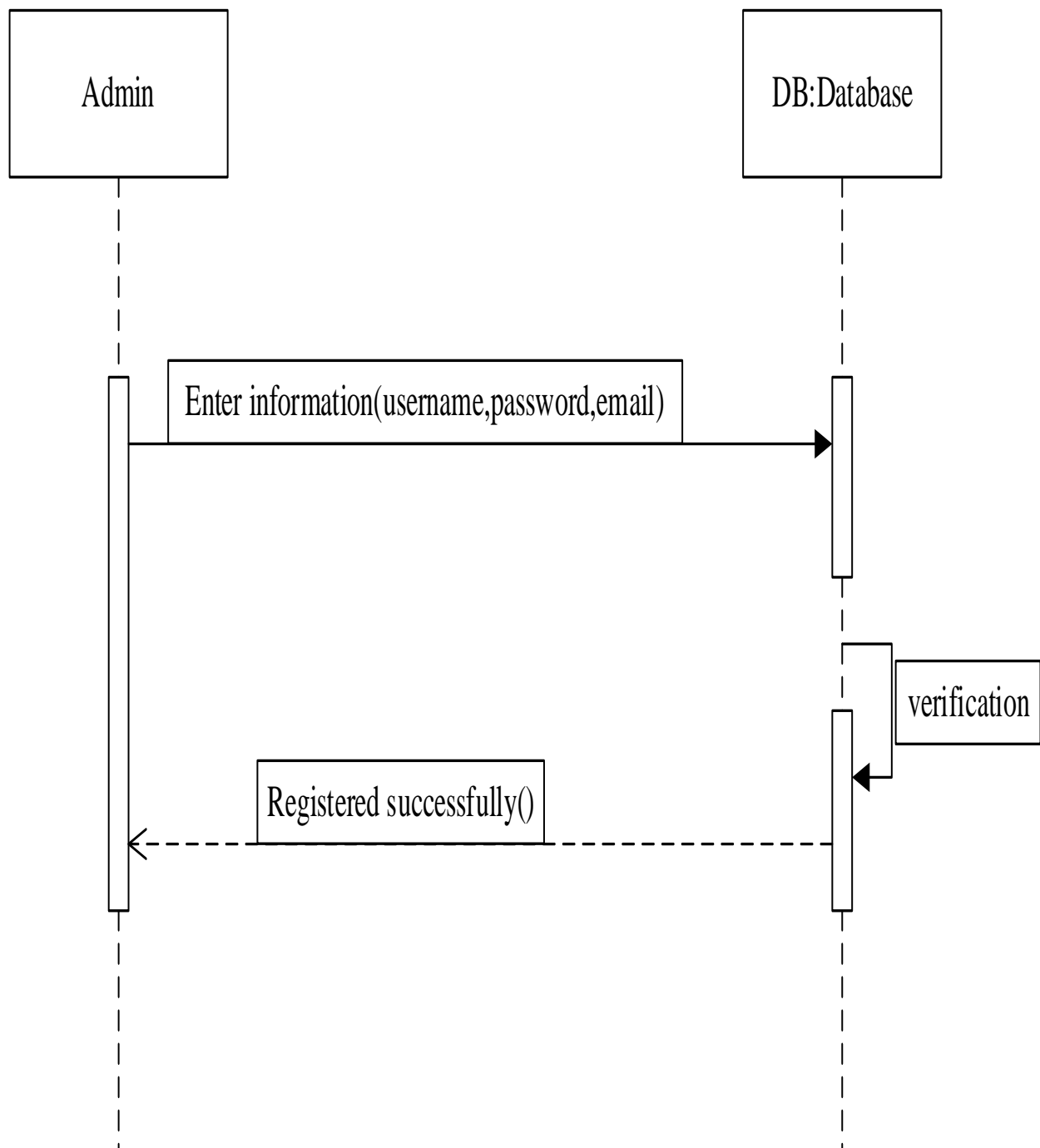
Sequence Diagram for Admin Register

Figure 19: Admin Register

Description:

Admin Registers by entering valid information i.e, username, Email and password and it reaches to Database and after verification registration done successfully.

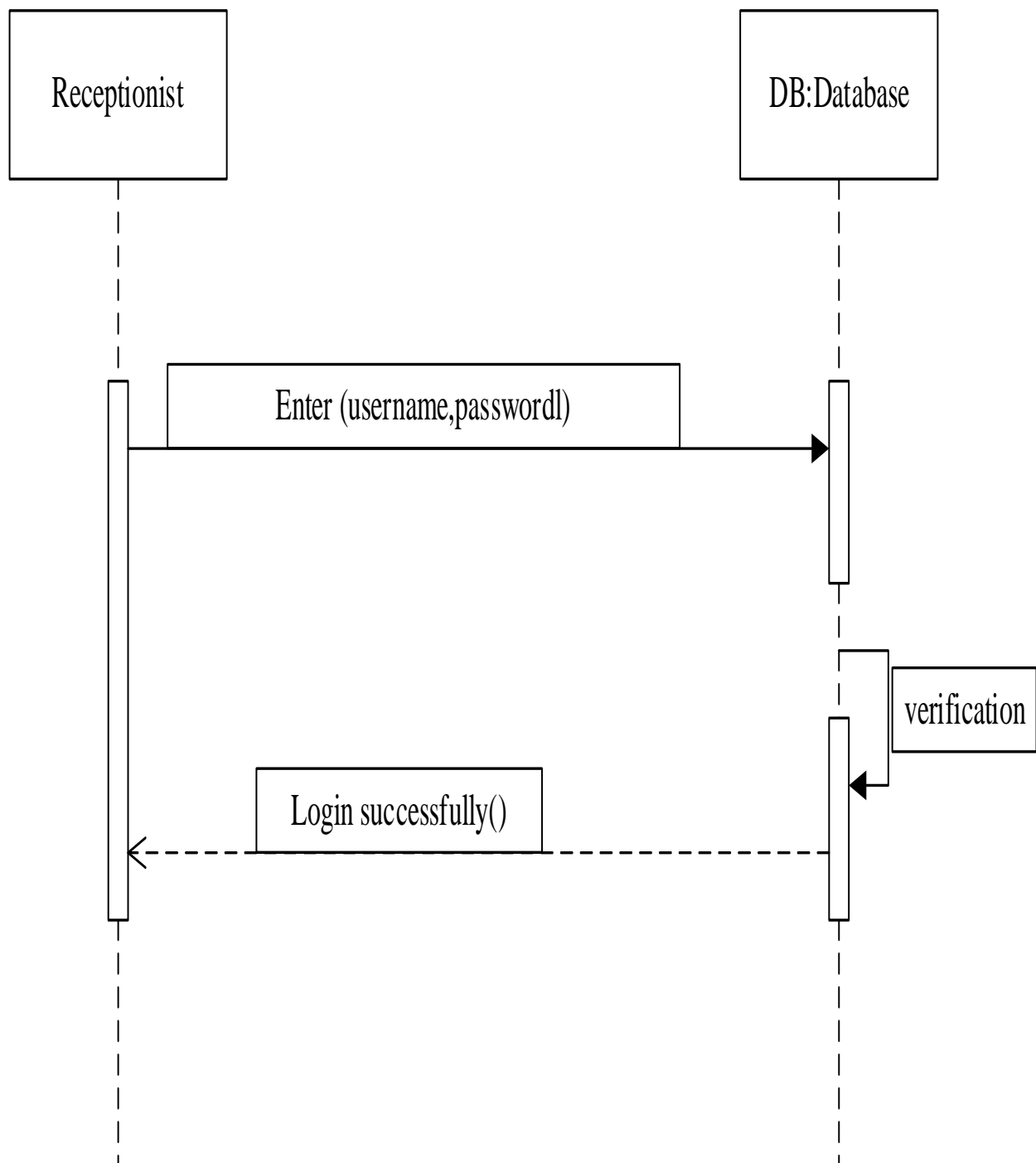
Sequence Diagram for Receptionist Login

Figure 20: Receptionist Login

Description:

Receptionist logs in by entering valid information i.e, username and password and it reaches to Database and after verification login done successfully.

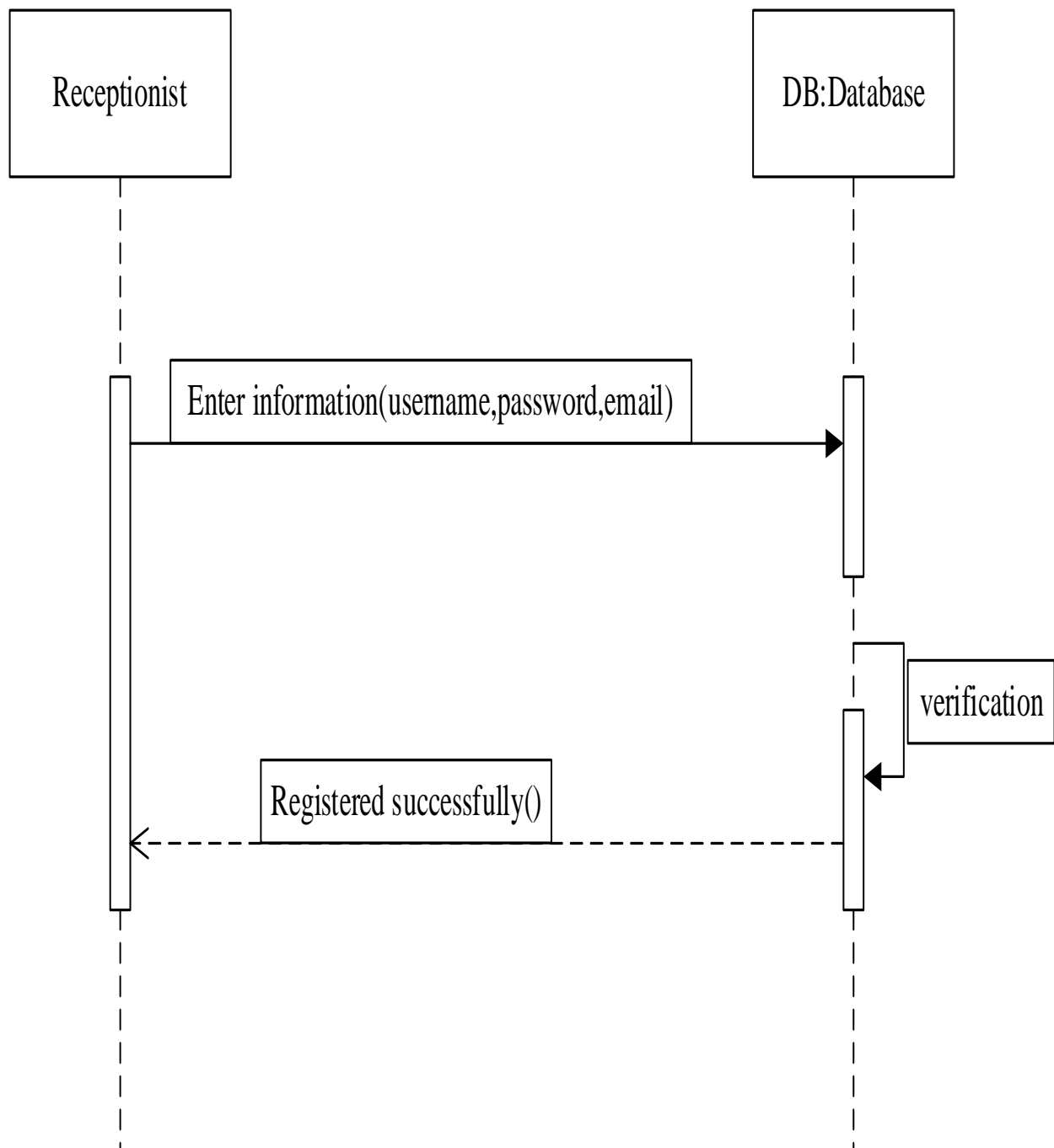
Sequence Diagram for Receptionist Registration

Figure 21: Receptionist Registration

Description:

Receptionist Registers by entering valid information i.e, username, Email and password and it reaches to Database and after verification registration done successfully.

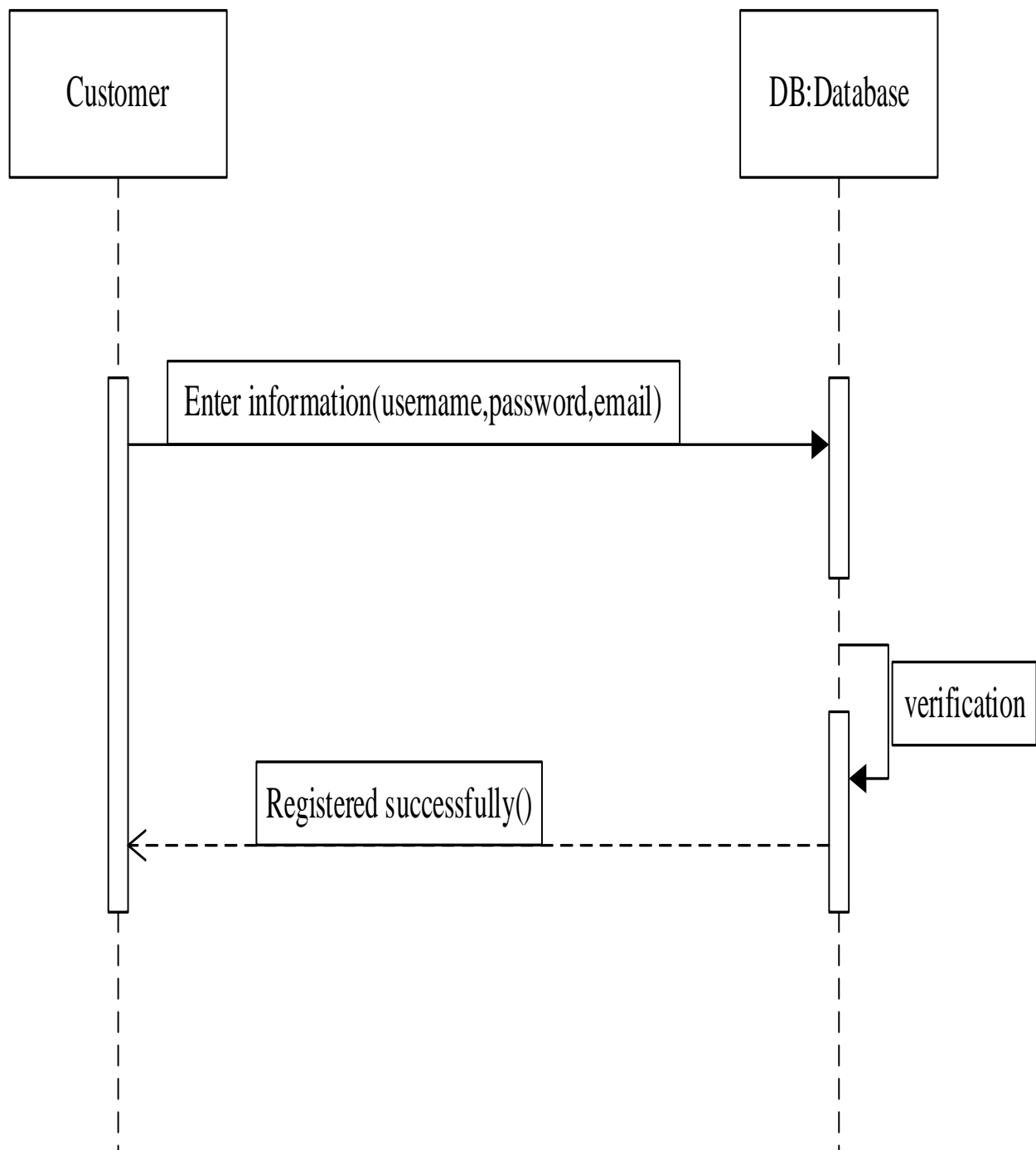
Sequence Diagram for Customer Registration

Figure 22: Customer Registration

Description:

Customer Registers by entering valid information i.e, username and password and it reaches to Database and after verification registration done successfully.

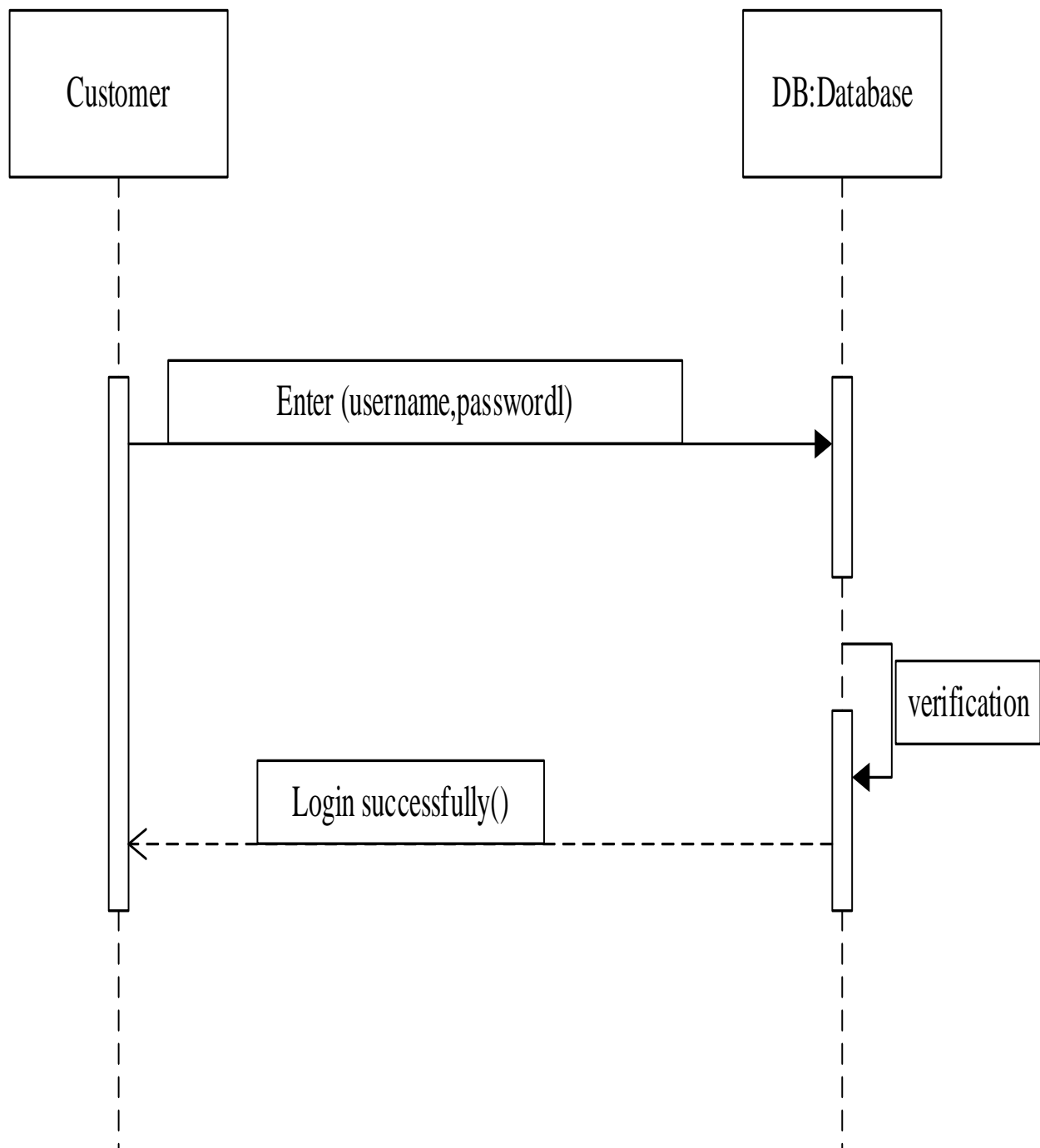
Sequence Diagram for Customer Login

Figure 23: Customer Login

Description:

Customer logs in by entering valid information i.e, username and password and it reaches to Database and after verification login done successfully.

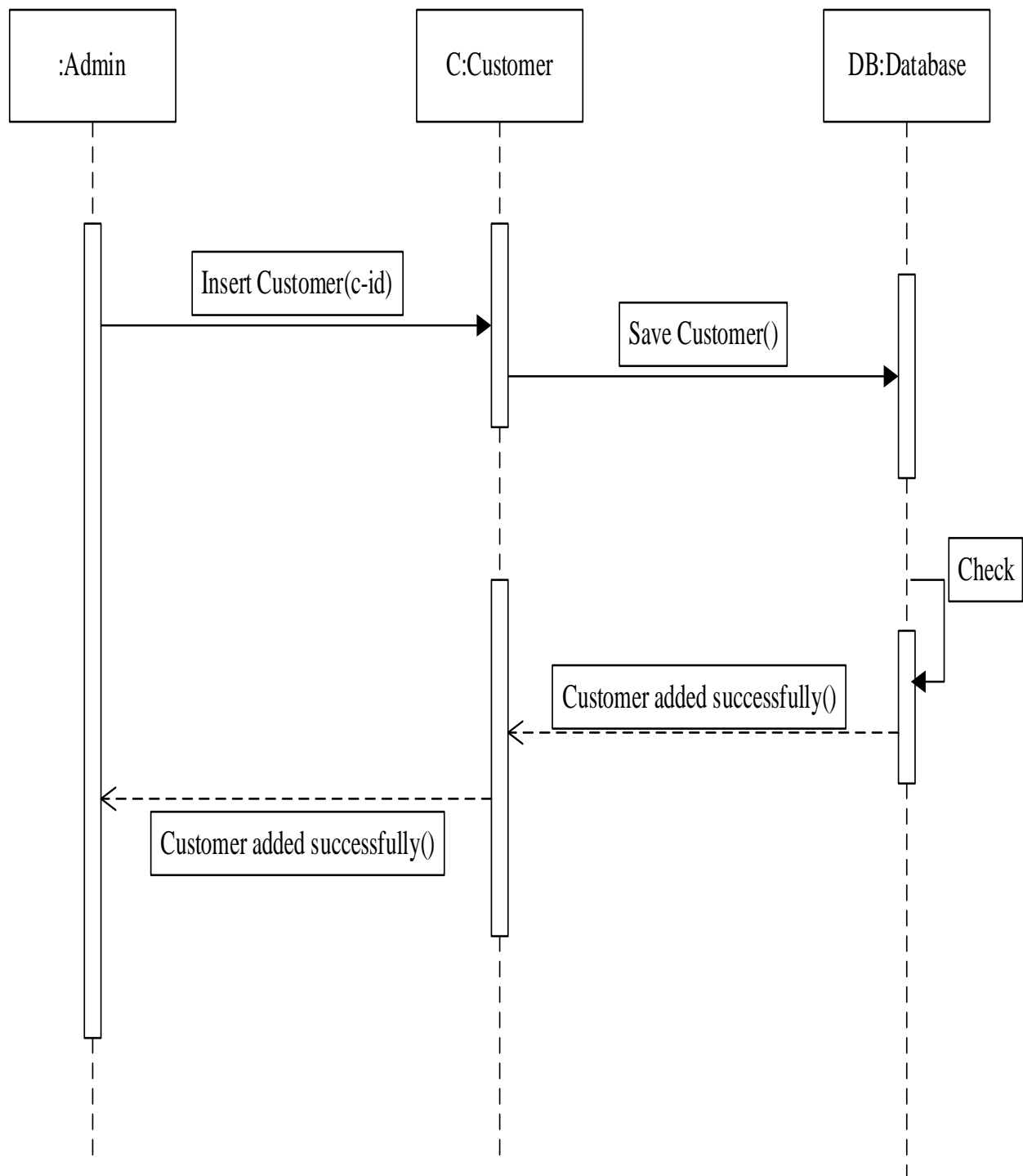
Sequence Diagram for Customer Add

Figure 24: Customer Add

Description:

Admin can insert and save customer to database and then check (verify) and customer gets added successfully.

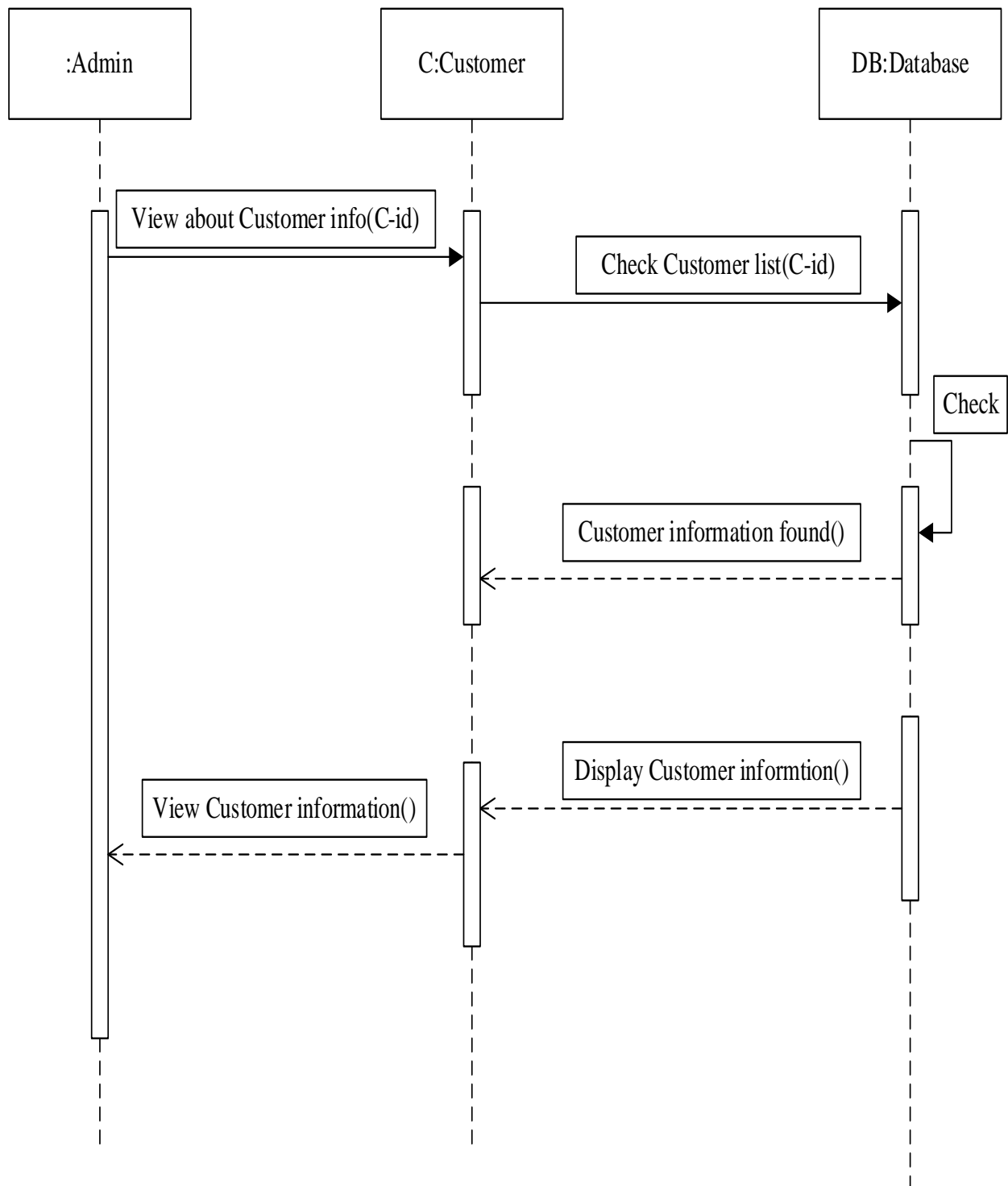
Sequence Diagram for Customer Info

Figure 25: Customer Info

Description:

Admin can view customer info by checking proper credentials into database.

