Final Report

Hotel Management System

Instructor: Shaheer Ahmad Khan

Submission Date: 08-05-25

Group Members:

Sania Ushaa	**22k-4382**
Shermeen Ziauddin	**22k-4206**
Eesha Fatima	**22k-4226**

Project Code Github Link:

https://github.com/saniaUshaa/HotelManagementSystemInPHP

1. Introduction

1.1. Purpose of Document

The purpose of this document is to outline the requirements, architecture, and design for a **Hotel Management System** that automates hotel operations like bookings, billing, guest management, and admin management.

1.2. Intended Audience

- Developer: To build and maintain the system
- Project Managers: To oversee the development and ensure the project meets requirements.
- Hotel Staff/Admins: To understand system capabilities for managing hotel operations.
- End-user/Guests: To learn how the system facilitates room bookings and other services.

2. Overall System Description

2.1. Project Background

Hotels face challenges like manual bookings, billing inaccuracies, and lack of real-time updates on room availability. This system addresses these inefficiencies by offering an automated solution to enhance operational productivity and guest experience.

2.2. Project Scope

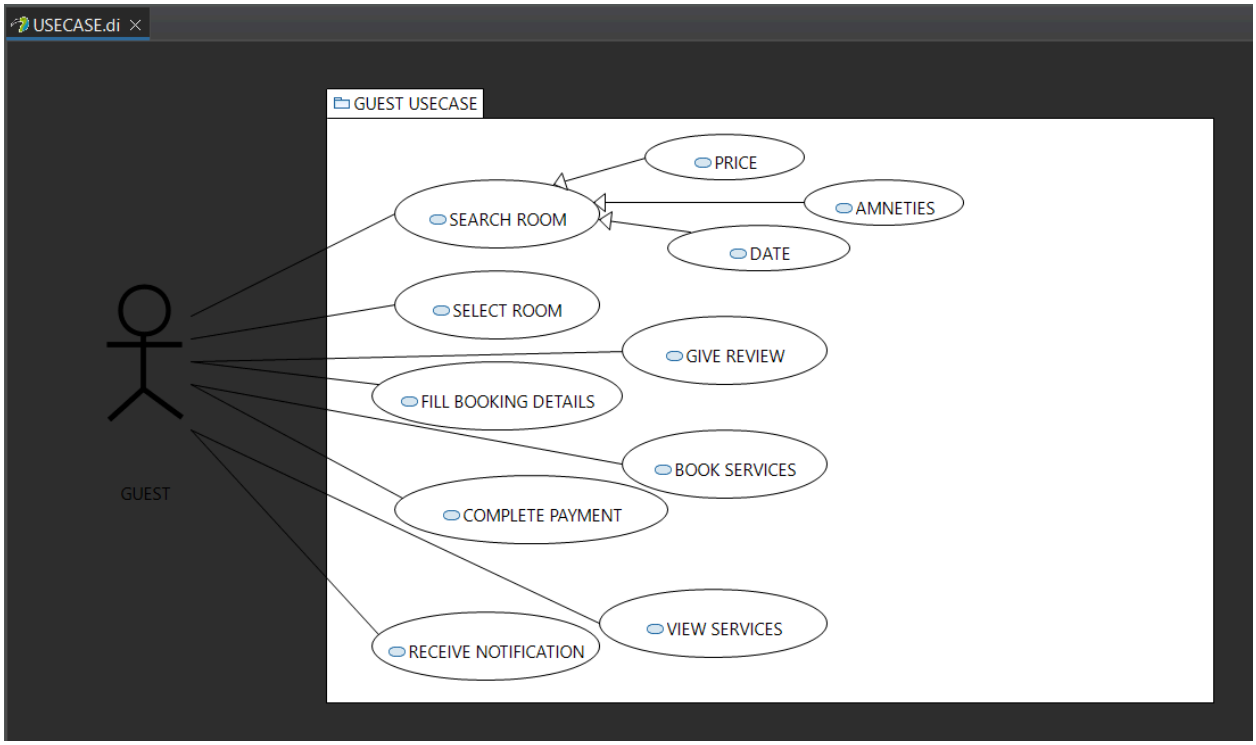
The system will:

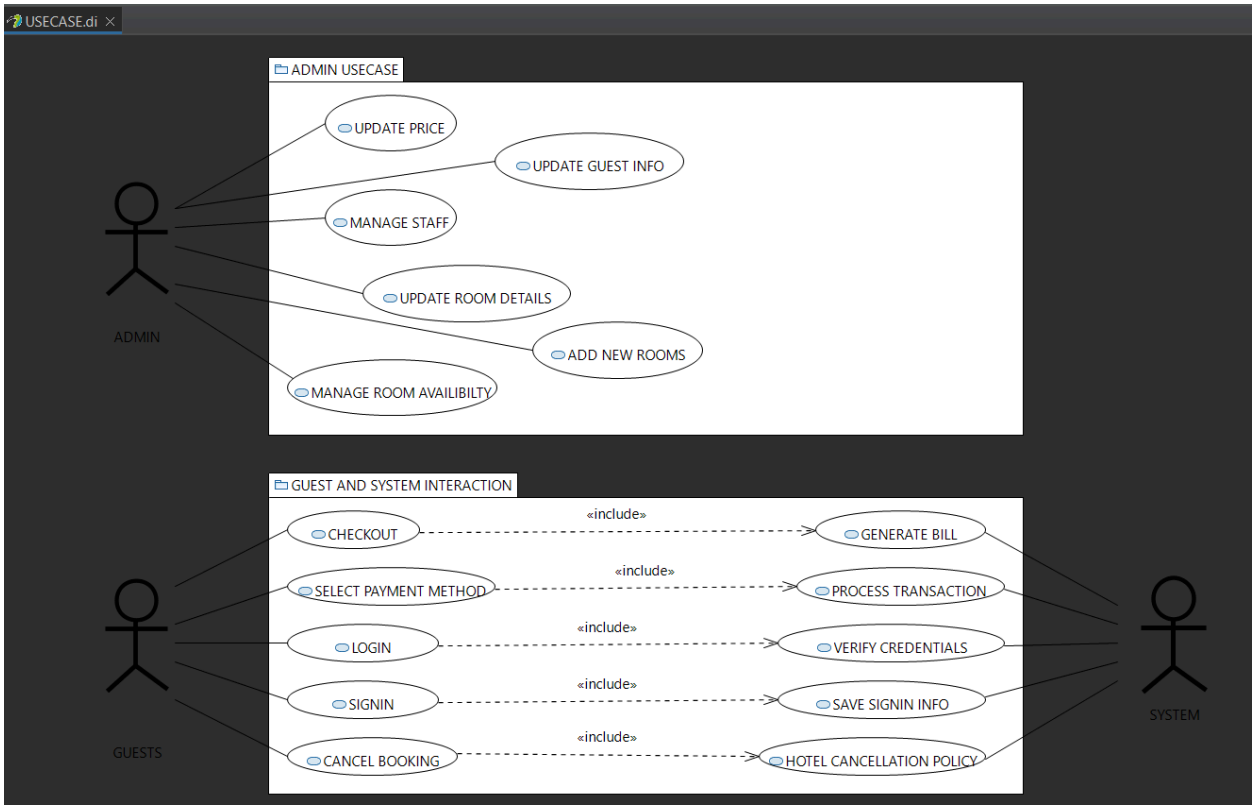
- Enable online room booking with real-time availability.
- Manage billing, including discounts and loyalty programs.
- Handle staff coordination and housekeeping tasks.
- Maintain guest reviews and feedback

