Smart Hotel Management System

Submitted By

Ziaul Rashid Elham

Student Id: 221-35-815

Department of Software Engineering

Daffodil International University

Supervised By

Mr. Nuruzzaman Faruqui

Designation: Assistant Professor

Faculty of Science and Information Technology

Department of Software Engineering Daffodil

International University

This project report has been submitted in fulfilment of the requirements for the degree a **Bachelor of Science in Software Engineering**

APPROVAL

This Project titled on **"Hotel Management System"**, submitted by **Ziaul Rashid Elham (ID: 221-35-815)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. A. H. M. Saifullah Sadi
Professor
Department of Software Engineering
Faculty of Science and Information Technology Daffodil International University

Internal Examiner 1

Dr. Rubaiyat Islam
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Dr. Md. Abdul Kader
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 3

Nuruzzaman Faruqui
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Md. Mostafiz Khan
Managing Director
Tecognize Solutions Limited

Smart Hotel Management System

Ziaul Rashid Elham

Bachelor of Science

DAFFODIL INTERNATIONAL UNIVERSITY

DAFFODIL INTERNATIONAL UNIVERSITY

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : Ziaul Rashid Elham
Date of Birth : 28-02-2000
Title : Smart Hotel Management System

Academic Session : 2022-2025

I declare that this thesis is classified as:

□ CONFIDENTIAL (Contains confidential information under the Official Secret A Act 1997)*
□ RESTRICTED (Contains restricted information as specified by the organization where research was done)*
□ OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Daffodil International University reserves the following rights:

1. The Thesis is the Property of Daffodil International University.
2. The Library of Daffodil International University has the right to make copies of the thesis for the purpose of research only.
3. The Library of Daffodil International University has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature)

ID:221-35-815
Date:8-12-21

(Supervisor's Signature)

Mr. Nuruzzaman Faruqui
Assistant Professor
Date: 21-12-25

PROJECT DECLARATION LETTER (OPTIONAL)

Librarian,
Daffodil International University,
Daffodil Smart City,
Ashulia.Dhaka,Bangladesh

Dear Sir,
CLASSIFICATION OF Project AS RESTRICTED

Please be informed that the following project is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.
Author's Name: Ziaul Rashid Elham
Project Title: Smart Hotel Management System

Reasons (i)

(ii)

(iii)

Thank you.
Yours faithfully,

(Supervisor's Signature)
Date:
Stamp:

Note: This letter should be written by the supervisor and addressed to the Librarian, Daffodil International University with its copy attached to the thesis.

SUPERVISOR's DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Science.

Mr. Nuruzzaman Faruqui
Assistant Professor
Department of Software Engineering

STUDENT'S DECLARATION

I hereby declare that the work in this project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution.

Elham

Ziaul Rashid Elham
ID: 221-35-815
Batch: 37
Date: 22-11-25
Department of Software Engineering

Smart Hotel Management System

Ziaul Rashid Elham

Project submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Science

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

DECEMBER 2024

ACKNOWLEDGEMENT

I thank Almighty Allah sincerely because He provided me with the power and chance to say these words. and complete this project. I would like to take this opportunity to thank my supervisor, Mr. Nuruzzaman Faruqui, Assistant Professor at the Department of Software Engineering of Daffodil International University. His steadfast backing, advice, and encouragement have played a crucial role in my academic success. determining the result of this work. I thank each of the individuals who provided their knowledge and offered their help during different phases of this project. Their help and valuable suggestions have greatly helped its successful completion. Finally, I must take the opportunity to offer my gratitude first of all to the entire faculty of the Department of Software Engineering. unbroken motivation and encouragement throughout my academic experience.

DEDICATION

I therefore declare that I have done this project under the oversight of Mr. Nuruzzaman Faruqui ,Assistant Professor, Department of Software Engineering, Daffodil International University. Also declare that neither entire record nor any portion of this record has been submitted somewhere else for my degree.

ABSTRACT

Smart Hotel Management Systems use the latest technologies, including the Internet of Things (IoT), Artificial Intelligence (AI), cloud computing, and data analytics to optimize the work of hotels, better their experience with guests, and increase the efficiency of resources use. This system allows managing rooms automatically, smart check-in and check-out, personalized guest management, and light and temperature control which is energy-efficient and real-time monitoring of the hotel facilities. Smart hotel management decreases the amount of human intervention, operational expenses and errors by combining smart reservation system, online payment system, and customer review analysis. Also, AI-based trends can be used to aid decision-making when it comes to demand prediction, personnel management, and optimization of maintenance. The Smart Hotel Management System in general is a scalable, secure, and efficient system that is able to turn traditional hotels into intelligent, customer-centric service spaces.

Table of Contents

Chapter 1

Introduction

1. About the Project

The project is a Hotel Management System created on the basis of a web-application (Laravel + PHP) to administer different hotel activities: rooms and bookings, check-in/out, possibly billing, and administration activities. It is a full-stack web application as shown by the folder structure (routes, database, public, resources).

Key aspects likely include

• Room management: create, edit, delete rooms

.• Room reservation: Customers are able to book rooms; employees are able to access reservations

.• User statuses: Admin, staff, maybe guest.Database-based: Permanent storage rooms, bookings, user information

.•Web interface: Both front-office (reception) and back-office (management).Testing: The repo contains test files that signify unit or integration tests.