Sequence Diagram for Customer Update

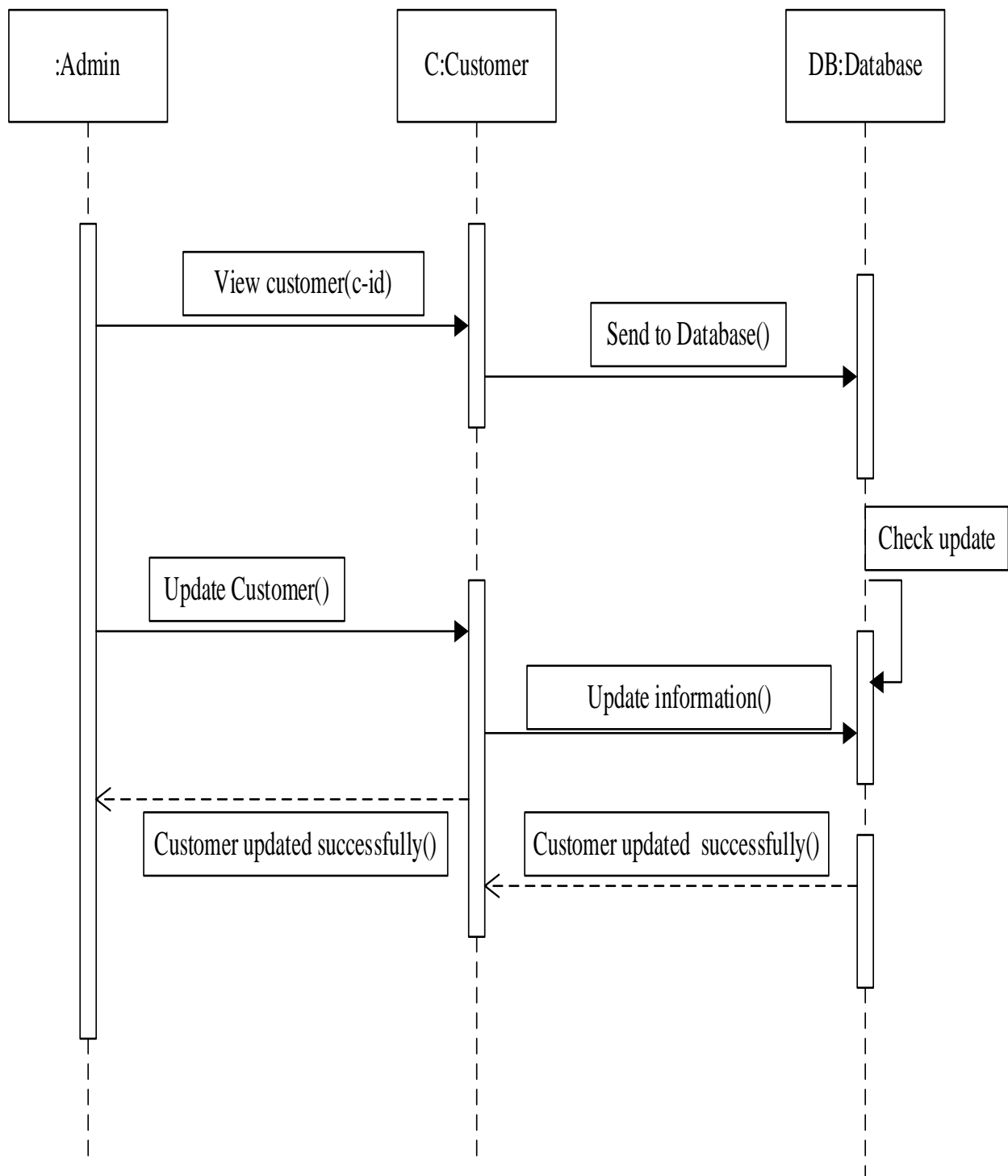


Figure 26: Customer Update

Description:

Admin can view customer from database and can update any customer anytime.

Sequence Diagram for Available Room Checking

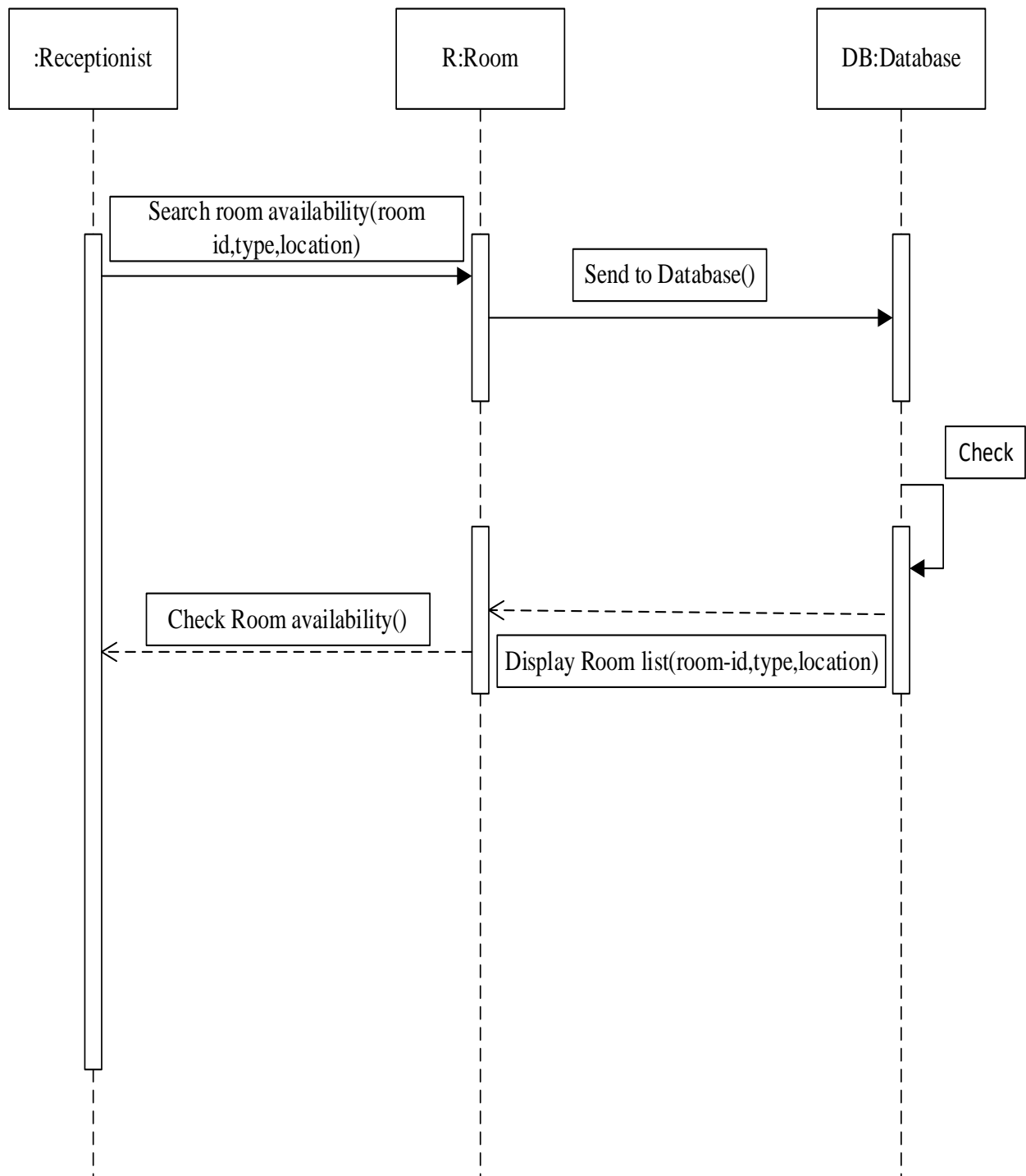


Figure 27: Available Room checking

Description:

Receptionist can search room availability from database and then display available rooms list to the new coming customers.

Sequence Diagram for Room Booking

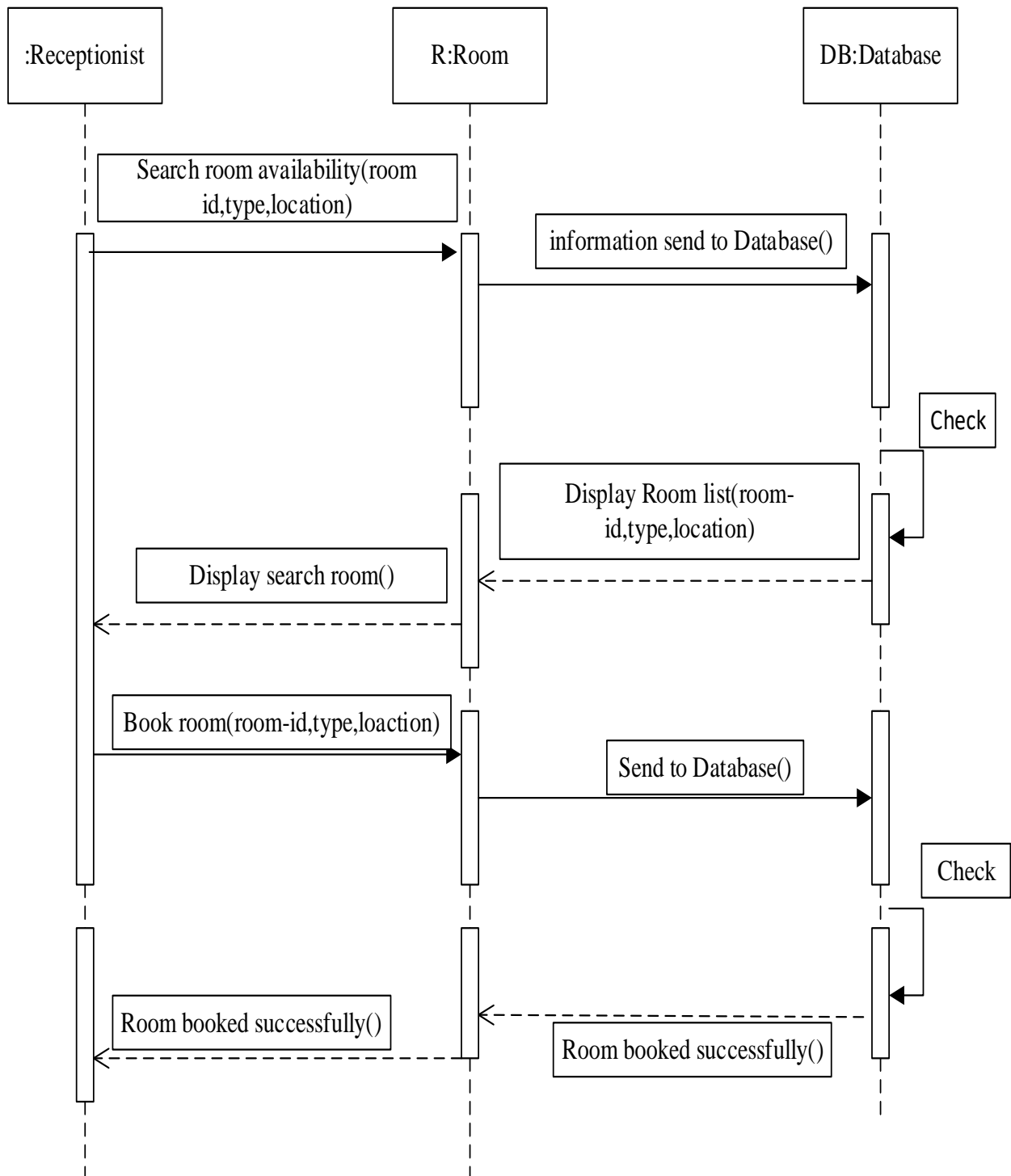


Figure 28: Room Booking

Description:

Receptionist can search room availability from database and then display available rooms list to the new coming customers and can book the required room by putting credentials for booking room.

Sequence Diagram for Payment

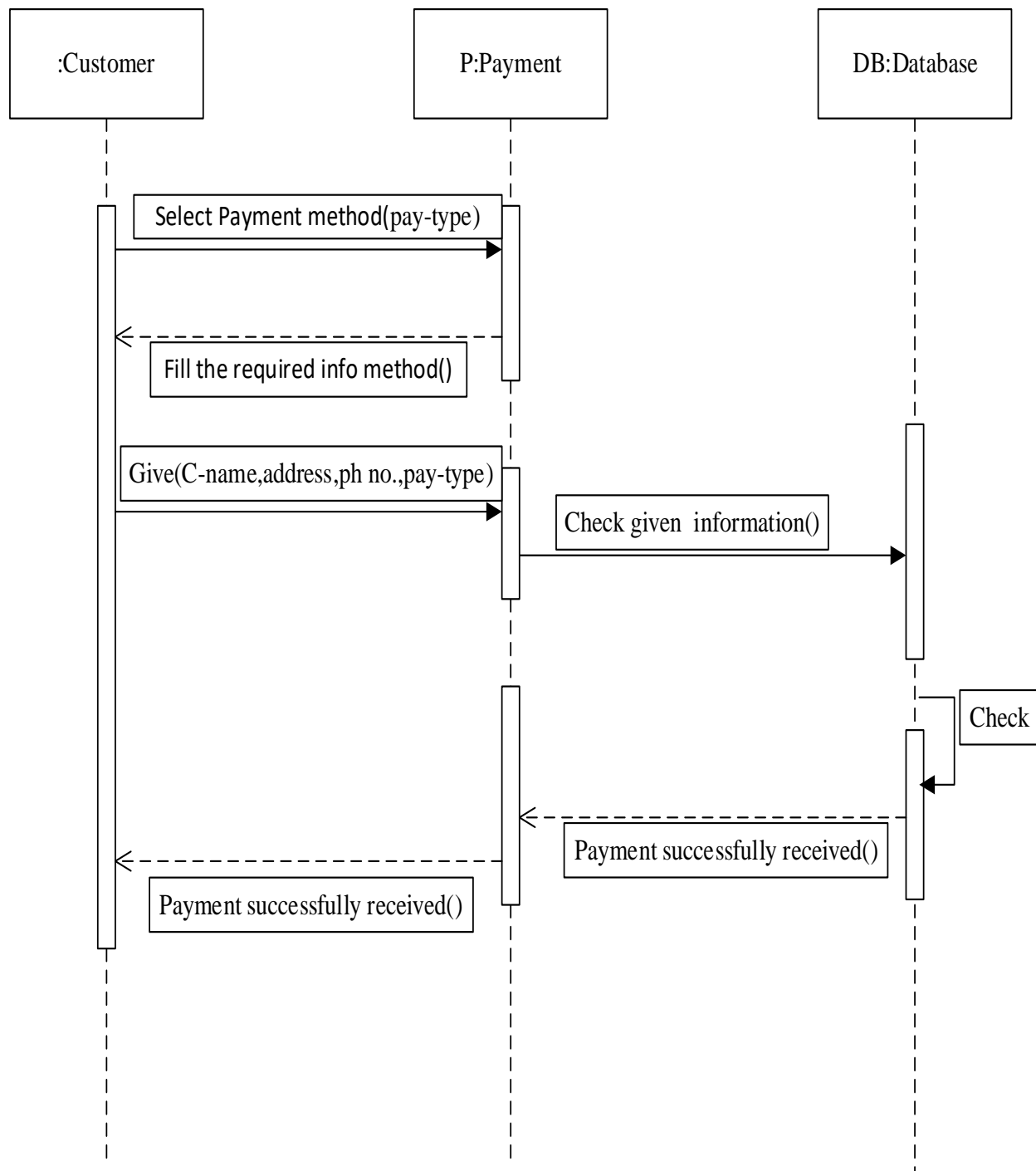


Figure 29: Payment

Description:

Customer can select payment method (Cash Payment/ Jazzcash/ Easypaisa) after filling the asked credentials and after verified payment gets the booked room.

Sequence Diagram for Generate Payment

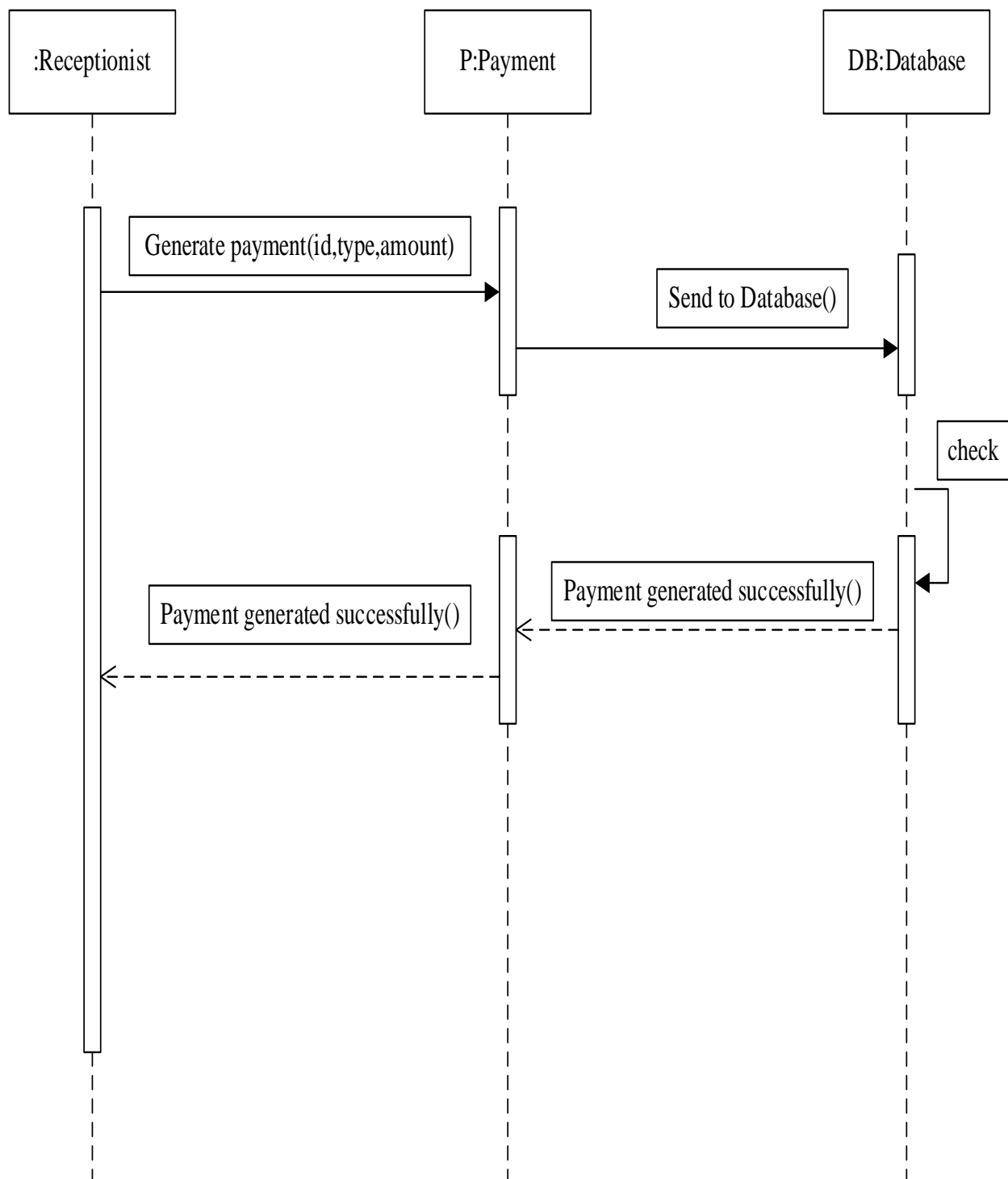


Figure 30: Generate Payment

Description:

Customer can select payment method (Cash payment/ Jazzcash/ Easypaisa) after filling the asked credentials and generate payment, after verified payment gets the booked room.

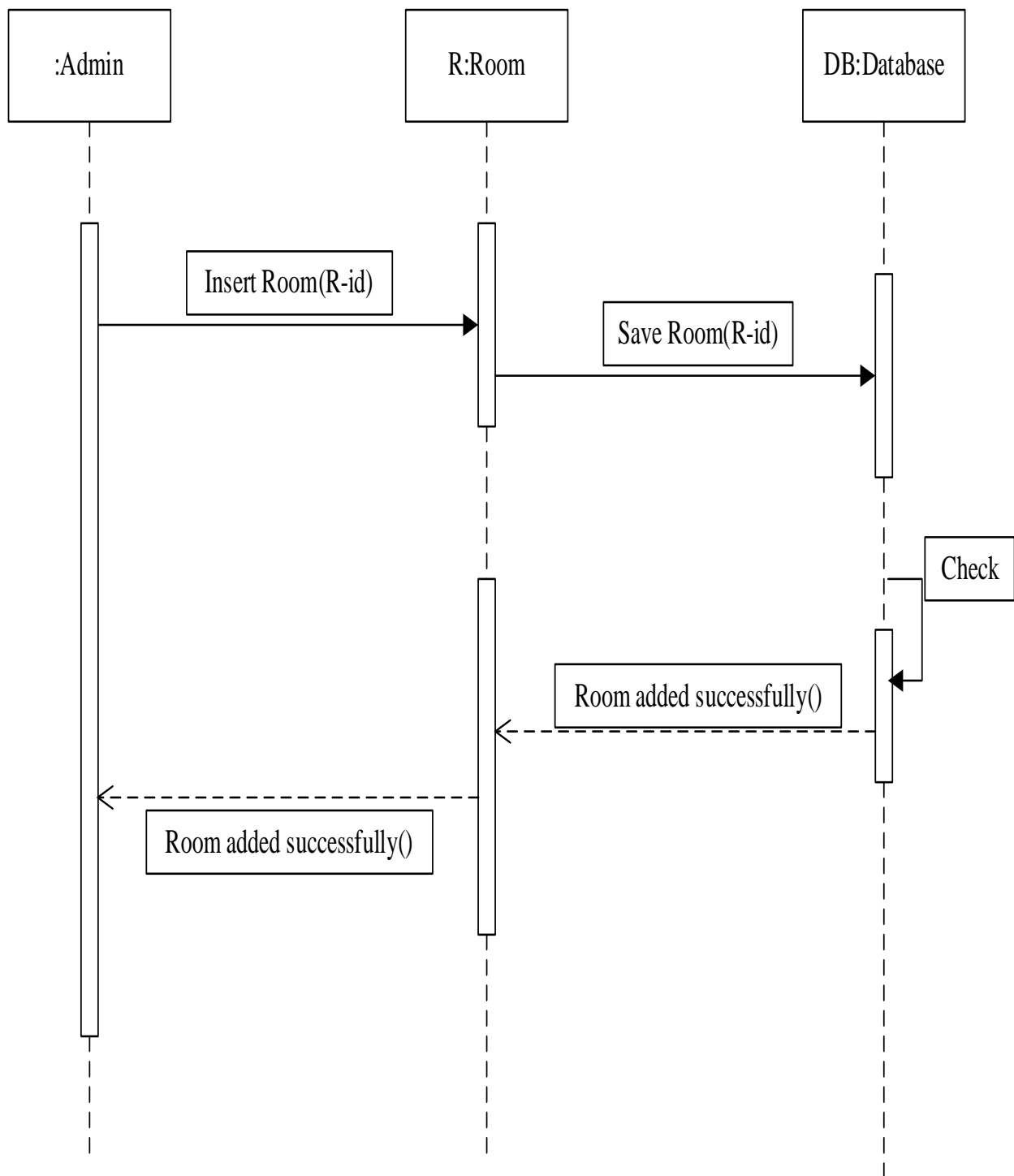
Sequence Diagram for Room Add

Figure 31: Add Room

Description:

Admin can insert room Id and save it to database and after verify check room gets added in the system.

Sequence Diagram for Room Update

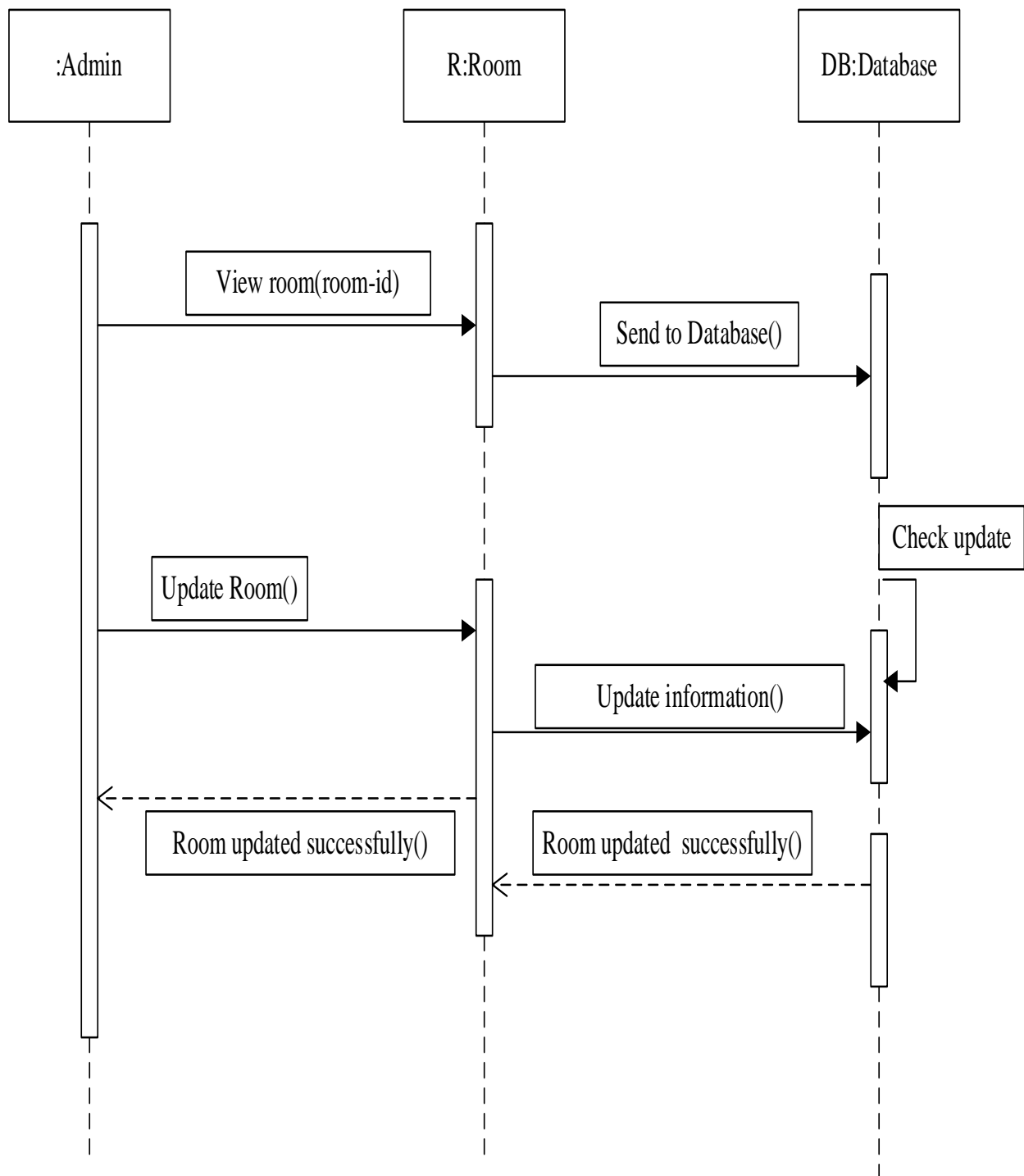


Figure 32: Update Room

Description:

Admin can view room Id and send to database the update information and room information gets updated instantly.