3. System Requirement Specification - SRS

3.1. Functional Requirements:

3.1.1. Use Cases





Use Case Description	
Use Case name:	**Hotel Management System**
Use Case Description: The Hotel Management System automates various hotel operations such as room bookings, guest management, billing, and staff coordination. It allows hotel admin to manage rooms and reservations,guests to make online bookings and payments.gateways The system also provides reviews, and notifications to enhance the guest experience.	
Primary actor: Hotel Guest	**Other actors:**Hotel Admin, Hotel Staff, Payment Gateway **Third Party Actors:**Government(tax from hotel)
Stakeholders:	**Hotel Guests:** Uses the system for booking rooms, making payments, giving reviews, and managing their stay. **Hotel Admins:** Oversee room management, reservations, staff coordination, and billing. **Hotel Staff:** Manage guests information.
Relationships ▪ **Includes:** Admin and system,Guest and system.	
Pre-conditions: ● The staff, or admin must be registered and authenticated in the system. ● Room availability data is updated in real-time. ● User must be a registered citizen.	
Flow of Events: **1) The staff, or admin logs into the system using their credentials and the guest sign in by providing their information to the system.** **2) The system verifies their role and grants access accordingly.** **3) The guest searches for available rooms using filters like price, dates, and amenities.** **4) The rooms that are Available are displayed on screen** **5) The guest then selects one of the rooms according to their preference to book.** **6) The guests fill in necessary details like check-in date and time along with check-out details as well.** **7) The guest then selects a payment method, and the system processes the transaction.** **8) The system can apply discounts if any offers are available for customers.** **9) Upon successful payment, the system confirms the booking and sends a notification.** **10) Admins or staff manage room availability, add new rooms, update pricing and other room details.** **11) Staff is assigned tasks to perform before the guest arrives such as checking the room condition.**	

12) After the guest checks in, the staff assigns the room to guests and update the guests stay information in the system. **13) At check-out, a final bill is generated for the guest to review and complete the payment.** **14) After the stay, guests can leave reviews and ratings.** **15) The system adds loyalty points for frequent guests, which can be redeemed for future bookings.** **16) Hotel staff coordinates housekeeping and other services, keeping room statuses updated in the system.**
Alternative and exceptional flows: **1.1) If the guest account is not already registered then the guest will register a new account.** **3.1) If no rooms are available for the selected dates, the system notifies the guest and suggests alternate dates or room types.** **9.1) If the payment fails, the system prompts the guest to retry with a different method or correct payment details.**
Post-conditions: ● The guest's booking is confirmed, and the reservation data is saved in the system. ● The guest receives a confirmation notification, and the room availability is updated accordingly.

3.2. Non-Functional Requirements:

3.2.1. Performance Requirements

- The system must respond to user inputs within 2 seconds under normal operating conditions.
- It should support at least 50 concurrent users without noticeable performance degradation.
- Room availability searches and booking confirmations must be processed in under 3 seconds.

3.2.2. Safety Requirements

- The system must validate all user inputs to prevent errors that could corrupt booking, billing, or guest records.
- Data backups must be taken daily to prevent data loss in case of system failure.

3.2.3. Security Requirements

- User authentication must be enforced using secure login credentials (username and password).
- Access to admin and staff features should be role-based and restricted accordingly.
- The system must log all critical actions (e.g., booking, cancellations, payments) for audit and traceability.

3.2.4. User Documentation

A **User Manual** will be provided with step-by-step instructions for:

- Guest registration, login, and password recovery.
- Room search, filtering, and booking process.
- Payment procedures and troubleshooting failed transactions.
- Leaving reviews, checking loyalty points, and canceling bookings.