2. Project Objectives

The project is a Hotel Management System created on the basis of a web-application (Laravel + PHP) to administer different hotel activities: rooms and bookings, check-in/out, possibly billing, and administration activities. It is a full-stack web application as shown by the folder structure (routes, database, public, resources)

Key aspects likely include:

• Room management: create, edit and delete rooms

• Room reservation: Customers are able to book rooms; employees are able to access reservations

.• User statuses: Admin, staff, maybe guest.Database-based: Permanent storage rooms, bookings, user information

.•Web interface: Both front-office (reception) and back-office (management).Testing: The repo contains test files that signify unit or integration tests.

. Scalability: Construct a system that is scalable with the needs of the expansion of the hotel (more rooms, more bookings).
. Security and reliability: Data of guests and hotels must be in a secure place, data consistency, and recovery (backups, error handling).

3. Project Scope Definition

1.Room inventory are management CRUD (create, read, update, delete) operations on rooms, room types (single, double, suite), room status (available, occupied, maintenance)
2.Booking/reservation system: Customers and the staff can make and view, change and cancel bookings.
3. Check-in/check-out module: rooms assignment of guests, registration of their stay, calculation of time.
4.Billing/invoicing: Creating invoices of rooms, computing the charges (per night, additional services), and even taking note of payments.
5.Role management: Administrator, receptionist, housekeeper; authorization.
6.Reporting: Simple reports such as occupancy reports, revenue reports and future bookings.
7.Web UI: Cockpit, booking, and administration.
8.Database: A relational database where all the relevant entities have to be stored.
9. Testing: Automated tests of critical functionality (unit tests, maybe integration tests).

Out of Scope (in original version);

1.Channel management (alignment with online travel agents, OTAs)

2.Dynamic (yield) management or advanced revenue management

.3. Mobile application (not the web responsive).

4.State-of-the-art analytics / predictive forecasting

5.Scheduling of housekeeping work outside room condition.

6.Multi-property operation (when targeted at one hotel).

4. Market Research

1.4.1 Market Analysis

Hotel and hospitality business is becoming digital at a rapid pace. Small guesthouses to luxury hotels are abandoning manual registers and old-fashioned desktop based systems and migrating to web based solutions of hotel management. Reportedly, the Hotel Management Software market has a valuation of the several billion USD and it is rising steadily, at a CAGR of 78. The need to automate and enhance the guest experience and efficiency in operations drives this growth.

Hotels are now focusing on:

- Updates on room booking in real-time.
- Fast check-in/check-out
- Centralized guest data
- Efficient billing systems
- Personnel organization and reporting.

This also presents a great demand of the contemporary, basic, and cost-effective HMS solutions, particularly in developing nations such as Bangladesh.

Comparing Current Systems

Feature	Legacy / Manual	Standard HMS	Smart HMS (AI-Powered)
AI Customer Support (24/7, Multilingual) Chat/voice automation, intent routing, human handoff	No	Limited	Yes
Reservations & Channel Management OTA sync, availability, overbooking protection	Limited	Yes	Yes
Real-Time Notifications Guest updates, staff alerts, SLA reminders	No	Limited	Yes
Feedback & Analytics Surveys, trends, anomaly detection	No	Basic	Advanced
User Role Management RBAC, audit logs, SSO	No	Limited	Yes

Fig: 1.1

1.4.2 Market Feasibility

Market Feasibility is a study that examines the possibility of the proposed Hotel Management System to succeed in the existing market taking into consideration the demand, competition, customer reception and price. It assists in establishing whether the project is viable, profitable as well as sustainable.

. **1. Market Demand Analysis**

The demand of inexpensive and easy to use hotel management software, in particular, among:

A. Small & Medium Hotels

• Within the small hotels, most of them continue to use paper registers or excel sheets

.These systems lead to booking errors, and booking in and into as well as slow check-ins.

.Owners are searching to have low cost software to automate these functions.

B. Guest Houses, Motels, Resorts.

To manage the following businesses, lightweight systems are required:

• Room availability

• Guest details • Bills & payments

C. Local Hotels (Asia Region/Bangladesh)

The percentage of hotels in the area, which are not equipped with modern property management software, is large. (PMS) developing a strong market opportunity of easy and economic solutions

2. Target Customer Acceptance

This system is likely to be acceptable to the target customers due to the following reasons:

• It reduces manual work

• Increases staff efficiency

• No need of high-level technical expertise

1.5 System Feasibility Analysis

1.5.1 Feasibility Study Overview

Feasibility analysis looks at the feasibility of this system and its value of building. The following is a breakdown on various dimensions:

1.5.2 Operational Feasibility

Operational Feasibility assesses whether the proposed system will run effectively in the hospitality environment of an average hotel and whether the staff will manage the system without too many difficulties.

1. Ease of Use for Hotel Personnel

This system enables the hotel staff, whether receptionist staff, management staff, or admin staff, to carry out their daily tasks effortlessly; these tasks include:

- Room status management
- Browsing customer records
- Handling Payments
- A user-friendly interface makes it easy, even for staff with only elementary knowledge of computers to operate the system without training.