Sequence Diagram for Reservation Update

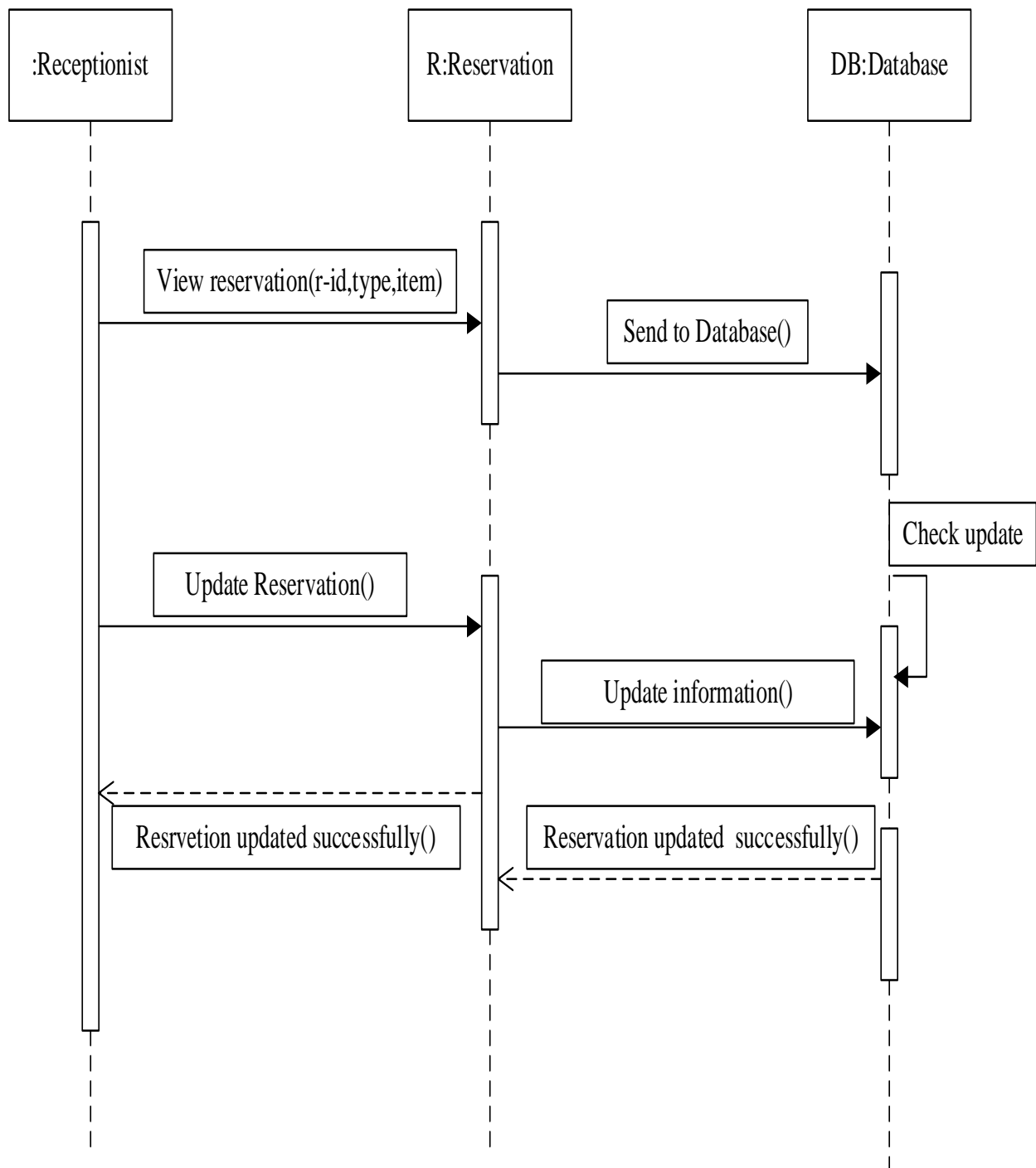


Figure 33: Reservation Update

Description:

Receptionist can view reservation from database and after checking update, inserts the new information and Reservation gets updated.

Sequence Diagram for Check In

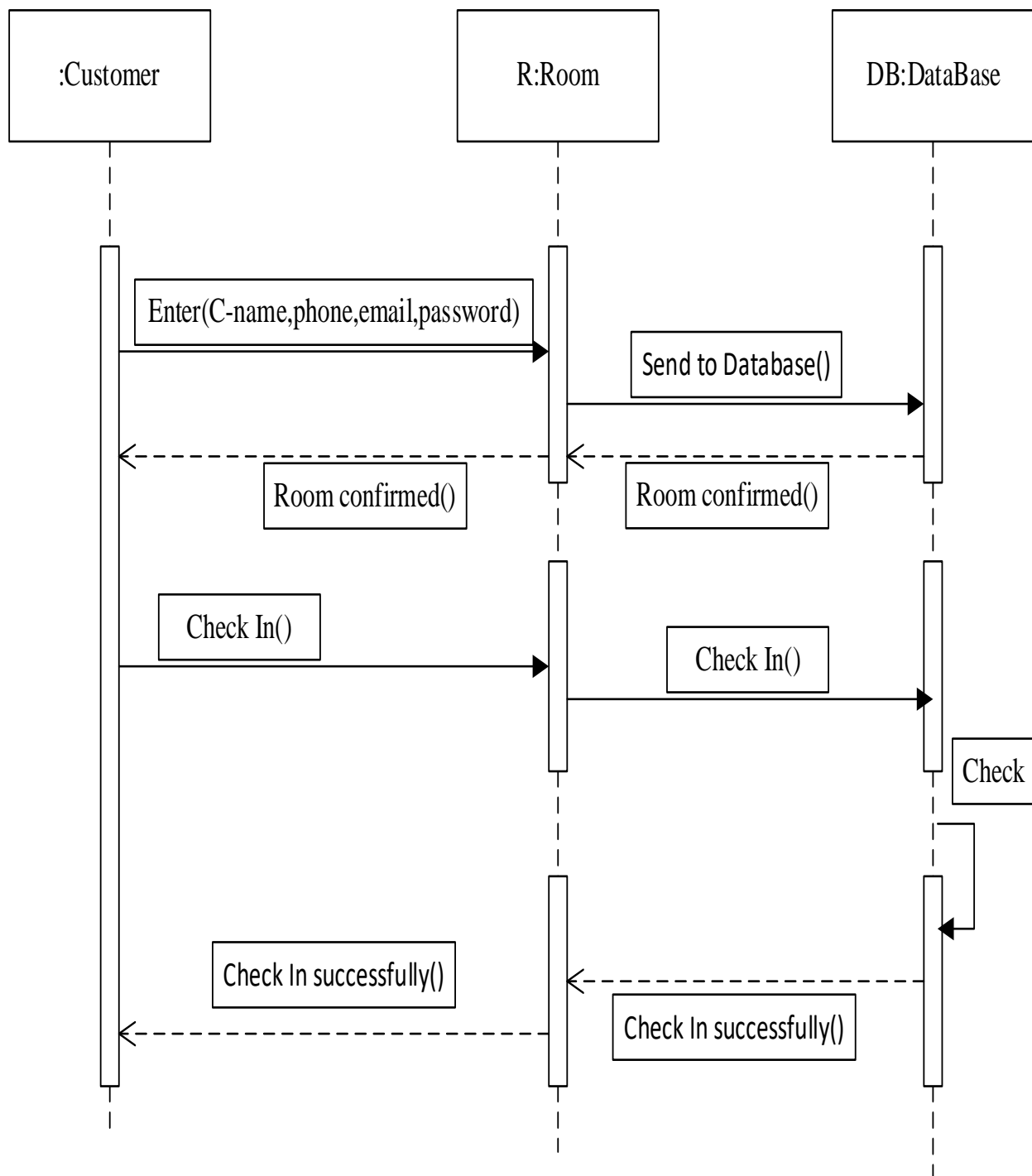


Figure 34: Check In

Description:

Customer enters valid credentials and gets sent to database after verification, customer allowed to check in successfully.

Sequence Diagram for Customer Feedback

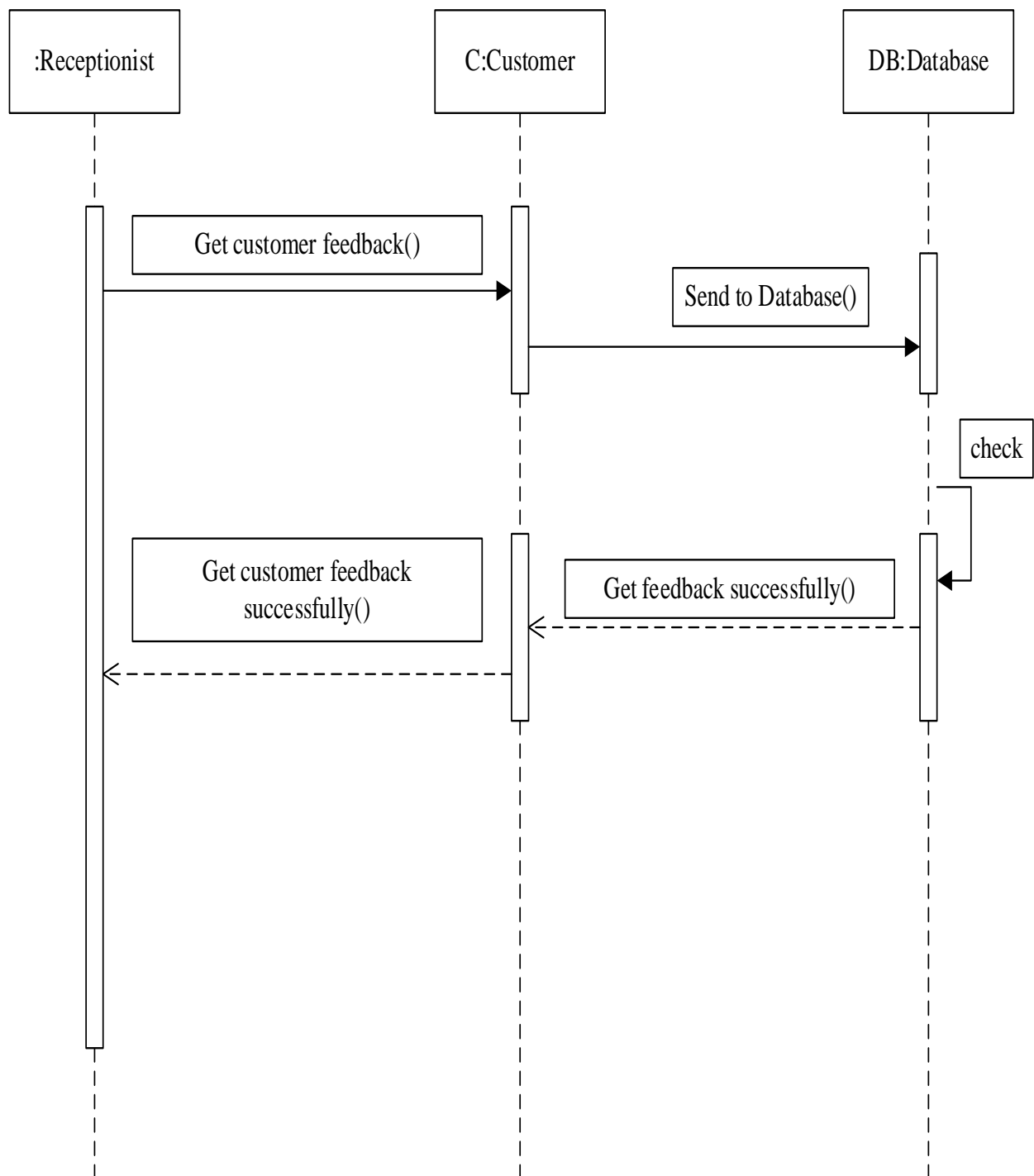


Figure 35: Customer Feedback

Description:

Receptionist asks for getting feedback from customer and database delivers command to customer and feedback page displayed.

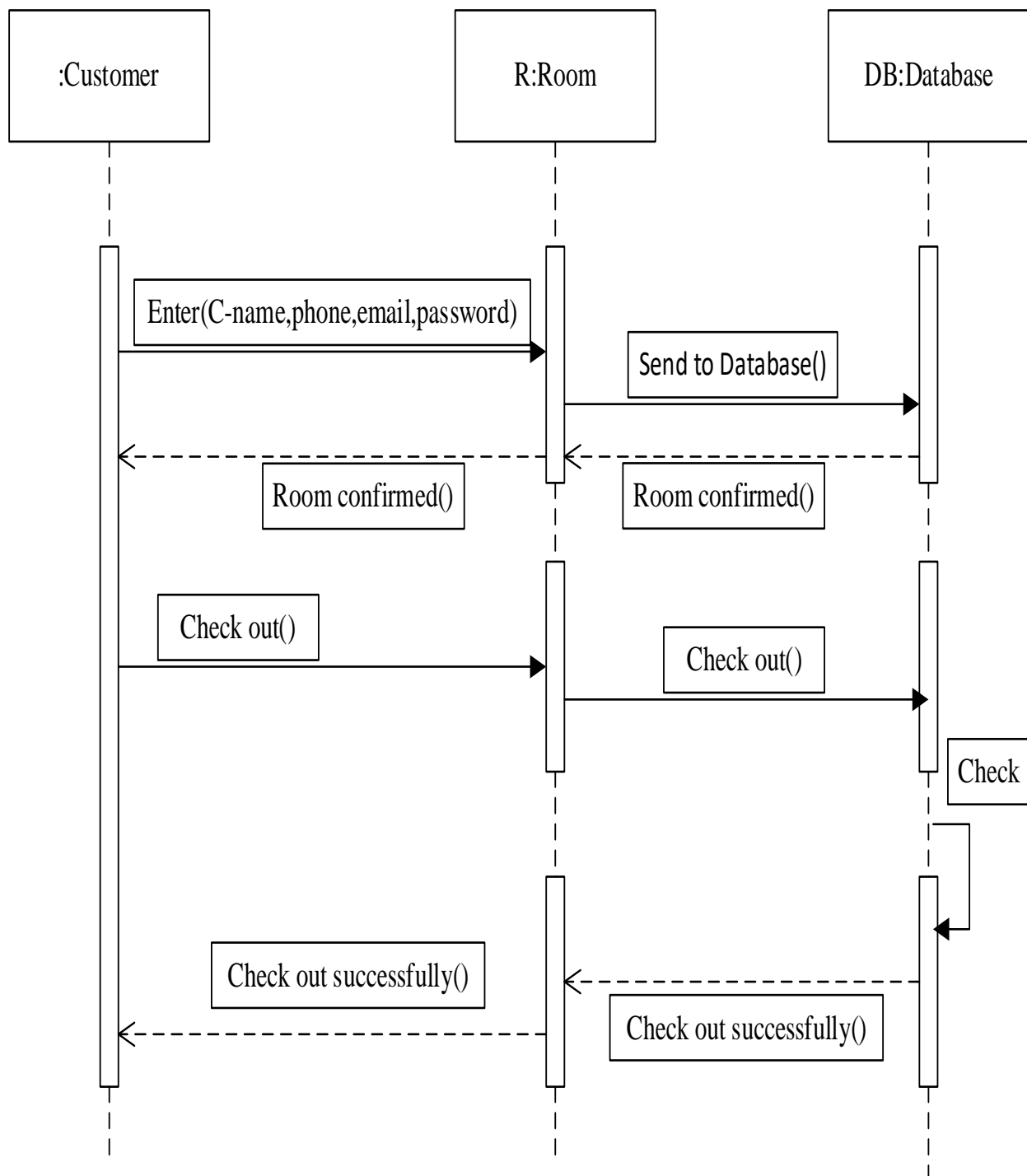
Sequence Diagram for Check Out

Figure 36: Check Out

Description:

Customer enters valid credentials and gets sent to database after verification, customer allowed to check out successfully.

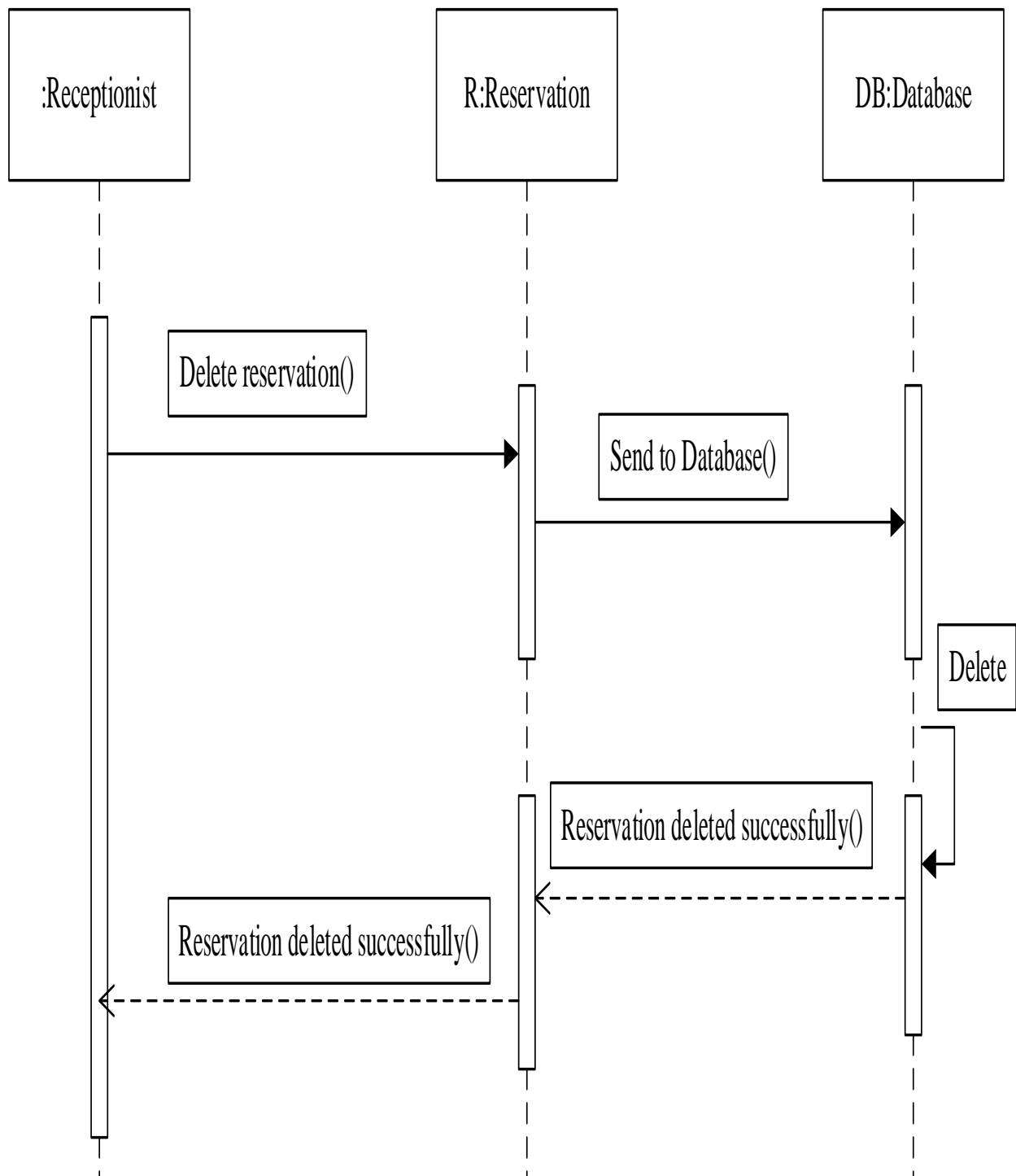
Sequence Diagram for Reservation Delete

Figure 37: Reservation Delete

Description:

Receptionist deletes the reservation by sending information to database and deletion successful.

4.7 Activity Diagram

Activity Diagram for User Login

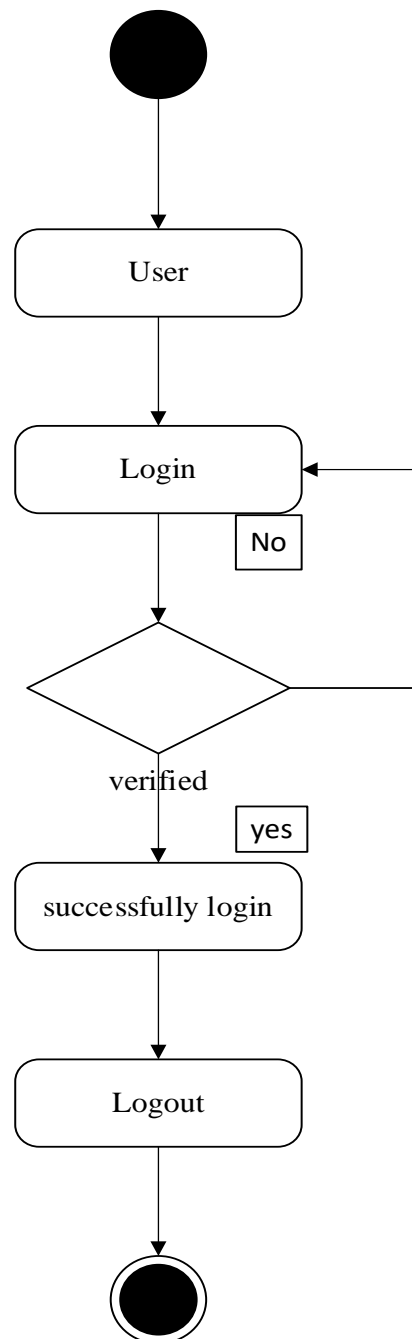


Figure 38: User Login

Description:

Users login to the system. The username and password gets verified. If credentials wrong then no and again try to login with right credentials. If yes credentials are right and user gets login to website system.

Activity Diagram for Admin Login

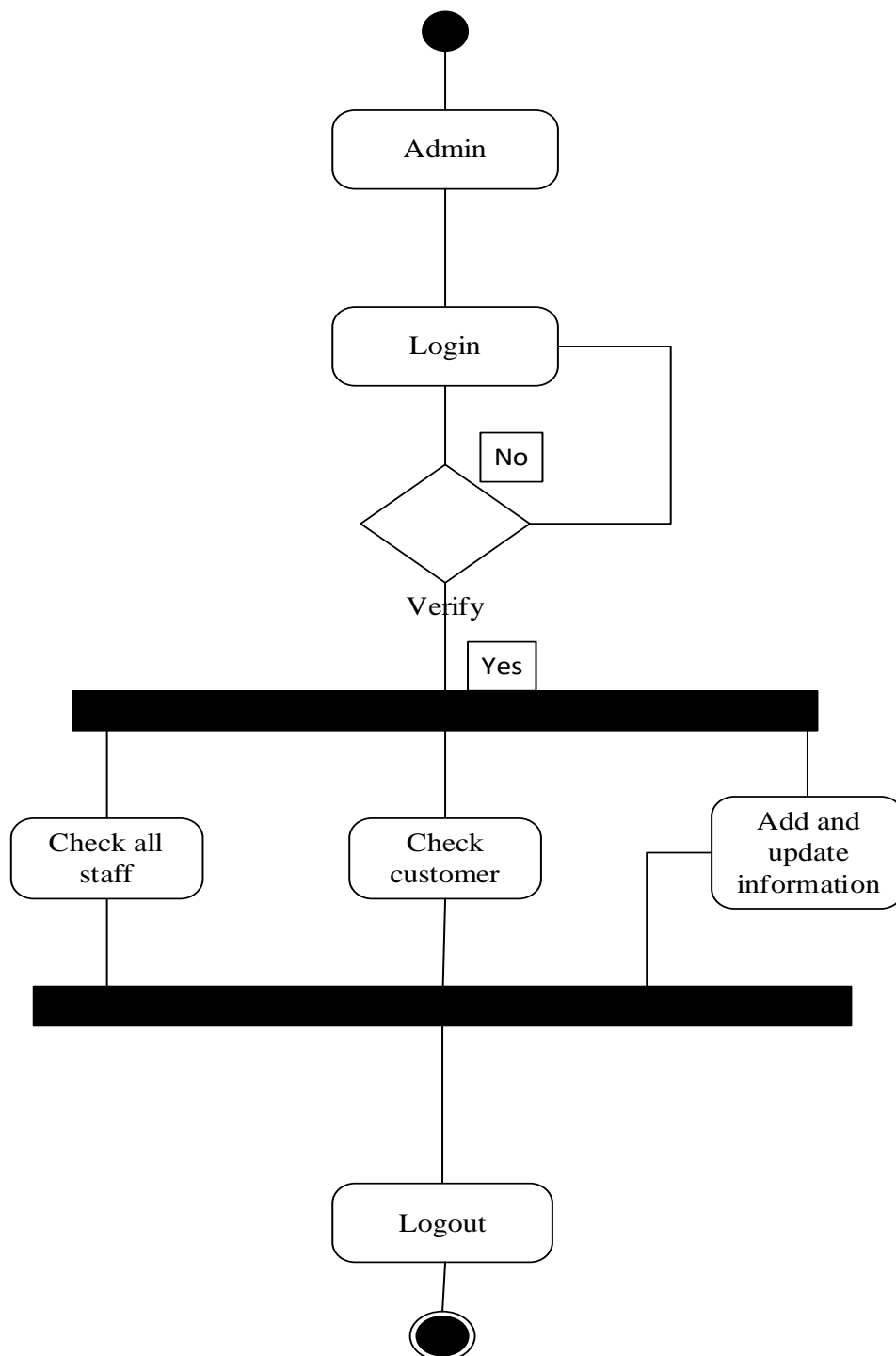


Figure 39: Admin Login

Description:

Admin login to the system. The username and password gets verified. If credentials wrong then no and again try to login with right credentials. If yes credentials are right, gets access to the admin system.

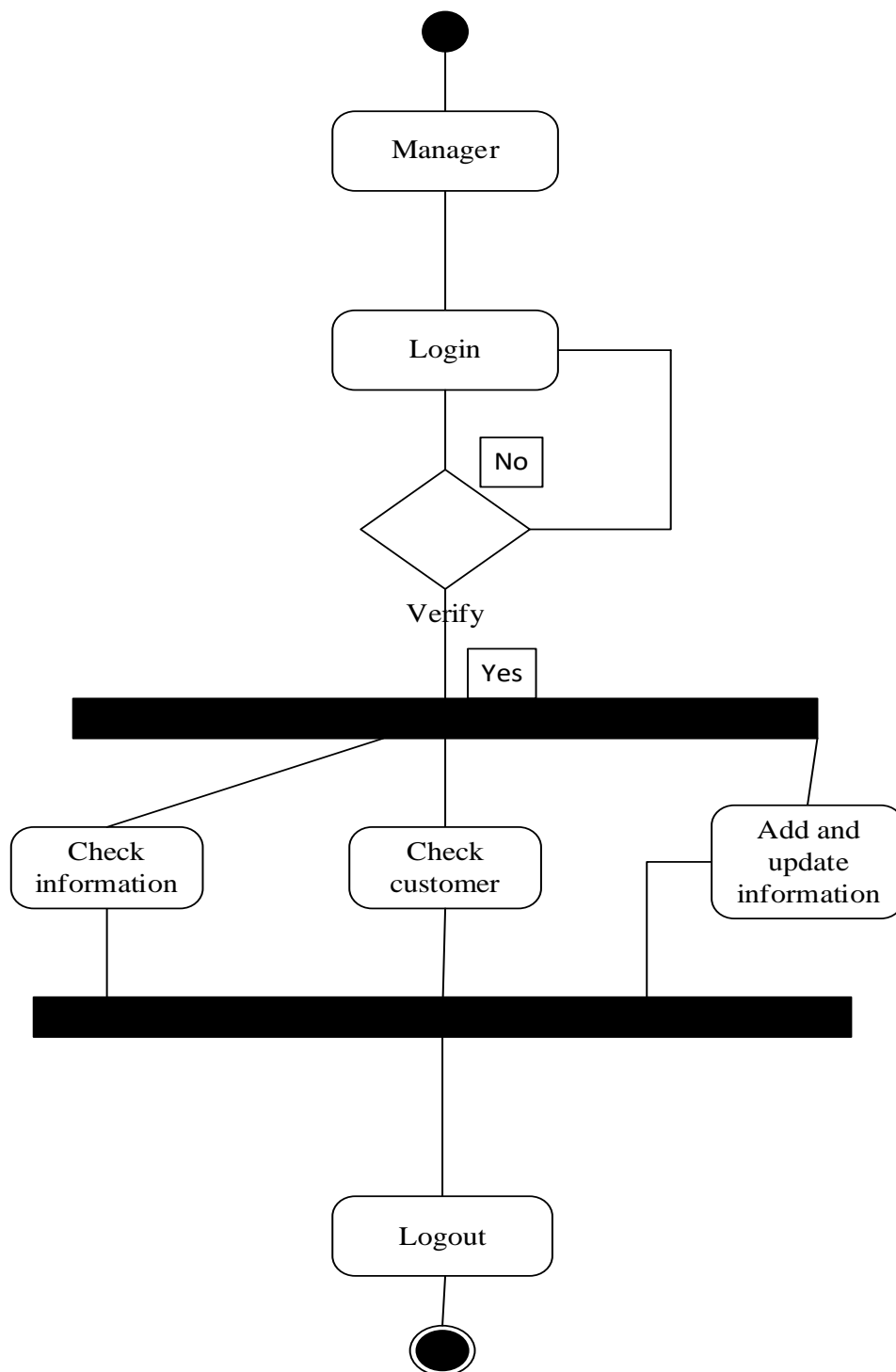
Activity Diagram for Manager Login

Figure 40: Manager Login

Description:

Manager login to the system. The username and password gets verified. If credentials wrong then no and again try to login with right credentials. If yes credentials are right Manager gets access to the system.