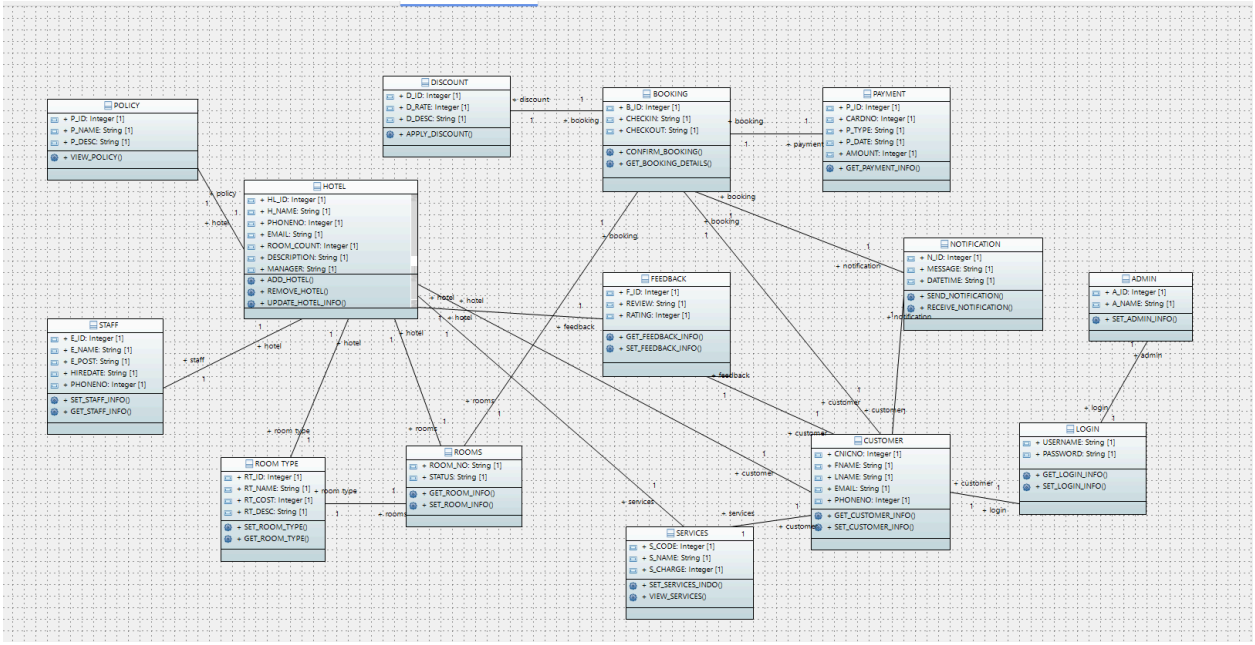
Admin Documentation will cover:

- Room inventory management.
- User account and booking oversight.
- Staff task assignment and updates.
- Generating and reviewing reports.