1.5.3 Technical Feasibility

Feasibility Type	Description	Analysis for This Project (Hotel Management System)	Result
1. Technical Feasibility	Assures the availability of the necessary technology, tools and skills to develop	- Built using Laravel/PHP, easy to develop and maintain. -	Feasible
2. Operational Feasibility	Determines whether the system will be accepted by users and whether it will improve current	- Simple UI, easy for hotel staff to learn. - Automates manual tasks (booking, billing, check-	Highly Feasible
3. Economic Feasibility (Cost–Benefit Analysis)	Evaluates whether the system is financially viable.	Costs: Development effort, minimal hosting cost. Benefits: Saves staff	Cost-Effective / Feasible
4. Time/Schedule Feasibility	Checks whether the system can be developed within a reasonable	The project modules (rooms, bookings, billing, users) can be completed	Feasible
5. Legal Feasibility	Ensures the system complies with legal, data, and privacy standards.	- Stores customer data securely. - No violation of intellectual property (uses	Feasible
6. Operational Environment Feasibility	Checks whether the environment required to run the system is	- Works on any modern PC or mobile device. - Can run on local	Feasible
7. Social Feasibility	Determines whether users will accept the change from manual to digital	- Hotel staff prefer automation to reduce workload. - Owners	Feasible
8. Security Feasibility	Evaluates whether the system can protect user and hotel data.	- Laravel authentication system provides built-in security. - Role-based	Feasible

1.5.4 Development Environment & Technology Stack

Layer	Technology / Tool	Purpose / Role
Backend / Framework	Laravel (PHP)	The project uses PHP with Laravel, as indicated by composer.json, artisan, and Laravel folder structure. Laravel is a popular MVC PHP framework.
Frontend (CSS / UI)	Tailwind CSS	There's a tailwind.config.js in the repo, which suggests Tailwind is used for styling. Tailwind is a utility-first CSS framework.
CSS Framework	Bootstrap	There is also a bootstrap folder in the repo, indicating some use of Bootstrap.
JavaScript / Frontend Build	Node / NPM	The repo has a package.json. This suggests use of Node.js tooling to manage JS dependencies or build Tailwind.
Database	MySQL (or another relational DB)	There’s a database folder in the repo, and Laravel often uses MySQL / MariaDB / PostgreSQL for relational data. (Exact DB type not explicitly specified in README.)
Routing / Web	Laravel Routing	The routes directory shows Laravel-style routing.
Storage	Laravel Storage / Filesystem	There’s a storage directory in the project, likely using Laravel’s filesystem for file uploads, caching, or logs.
Testing	PHPUnit	There’s a phpunit.xml file in the repo.
Version Control	Git / GitHub	The project is hosted on GitHub, using Git for version control.

Development Environment (What You Would Need to Develop)

The probable development set-up would be as follows based on the stack above:

Local Machine Requirements:

- PHP (compatible version for Laravel used in the project)
- Composer (for PHP dependency management)
- Node.js + npm or yarn (for JS / Tailwind)
- A relational database (e.g. MySQL) installed locally or via Docker / another host
- Web server (e.g. Apache / Nginx) — or use Laravel's built-in server (php artisan serve)
- IDE / Code Editor:
- VS Code, PHPStorm, or any editor that supports PHP and JS
- Optional: Laravel extension/plugins for better dev experience

1.5.5 Financial Feasibility Analysis

Financial feasibility analysis is done to determine whether the proposed system is cost effective as well as whether the benefits are more than the costs. This analysis assists the stakeholders in making a decision or on whether a project is viable or not economically.

1. Objective

To calculate the are financial viability of development and implementation of Get Hotel Management System through the expected costs, potential savings and the return on investment (ROI).

2. Cost Analysis

The total cost includes development costs, operational costs, and maintenance costs.

Cost Category	Details	Estimated Cost (USD)
Software Development	Developer salaries,	$3,000
Hardware & Infrastructure	Computers, servers, database hosting	$1,000
Licenses & Tools	IDEs, frameworks, libraries	$200
Training & Support	Training staff to use the system, documentation preparation	$300
Maintenance	Bug fixes, updates, database maintenance (annual)	$500
Miscellaneous	Contingency for unforeseen expenses	$200
Total Estimated Cost		$5,200

3. Expected Benefits

The system will be in place to streamline the operations of the hotels in the sense that it will reduce the cost of labor, as well as, enhance efficiency. Benefits include:

Benefit	Details	Estimated Savings / Value (USD)
Labor Cost Reduction	Automating check-ins, room allocation, and billing reduces staff workload	$1,500/year
Operational Efficiency	Faster booking, billing, and inventory management	$1,200/year
Revenue Increase	Improved customer experience increases bookings and customer retention	$2,000/year
Error Reduction	Minimizes mistakes in billing and reservations	$500/year
Total Expected Annual Benefits		**$5,200/year**

4. Break-even Analysis

Break-even point is time within which the system breaks even i.e. breaks even at this point.

Break-even Point=Annual Savings / BenefitsTotal Estimated Cost. Break-even Point=5200\5200=1 year

Interpretation: The system will break even in 1 Yr and hence it can be economically viable

5. ROI (Return on Investment)

ROI(%)=Total CostNet Benefits×100 And annual benefits = 5,200 and cost = 5,200: ROI=5200\5200−5200×100=0%

Nevertheless, in later years, ROI rises due to the fact that the system remains profitable with the less cost incurred.

The financial feasibility analysis has shown that:
Initial development cost is $5,200.
.It is expected that the benefits will take 1 year to cover the cost, which is expected to be 5,200 per year.
.System saves labor, increases efficiency, and might result in increased revenue.
. Therefore, financially, Hotel Management System is viable.

1.6 Target Users

The Hotel Management System will accommodate many users who will be using the system depending on the purposes.