Activity Diagram for Receptionist Login

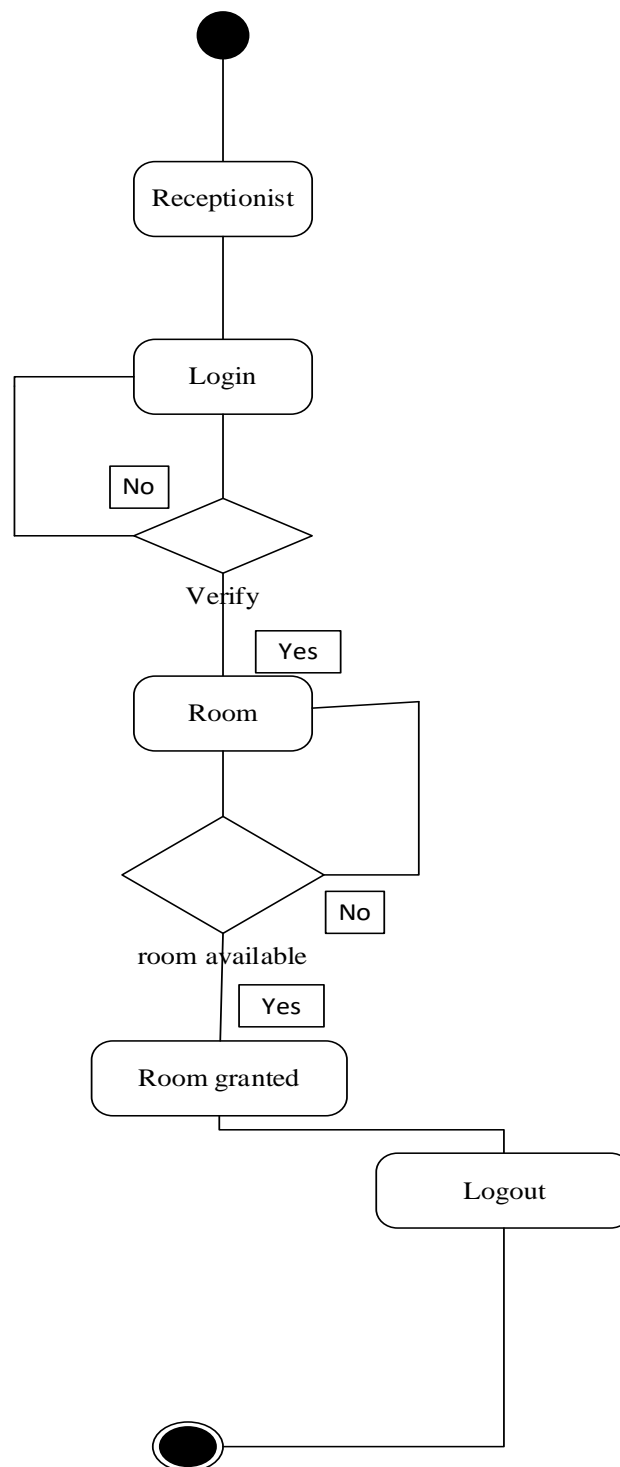


Figure 41: Receptionist Login

Description:

Receptionist login to the system. The username and password gets verified. If credentials wrong then no and again try to login with right credentials. If yes credentials are right receptionist gets access to the system.

Activity Diagram for Customer Login

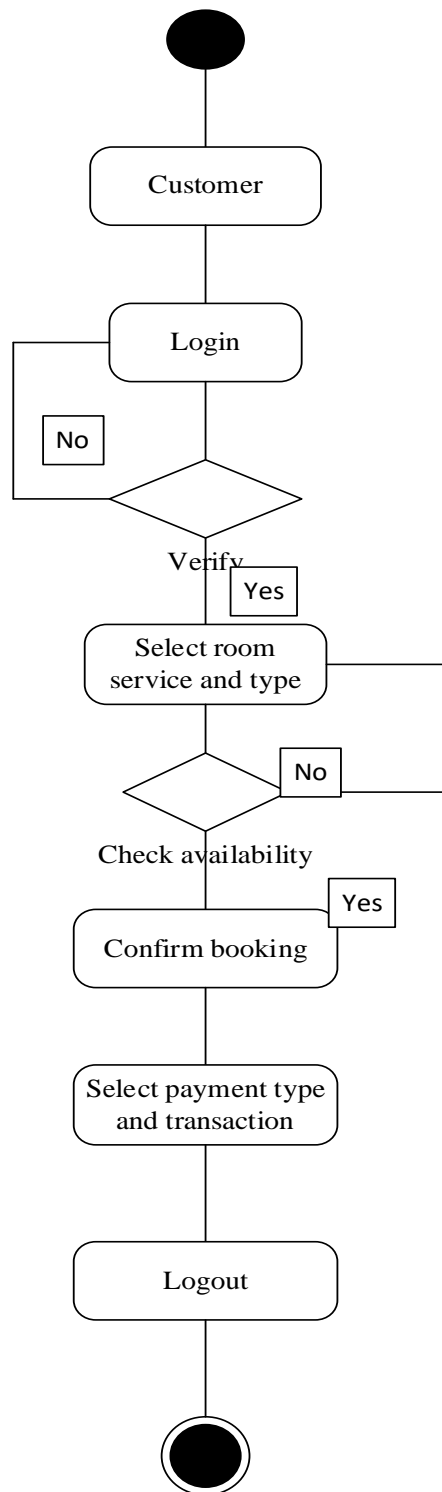


Figure 42: Customer Login

Description:

Customer login to the system. The username and password gets verified. If credentials wrong then no and again try to login with right credentials. If yes credentials are right customer gets access and can book room.

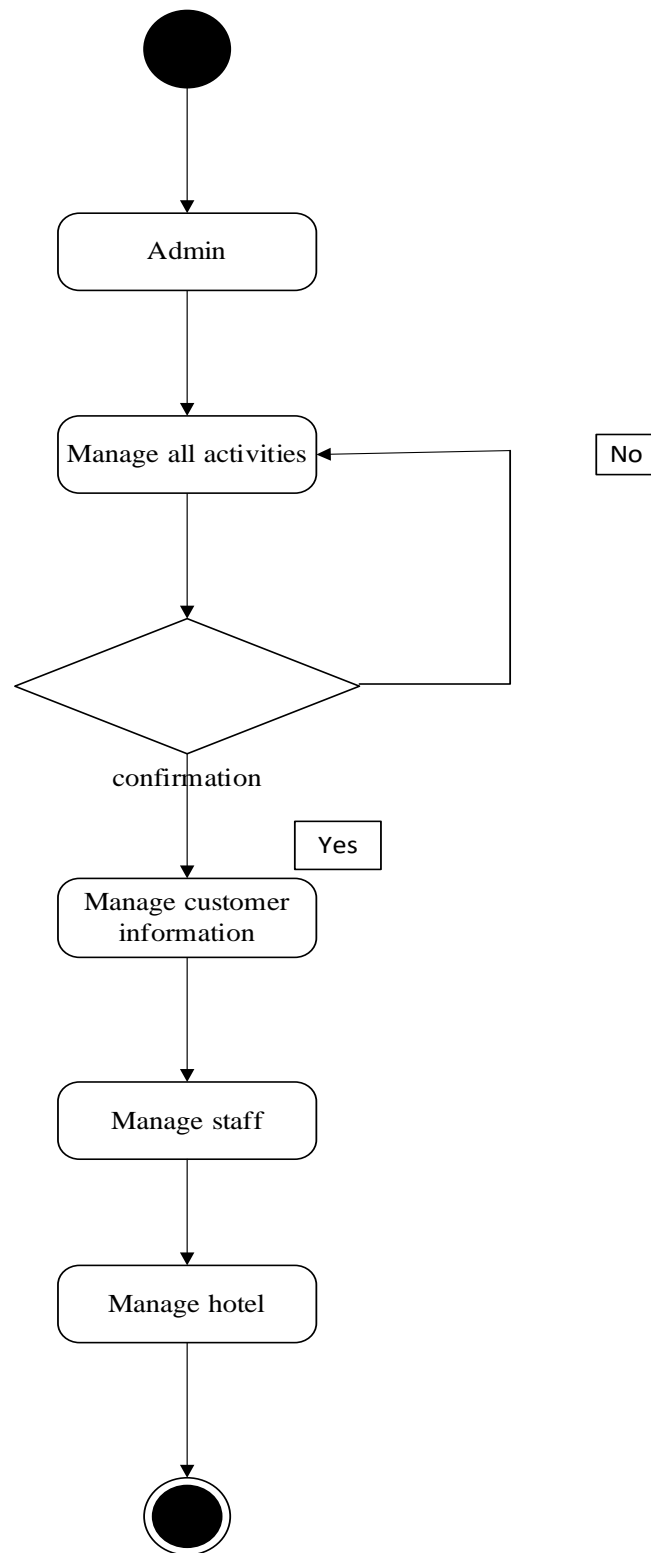
Activity Diagram for Manage Activities

Figure 43: Manage Activity

Description:

Admin Manage all Activities. If No, Return back to Manage activities. If Yes, manage customer and staff information and manage hotel.

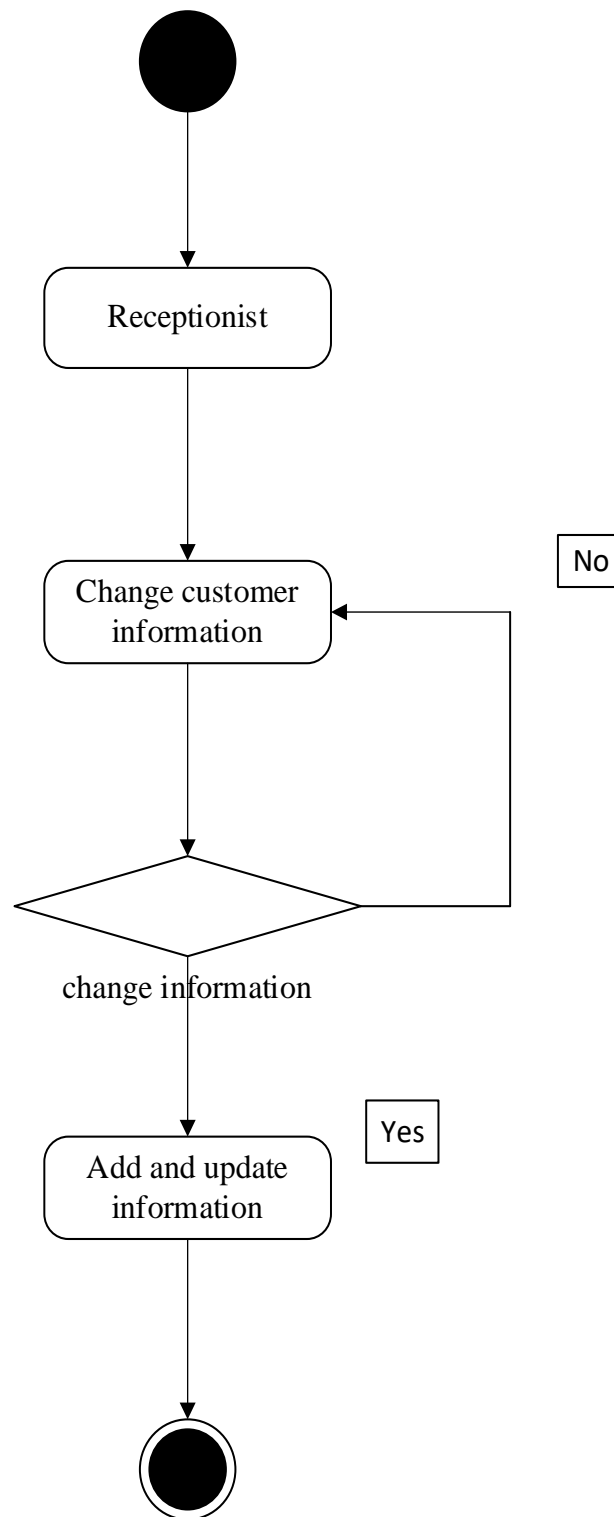
Activity Diagram for Customer Information

Figure 44: Customer Info

Receptionist change customer information. If No, Return back to change customer information. If Yes, add and update information.

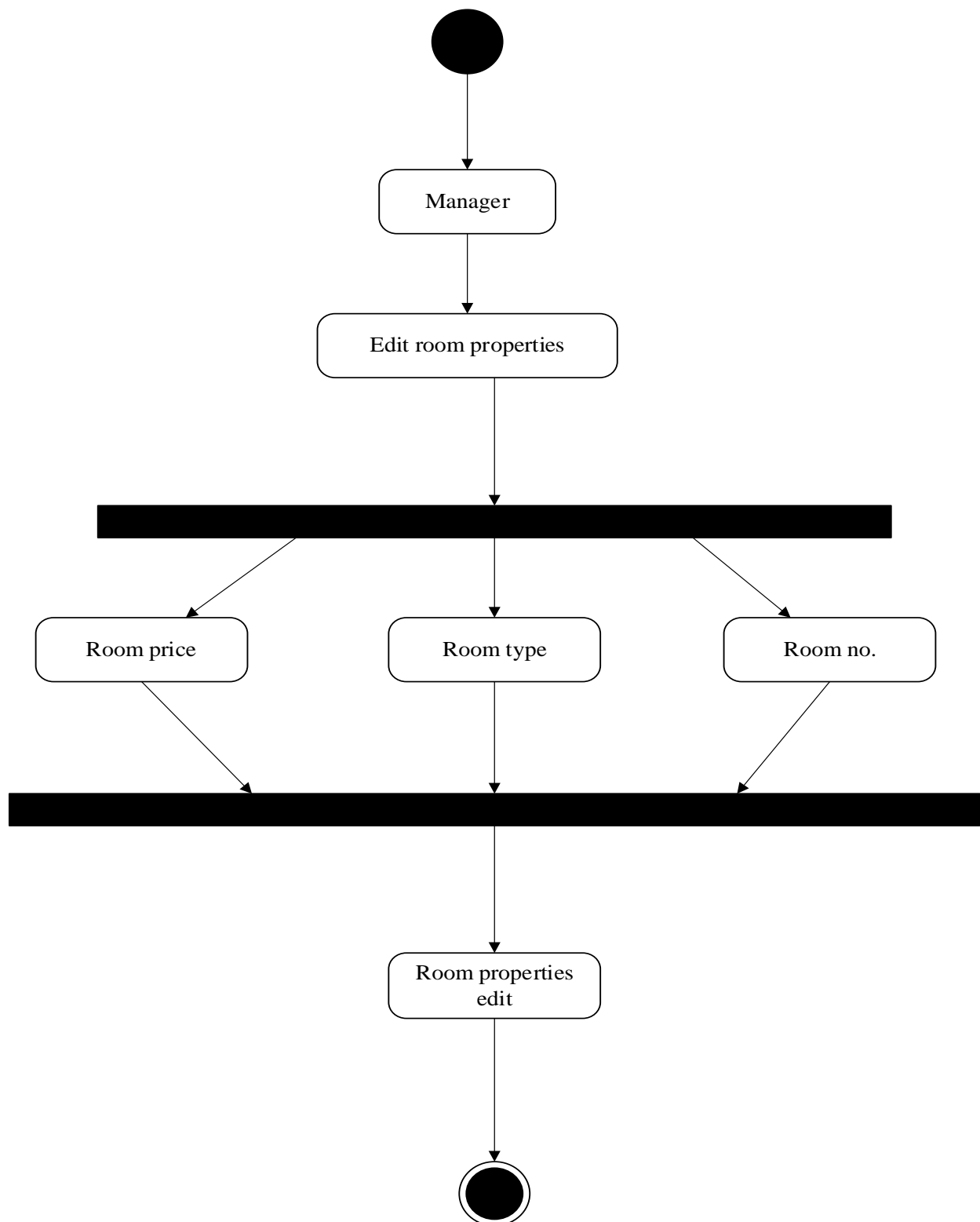
Activity Diagram for Room Properties

Figure 45: Room Properties

Description:

Manager edit room properties i.e, Room price, Room type, Room number.

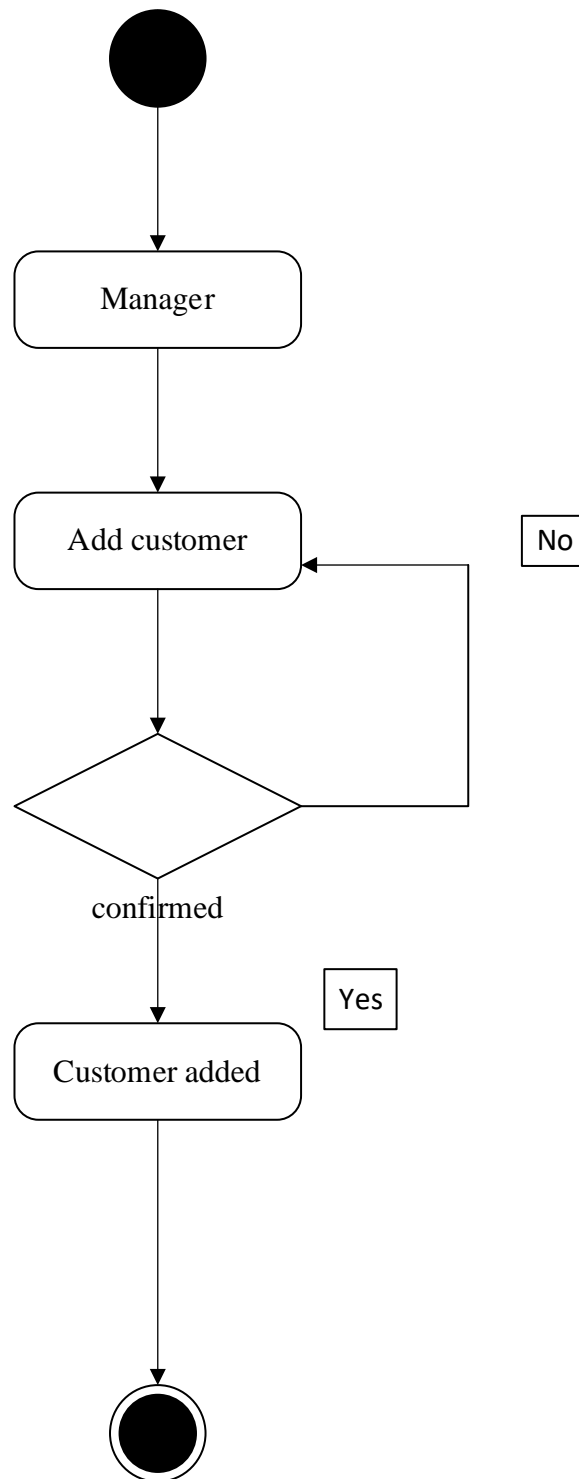
Activity Diagram for Customer Add

Figure 46: Customer Add

Description:

Manager adds customer. If No, Return back to Add customer. If Yes, customer added confirmation.

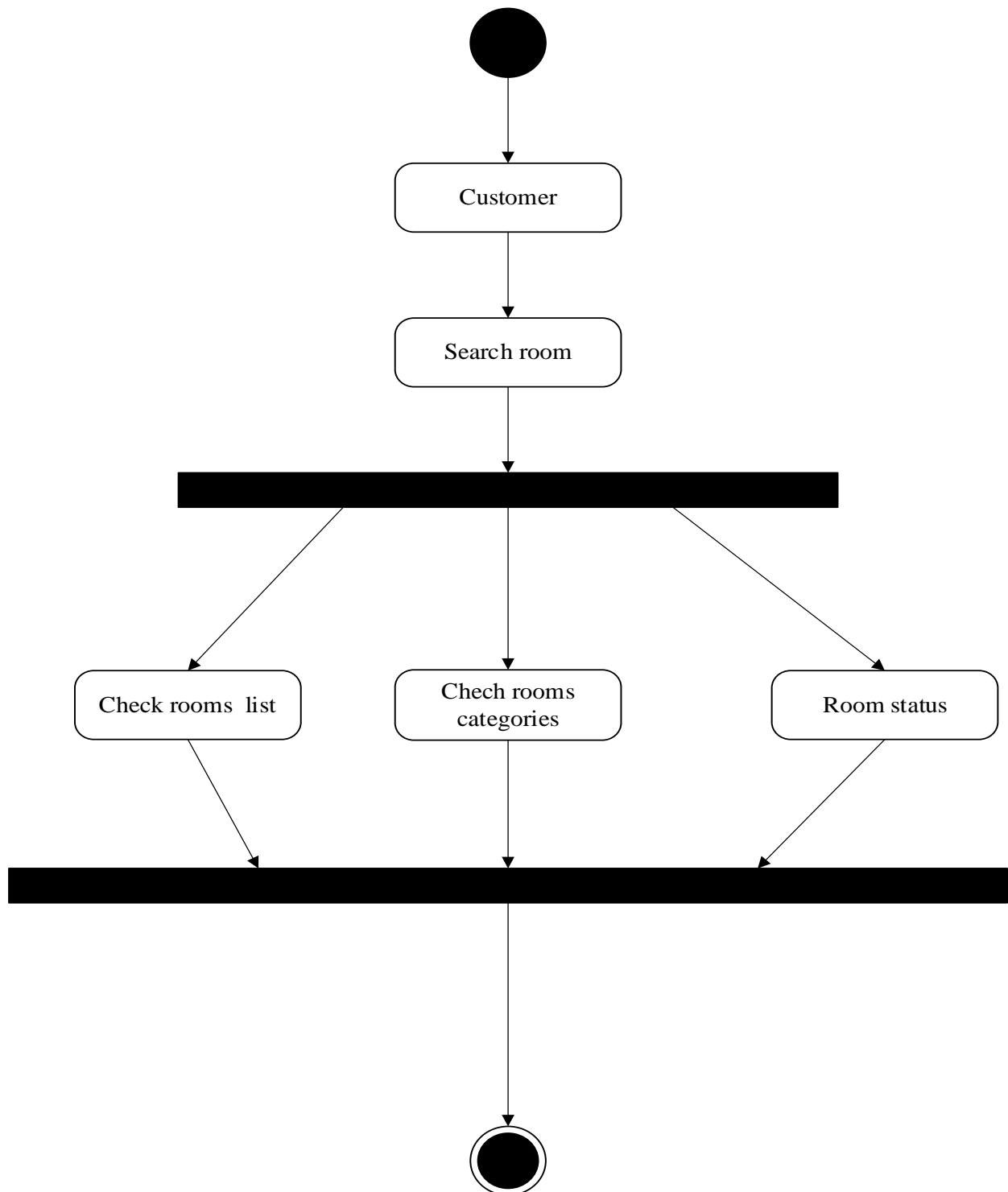
Activity Diagram for Room Search

Figure 47: Room Search

Description:

Customer Search rooms available and checks room list, categories and status of rooms.

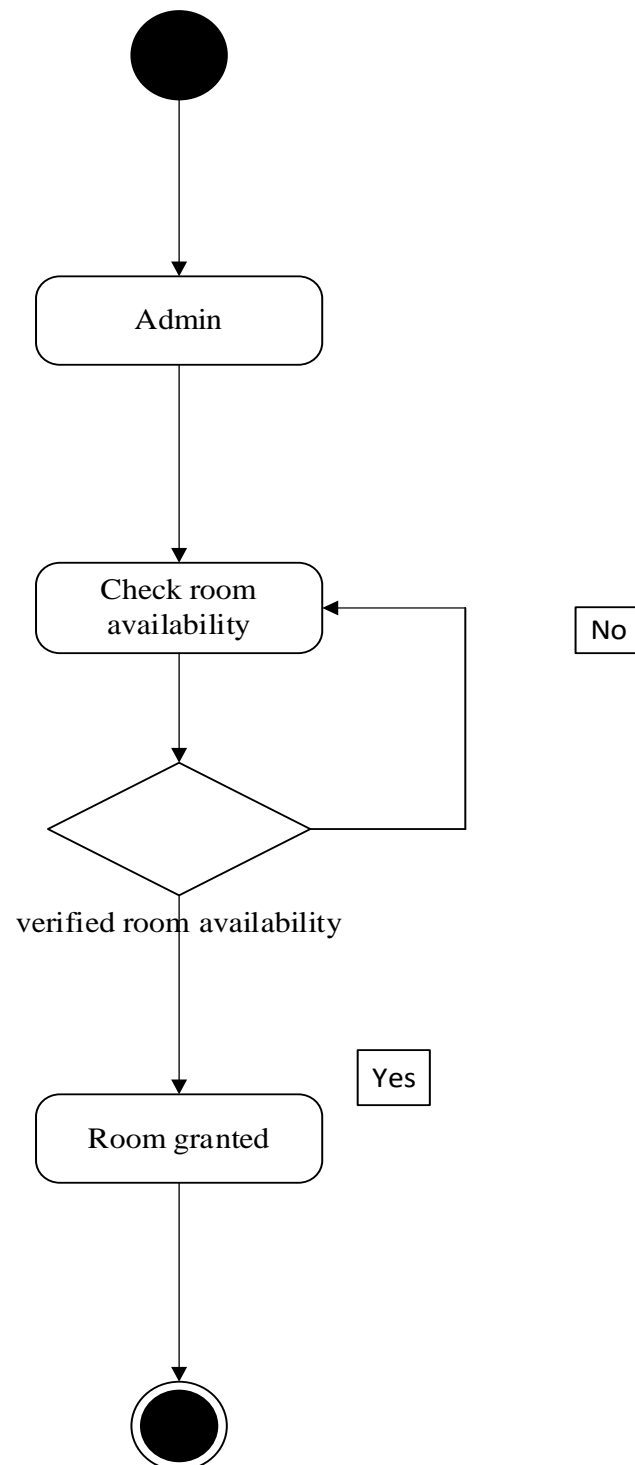
Activity Diagram for Rooms Available

Figure 48: Rooms Available

Description:

Admin checks room availability. If No, Return back to room checking page. If Yes, room availability verified and room granted.

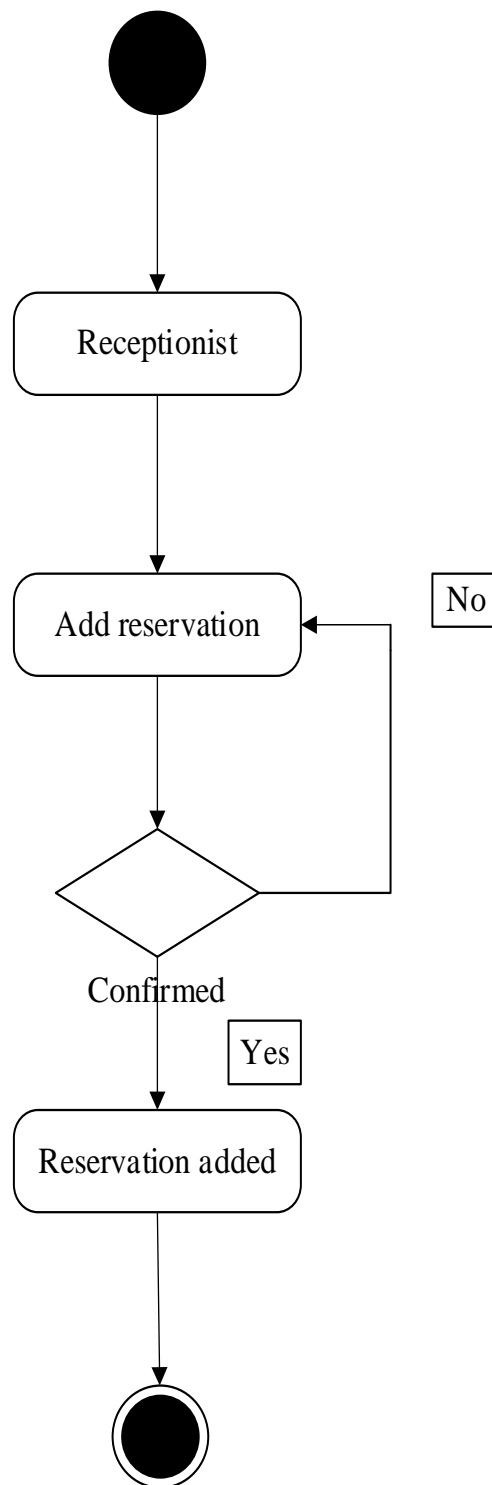
Activity Diagram for Add Reservation

Figure 49: Add Reservation

Description:

Receptionist adds reservation. If No, Return to add reservation. If Yes, Reservation added.