4. System Design Specification

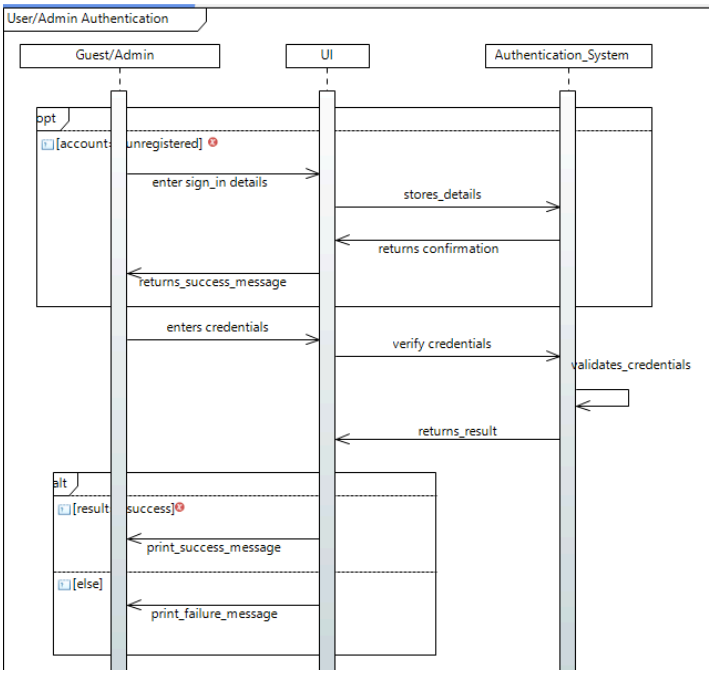
4.1. System Architecture

4.1.1. Class Diagram

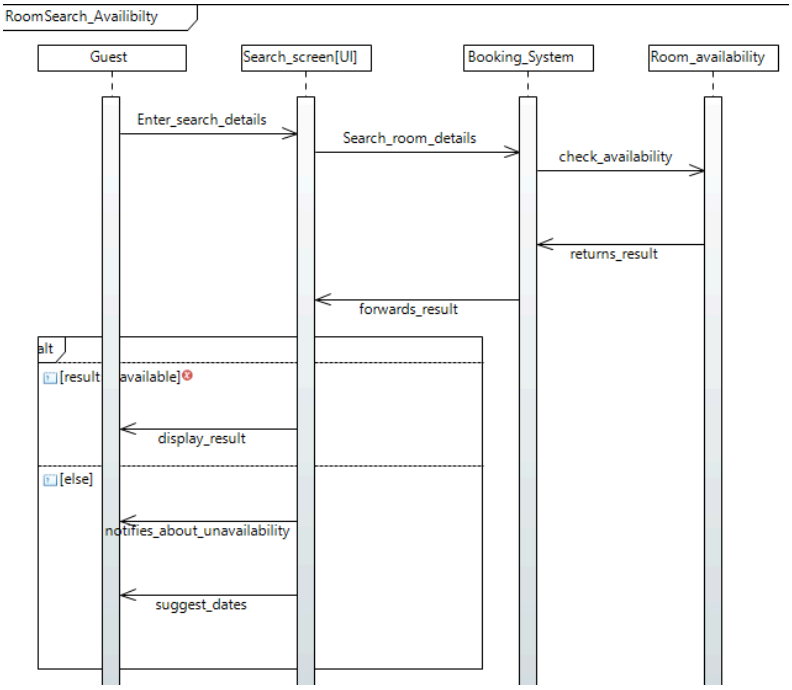


4.1.2. Sequence Diagram

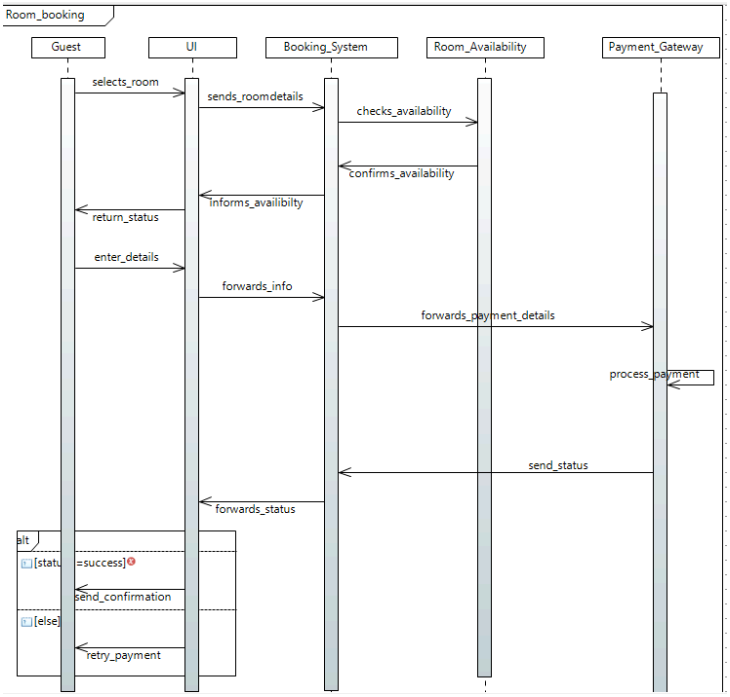
- The user sends login credentials to the LoginSignup UI component.LoginSignup forwards these credentials to AuthenticationSystem to validate them.AuthenticationSystem checks and returns the validation result, and a success message is returned upon success



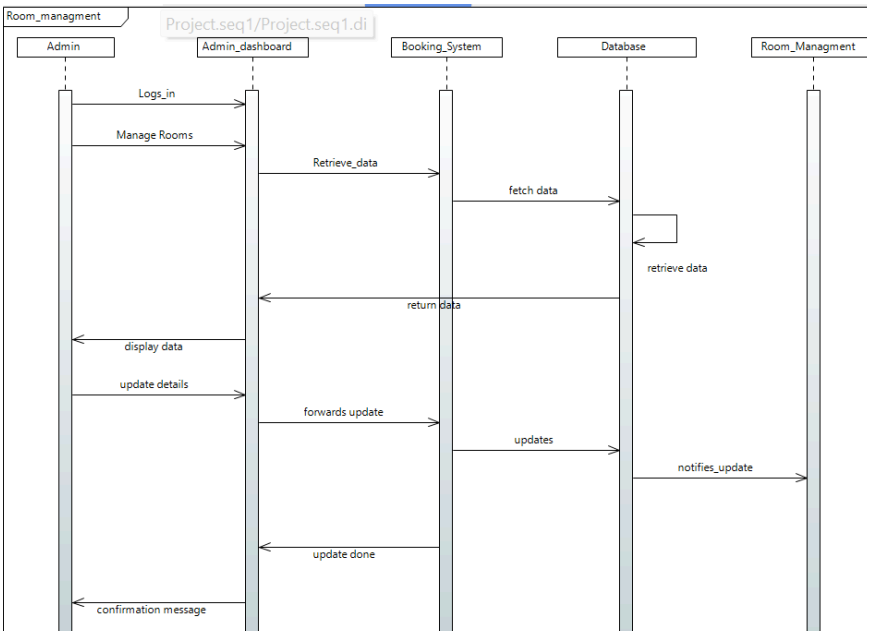
- The guest enters search details and the system checks room availability through UI and returns status to the guest.



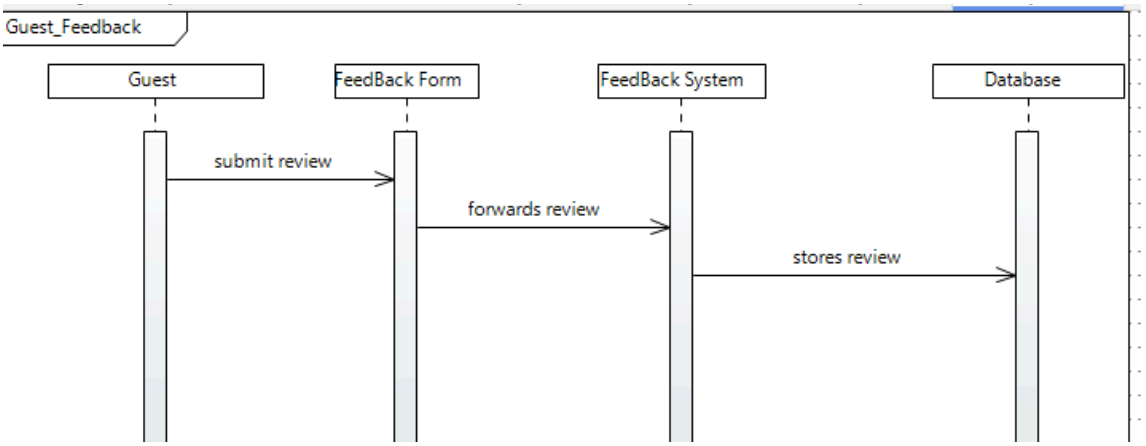
- The guest initiates a room booking in BookingSystem through UI.BookingSystem Checks room availability and confirms it to guests and proceeds to payment ,if a room is unavailable it tells the user through status. After successful booking, it confirms booking to guest everelse retry payment message.



- The admin logs in and retrieves data related to rooms and updates room availability etc .

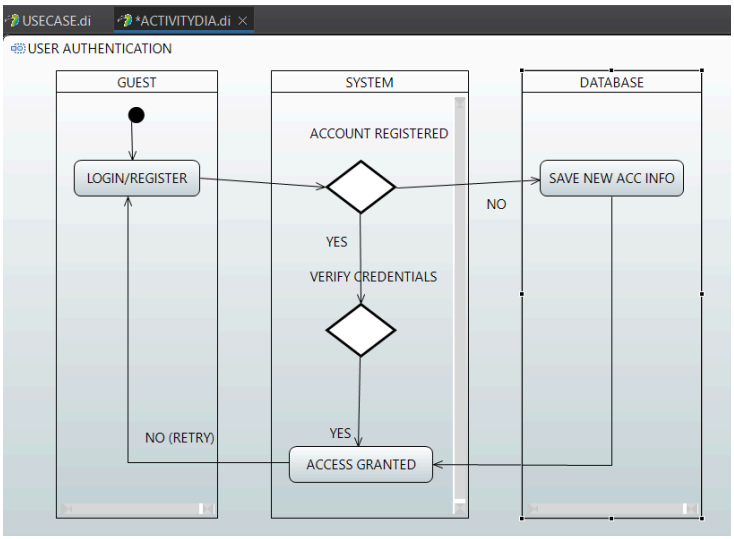


- Guests can give review and feedback of their stay and the system stores it in a database.