1.Hotel Manager / Admin
2.Front Desk Personnel / Receptionist.
3.Housekeeping Staff
4.Customers / Guests
5.Accounts / Finance Staff

1.6.1 Target User Roles

User Type	Role in the System	Responsibilities / Tasks
Hotel Manager / Admin	System Administrator	- Manage hotel operations- Oversee bookings, staff, and room availability- Generate reports on revenue, occupancy, and performance- Approve or monitor staff actions
Receptionist / Front Desk Staff	Booking & Customer Management	- Check-in and check-out guests- Handle reservations and cancellations- Issue invoices and bills- Manage customer queries and complaints
	Room Management	- Update room status (cleaned, occupied, vacant)- Report maintenance issues- Coordinate with reception for room readiness
Customers / Guests	Service User	- Make online or offline room bookings- View room availability and pricing- Provide feedback or request services- Access personal booking history
Accounts / Finance Staff	Financial Management	- Manage payments and billing- Track hotel revenue and expenses- Generate financial reports for management review

1.6.2USER PROFILES

- **. Hotel Manager / Admin**

Attribute	Description
Role	System Administrator
Access Level	Full access to all modules (rooms, staff, bookings, finances)
Responsibilities	Manage hotel operations, monitor staff performance, generate reports, oversee bookings and finances
Technical Skills	Basic computer literacy; comfortable with web interfaces
Goals	Ensure smooth operation of the hotel, track performance, maximize revenue
Usage Frequency	Daily

2. Receptionist / Front Desk Staff

Attribute	Description
Role	Booking & Customer Management
Access Level	Access to booking, check-in/check-out, customer management, invoice generation
Responsibilities	Handle reservations, customer queries, billing, and check-ins/check-outs
Technical Skills	Moderate computer literacy, familiarity with booking systems
Goals	Provide fast, accurate service to guests and maintain accurate records
Usage Frequency	Daily

3. Housekeeping Staff

Attribute	Description
Role	Room Management
Access Level	Access to room status updates and maintenance reports
Responsibilities	Update room cleanliness status, report maintenance issues
Technical Skills	Basic computer literacy, mobile device usage
Goals	Maintain room readiness and support hotel operations
Usage Frequency	Daily

4. Customers / Guests

Attribute	Description
Role	Service User
Access Level	Limited access to booking system, personal booking history, feedback
Responsibilities	Book rooms, provide feedback, make payments
Technical Skills	Basic digital literacy for web or mobile booking
Goals	Make reservations easily, get timely service, and enjoy a smooth stay
Usage Frequency	Occasional (based on bookings)

5. Accounts / Finance Staff

Attribute	Description
Role	Financial Management
Access Level	Access to billing, payments, revenue reports, and financial data
Responsibilities	Track financial transactions, generate invoices, manage revenue and expenses
Technical Skills	Moderate computer literacy, familiarity with accounting software
Goals	Maintain accurate financial records and support decision-making
Usage Frequency	Daily

1.6.3 Elicitation Technique

Technique	Description	Purpose	Stakeholders Involved	Example in Project
Interviews	Direct conversations with stakeholders	Gather detailed insights about current workflow, challenges, and expectations	Hotel Manager, Receptionist, Housekeeping Staff	Asking receptionists about difficulties in managing bookings and check-ins
Questionnaires / Surveys	Structured forms for quantitative and qualitative data collection	Understand user preferences, common issues, and desired features	Customers, Staff	Survey asking guests about preferred online booking features
Observation	Observing day-to-day hotel operations	Identify pain points and inefficiencies in real-time	Receptionist, Housekeeping Staff	Observing how multiple bookings are handled simultaneously
Document Analysis	Reviewing existing documents, reports, invoices, and records	Understand current processes, data formats, and operational requirements	Hotel Manager, Accounts Staff	Analyzing past booking logs and invoices to identify patterns and errors
Prototyping	Creating low or high-fidelity system mock-ups	Validate requirements and gather feedback on interface and functionality	Hotel Manager, Receptionist, Customers	Showing booking module mock-ups to receptionists and managers
Focus Group	Group discussions with stakeholders from different roles	Collect diverse perspectives and agree on priority features	Hotel Manager, Receptionist, Housekeeping Staff	Discussion about room allocation, maintenance, and service features

1.7 System Requirements

Category	Minimum Requirement	Recommended / Notes
Operating System	Windows, Linux, macOS	Any modern OS that supports PHP + web server + database
Web Server	Apache or equivalent (e.g., Nginx with PHP)	Apache recommended since Laravel-based stack present
PHP Version	PHP 7.4 or newer (repository shows PHP 7.4)	Use latest PHP 8.x for better performance & security
Database	MySQL / MariaDB (or any DB supported by Laravel)	Scheme present under database/ folder
RAM	At least 2 GB	4 GB or more recommended for smoother operation
Disk Space	~500 MB (code + dependencies)	More space for logs, uploads, bigger database
Node.js / NPM	For front-end build (if used)	Files like package.json, vite.config.js present
Web Browser (for Access)	Modern browser (Chrome, Firefox, Edge)	Latest stable version
Network	Local or/and remote web-hosting environment	Consider HTTPS if deploying publicly

1.8 Project Schedule

1.8.1 Project Schedule: May 2025 – November 2025

1. Project Planning & Research (May 2025)

- Tasks:
  - ○ Finish the objectives and scope of a project.
  - ○ Facebook apps: HTML,CSS,JS, PHP,Laravel.
  - ○. unfunctional and functional Requirements Explain the requirements
  - ○ Develop frontend and design the administrative panel
  - ○ Develop the environment by configuring the Laravel, MySQL. and IDE.

- Deliverables:
  - ○ proposal of the undertaking.
  - ○ created the user interfaces and wireframes.
  - ○ The environment of development is ready.

1.8.2 Backend Setup & MySQL Integration (June 2025)

- Tasks:

○ Larval project development and addition of db Storage and Authentication. Frontend Use firebase to arrange Firebase with larval command.

○ Email and password user identification

- Deliverables:

1.established and joined a frontend to alaravalf project

2. The authentication of the user is functional.

3.The database is organized.