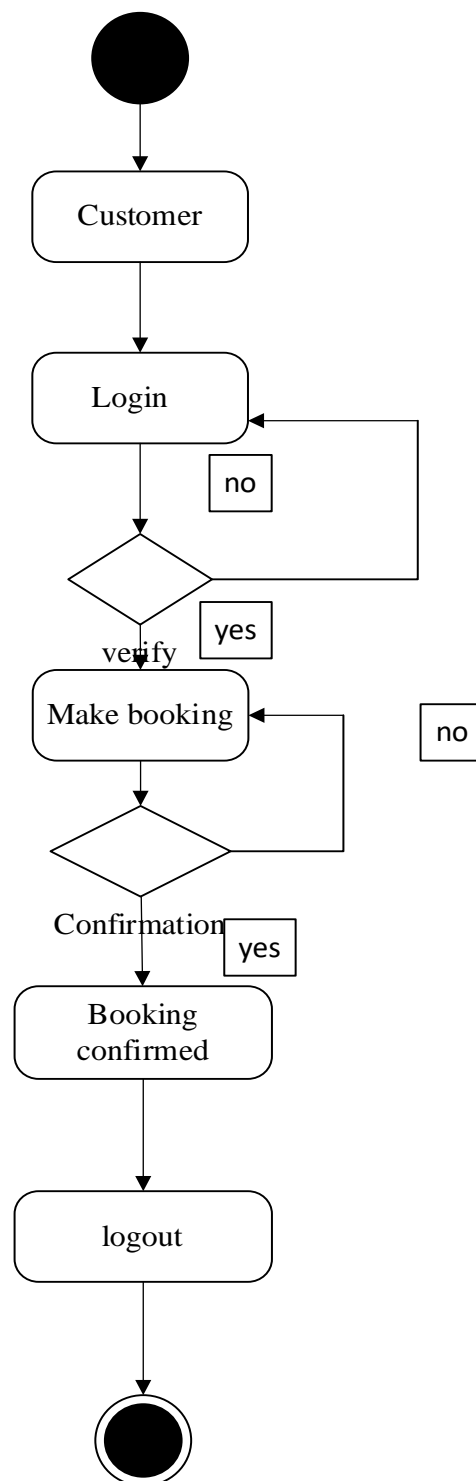
Activity Diagram for Make Booking

Figure 50: Make Booking

Description:

Customer Login and makes booking. If No, Return back. If Yes, Booking confirmed.

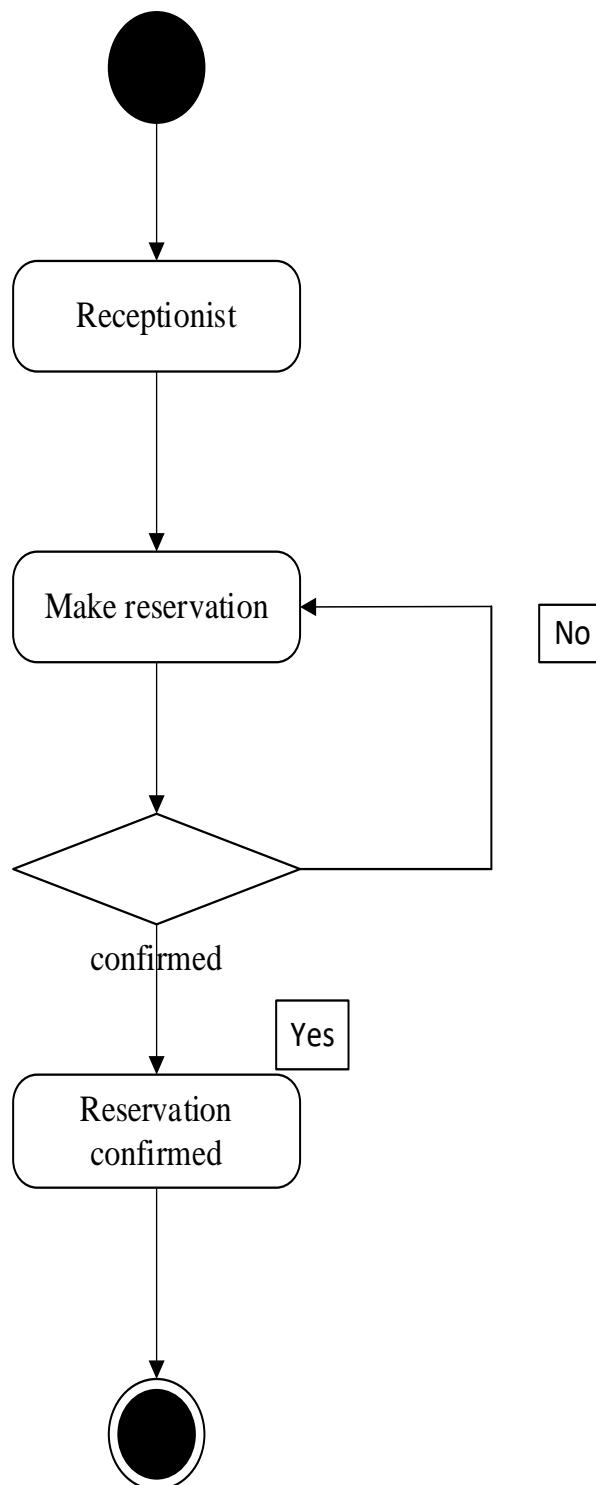
Activity Diagram for Make Reservation

Figure 51: Make Reservation

Description:

Receptionist makes reservation. If No, Return to make reservation. If Yes, Reservation confirmed.

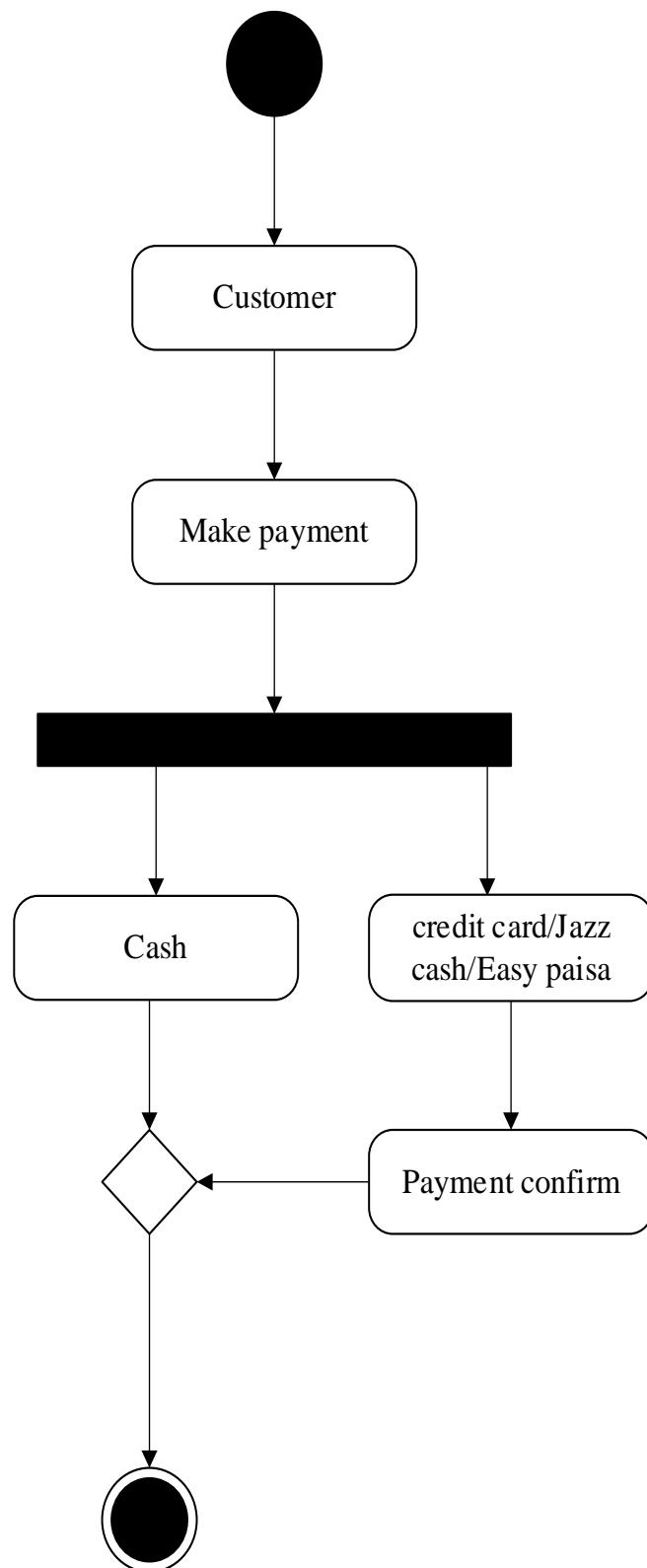
Activity Diagram for Make Payment

Figure 52: Make Payment

Description:

Customer make payment with cash and credit card and payment confirmed.

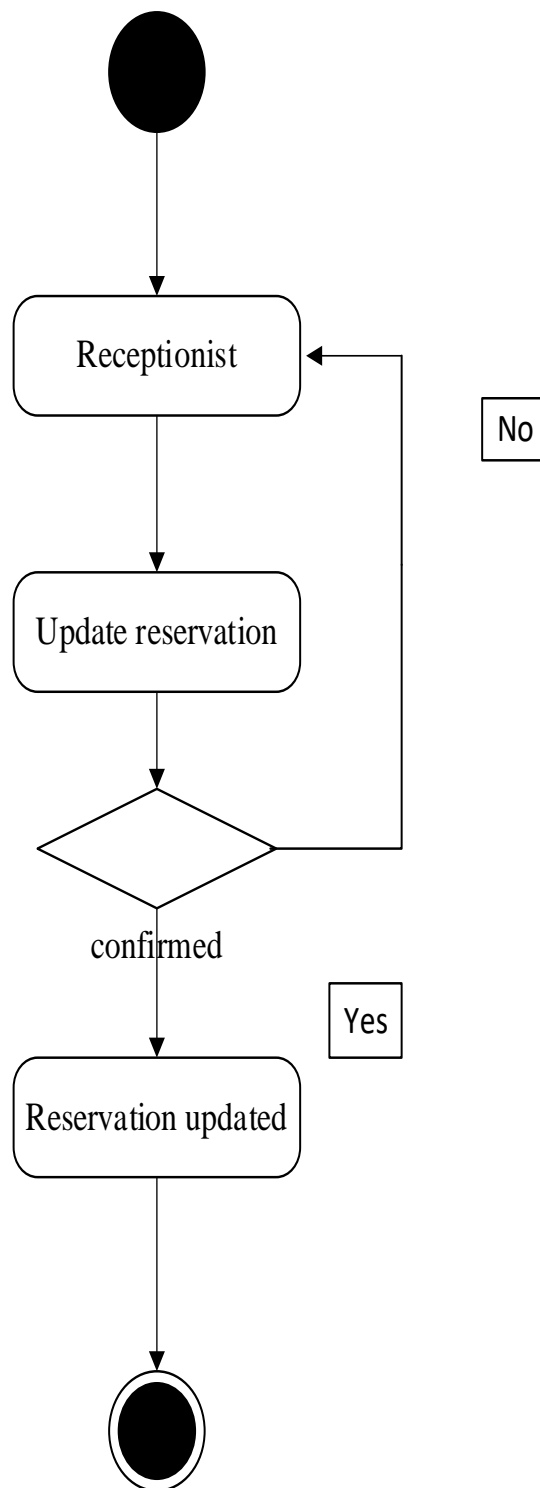
Activity Diagram for Reservation Update

Figure 53: Reservation Update

Description:

Receptionist update reservation. If No, Return to update reservation. If Yes, Reservation updated.

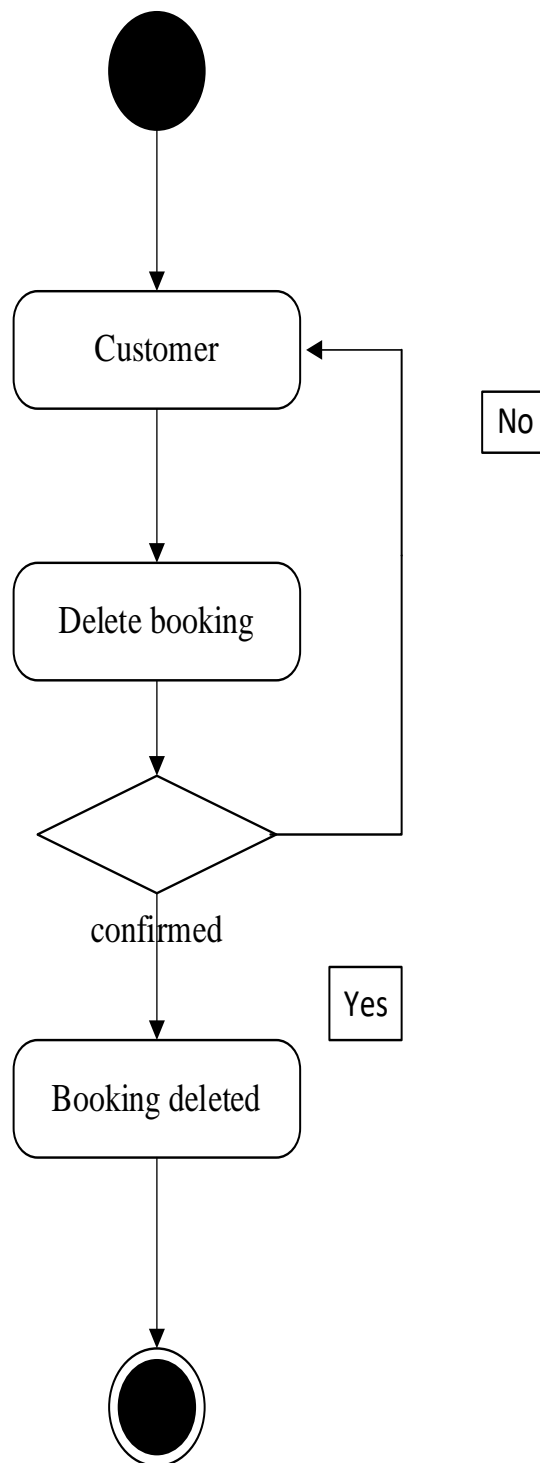
Activity Diagram for Delete Booking

Figure 54: Delete Booking

Description:

Customer deletes booking. If No, Return to delete booking page. If Yes, Booking deleted.

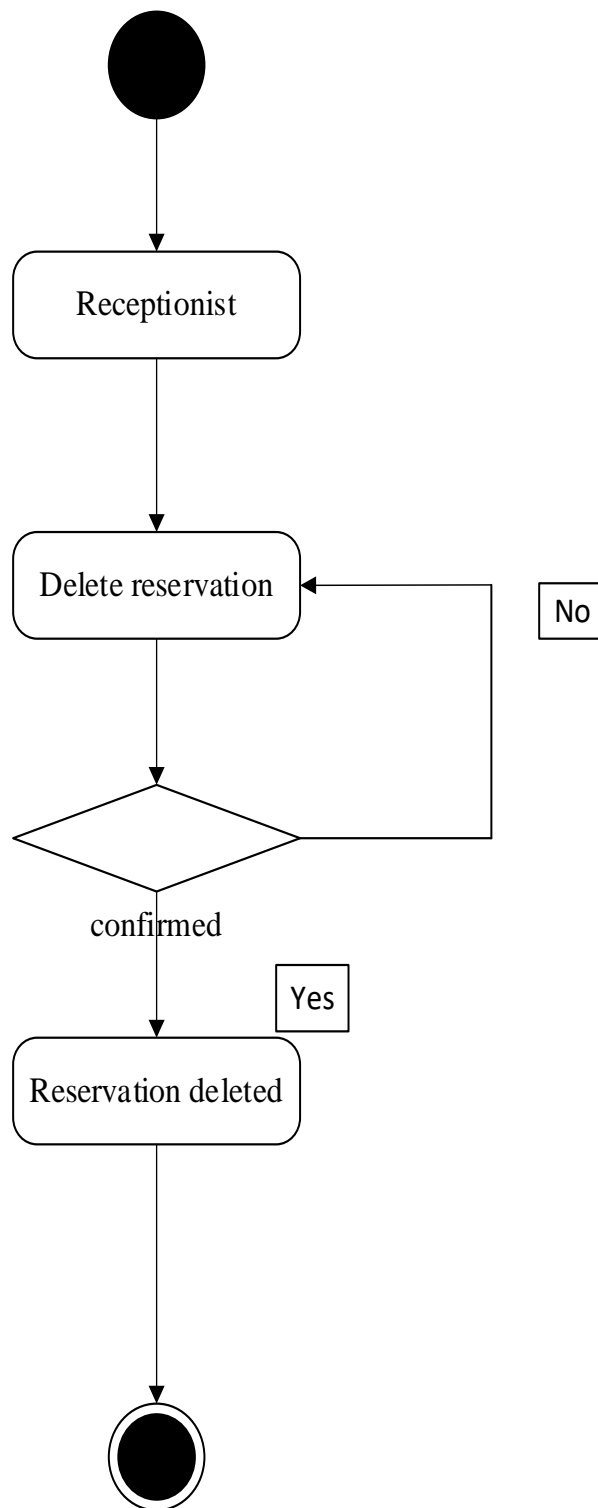
Activity Diagram for Delete Reservation

Figure 55: Delete Reservation

Description:

Receptionist deletes reservation. If No, Return to delete reservation page. If Yes, Reservation deleted.

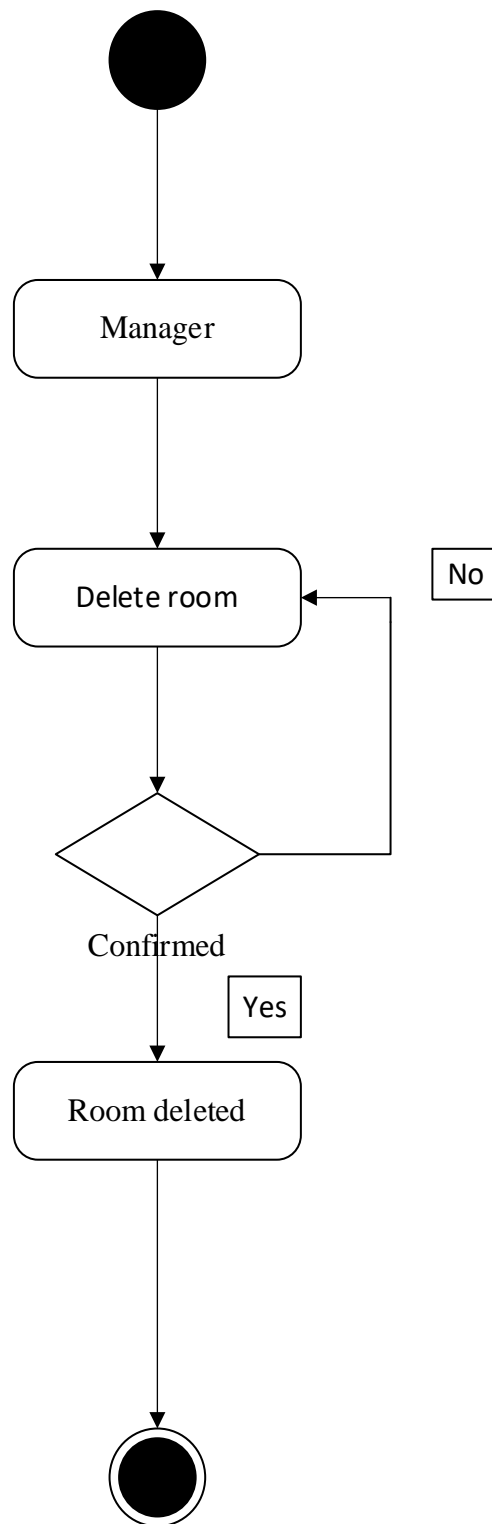
Activity Diagram for Delete Room

Figure 56: Delete Room

Description:

Manager deletes Room. If No, Return to delete room page. If Yes, Room deleted.

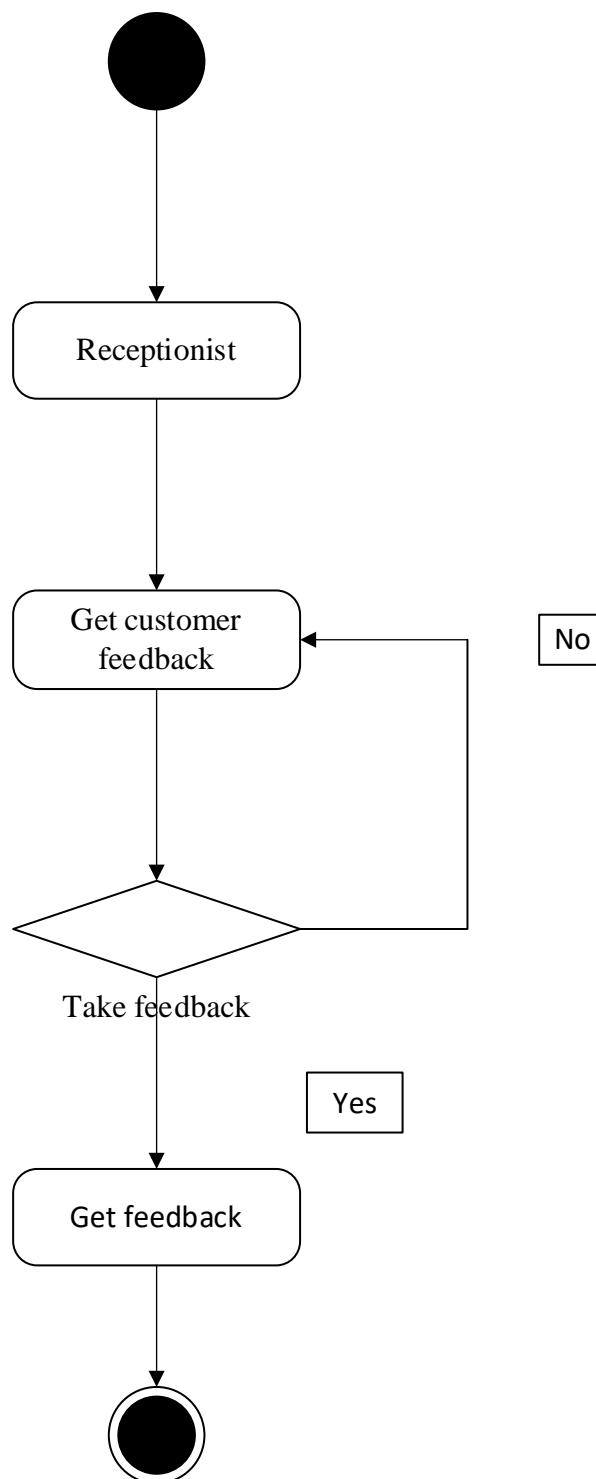
Activity Diagram for Customer Feedback

Figure 57: Customer Feedback

Description:

Receptionist get customer feedback. If No, Return to customer feedback page. If Yes, Feedback got.

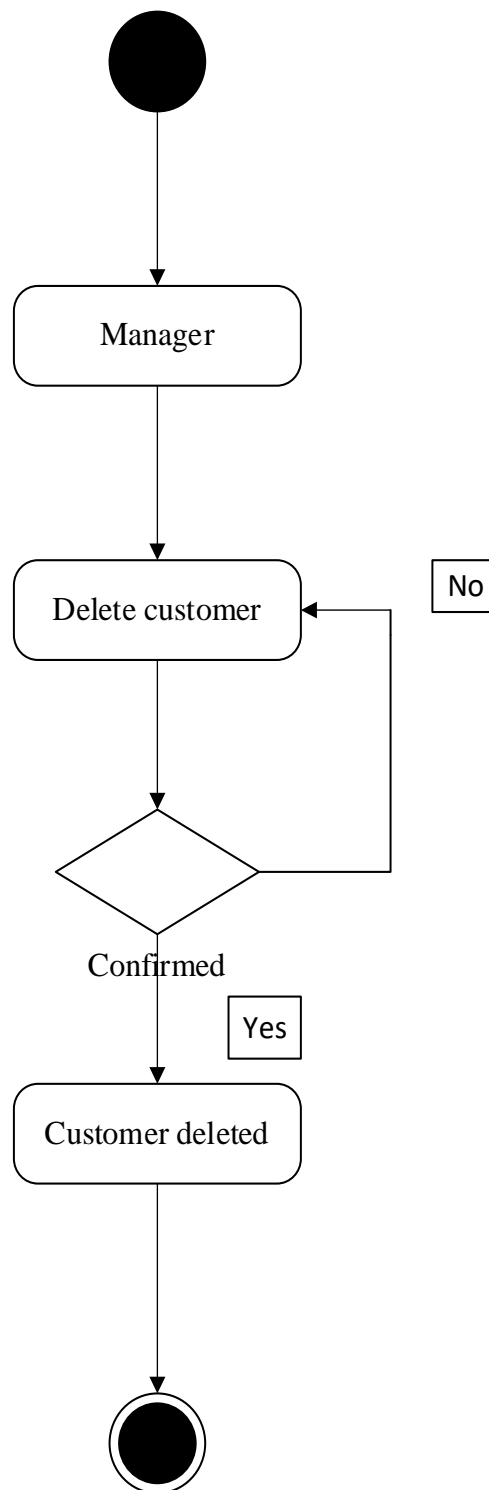
Activity Diagram for Delete Customer

Figure 58: Delete Customer

Description:

Manager deletes customer. If No, Return to delete customer page. If Yes, Customer deleted.

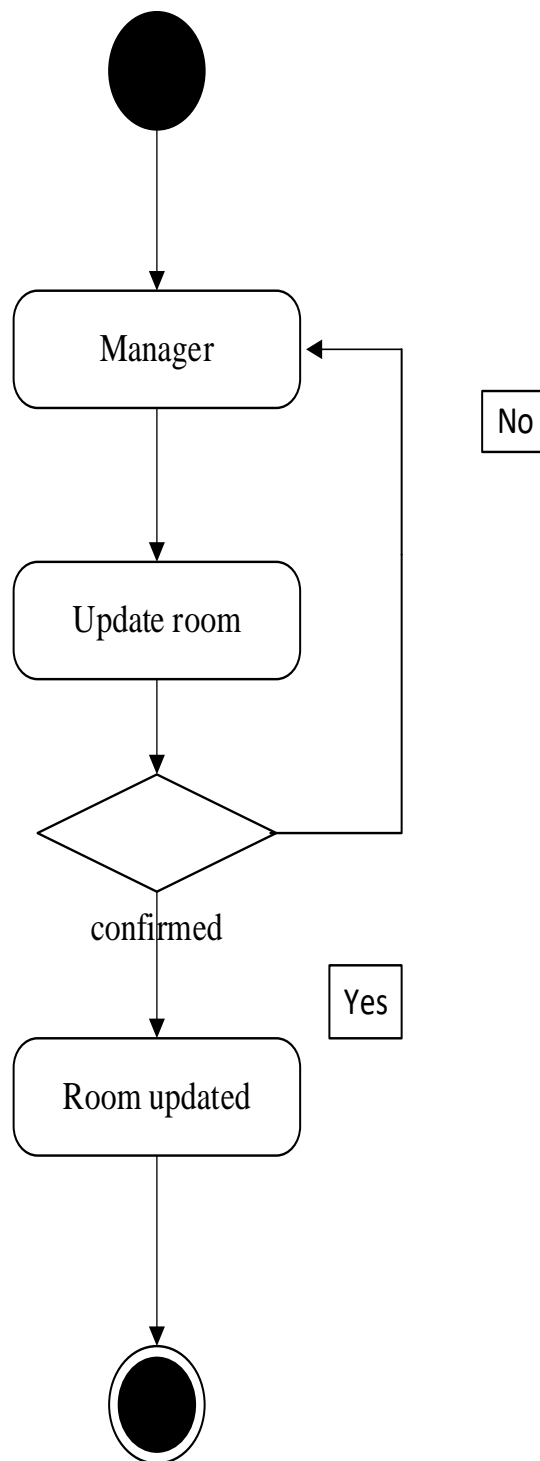
Activity Diagram for Room Update

Figure 59: Update Room

Description:

Manager updates room. If No, Return to update room page. If Yes, Room updated.