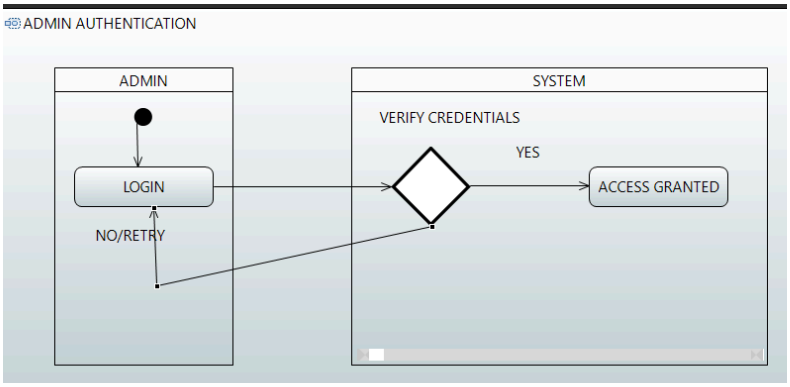


4.1.3. Activity Diagram

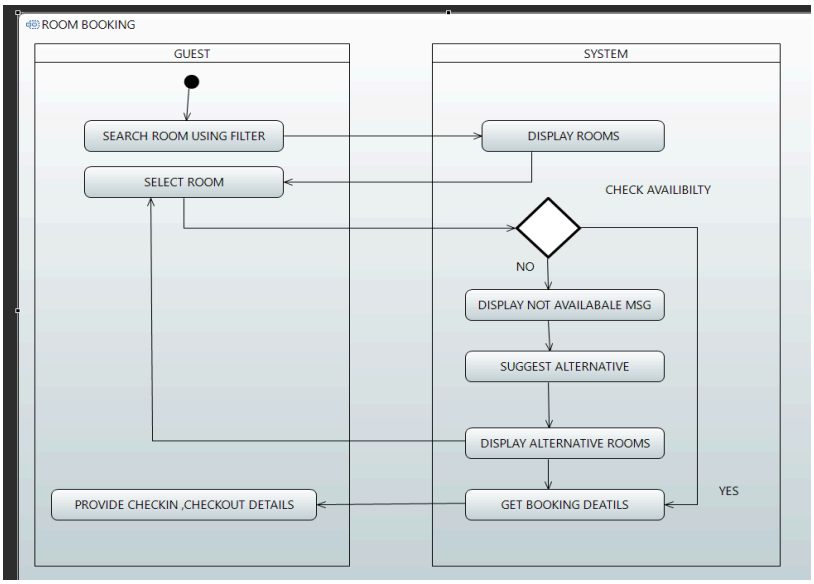
- This diagram illustrates the steps a user takes to log into the system, starting from entering credentials to gaining access based on role verification.



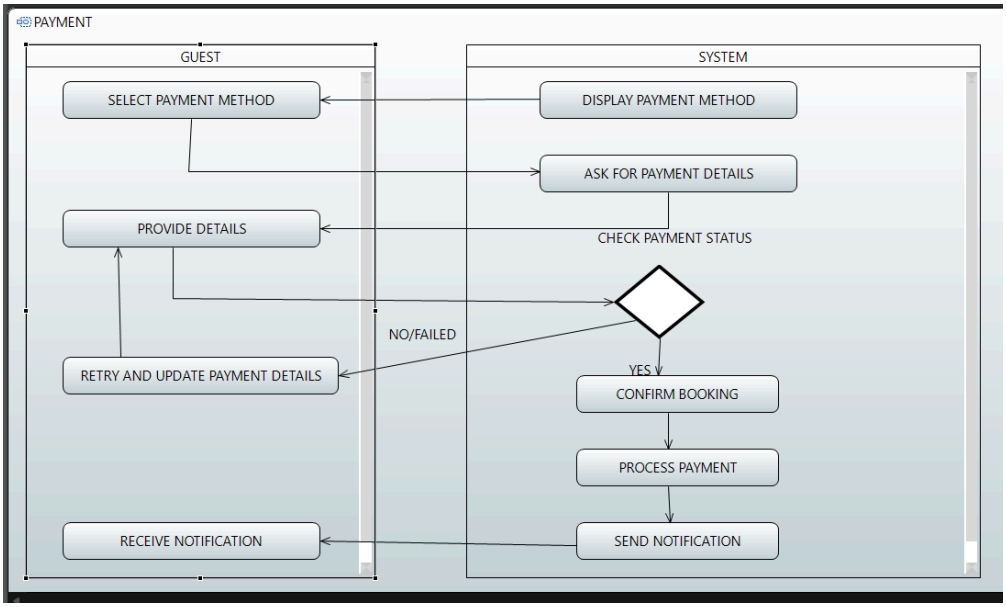
- This diagram illustrates the steps an admin takes to log into the system, starting from verifying credentials to gaining access.



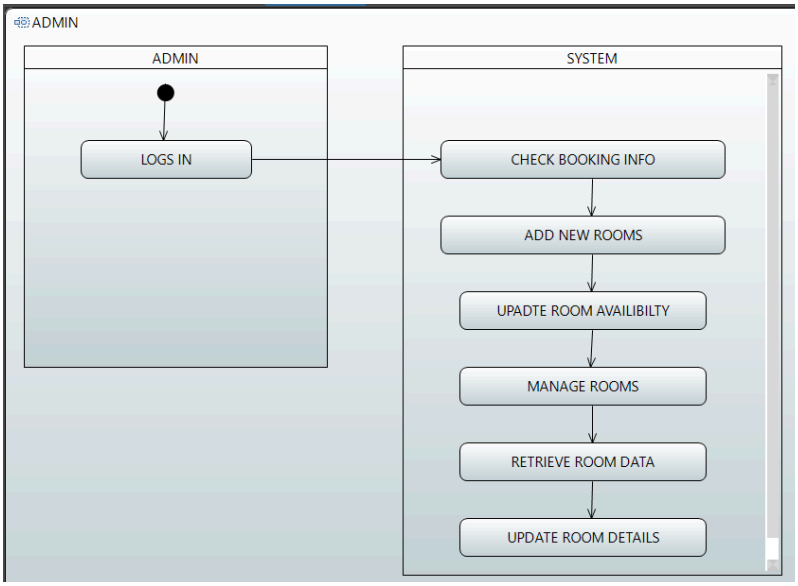
- This diagram shows the process of searching for available rooms by using filters and confirming the booking and payment.