1.8.3 Web Development – Core Features (July 2025)

- Tasks:

  The app for students is:

  - Using the home screen, the events are displayed
  - screen with the details on the event.
  - The ability to register events
  - Control user profiles.

  Firebase Cloud Messaging (FCM) has the ability to enable Push alerts, which can be sent.

  - Get to know the key characteristics of the Smartphones (iOS and Android)

- Deliverables:

Basic navigation features such as browsing of events, event registration, and profile editing are included in the mobile app.

- Push alerts are operational.

1.8.4 Web Development – Advanced Features (August 2025)

- Tasks:

  Add advanced features:

  - Calendar integration
  - A venue to lend support as well as provide criticism.
  - Enhance operations and solve issues.

- Deliverables:

  - A smartphone application that has more features.
  - Report on the usability test

1.8.5 Admin Panel Development (September 2025)

- Tasks:

  Develop the admin panel:

. User/Registration tracking dashboard.

○ Connect db to the admin panel.

○ Browse the administrator panel of the

- Deliverables:

  ○ fully functioning administration panel.

  ○ Database can be running at the same time as an administration panel

1.8.6 Testing & Debugging (October 2025)

- Tasks:
  ○ do a thorough test of the web application and the admin panel

  . × Mend problems and improve productivity.

  ○ Groundlevel security testing.

  ○ Prepare user information and carry out modifications.
- Deliverables:
  ○ bug-free application.
  ○ Report on testing.

1.8.7 Documentation & Final Submission (November 2025)

- Tasks:

Write documentation of project:
○ This is both the user and the administration manual.
○ Technical documentations (structure of the code, system architecture).
ADDRESS: The project should be awarded to your university.

- Deliverables:
    - full project records.
    - Presentation.
    - completion of the project

1.9 Grantt Chart

Fig: 1.2

Chapter 2

Design and Implementation

2.1 Functional Requirements

FR-1	User Registration & Login
Description	Users can register, log in/out, recover password
Stakeholder	Guest, Admin, Receptionist

FR-2	Role-based Access Control
Description	Each role has different permissions & dashboards
Stakeholder	Admin, Manager

FR-3	Room Management
Description	Add/update/delete rooms, set room type, price
Stakeholder	Admin, Manager

FR-4	Room Availability Search
Description	Search for available rooms by date/type
Stakeholder	Guest, Receptionist

FR09	Billing / Invoice Generation
Description	Generate invoices that include room charges, service charges, taxes
Stakeholder	Finance Team, Manager, Guests

FR10	Housekeeping Status Tracking
Description	Track room cleaning status (dirty, cleaning, clean) and update
Stakeholder	Housekeeping Staff, Receptionist, Manager

FR11	Employee / Staff Management
Description	Admin can manage staff profiles, assign roles
Stakeholder	Admin, Manager, IT Support

FR12	Guest Profile Management
Description	Store guest information (contact details, preferences)
Stakeholder	Guest, Receptionist, Manager

FR-5	Reservation / Booking
Description	Guests can create bookings, system generates booking IDs, check-in and check-out dates
Stakeholder	Guests, Receptionist, Manager

FR-6	Modify / Cancel Booking
Description	Allow users to change or cancel their reservations
Stakeholder	Guests, Receptionist

FR07	Check-In & Check-Out
Description	Receptionists register guest arrival and departure, assign rooms
Stakeholder	Receptionist, Housekeeping, Manager

FR08	Payment Processing
Description	Accept payments (advance / full), support payment status tracking
Stakeholder	Guests, Finance Team, Admin

FR13	Reporting / Analytics
Description	Generate reports for occupancy, revenue, performance
Stakeholder	Manager, Admin, Owner / Investor

FR14	Notifications / Alerts
Description	Send notifications: booking confirmation to guests, housekeeping alerts, etc.
Stakeholder	Guests, Housekeeping, Manager

FR15	Service Management
Description	Manage extra services: room service, minibar, meals
Stakeholder	Guests, Receptionist, Housekeeping, Finance

FR16	CRUD for Core Entities
Description	Create, Read, Update, Delete operations for rooms, users, bookings, services
Stakeholder	Admin, IT Support, Manager

FR17	Room Type Management
Description	Create and customize room categories (Deluxe, Single, Suite, Family Room)
Stakeholder	Admin, Manager

FR18	Seasonal Pricing Control
Description	Set dynamic pricing based on season, holidays, occupancy
Stakeholder	Admin, Manager

FR19	Discount / Coupon Management
Description	Add promotional codes or discounts for guests
Stakeholder	Admin, Receptionist

FR20	Food & Beverage Ordering
Description	Guests can request food or room service from the system
Stakeholder	Guests, Kitchen Staff, Receptionist

FR21	Service Charge Tracking
Description	Add service charges to a guest's bill automatically
Stakeholder	Finance, Receptionist

FR22	Inventory Management
Description	Track hotel inventory (bedsheets, room supplies, toiletries, minibar items)
Stakeholder	Housekeeping, Manager