Activity Diagram for Logout

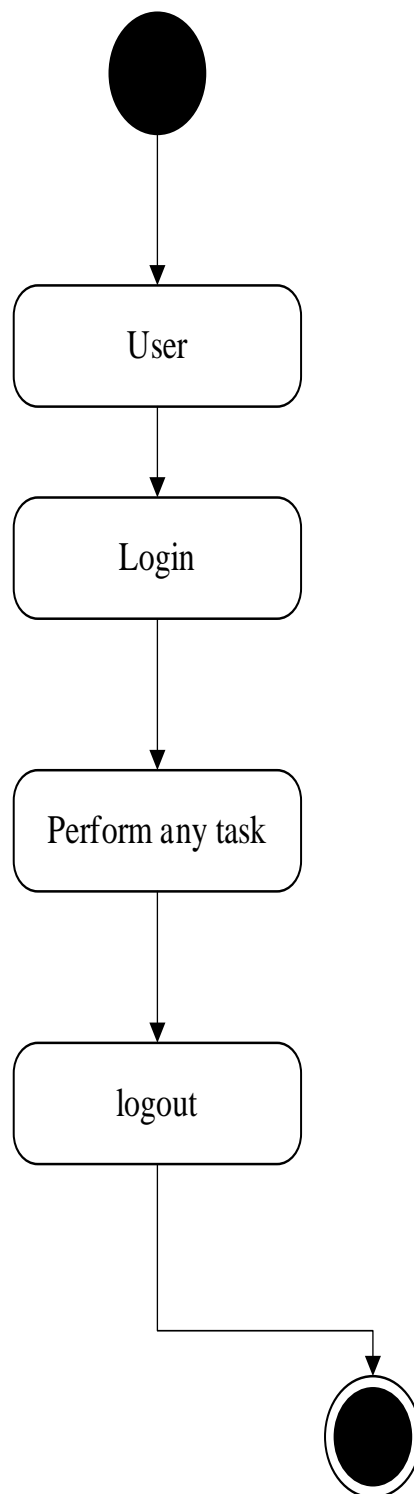


Figure 60: Logout

Description:

User login and perform any task and logouts.

4.8 Collaboration Diagram

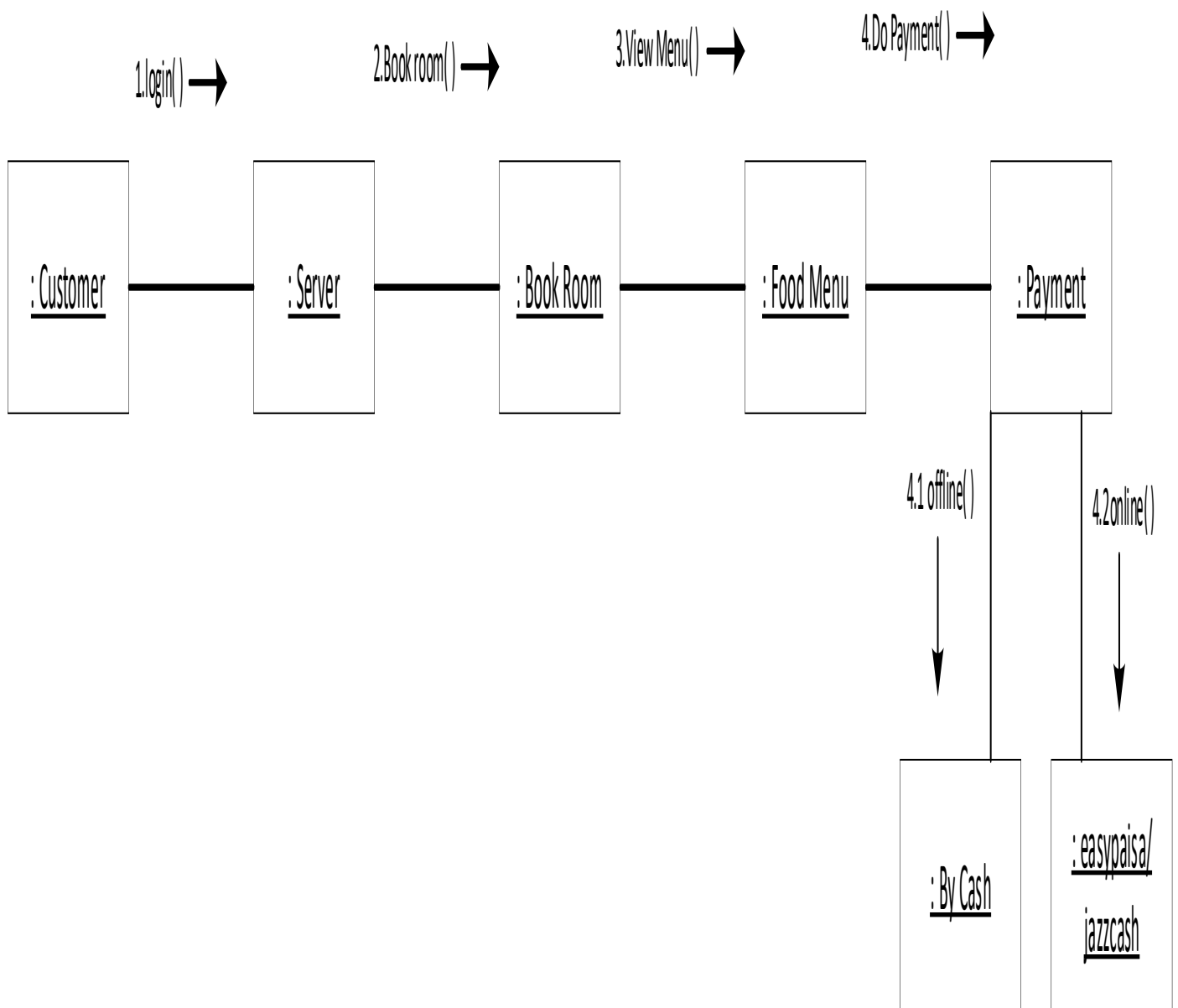


Figure 61: Collaboration diagram

Description:

In HMS, this diagram works as a communication for our roles in our system. It shows an illustration of the relationships and interactions among system objects.

4.9 State Transition Diagram

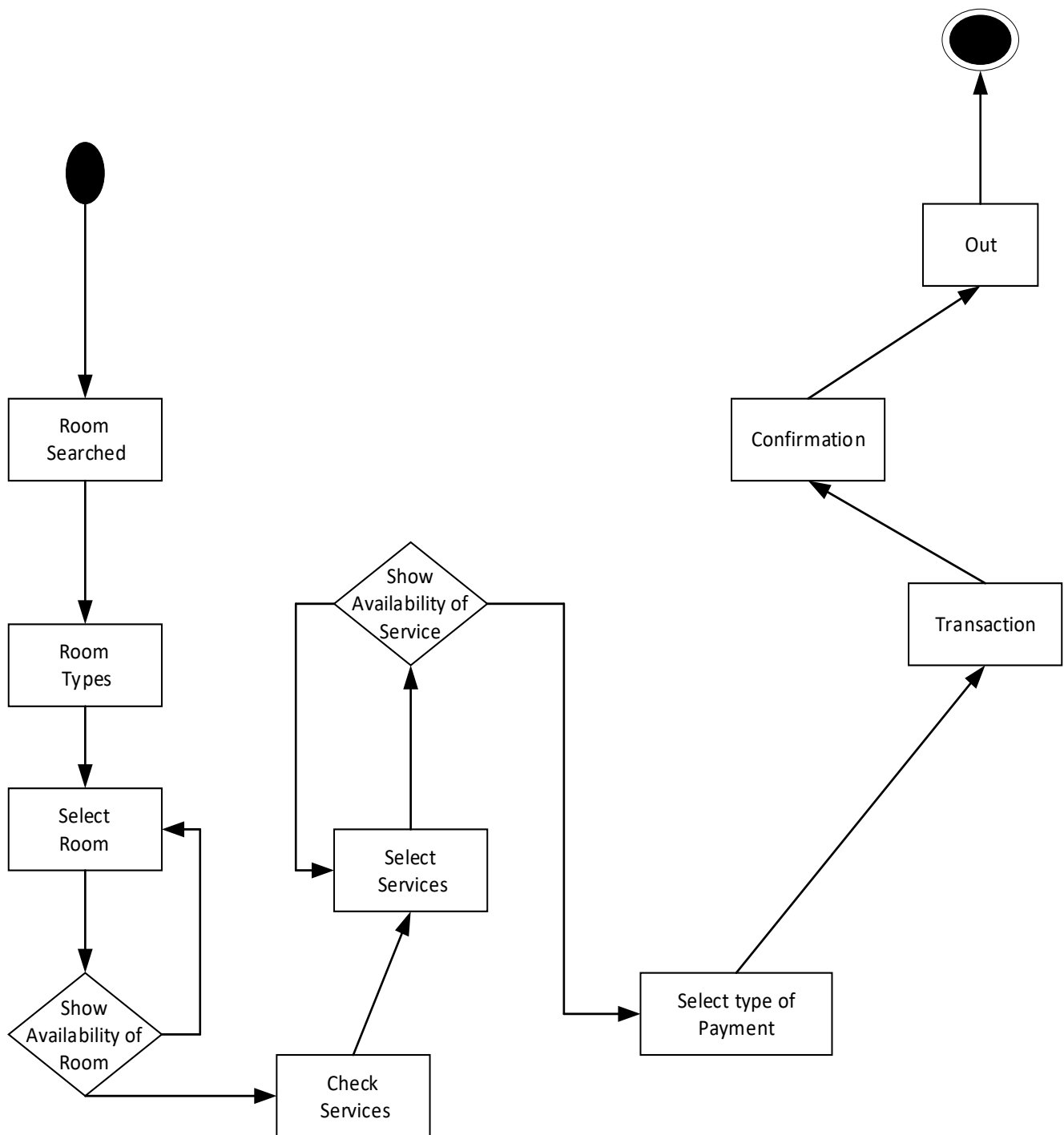


Figure 62: State Transition Diagram

Description:

State transition diagram is analyzed and represented by a series of event that can occur in possible states.

5. Implementation

5.1 Component Diagram

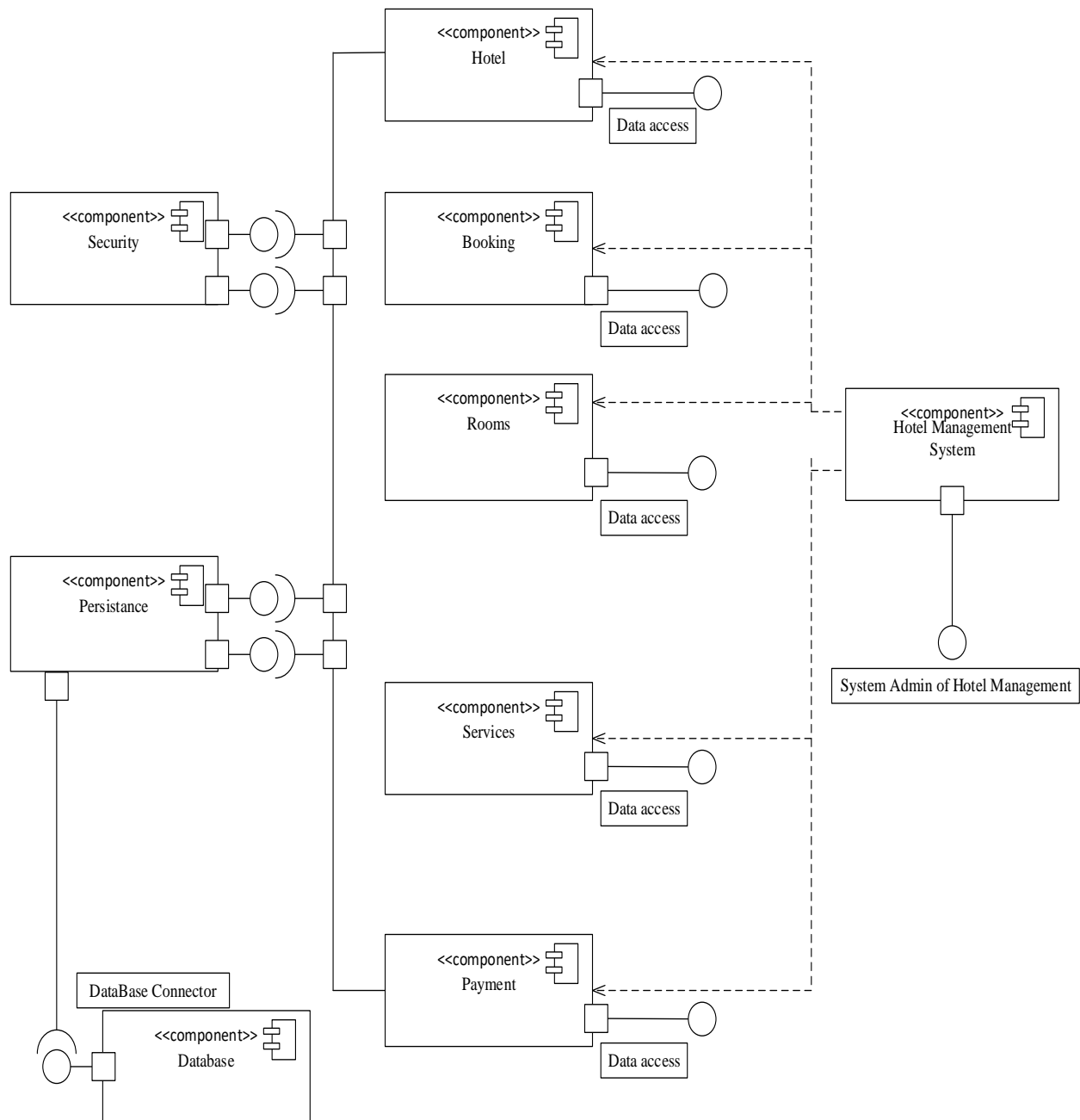


Figure 63: Component Diagram

Description:

In HMS, this diagram does not describe the functionality of the system but it describes the components used to make those functionalities.

5.2 Deployment Diagram

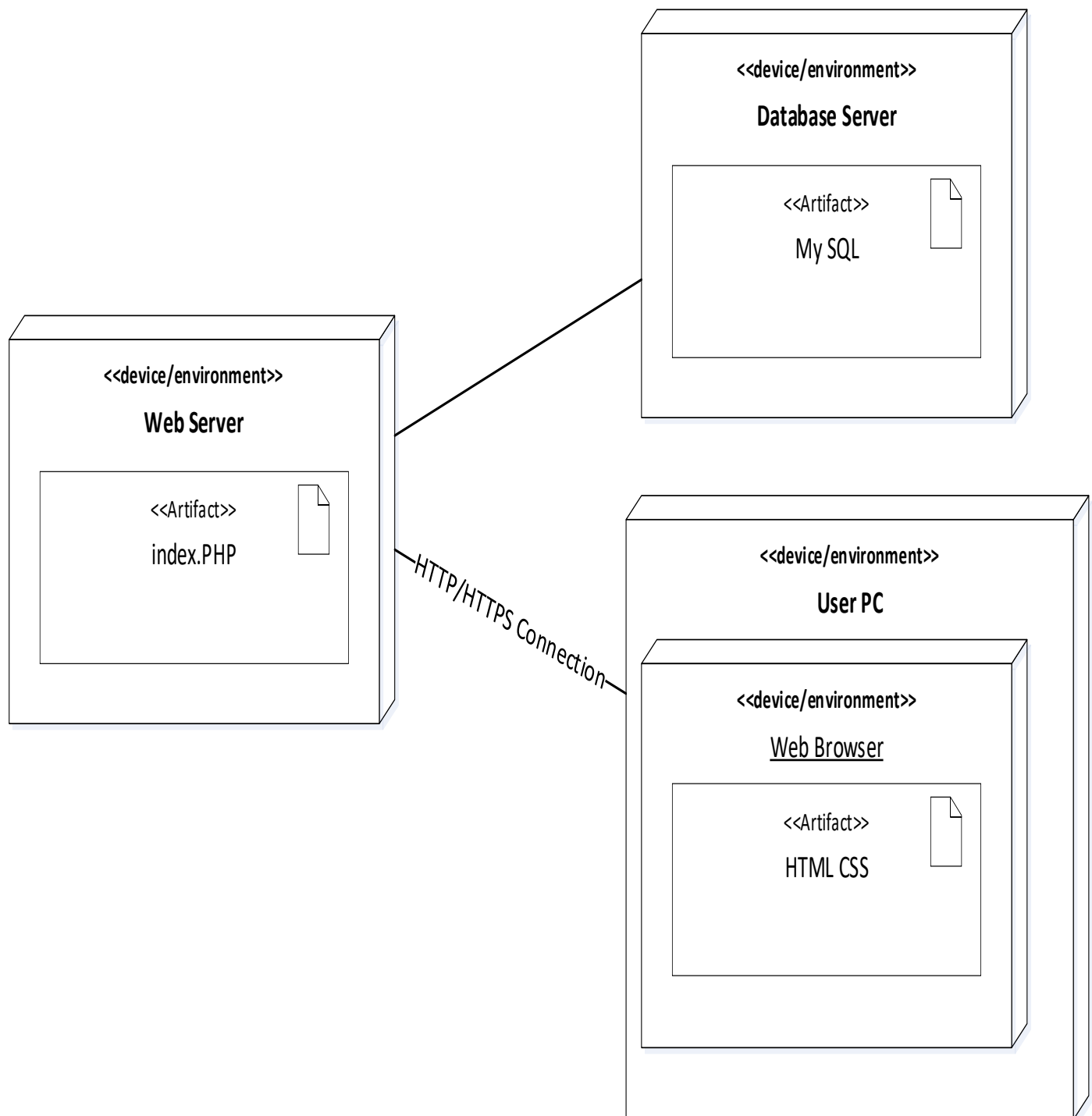


Figure 64: Deployment Diagram

Description:

In HMS, this diagram describes the execution architecture of HMS. This deployment diagram visualizes the physical hardware and software.

5.3 Database Architecture

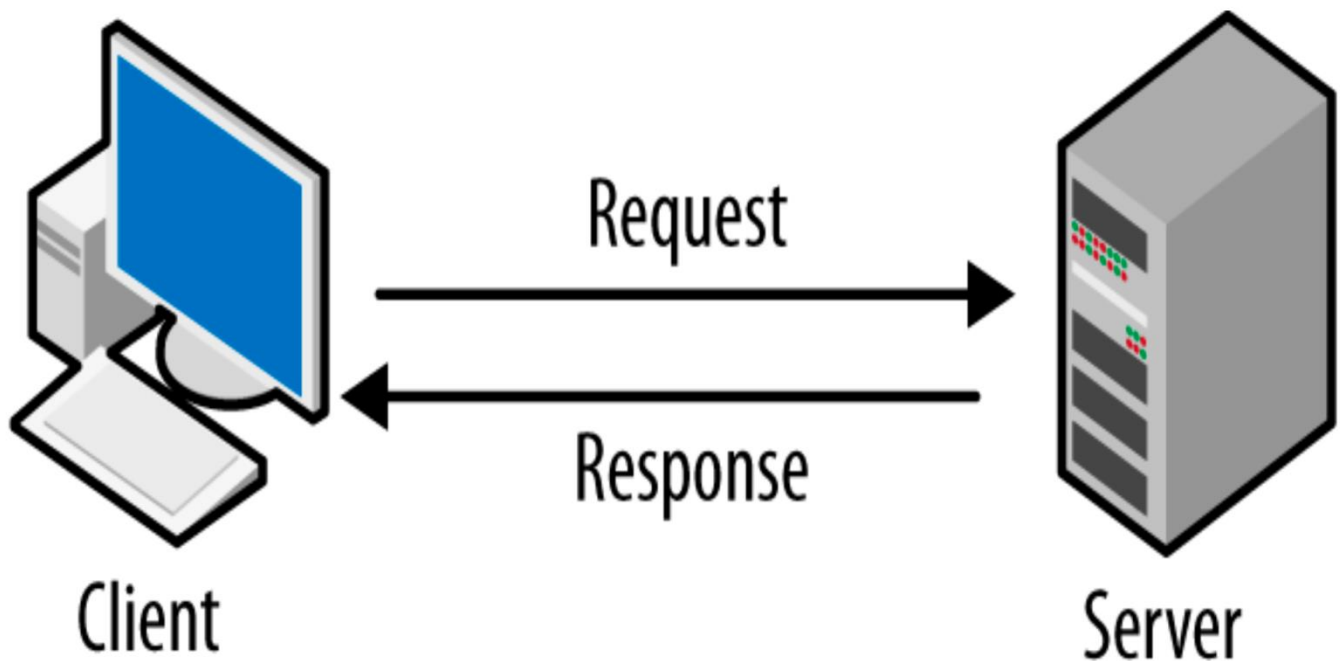


Figure 65: Database Architecture

A 3 Tier Architecture in DBMS is the most popular client server architecture in DBMS for development and maintenance of functional processes, logic, and data access, data storage and user interface is done independently being separate modules. 3 Tier architecture contains a presentation layer, an application layer and a database server.

3- Tier database Architecture design is an extension of the 2- tier client-server architecture. A 3- tier architecture has the following layers:

1. Presentation layer (for PC, Tablet, Mobile etc)
2. Application layer (server)
3. Database Server

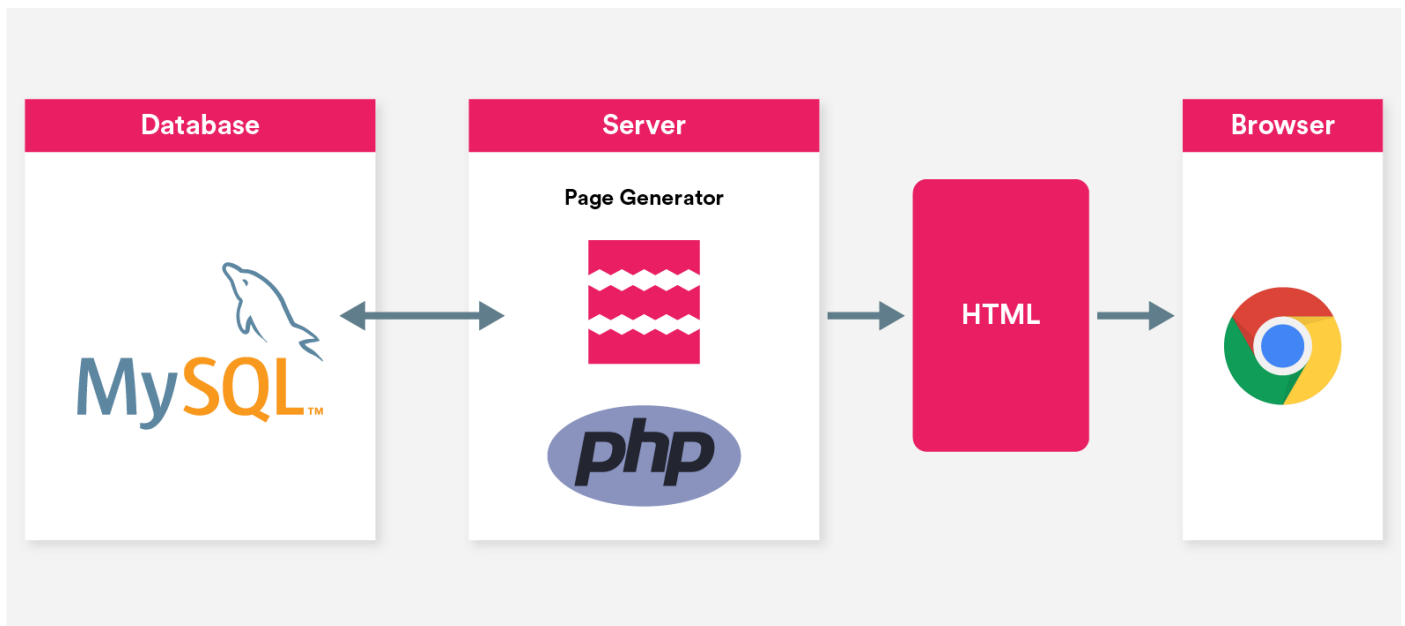


Figure 66: 3-tier architecture

The Application layer resides between the user and the DBMS, which is responsible for communicating the user's request to the DBMS system and send the response from the DBMS to the user. The application layer (business logic layer) also processes functional, logic, constraint, and rules before passing data to the user or down to the DBMS.

The goal of 3- tier client-server architecture is:

- To separate the user applications and physical database
- To support DBMS characteristics
- Program-data independence
- Supporting multiple views of the data

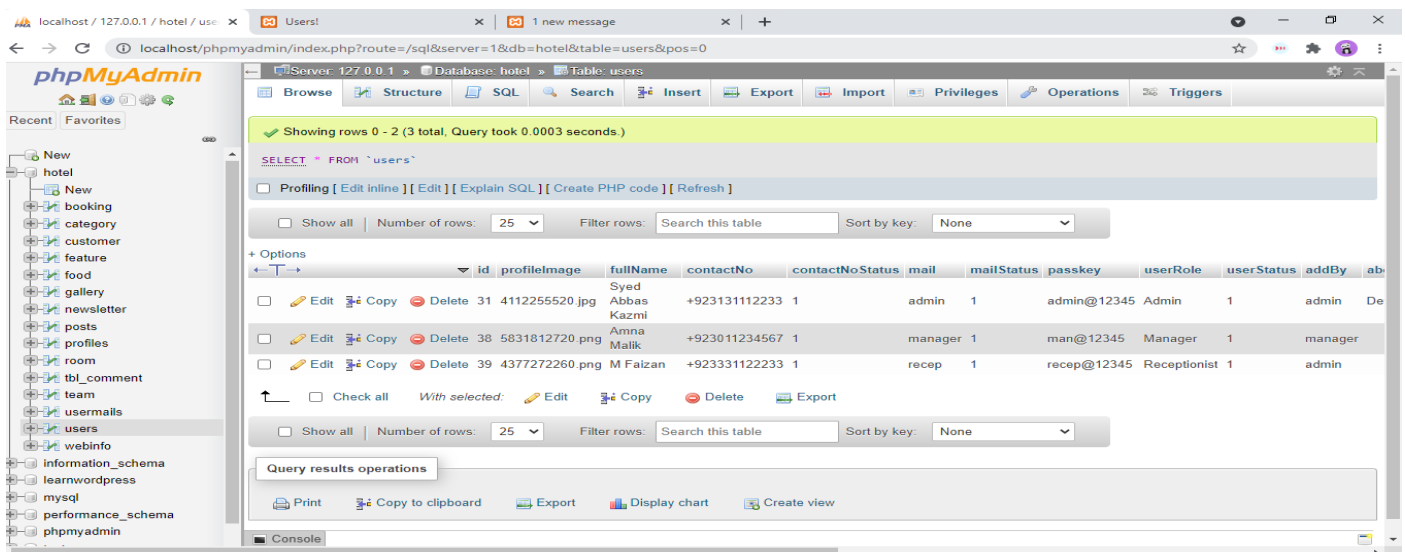


Figure 67: Database Table for Users

The screenshot shows the phpMyAdmin interface for the 'hotel' database. The 'webinfo' table is selected, and its structure and data are displayed. The table has 8 columns: id, logo, companyName, companyTagLine, companyContactNo, companyEmail, companyAddress, and date. The data shows one record for 'Royal Sheikhpura Hotel'.

id	logo	companyName	companyTagLine	companyContactNo	companyEmail	companyAddress	date
1	4839859805.png	Royal Sheikhpura Hotel	We deliver trust	+923011234567	info@royal.com	Batti Chowk Sheikhpura	2021-11-03 21:31:31

Figure 68: Database Table for Webinfo

The screenshot shows the phpMyAdmin interface for the 'hotel' database. The 'room' table is selected, and its structure and data are displayed. The table has 8 columns: id, roomImage, roomTitle, roomType, roomSubType, roomAbout, noORoom, and date. The data shows 9 records for various room types and prices.

id	roomImage	roomTitle	roomType	roomSubType	roomAbout	noORoom	date
1	9219492531.jpg	A-1	Queen room	Basic	Fantastic room	8	0000-00-00 00:00:00
22	6848481456.jpg	A-2	Queen room	Standard	Fantastic room	7	0000-00-00 00:00:00
23	8746832056.jpg	A-3	Queen room	Premium	Fantastic room	3	0000-00-00 00:00:00
24	2667475527.jpg	B-1	Single room	Basic	Fantastic room	10	0000-00-00 00:00:00
25	2989597307.jpg	B-2	Single room	Standard	Fantastic room	5	0000-00-00 00:00:00
26	1355426114.jpg	B-3	Single room	Premium	Fantastic room	5	0000-00-00 00:00:00
27	2538969005.jpg	C-1	Double room	Basic	Fantastic room	7	0000-00-00 00:00:00
28	6769121869.jpg	C-2	Double room	Standard	Fantastic room	5	0000-00-00 00:00:00
29	230804374.jpg	C-3	Double room	Premium	Fantastic room	5	0000-00-00 00:00:00

Figure 69: Database Table for Rooms

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
SELECT * FROM `posts`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	postImage	postName	postCategory	postText	date
<input type="checkbox"/>	11	3798386382.jpg	About our Royal Sheikhpura Hotel	About us	<h3>Our Royal Sheikhpura Hotel is one of ...	2021-11-05 22:43:23
<input type="checkbox"/>	12	5387653252.png	Our Vision & Mission	About us	<h2>We (Royal Sheikhpura Hotel) deliver t...	2021-11-05 22:47:08

Query results operations: Print, Copy to clipboard, Export, Display chart, Create view

Figure 70: Database Table for Posts

Showing rows 0 - 24 (39 total, Query took 0.0003 seconds.)

```
SELECT * FROM `food`
```

Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	foodName	foodSize	foodType	foodPrice	foodImage	date
<input type="checkbox"/>	5	32 EGG MCMUFFIN	Burger		350	8301299372.jpg	2021-10-28 17:34:16
<input type="checkbox"/>	6	CHEESE2BURGER	Burger		300	3469700816.jpg	2021-10-28 17:35:43
<input type="checkbox"/>	7	CHEESEBURGER	Burger		320	7762055349.jpg	2021-10-28 17:37:05
<input type="checkbox"/>	8	DOUBLE SHACK	Burger		480	4612719016.jpg	2021-10-28 17:37:38
<input type="checkbox"/>	9	FILET-O-FISH	Burger		560	2838198619.jpg	2021-10-28 17:38:59
<input type="checkbox"/>	10	MCGRIDDLE	Burger		660	6668521549.jpg	2021-10-28 17:44:22
<input type="checkbox"/>	11	QUARTER POUNDER	Burger		750	3472859504.jpg	2021-10-28 17:47:42
<input type="checkbox"/>	12	SHAKESHACK	Burger		750	3669634445.jpg	2021-10-28 17:49:15
<input type="checkbox"/>	13	STEAKBURGER	Burger		850	4167291247.jpg	2021-10-28 17:50:00
<input type="checkbox"/>	14	WHOPPER	Burger		710	6090378852.jpg	2021-10-28 17:50:44
<input type="checkbox"/>	15	ANIMAL STYLE	Burger		350	4879704913.jpg	2021-10-28 17:51:27
<input type="checkbox"/>	16	BACON CHEESEBURGER	Burger		320	7673667048.jpg	2021-10-28 17:52:08
<input type="checkbox"/>	17	BIG MAC	Burger		450	125423450.jpg	2021-10-28 17:53:58

Figure 71: Database Table for Food

localhost / 127.0.0.1 / hotel / cus

Users!

localhost/phpmyadmin/index.php?route=/sql&server=1&db=hotel&table=customer&pos=0

Server: 127.0.0.1 Database: hotel Table: customer

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.)