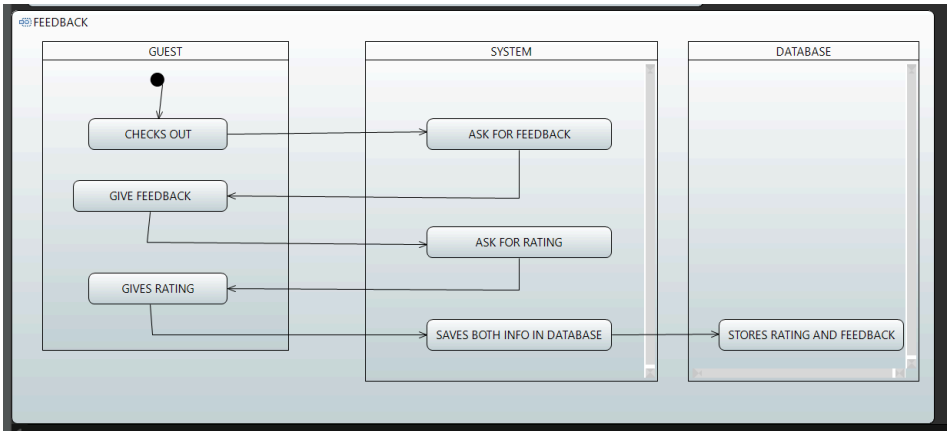
- Illustrates the steps for processing payments, including selecting a payment method, verifying details, and handling successful or failed transactions.



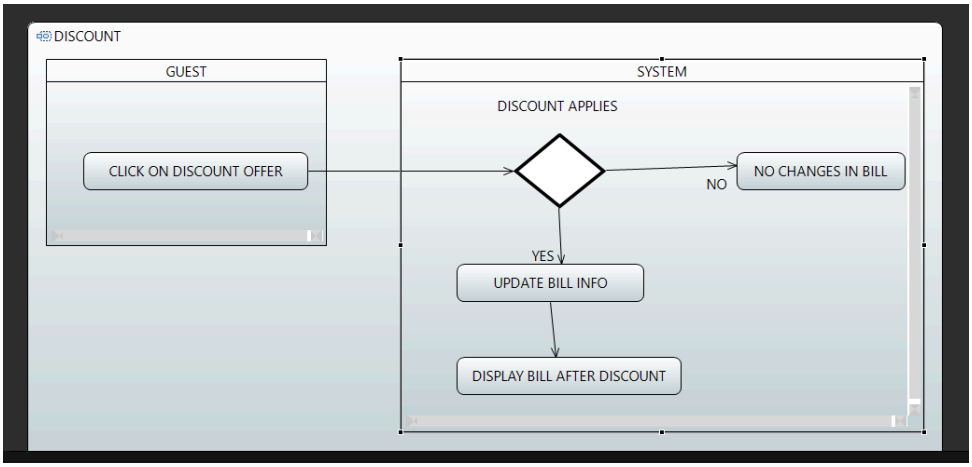
- Represents the process where the admin manages tasks like updating room availability, monitoring bookings, handling guest feedback, and coordinating with staff.



- This diagram helps visualize how feedback is collected, processed, and handled within the system, enhancing user satisfaction and improving services.



- Describes the actions taken when a guest clicks on any available discount offers available and the system applying the discount amount on the bill.



RMMM (Risk Mitigation, Monitoring, and Management) Table:

Risk ID	Risk Description	Probability	Impact	Mitigation Strategy	Monitoring/Management Plan
R1	Server downtime during peak booking season	High	High	Implement load balancing and failover systems	Use uptime monitoring tools e.g alert IT team on downtime
R2	Data breach of customer personal and payment info	Low	High	Use strong encryption, firewalls, and regular security audits	Conduct periodic vulnerability assessments and penetration testing
R3	Double bookings due to race conditions in software	Medium	High	Use atomic transactions and database locking mechanisms	Log booking conflicts and review regularly
R4	Unavailability of payment gateway	Medium	High	Integrate multiple payment gateways as fallback	Monitor gateway uptime via APIs, switch dynamically if failure detected
R5	Inaccurate billing due to calculation bugs	Low	Medium	Write unit tests for billing modules and validate them with dummy data	Monthly manual checks and logging of anomalies
R6	Lack of staff training on new system	Medium	Medium	Conduct regular training sessions and provide documentation	Monitor support ticket frequency related to staff errors

R7	Failure in backup/restore mechanism	Low	High	Automate daily backups with redundancy in cloud and local storage	Perform routine restore tests monthly
R8	Regulatory compliance failure (GDPR, local laws)	Low	High	Hire compliance consultant, keep policies updated	Schedule quarterly compliance reviews
R9	Negative customer reviews due to buggy UI/UX	Medium	Medium	Perform usability testing, collect user feedback early	Track ratings and reviews, prioritize fixes in sprints
R10	Network issues affecting remote access/control	Medium	Medium	Use network monitoring tools, provide mobile data based fallback	IT to investigate all disconnections and maintain incident logs

TEST CASES:

TestID	TestData	ExpectedResult	Status
TC01	1. Open http://localhost/DBProject/home.php 2. Set window size to 1295x687 3. Click on linkText=SignUp 4. Click on id=firstname 5. Type "Ees" in id=firstname 6. SendKeys ${KEY_DOWN} on id=firstname 7. Type "Eesha" in id=firstname 8. Click on id=lastname 9. Click on id=password 10. Click on .col-sm-7 > .form-control 11. Type "Khi" 12. Click on .col-sm:nth-child(2) > .form-control 13. Type "Sindh" 14. Click on .col-sm:nth-child(3) > .form-control 15. Type "75300" 16. Click on .form-check:nth-child(2) 17. Click on id=radio2 18. Click on .btn 19. Click on id=firstname 20. Type "Sania" in id=firstname 21. Click on id=lastname 22. Type "Ushaa" 23. Click on id=password 24. Click on .col-sm-7 > .form-control 25. Type "lahore" 26. Click on .col-sm:nth-child(2) > .form-control 27. Type "punjab" 28. Click on .col-sm:nth-child(3) > .form-control 29. Type "75300" 30. Click on .btn	The full signup flow should complete with valid entries for user and admin.	OK

TestID	TestData	ExpectedResult	Status
TC02	1.open on http://localhost/DBProject/login.php OK 2.setWindowSize on 1266x672 OK 3.click on linkText=SignUp OK 4.click on id=firstname OK 5.type on id=firstname with value Eesha OK 6.click on id=lastname OK 7.click on id=password OK 8.click on css=.col-sm-7 OK 9.click on css=.col-sm-7 > .form-control OK 10.type on css=.col-sm-7 > .form-control with value Khi OK 11.click on css=.col-sm\:nth-child(2) > .form-control OK 12.type on css=.col-sm\:nth-child(2) > .form-control with value Sindh OK 13.click on css=.col-sm\:nth-child(3) > .form-control OK 14.type on css=.col-sm\:nth-child(3) > .form-control with value 75300 OK 15.click on id=radio2 OK 16.click on css=.btn OK 17.click on css=strong\:nth-child(2) OK 18.click on linkText=Login OK 19.click on id=username OK 20.type on id=username with value Ees OK 21.sendKeys on id=username with value ${KEY_DOWN} OK 22.sendKeys on id=username with value ${KEY_DOWN} OK 23.sendKeys on id=username with value ${KEY_DOWN} OK 24.type on id=username with value EeshaFatima_3 OK 25.click on id=password OK 26.click on css=.btn OK 27.click on id=destination OK 28.type on id=destination with value K OK 29.sendKeys on id=destination with value ${KEY_DOWN} OK 30.type on id=destination with value Karachi OK 31.click on id=checkInDate OK	The complete user interface will be displayed along with the successful hotel booking and payment process.	OK

TestID	TestData	ExpectedResult	Status
	32.click on id=checkInDate OK 33.type on id=checkInDate with value 2025-05-09 OK 34.click on id=checkOutDate OK 35.click on id=checkOutDate OK 36.type on id=checkOutDate with value 2025-05-16 OK 37.mouseDownAt on id=room_type with value -480.5333251953125,-346.8999938964844 OK 38.mouseMoveAt on id=room_type with value -480.5333251953125,-346.8999938964844 OK 39.mouseUpAt on id=room_type with value -480.5333251953125,-346.8999938964844 OK 40.click on id=room_type OK 41.select on id=room_type with value label=Double Room OK 42.click on css=option\:nth-child(3) OK 43.click on id=price OK 44.type on id=price with value 750000 OK 45.click on css=button OK 46.click on linkText=View Details OK 47.click on linkText=Book Now OK 48.click on id=cnic OK 49.type on id=cnic with value 12 OK 50.sendKeys on id=cnic with value ${KEY_DOWN} OK 51.type on id=cnic with value 42201-1234567-6 OK 52.click on id=fname OK 53.type on id=fname with value Eesha OK 54.click on id=lname OK 55.type on id=lname with value Fatima OK 56.click on id=phone OK 57.type on id=phone with value 12 OK 58.sendKeys on id=phone with value ${KEY_DOWN} OK 59.type on id=phone with value 1234-5678932 OK 60.click on id=email OK 61.type on id=email with value fat OK 62.sendKeys on id=email with value ${KEY_DOWN} OK 63.type on id=email with value fatimaeesha@gmail.com OK 64.sendKeys on id=email with value ${KEY_ENTER} OK		

TestID	TestData	ExpectedResult	Status
	65.click on id=checkin OK 66.type on id=checkin with value 2025-05-09 OK 67.click on id=checkout OK 68.type on id=checkout with value 2025-05-16 OK 69.click on id=discount OK 70.click on css=.btn-pay OK 71.click on id=cardNumber OK 72.type on id=cardNumber with value 1 OK 73.sendKeys on id=cardNumber with value ${KEY_DOWN} OK 74.sendKeys on id=cardNumber with value ${KEY_DOWN} OK 75.sendKeys on id=cardNumber with value ${KEY_UP} OK 76.type on id=cardNumber with value 1234 7895 1234 67584 OK 77.sendKeys on id=cardNumber with value ${KEY_ENTER} OK 78.type on id=cardNumber with value 1234 7895 1234 6758 OK 79.sendKeys on id=cardNumber with value ${KEY_ENTER} OK 80.type on id=cardName with value Eesha OK 81.sendKeys on id=cardName with value ${KEY_DOWN} OK 82.type on id=cardName with value Eesha Fatima OK 83.click on id=expiryDate OK 84.type on id=expiryDate with value 11/29 OK 85.click on id=cvv OK 86.type on id=cvv with value 123 OK 87.click on css=.btn OK 88.click on linkText=My Account OK 89.mouseOver on css=.float-end OK 90.click on css=.float-end OK 91.mouseOut on css=.float-end OK 92.select on id=hotelId with value label=Avari Towers Karachi OK 93.click on css=option\:nth-child(3) OK		