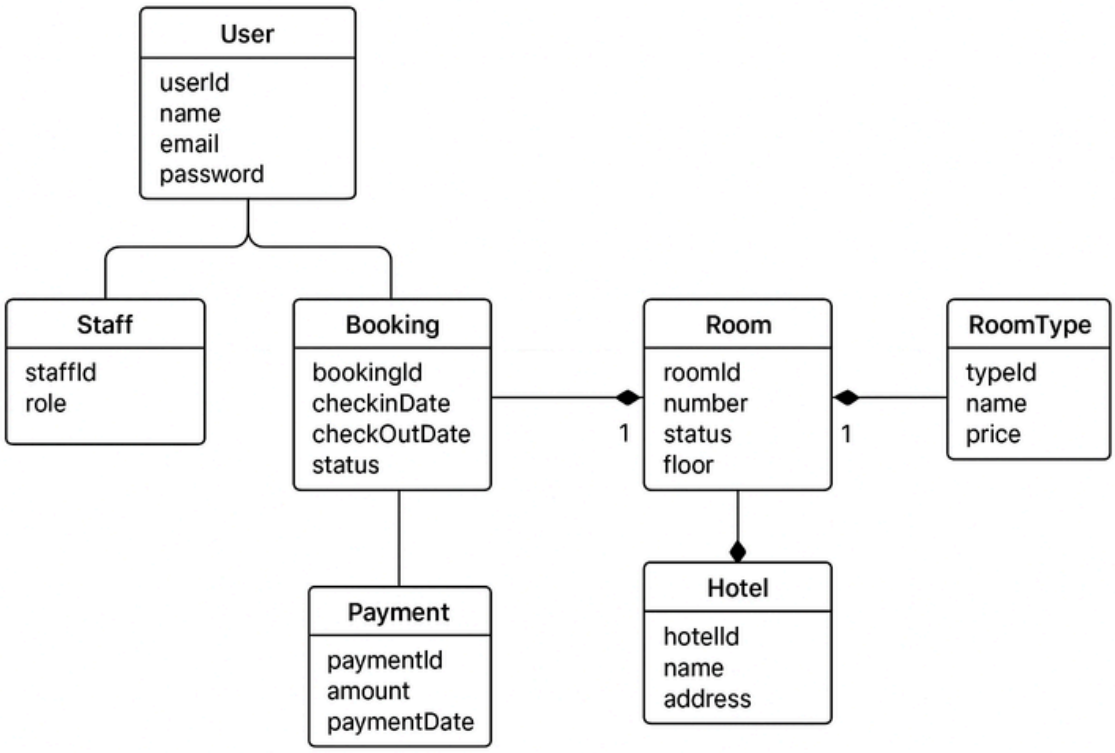
FR23	Minibar Consumption Tracking
Description	Record items consumed from minibar and auto-add to bill
Stakeholder	Housekeeping, Finance

FR24	Laundry Service Management
Description	Manage laundry requests and add charges to bill
Stakeholder	Guests, Housekeeping, Finance