SELECT * FROM `customer`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table

+ Options

id	customerName	customerImage	CustomerMail	CustomerContact	date
0	Ahsan Jutt	https://lh3.googleusercontent.com/a-/AOh14GiY-2-q9...	jutt.mail070@gmail.com		2021-10-14 09:58:46
0	Muhammad Ahsan Yousaf	https://lh3.googleusercontent.com/a-/AOh14GgEFC1Fy...	m.ahsanyousaf5@gmail.com		2021-10-19 09:39:15
0	Muhammad Ahsan Yousaf	https://lh3.googleusercontent.com/a-/AOh14GgEFC1Fy...	m.ahsanyousaf5@gmail.com		2021-10-19 10:16:56
0	Muhammad Ahsan Yousaf	https://lh3.googleusercontent.com/a-/AOh14GgEFC1Fy...	m.ahsanyousaf5@gmail.com		2021-10-19 14:38:25
0	Muhammad Ahsan Yousaf	https://lh3.googleusercontent.com/a-/AOh14GgEFC1Fy...	m.ahsanyousaf5@gmail.com		2021-10-19 17:24:21

Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Figure 72: Database Table for Customer

localhost / 127.0.0.1 / hotel / boo

Users!

1 new message

localhost/phpmyadmin/index.php?route=/sql&server=1&db=hotel&table=booking&pos=0

Server: 127.0.0.1 Database: hotel Table: booking

Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.)

SELECT * FROM `booking`

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	profileImage	fullName	mail	contactNo	cnic	checkin	checkout
<input type="checkbox"/> Edit Copy Delete	13	https://lh3.googleusercontent.com/a/AATXAJxkhfvcw6...	InnoPrince	innoprince7@gmail.com	+923131112233	35404-11122333-1	2021-11-10	2021-11-17
<input type="checkbox"/> Edit Copy Delete	14	https://lh3.googleusercontent.com/a/AATXAJxkhfvcw6...	InnoPrince	innoprince7@gmail.com	+923331122333	35404-11122333-1	2021-11-13	2021-11-14
<input type="checkbox"/> Edit Copy Delete	15	https://lh3.googleusercontent.com/a/AATXAJxkhfvcw6...	InnoPrince	innoprince7@gmail.com	031011122333	35404-1111111-1	2021-11-15	2021-11-16
<input type="checkbox"/> Edit Copy Delete	16	https://lh3.googleusercontent.com/a/AATXAJxkhfvcw6...	InnoPrince	innoprince7@gmail.com	030111122333	35303-4111111-1	2021-11-12	2021-11-13
<input type="checkbox"/> Edit Copy Delete	17	https://lh3.googleusercontent.com/a/AATXAJxkhfvcw6...	InnoPrince	innoprince7@gmail.com	030111122333	35303-4111111-1	2021-11-12	2021-11-13

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Figure 73: Database Table for Booking

Chapter No 6: Testing (Software Quality Attributes)

6.1 Test Case Specification

Main Screen

Screen Name	Main Screen
Test Case ID	TC:01
Engineer	SYED ABBAS KAZMI M FAIZAN AMNA MALIK
Website Name	RSH
Testing Cycle	01
Purpose	For checking that all controls or properly display
Scenario	All controls are visible and button are shown
Environment	Online
Pre-Requisite	No
Strategy	Click On Exe File
Expected Result	Main Screen Will Display
Observations	Screen width may be ask to you in case of running by exe file
Result	NO bug on Main Screen
Re-Test	As Above

Table 17: Test Case for Main Screen

Registration

Test Name	Registration
Test Case ID	TC:02
Engineer	SYED ABBAS KAZMI M FAIZAN AMNA MALIK
Website Name	RSH
Testing Cycle	01
Purpose	For checking that Play Button Working Properly
Environment	Online
Pre-Requisite	Main Screen Visible
Strategy	Click On Play icon Then this window will open
Expected Result	When Click on Register button
Observations	Register Button
Result	NO bug
Re-Test	As Above

Table 18: Test Case for Registration

Room Booking

Test Name	Room Booking
Test Case ID	TC:03
Engineer	SYED ABBAS KAZMI M FAIZAN AMNA MALIK
Website Name	RSH
Testing Cycle	01
Purpose	Select Room
Scenario	Select Available Days & Date
Environment	Online
Strategy	After select then book
Expected Result	Done button is properly working
Result	NO bug
Re-Test	As Above

Table 19: Test Case for Room Booking

Login

Test Name	Login Success Window
Test Case ID	TC:04
Engineer	SYED ABBAS KAZMI M FAIZAN AMNA MALIK
Website Name	RSH
Testing Cycle	010
Purpose	For checking all controls on it working properly or not
Scenario	In this screen you see 3 buttons (Login, Register , booking)
Environment	Online
Pre-Requisite	Home Page Show on 4 page
Expected Result	All 3 buttons properly working
Result	NO bug
Re-Test	As Above

Table 20: Test Case for Login Success

Login Failed Window

Test Name	Login Failed Window
Test Case ID	TC:05
Engineer	SYED ABBAS KAZMI M FAIZAN AMNA MALIK
Website Name	RSH
Testing Cycle	01
Purpose	For checking buttons are working properly
Scenario	In this screen you see 2 buttons (Login, Register)
Environment	Online
Expected Result	All 2 buttons properly working
Result	NO bug
Re-Test	As Above

Table 21: Test Case for Login fail

6.2 Black Box Test Cases

Registration of All Actors

Use Case Title	Registration of All Actors (Register)	
Use Case ID	1	
Action	<p>User will go to the RSH Web Application. And click Register button.</p> <p>The user will fill it and after sign up submits it.</p> <p>Web Application will verify and check it</p> <p>If it is verified and all data entered is valid then Web Application will create an account for user.</p> <p>Web Application will display a congratulation message if the account is created</p> <p>Web Application will display a message for refill or resubmit in case of process failed</p>	
Description	<p>Registration feature will allow the user to be registered.</p> <p>Registered user will be able to login to the Web Application to get access to all information available in it.</p>	
Alternative Path	N/A	
Pre- Conditions	The user is connected to internet and on the ARS Web Application's home screen and clicks sign up for registration.	
Post- Conditions	The user become registered and now (s)he can perform all operations available in Web Application	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	If the user entered invalid data in the form then the exception of invalid field will occur	

Table 22: Black Box Test Case for Registration

Login of All Actors

Use Case Title	Login of All Actors	
Use Case ID	2	
Action	<p>The user will click login button.</p> <p>The user will enter his email and password and will click the button of login</p> <p>System will check the user id and password and verify it</p> <p>If the id and password is valid then user will enter in his account and can access the website.</p> <p>If the id and password is not valid then system will display a message of invalidation.</p>	
Description	<p>Login option will allow user to enter in his account.</p> <p>It will allow the user to get access to the ARS by providing valid registered id and password</p>	
Alternative Path	N/A	
Pre- Conditions	The user will enter in his account for performing any task by providing id and password	
Post- Conditions	The user will enter his/her account and can perform any task	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	When user will enter a wrong id or password then exception of invalid id or password will come.	

Table 23: Black Box Test Case for Login

Logout of All Actors

Use Case Title	Logout of All Actors	
Use Case ID	3	
Action	<p>An already logged in user clicks on the Log out button.</p> <p>System sends the session out request to the database.</p> <p>User logout successfully.</p> <p>User is routed to the home screen of the Web Application.</p>	
Description	This feature will allow the user to log out from the Web Application after login	
Alternative Path	<p>User cancels the logout action.</p> <p>User has already successfully logout from the Web Application</p>	
Pre- Conditions	<p>User is logged in to the Web Application before log out.</p> <p>User is a registered user</p>	
Post- Conditions	User is logged out from the Web Application and routed to the Home screen of the Web Application.	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	User cannot logout if he is not logged in or registered already.	

Table 24: Black Box Test Case for Logout

Change Password

Use Case Title	Change Password	
Use Case ID	4	
Action	<p>Already logged in User click on change password.</p> <p>Web Application will require old password.</p> <p>User will enter required data and click on change password.</p> <p>Web Application will verify provided details.</p> <p>Password changed successfully.</p>	
Description	This feature will allow the user to change password from the Web Application.	
Alternative Path	User can cancel password change action.	
Pre- Conditions	<p>User must visit the Web Application.</p> <p>User has valid email.</p> <p>User must be register member of the Web Application.</p> <p>User must click change password button.</p>	
Post- Conditions	Password changed successfully.	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	User cannot change password if it is not registered.	

Table 25: Black Box Test Case for Change Password

Update Profile

Use Case Title	Update Profile	
Use Case ID	5	
Action	<p>Already registered and logged in User click on Update Profile Option.</p> <p>Web Application will show the profile form.</p> <p>User will enter relevant details.</p> <p>User will click on Update Profile button.</p> <p>Profile updated successfully.</p>	
Description	This feature will allow the user to update his/her profile from the Web Application.	
Alternative Path	User can cancel update profile action.	
Pre- Conditions	<p>User must visit the Web Application with correct login details.</p> <p>User must be register member of the Web Application.</p> <p>User must click Update Profile button.</p>	
Post- Conditions	<p>Profile Updated successfully.</p> <p>Success message shown to user</p> <p>User is routed to the Home screen of the Web Application</p>	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	User cannot Update Profile if it is not registered or logged in.	

Table 26: Black Box Test Case for Update Profile

Manage Users

Use Case Title	Manage Users	
Use Case ID	6	
Action	<p>User (Admin) click on Manage Users Option.</p> <p>Web Application will show the List of all registered users.</p> <p>User can delete any user and suspend his/her login credentials.</p> <p>Action is done successfully.</p>	
Description	This feature will allow the user to Manage (Edit, Update, and Delete) Users in Web Application.	
Alternative Path	User can cancel managing users in the Web Application.	
Pre- Conditions	<p>User must visit the Web Application with correct login details.</p> <p>User must be register member (admin) of the Web Application.</p> <p>User must click on relevant options.</p>	
Post- Conditions	Success message shown to user (admin) if user is deleted	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	User cannot Manage Users if he/she is not admin or logged in.	

Table 27: Black Box Test Case for Manage Users

Add Booking

Use Case Title	Add Booking	
Use Case ID	7	
Action	<p>User (admin) will click on bookings and then click on Add booking in left Side Action Bar.</p> <p>Web Application will show the form to fill details in.</p> <p>User will fill form and click save button.</p> <p>Booking is saved successfully.</p>	
Description	This feature will allow the admin to add new booking in the system.	
Alternative Path	User can cancel this action.	
Pre- Conditions	<p>User must visit the Web Application with admin credentials.</p> <p>User must click on Add bookins in side bar menu options.</p>	
Post- Conditions	Booking is Added Successfully	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	User cannot Add booking if he/she is not admin of Web Application.	

Table 28: Black Box Test Case for Add Booking

Manage Booking

Use Case Title	Manage Booking	
Test Case ID	8	
Action	<p>User (admin) click on Manage booking option from left side menu bar.</p> <p>System will show the list of bookings having paid payment and non-paid.</p> <p>User will click on any booking to change its status from unconfirmed to confirmed, from confirmed to rejected on the bases of payments.</p> <p>Booking is updated accordingly.</p>	
Description	This feature will allow the user (admin) to manage booking.	
Alternative Path	User can cancel managing booking option any time.	
Pre- Conditions	User must visit the Web Application with correct credentials and role permissions must be admin.	
Post- Conditions	User can cancel this option.	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	User cannot manage booking if he/she not the admin of the system.	

Table 29: Black Box Test Case for Manage Booking

Find Rooms

Use Case Title	Find Rooms
Use Case ID	9
Action	<p>User (customer) will see the search box on the front screen of the web app.</p> <p>User will fill the relevant details to filter the desired room package.</p> <p>Web App will show the related results to the user with room information and packages.</p>
Description	This feature will allow the user to find/search rooms.
Alternative Path	User can cancel this action any time.
Pre- Conditions	<p>User must visit the Web Application and should be on main screen of the web app.</p> <p>User must click and fill search options.</p>
Post- Conditions	N/A
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>
Exception	User cannot search room details if he/she not on the main page of the web app.

Table 30: Black Box Test Case for Find Rooms

Book Room Package

Use Case Title	Book Room Package	
Use Case ID	10	
Action	<p>User (customer) will see the room booking on the front screen of the web app.</p> <p>User will fill the relevant details to filter the desired room package.</p> <p>Web App will show the related results to the user with booking information and packages.</p> <p>User will select the desired package and proceed to the room booking.</p> <p>There will be three payment options. i- via Jazzcash, ii- via Easypaisa, iii- via Hand on check-in.</p> <p>User will select the option according to his/her ease and submit the room booking form.</p>	
Description	This feature will allow the User to book a room package.	
Alternative Path	User can cancel this option before submission any time.	
Pre- Conditions	<p>User/Visitor must visit the Web Application with having internet connection.</p> <p>User/Visitor must click on relevant button for options.</p>	
Post- Conditions	N/A	
Author	<p>ABBAS KAZMI 2017-GCUF-059754</p> <p>AMNA MALIK 2017-GCUF-059720</p> <p>MOHAMMAD FAIZAN 2017-GCUF-059730</p>	
Exception	User/Visitor cannot Find and add room booking form if he/she is not on relevant option of the web application.	

Table 31: Black Box Test Case for Book Room

Cancel Booking

Use Case Title	Cancel Booking	
Use Case ID	11	
Action	User click on booking option from left side menu bar after successful login. Web Application will show the relevant option for cancelling booking. User will click on delete option for cancelling booking.	
Description	This feature will allow the User to cancel booking.	
Alternative Path	User can cancel this action any time.	
Pre- Conditions	User must visit the Web Application. User must click on relevant option.	
Post- Conditions	N/A	
Author	ABBAS KAZMI 2017-GCUF-059754 AMNA MALIK 2017-GCUF-059720 MOHAMMAD FAIZAN 2017-GCUF-059730	
Exception	User cannot cancel booking if he/she is not admin.	

Table 32: Black Box Test Case for Cancel Booking

6.3 White Box Testing

6.3.1 Statement Coverage

Statement coverage ensures whether each and every line of the code is executed at least once during testing.

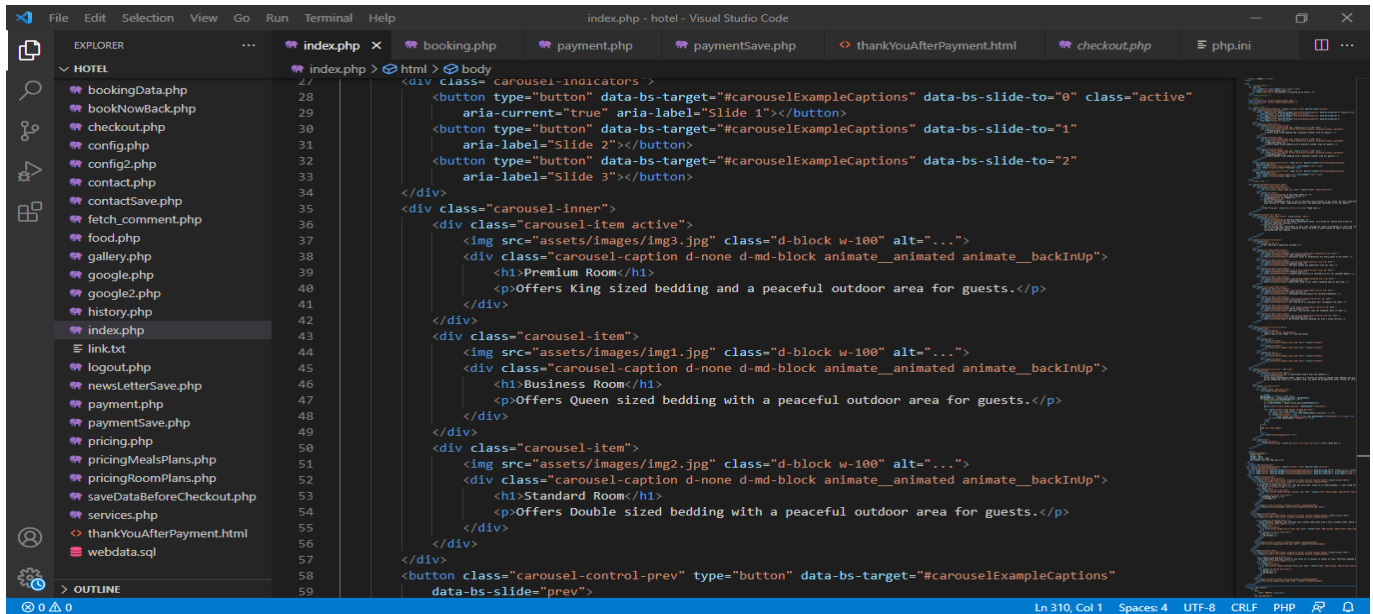


Figure 74: Statement Coverage I

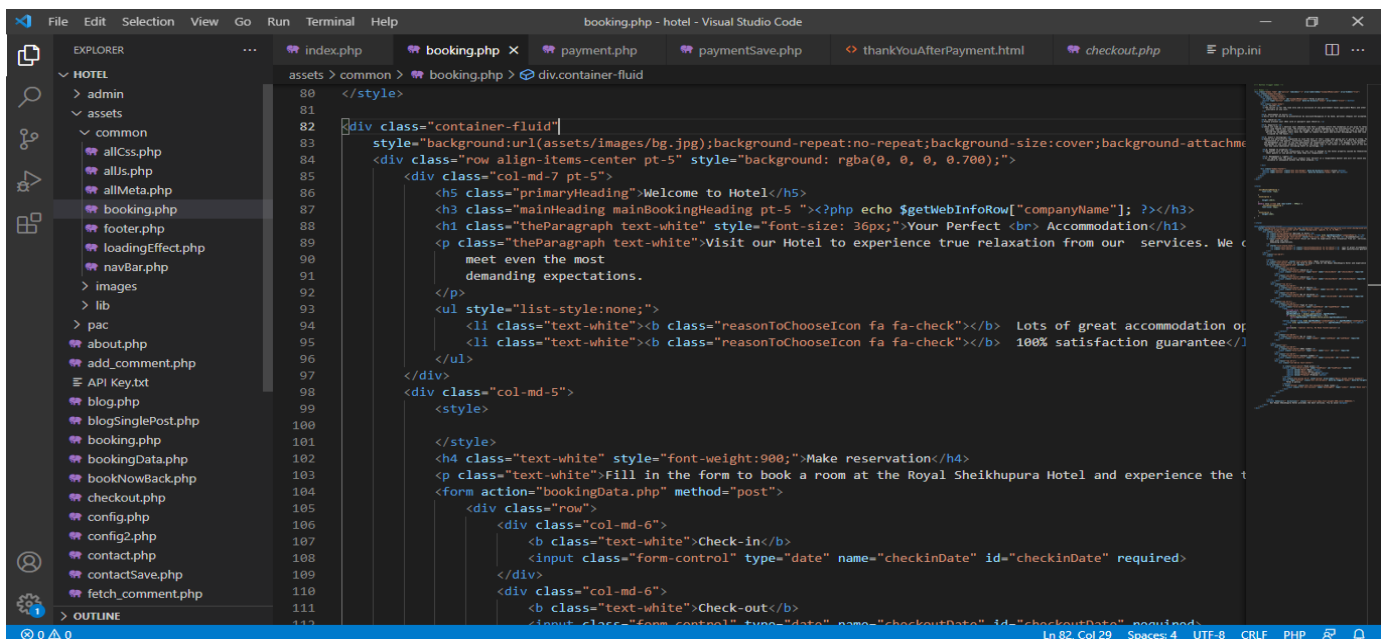


Figure 75: Statement Coverage II

6.3.2 Branch Coverage

Branch coverage ensures that every branch from each decision point is executed.

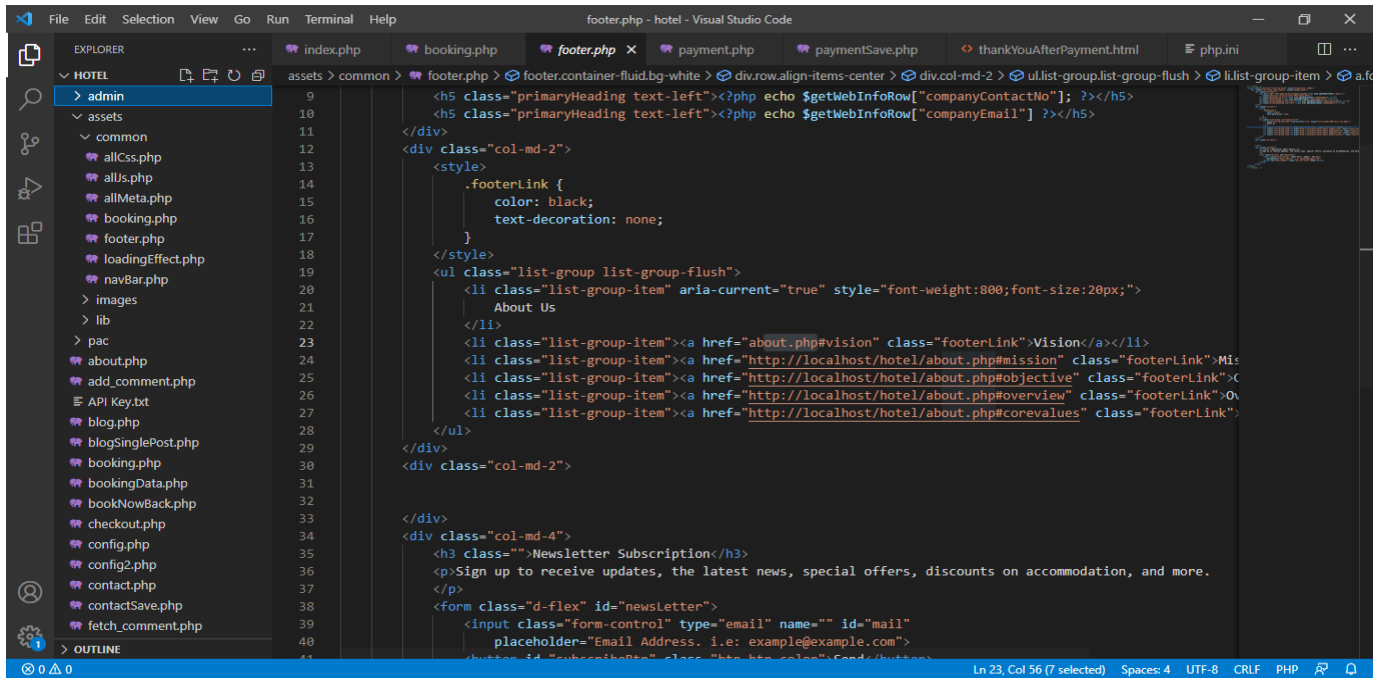


Figure 76: Branch Coverage

6.3.3 Path Coverage

Path coverage ensures that every path is traversed at least once. This technique is used for testing complex programs.