TestID	TestData	ExpectedResult	Status
	94.click on id=review OK 95.type on id=review with value Nice Hotel OK 96.click on id=rating OK 97.type on id=rating with value 5 OK 98.click on css=.btn\:nth-child(2) OK 99.click on css=.float-end OK 100.click on css=.btn-close\:nth-child(2) OK 'testuserlogin_1' completed successfully OK		
TC03	1.open on http://localhost/DBProject/login.php OK 2.setWindowSize on 1266x672 OK 3.click on linkText=SignUp OK 4.click on id=firstname OK 5.type on id=firstname with value S OK 6.sendKeys on id=firstname with value ${KEY_DOWN} OK 7.type on id=firstname with value Sania OK 8.click on id=lastname OK 9.type on id=lastname with value Ushaa OK 10.click on id=password OK 11.click on css=.col-sm-7 > .form-control OK 12.type on css=.col-sm-7 > .form-control with value Lahore OK 13.click on css=.col-sm:nth-child(2) > .form-control OK 14.type on css=.col-sm:nth-child(2) > .form-control with value Punjab OK 15.click on css=.col-sm:nth-child(3) > .form-control OK 16.type on css=.col-sm:nth-child(3) > .form-control with value 55000 OK 17.click on css=.btn OK 18.click on linkText=Login OK 19.click on id=username OK 20.type on id=username with value S OK 21.sendKeys on id=username with value ${KEY_DOWN} OK 22.sendKeys on id=username with value ${KEY_DOWN} OK	The expected output includes successful sign-up and login, allowing admins to manage hotel, employee, and guest details. Actions like adding, updating, and displaying information should work correctly, with proper redirects and success messages. The system should be stable and responsive without errors.	OK

TestID	TestData	ExpectedResult	Status
	23.sendKeys on id=username with value ${KEY_DOWN} OK 24.type on id=username with value SaniaUshaa_7 OK 25.click on id=password OK 26.click on css=.btn OK 27.click on css=.toggle-btn OK 28.click on linkText=Manage Hotels OK 29.mouseOver on css=.container > .btn OK 30.click on css=.container > .btn OK 31.mouseOut on css=.container > .btn OK 32.click on css=.btn-success OK 33.click on id=name OK 34.type on id=name with value t OK 35.sendKeys on id=name with value ${KEY_DOWN} OK 36.type on id=name with value TajHotel OK 37.click on id=email OK 38.type on id=email with value r OK 39.sendKeys on id=email with value ${KEY_DOWN} OK 40.type on id=email with value royal.taj@gmail.com OK 41.click on id=phone OK 42.type on id=phone with value 0 OK 43.sendKeys on id=phone with value ${KEY_DOWN} OK 44.type on id=phone with value 03336543218 OK 45.click on id=loc OK 46.type on id=loc with value 7 OK 47.click on id=manager OK 48.type on id=manager with value Shermeen OK 49.click on css=.col-12 > .btn-primary OK 50.click on css=.toggle-btn OK 51.click on linkText=Manage Employees OK 52.click on css=.container > .btn OK 53.click on css=.btn-warning OK 54.click on id=empID OK 55.type on id=empID with value 2 OK 56.click on css=.btn-danger OK 57.click on css=.container > .btn OK 58.click on css=.btn-success OK 59.click on id=Name OK		

TestID	TestData	ExpectedResult	Status
	60.type on id=Name with value Sara OK 61.click on id=post OK 62.type on id=post with value Chef OK 63.click on id=hiredate OK 64.type on id=hiredate with value 2025-05-02 OK 65.click on css=#addForm > h5 OK 66.click on id=phone OK 67.type on id=phone with value 0 OK 68.sendKeys on id=phone with value ${KEY_DOWN} OK 69.type on id=phone with value 03336543218 OK 70.click on id=hid OK 71.type on id=hid with value 4 OK 72.click on css=.col-12 > .btn-primary OK 73.click on css=tr:nth-child(28) > td:nth-child(2) OK 74.click on css=tr:nth-child(34) OK 75.click on css=.fa-hotel OK 76.click on linkText=Manage Guests OK 77.click on css=.container > .btn OK 78.click on css=.btn:nth-child(7) OK 79.click on css=#updateForm #cnic OK 80.type on css=#updateForm #cnic with value 10234-5678000-0 OK 81.mouseDownAt on id=updateField with value -868,-280.3333511352539 OK 82.mouseMoveAt on id=updateField with value -868,-280.3333511352539 OK 83.mouseUpAt on id=updateField with value -868,-280.3333511352539 OK 84.click on id=updateField OK 85.click on css=option:nth-child(1) OK 86.click on id=newValue OK 87.type on id=newValue with value Sam OK 88.type on id=newValue with value Sameer OK 89.click on css=#updateForm > .row OK 90.click on css=.btn-info:nth-child(1) OK 91.click on css=tbody > tr:nth-child(1) OK 92.click on css=.fa-hotel OK 93.click on css=.sidebar-item:nth-child(5) span OK		

TestID	TestData	ExpectedResult	Status
	94.mouseOver on css=.container > .btn OK 95.click on css=.container > .btn OK 96.mouseOut on css=.container > .btn OK 97.click on css=.btn:nth-child(7) OK 98.click on css=#updateForm #roomNo OK 99.type on css=#updateForm #roomNo with value 102 OK 100.click on css=#updateForm #hlId OK 101.type on css=#updateForm #hlId with value 1 OK 102.mouseDownAt on id=updateField with value -868,-366 OK 103.mouseMoveAt on id=updateField with value -868,-366 OK 104.mouseUpAt on id=updateField with value -868,-366 OK 105.click on id=updateField OK 106.select on id=updateField with value label=Room Status OK 107.click on css=#updateField > option:nth-child(2) OK 108.click on id=newValue OK 109.mouseDownAt on id=updateField with value -868,-360 OK 110.mouseMoveAt on id=updateField with value -868,-360 OK 111.mouseUpAt on id=updateField with value -868,-360 OK 112.click on id=updateField OK 113.select on id=updateField with value label=Room ID OK 114.click on css=option:nth-child(3) OK 115.click on id=newValue OK 116.type on id=newValue with value 1 OK 117.sendKeys on id=newValue with value ${KEY_ENTER} OK 118.click on css=.fa-hotel OK 119.click on linkText=Show Database OK 'testadminlogin_1' completed successfully		

TestID	TestData	ExpectedResult	Status
FTC_01	Verify login fails when an incorrect username is entered with a valid password on the login screen.	System should display “Invalid username or password” and deny access.	Fail
FTC_02	Verify that hotel booking fails when the user enters a phone number not matching the 11-digit format.	System should display “Meet the required format phone number” and prevent booking from proceeding.	Fail
FTC_03	Verify that the admin cannot add a new hotel when the hotel phone number does not meet the required format.	System should display “Invalid phone number format” and prevent the hotel from being added.	Fail