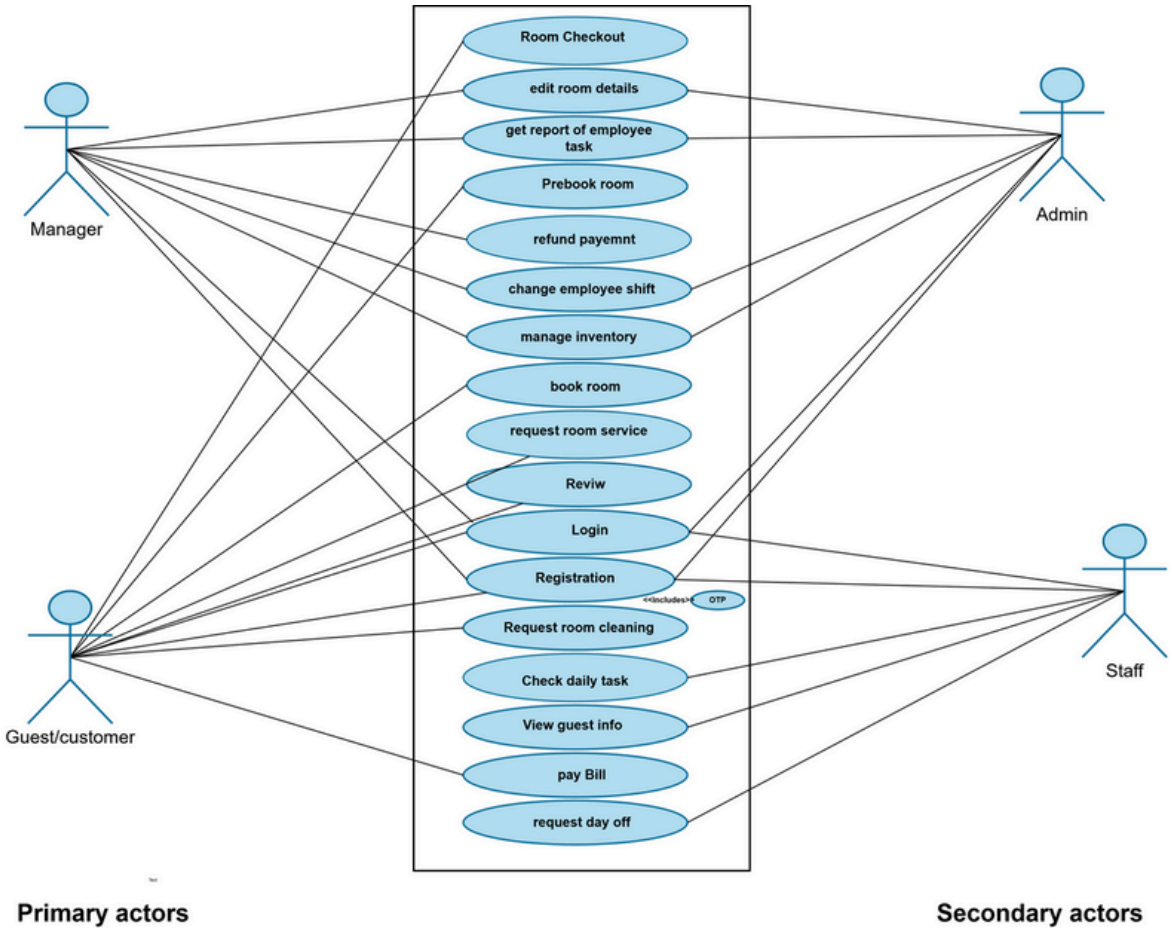
2.2 Non-Functional Requirements

ID	Non-Functional Requirement	Description / Details	Stakeholders
NFR-1	**Performance / Response Time**	The system must respond to user requests within 2–3 seconds for bookings, searches, and data retrieval	Guests, Receptionist, Admin
NFR-2	**Scalability**	System should support at least 500 concurrent users without performance degradation	Admin, Manager
NFR-3	**Availability / Uptime**	System should be available 24/7 with minimum downtime (<1% monthly)	Guests, Staff, Manager
NFR-4	**Reliability**	Ensure correct and consistent operation of booking, billing, and room management functions	Guests, Receptionist, Finance
NFR-5	**Security / Authentication**	Password encryption, secure login, role-based access, and session management	Admin, IT Support, Guests
NFR-6	**Data Privacy / Compliance**	Compliance with GDPR / local data protection laws for guest data	Admin, Manager, Guests
NFR-7	**Maintainability**	Code should be modular and well-documented to allow easy updates and bug fixes	IT Support, Developer Team
NFR-8	**Backup & Recovery**	Daily automated backups; system can restore data within 1 hour of failure	Admin, IT Support
NFR-9	**Usability / User-Friendliness**	Simple navigation, clear instructions, mobile-friendly design	Guests, Receptionist, Housekeeping
NFR-10	**Portability / Compatibility**	Accessible on multiple devices: PC, tablet, mobile, and all major browsers	Guests, Staff
NFR-11	**Interoperability**	System can integrate with payment gateways, email/SMS services, and external reporting tools	Finance Team, Admin, Guests
NFR-12	**Auditability / Logging**	Track all critical user actions for security and compliance purposes	Admin, IT Support, Manager
NFR-13	**Localization / Multilingual Support**	Support multiple languages for international guests	Guests, Receptionist
NFR-14	**Accessibility**	System should comply with accessibility standards (e.g., WCAG) for users with disabilities	Guests, Staff
NFR-15	**Data Integrity**	Ensure accurate booking, payment, and service records; prevent duplicate entries	Admin, Finance, Manager
NFR-16	**Error Handling / Fault Tolerance**	Gracefully handle system errors, with meaningful messages for users	Guests, Receptionist, IT Support

2.3 UML DIAGRAM



2.4 Use Case Diagram



2.5 CASE DESCRIPTIONS

Use Case 1: Make Room Reservation

Field	Description
Use Case ID	UC001
Use Case Name	Make Room Reservation
Actor	Guest
Description	Guest books a room for specific dates
Precondition	Guest has valid contact information
Main Flow	1. Guest selects dates 2. System shows available rooms 3. Guest selects room and provides details 4. System processes payment 5. Confirmation sent
Postcondition	Reservation created, room reserved

Use Case 2: Check-in Guest

Field	Description
Use Case ID	UC002
Use Case Name	Check-in Guest
Actor	Front Desk Staff
Description	Register guest arrival and provide room access
Precondition	Valid reservation exists
Main Flow	1. Verify guest identity 2. Complete registration 3. Collect deposit 4. Issue room keys 5. Update room status
Postcondition	Guest checked in, room assigned

Use Case 3: Process Room Service

Field	Description
Use Case ID	UC003
Use Case Name	Process Room Service
Actor	Room Service Staff
Description	Handle guest food and beverage orders
Precondition	Guest is checked in
Main Flow	1. Receive order 2. Confirm details and cost 3. Prepare order 4. Deliver to room 5. Process payment
Postcondition	Order delivered and billed

Use Case 4: Manage Housekeeping

Field	Description
Use Case ID	UC004
Use Case Name	Manage Housekeeping
Actor	Housekeeping Staff
Description	Clean and maintain room standards
Precondition	Room requires cleaning
Main Flow	1. Receive task assignment 2. Clean room per standards 3. Report issues 4. Update room status 5. Quality inspection
Postcondition	Room clean and ready

Use Case 5: Check-out Guest

Field	Description
Use Case ID	UC005
Use Case Name	Check-out Guest
Actor	Front Desk Staff
Description	Process guest departure and final billing
Precondition	Guest ready to leave
Main Flow	1. Review final bill 2. Process payment 3. Collect keys 4. Refund deposit 5. Update room status
Postcondition	Guest departed, bill settled

Use Case 6: Handle Maintenance

Field	Description
Use Case ID	UC006
Use Case Name	Handle Maintenance
Actor	Maintenance Staff
Description	Repair facility and equipment issues
Precondition	Maintenance request submitted
Main Flow	1. Receive work order 2. Assess problem 3. Gather tools/parts 4. Perform repair 5. Update maintenance log
Postcondition	Issue resolved and documented

Use Case 7: Manage Staff Schedule

Field	Description
Use Case ID	UC007
Use Case Name	Manage Staff Schedule
Actor	Department Manager
Description	Create and maintain work schedules
Precondition	Staff availability known
Main Flow	1. Review occupancy forecast 2. Assess staffing needs 3. Create schedule 4. Handle change requests 5. Publish final schedule
Postcondition	Schedule created and distributed

Use Case 8: Process Guest Feedback

Field	Description
Use Case ID	UC008
Use Case Name	Process Guest Feedback
Actor	Guest Relations Manager
Description	Collect and respond to guest feedback
Precondition	Guest provides feedback
Main Flow	1. Receive feedback 2. Categorize and prioritize 3. Investigate issues 4. Prepare response 5. Implement improvements
Postcondition	Feedback addressed and logged