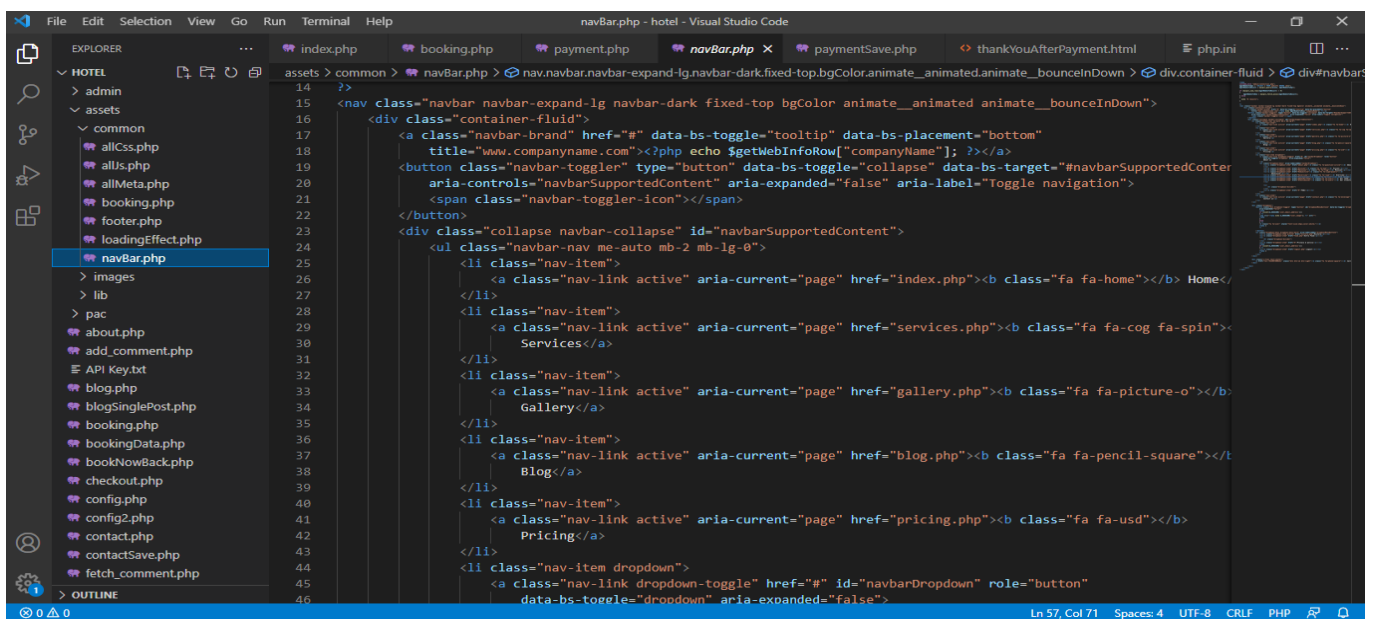




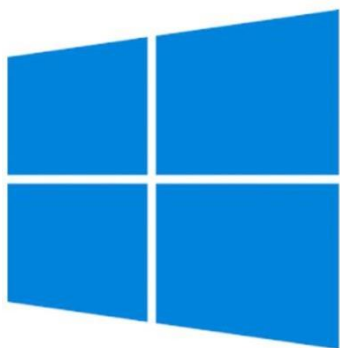
Figure 77: Path Coverage

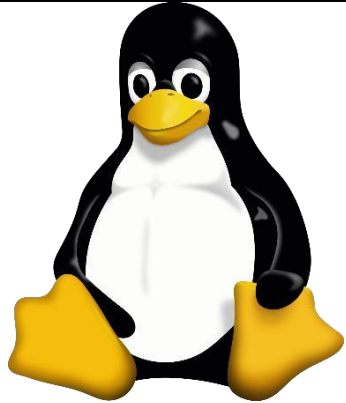


Chapter No 7: Tools and Technologies

7.1 Programming Languages

<p style="text-align: center;">PHP</p> <p>PHP is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group.</p>	
<p style="text-align: center;">Framework Code-Igniter</p> <p>Code-Igniter is an open-source software rapid development web framework, for use in building dynamic web sites with PHP.</p>	

7.2 Operating Environment

<p style="text-align: center;">Windows</p> <p>Microsoft Windows, commonly referred to as Windows, is a group of several proprietary graphical operating system families, all of which are developed and marketed by Microsoft. Each family caters to a certain sector of the computing industry.</p>	
---	---

<p style="text-align: center;">Linux</p> <p>Linux is a family of open-source Unix-like operating systems based on the Linux kernel, an operating system kernel first released on September 17, 1991, by Linus Torvalds. Linux is typically packaged in a Linux distribution.</p>	
<p style="text-align: center;">Unix</p> <p>Unix is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, whose development started in the 1970s at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.</p>	
<p style="text-align: center;">Mac OS</p> <p>macOS is a proprietary graphical operating system developed and marketed by Apple Inc. since 2001. It is the primary operating system for Apple's Mac computers.</p>	

Appendix A: User Documentation

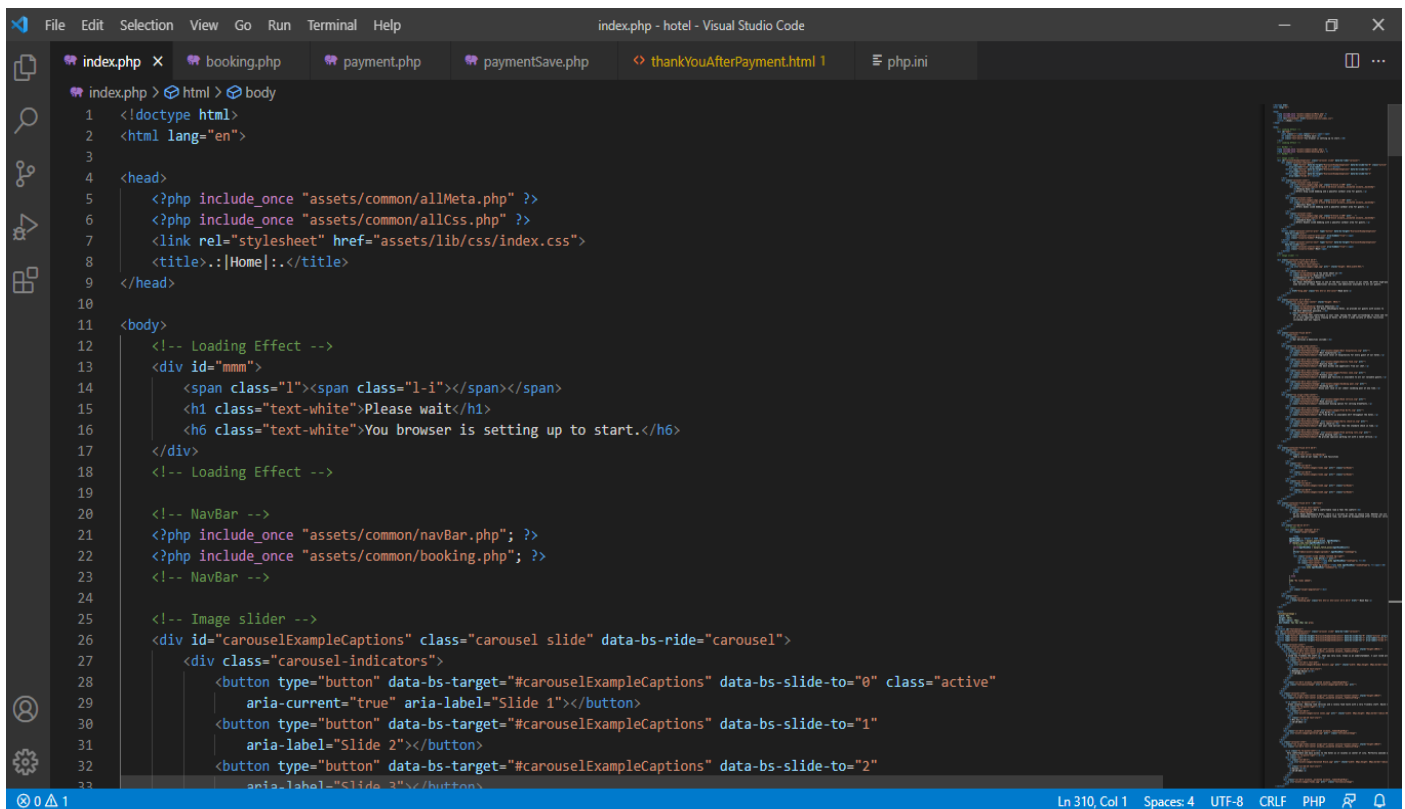
User documentation provided to the employee will give a clear idea in interacting with the system. It is written in understandable language. Hard copy of user documentation is delivered to the organization. The basic points that follow the user to interact with the system are as follow:

- **User:**
All the users login to the hotel management system. And search or filter the desired rooms.
- **Management of the software:**
The management of hotel are admin, managers, receptionist and whosoever we keep in our team.
- **Test the product:**
The administration and customers may check the software product for better understanding and this project may open on web application.

Software Development:

Here we describe the back end front end development. To develop the software, the front end developer GUI simply and the back end developer develops the coding and database of the software. The coding is in PHP language and database developed using My SQL, Apache server.

Appendix B: Source Code

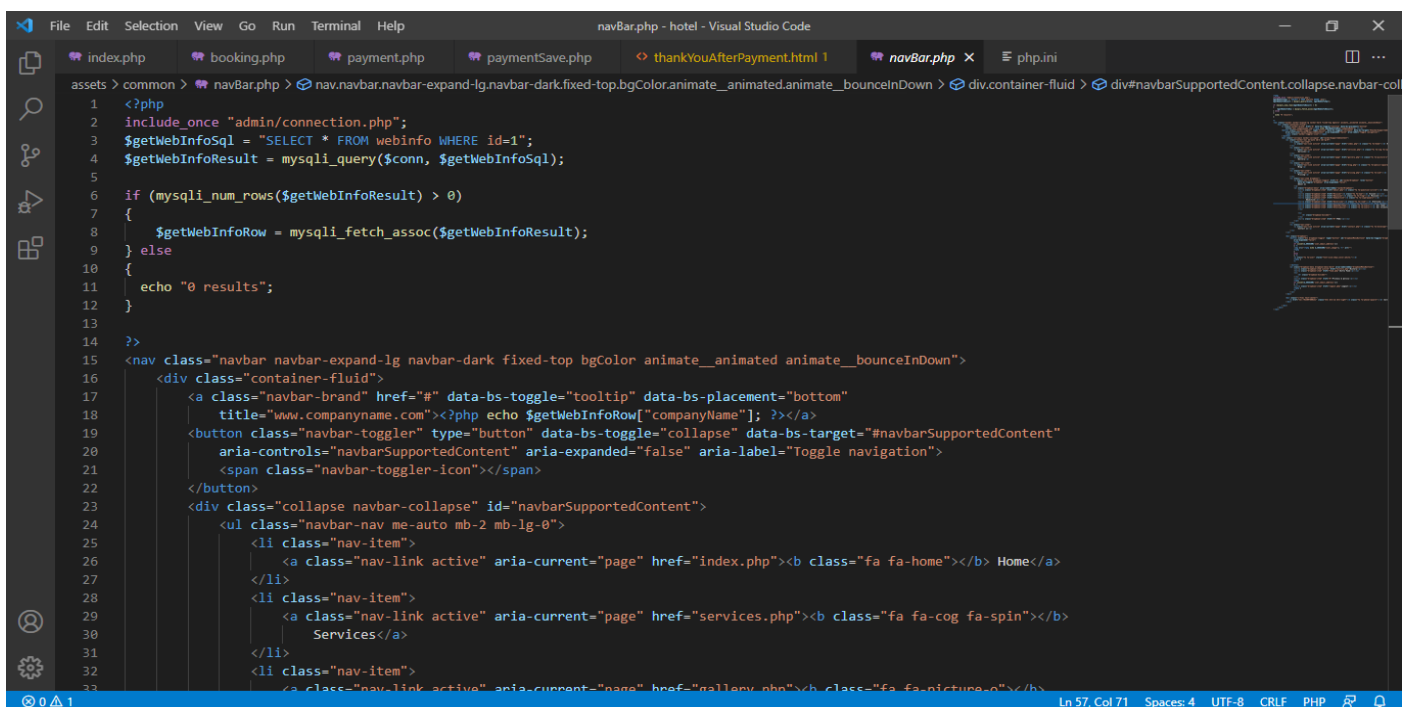


```

index.php - hotel - Visual Studio Code
index.php X booking.php payment.php paymentSave.php thankYouAfterPayment.html 1 php.ini
index.php > html > body
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5     <?php include_once "assets/common/allMeta.php" ?>
6     <?php include_once "assets/common/allCss.php" ?>
7     <link rel="stylesheet" href="assets/lib/css/index.css">
8     <title>.:|Home|.:</title>
9 </head>
10
11 <body>
12     <!-- Loading Effect -->
13     <div id="mmm">
14         <span class="l"><span class="l-i"></span></span>
15         <h1 class="text-white">Please wait</h1>
16         <h6 class="text-white">You browser is setting up to start.</h6>
17     </div>
18     <!-- Loading Effect -->
19
20     <!-- NavBar -->
21     <?php include_once "assets/common/navBar.php"; ?>
22     <?php include_once "assets/common/booking.php"; ?>
23     <!-- NavBar -->
24
25     <!-- Image slider -->
26     <div id="carouselExampleCaptions" class="carousel slide" data-bs-ride="carousel">
27         <div class="carousel-indicators">
28             <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="0" class="active"
29                 aria-current="true" aria-label="Slide 1"></button>
30             <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="1"
31                 aria-label="Slide 2"></button>
32             <button type="button" data-bs-target="#carouselExampleCaptions" data-bs-slide-to="2"
33                 aria-label="Slide 3"></button>

```

Figure 78: Code for index page



```

navBar.php - hotel - Visual Studio Code
index.php X booking.php payment.php paymentSave.php thankYouAfterPayment.html 1 navBar.php X php.ini
assets > common > navBar.php > nav.navbar.navbar-expand-lg.navbar-dark.fixed-top.bgColor.animate__animated.animate_bounceInDown > div.container-fluid > div#navbarSupportedContent.collapse.navbar-col
1 <?php
2 include_once "admin/connection.php";
3 $getWebInfoSql = "SELECT * FROM webinfo WHERE id=1";
4 $getWebInfoResult = mysqli_query($conn, $getWebInfoSql);
5
6 if (mysqli_num_rows($getWebInfoResult) > 0)
7 {
8     $getWebInfoRow = mysqli_fetch_assoc($getWebInfoResult);
9 } else
10 {
11     echo "0 results";
12 }
13
14 ?>
15 <nav class="navbar navbar-expand-lg navbar-dark fixed-top bgColor animate__animated animate_bounceInDown">
16     <div class="container-fluid">
17         <a class="navbar-brand" href="#" data-bs-toggle="tooltip" data-bs-placement="bottom"
18             title="www.companyname.com"><?php echo $getWebInfoRow["companyName"]; ?></a>
19         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
20             aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
21             <span class="navbar-toggler-icon"></span>
22         </button>
23         <div class="collapse navbar-collapse" id="navbarSupportedContent">
24             <ul class="navbar-nav me-auto mb-2 mb-lg-0">
25                 <li class="nav-item">
26                     <a class="nav-link active" aria-current="page" href="index.php"><b class="fa fa-home"></b> Home</a>
27                 </li>
28                 <li class="nav-item">
29                     <a class="nav-link active" aria-current="page" href="services.php"><b class="fa fa-cog fa-spin"></b>
30                     Services</a>
31                 </li>
32                 <li class="nav-item">
33                     <a class="nav-link active" aria-current="page" href="gallery.php"><b class="fa fa-picture-o"></b>

```

Figure 79: Code for Navbar

```

1 <!doctype html>
2 <html lang="en">
3
4 <head>
5 <?php include_once "assets/common/allMeta.php" ?>
6 <?php include_once "assets/common/allCss.php" ?>
7 <link rel="stylesheet" href="assets/lib/css/service.css">
8
9 <title>.:|Services|.:</title>
10 <style>
11
12 </style>
13 </head>
14
15 <body>
16 <!-- Loading Effect -->
17 <div id="mmm">
18 <span class="l"><span class="l-i"></span></span>
19 <h1 class="text-white">Please wait</h1>
20 <h6 class="text-white">You browser is setting up to start.</h6>
21 </div>
22 <!-- Loading Effect -->
23
24 <!-- NavBar -->
25 <?php include_once "assets/common/navBar.php"; ?>
26 <!-- NavBar -->
27 <div class="container-fluid reasonToChoose mb-5">
28 <div class="row align-items-center">
29 <div class="col-md-6 text-center">
30 
31 </div>
32 <div class="col-md-6">
33 <h5 class="animadHeading">Why customers love our services</h5>

```

Figure 80: Code for Services

```

1 <!doctype html>
2 <html lang="en">
3
4 <head>
5 <?php include_once "assets/common/allMeta.php" ?>
6 <?php include_once "assets/common/allCss.php" ?>
7 <link rel="stylesheet" href="assets/lib/css/gallery.css">
8 <title>.:|Pricing plans|.:</title>
9 </head>
10
11 <body>
12 <!-- Loading Effect -->
13 <div id="mmm">
14 <span class="l"><span class="l-i"></span></span>
15 <h1 class="text-white">Please wait</h1>
16 <h6 class="text-white">You browser is setting up to start.</h6>
17 </div>
18 <!-- Loading Effect -->
19
20 <!-- NavBar -->
21 <?php include_once "assets/common/navBar.php"; ?>
22 <!-- NavBar -->
23 <div class="container-fluid" style="margin-top: 150px;">
24 <div class="row align-items-center align-self-center justify-content-around d-flex">
25 <div class="col-md-4 shadow rounded text-center" style="height:50vh;background:url(assets/images/roomPlan.jpg);
26 background-size: cover;background-repeat: no-repeat;">
27 </div>
28 <div class="col-md-4 shadow rounded text-center" style="height:50vh;background:url(assets/images/mealsPlan.jpg);
29 background-size: cover;background-repeat: no-repeat;">
30 </div>
31 </div>
32 <div class="row">
33 <div class="col-md-6 text-center">

```

Figure 81: Code for Pricing

```

1 <!-- Button trigger modal -->
2
3
4 <!-- Modal -->
5 <div class="modal fade" id="policy" tabindex="-1" aria-labelledby="exampleModallabel" aria-hidden="true">
6   <div class="modal-dialog">
7     <div class="modal-content">
8       <div class="modal-header">
9         <h4 class="modal-title" id="exampleModallabel">Terms & policy</h4>
10        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
11      </div>
12      <div class="modal-body">
13        <h5>1. Tariff</h5>
14        <p>The tariff is for the room only and is exclusive of any government taxes applicable Meals and other services
15        available at any cost.
16      </p>
17
18        <h5>2. Settlement of bills</h5>
19        <p>Bills must be settled on presentation by Jazzcash/Easypaisa or by hand, personal cheques not accepted.</p>
20
21        <h5>3. Check-in</h5>
22        <p>Please present your CNIC card or passport upon Check-in.</p>
23
24        <h5>4. Departure</h5>
25        <p>Check out time is ( mention your checkout time here ) please inform the reception if you wish to retain
26        your room beyond this time. The extension will be given depending on the availability. If the room is
27        available, the normal tariff will be charged. On failure of the guest to vacate the room on expiry or
28        period the management shall have the right to remove the guest and his/her belongings from the room
29        occupied by the Guest.</p>
30
31        <h5>5. Guest's belongings</h5>
32        <p>Guests are particularly requested to lock the door of their rooms when going out or going to sleep. For the
33        convenience of the guests, electronic safety lockers are provided in the room to store any valuables.</p>

```

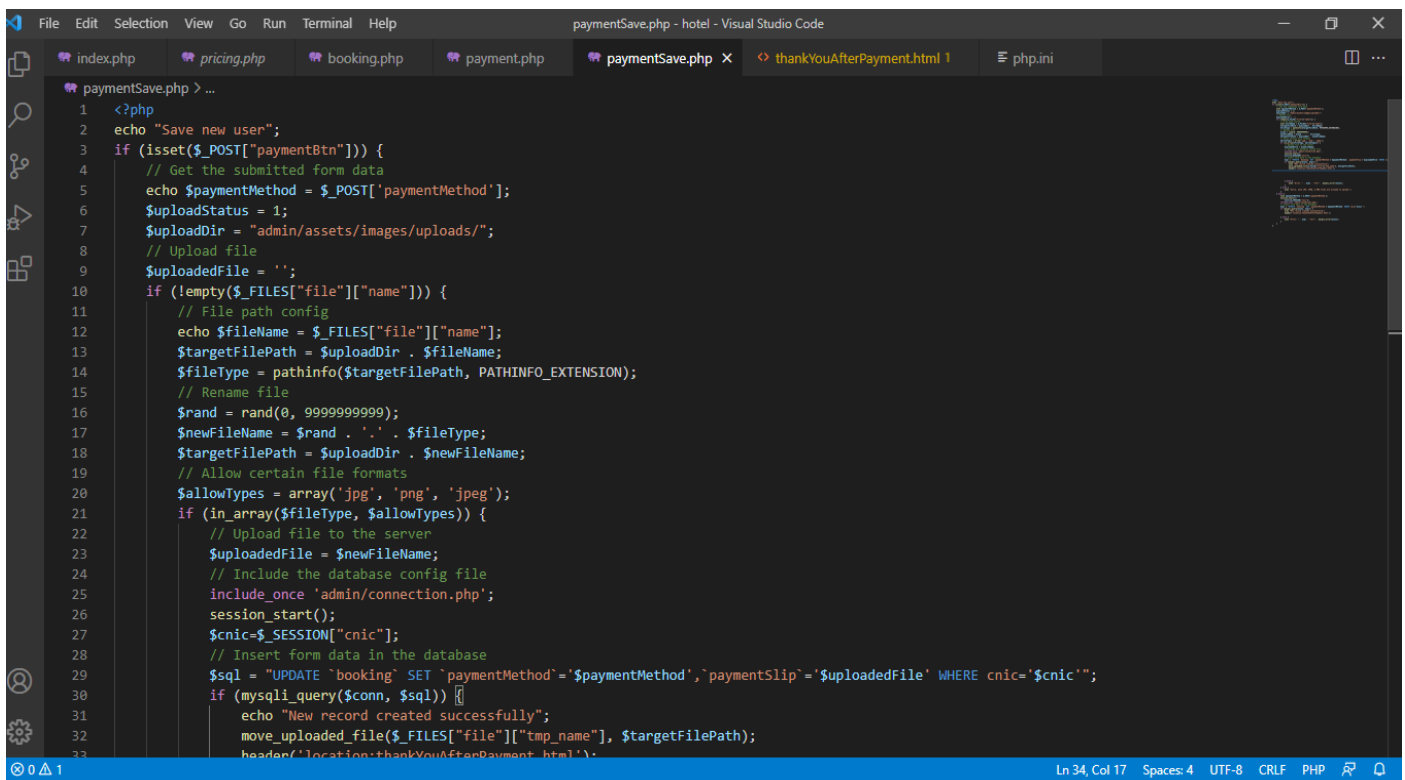
Figure 82: Code for Booking

```

1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <!-- Required meta tags -->
6   <meta charset="utf-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8
9   <!-- Bootstrap CSS -->
10  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
11    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztCQTWfSpd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
12
13  <title>Payment</title>
14 </head>
15
16 <body>
17   <div class="container">
18     <div class="row align-items-center align-self-center justify-content-center" style="height: 100vh;">
19       <div class="col-md-6">
20         <?php session_start(); ?>
21         
22         <?php echo $_SESSION['user_first_name']. $_SESSION['user_last_name']; ?>
23         <li class="list-group-item"><?php echo $_SESSION['user_email_address']; ?></li>
24         <table class="table">
25           <thead>
26             <tr>
27               <th scope="col">#</th>
28               <th scope="col">Description:</th>
29             </tr>
30           </thead>
31           <tbody>
32             <tr>
33               <th scope="row">Checkin Date:</th>

```

Figure 83: Code for Payment

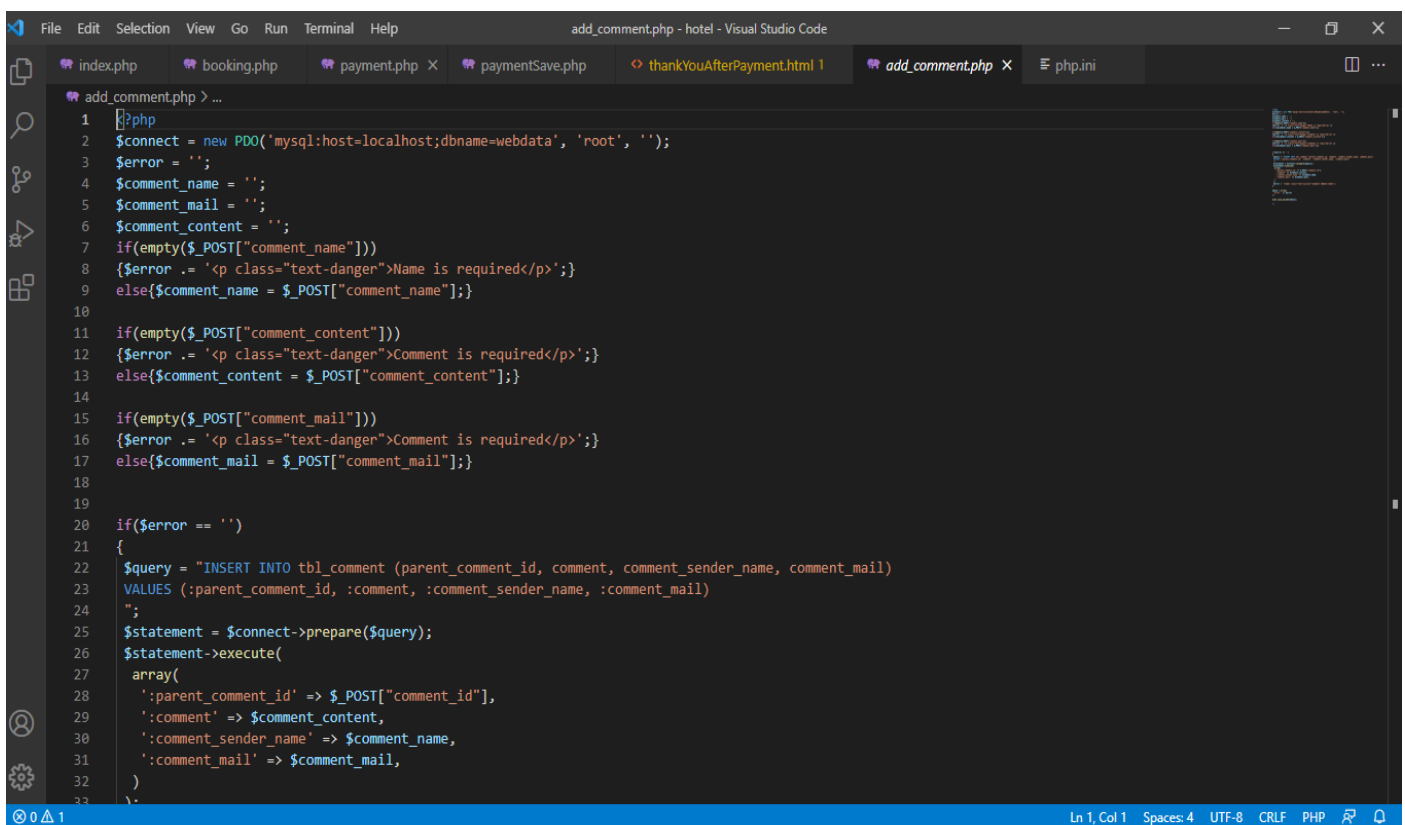


```

1 <?php
2 echo "Save new user";
3 if (isset($_POST["paymentBtn"])) {
4     // Get the submitted form data
5     echo $paymentMethod = $_POST["paymentMethod"];
6     $uploadStatus = 1;
7     $uploadDir = "admin/assets/images/uploads/";
8     // Upload file
9     $uploadedFile = '';
10    if (!empty($_FILES["file"]["name"])) {
11        // File path config
12        echo $fileName = $_FILES["file"]["name"];
13        $targetFilePath = $uploadDir . $fileName;
14        $fileType = pathinfo($targetFilePath, PATHINFO_EXTENSION);
15        // Rename file
16        $rand = rand(0, 9999999999);
17        $newFileName = $rand . '.' . $fileType;
18        $targetFilePath = $uploadDir . $newFileName;
19        // Allow certain file formats
20        $allowTypes = array('jpg', 'png', 'jpeg');
21        if (in_array($fileType, $allowTypes)) {
22            // Upload file to the server
23            $uploadedFile = $newFileName;
24            // Include the database config file
25            include_once 'admin/connection.php';
26            session_start();
27            $cnic=$_SESSION["cnic"];
28            // Insert form data in the database
29            $sql = "UPDATE `booking` SET `paymentMethod`='$paymentMethod',`paymentSlip`='$uploadedFile' WHERE cnic='$cnic'";
30            if (mysqli_query($conn, $sql)) {
31                echo "New record created successfully";
32                move_uploaded_file($_FILES["file"]["tmp_name"], $targetFilePath);
33                header("location:thankYouAfterPayment.html");

```

Figure 84: Code for Payment Save



```

1 <?php
2 $connect = new PDO('mysql:host=localhost;dbname=webdata', 'root', '');
3 $error = '';
4 $comment_name = '';
5 $comment_mail = '';
6 $comment_content = '';
7 if(empty($_POST["comment_name"]))
8 { $error .= '<p class="text-danger">Name is required</p>'; }
9 else{ $comment_name = $_POST["comment_name"]; }
10
11 if(empty($_POST["comment_content"]))
12 { $error .= '<p class="text-danger">Comment is required</p>'; }
13 else{ $comment_content = $_POST["comment_content"]; }
14
15 if(empty($_POST["comment_mail"]))
16 { $error .= '<p class="text-danger">Comment is required</p>'; }
17 else{ $comment_mail = $_POST["comment_mail"]; }
18
19
20 if($error == '')
21 {
22     $query = "INSERT INTO tbl_comment (parent_comment_id, comment, comment_sender_name, comment_mail)
23     VALUES (:parent_comment_id, :comment, :comment_sender_name, :comment_mail)
24     ";
25     $statement = $connect->prepare($query);
26     $statement->execute(
27         array(
28             ':parent_comment_id' => $_POST["comment_id"],
29             ':comment' => $comment_content,
30             ':comment_sender_name' => $comment_name,
31             ':comment_mail' => $comment_mail,
32         )
33     );

```

Figure 85: Code for Add Comment