Use Case 9: Manage Inventory

Field	Description
Use Case ID	UC009
Use Case Name	Manage Inventory
Actor	Inventory Manager
Description	Maintain adequate supply levels
Precondition	Inventory system operational
Main Flow	1. Monitor stock levels 2. Create purchase orders 3. Receive deliveries 4. Update inventory 5. Distribute supplies
Postcondition	Inventory maintained at optimal levels

Use Case 10: Generate Reports

Field	Description
Use Case ID	UC010
Use Case Name	Generate Financial Reports
Actor	Finance Manager
Description	Create financial performance reports
Precondition	Financial data available
Main Flow	1. Select report parameters 2. Extract data 3. Calculate metrics 4. Generate report 5. Distribute to stakeholders
Postcondition	Reports generated and distributed

Use Case 11: Process Bill Payment

Field	Description
Use Case ID	UC011
Use Case Name	Process Bill Payment
Actor	Guest, Front Desk Staff
Description	Handle guest bill payments during stay
Precondition	Guest has outstanding charges
Main Flow	1. Retrieve guest bill 2. Review charges with guest 3. Select payment method 4. Process payment 5. Update account balance
Postcondition	Payment processed and recorded

Use Case 12: Manage Room Information

Field	Description
Use Case ID	UC012
Use Case Name	Edit Room Information
Actor	Hotel Manager, Front Desk Staff
Description	Update room details, pricing, and amenities
Precondition	User has admin privileges
Main Flow	1. Select room to edit 2. Update room details 3. Modify pricing/amenities 4. Validate changes 5. Save updates
Postcondition	Room information updated

Use Case 13: Manage Employee Information

Field	Description
Use Case ID	UC013
Use Case Name	Edit Employee Information
Actor	HR Manager, Department Manager
Description	Update staff personal and work details
Precondition	Employee record exists
Main Flow	1. Search employee record 2. Edit personal details 3. Update job information 4. Modify access permissions 5. Save changes
Postcondition	Employee information updated

Use Case 14: Handle Advance Payments

Field	Description
Use Case ID	UC014
Use Case Name	Process Advance Payment
Actor	Front Desk Staff
Description	Accept partial payments before full bill
Precondition	Guest wants to make advance payment
Main Flow	1. Calculate current charges 2. Accept payment amount 3. Process payment 4. Update account credit 5. Provide receipt
Postcondition	Advance payment recorded

Use Case 15: Manage Room Status

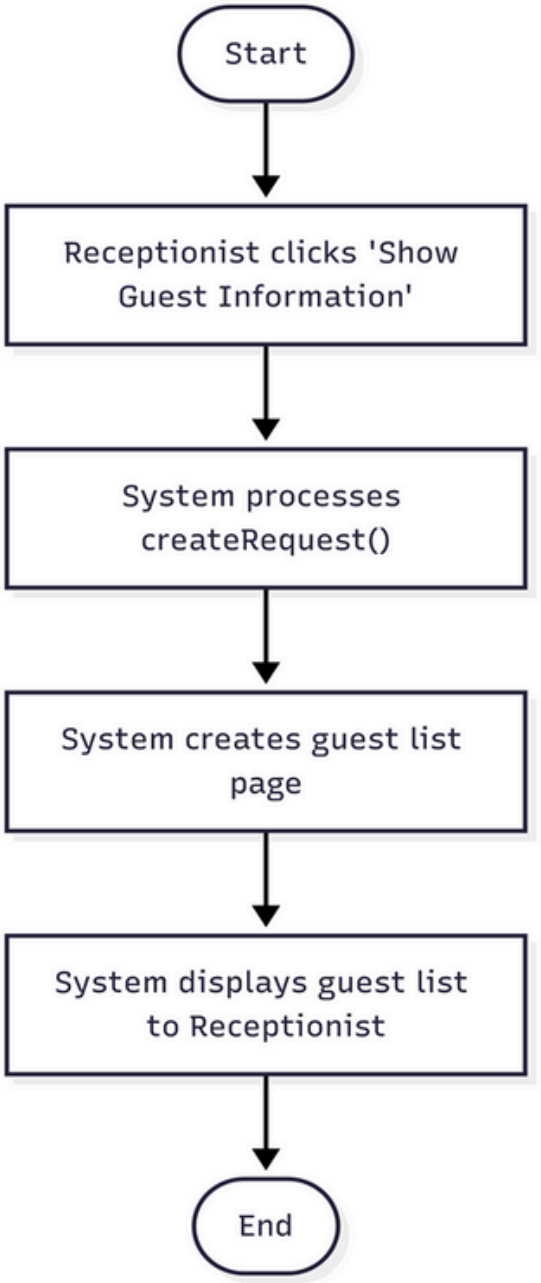
Field	Description
Use Case ID	UC015
Use Case Name	Update Room Status
Actor	Housekeeping Staff, Front Desk Staff
Description	Change room availability and condition status
Precondition	Room status needs updating
Main Flow	1. Select room 2. Update status (clean/dirty/maintenance) 3. Add status notes 4. Set availability 5. Notify relevant staff
Postcondition	Room status updated

Use Case 16: Process Refunds

Field	Description
Use Case ID	UC016
Use Case Name	Process Guest Refund
Actor	Finance Manager, Front Desk Staff
Description	Handle refund requests for cancellations or issues
Precondition	Valid refund request exists
Main Flow	1. Review refund request 2. Calculate refund amount 3. Get manager approval 4. Process refund 5. Update guest account
Postcondition	Refund processed and documented

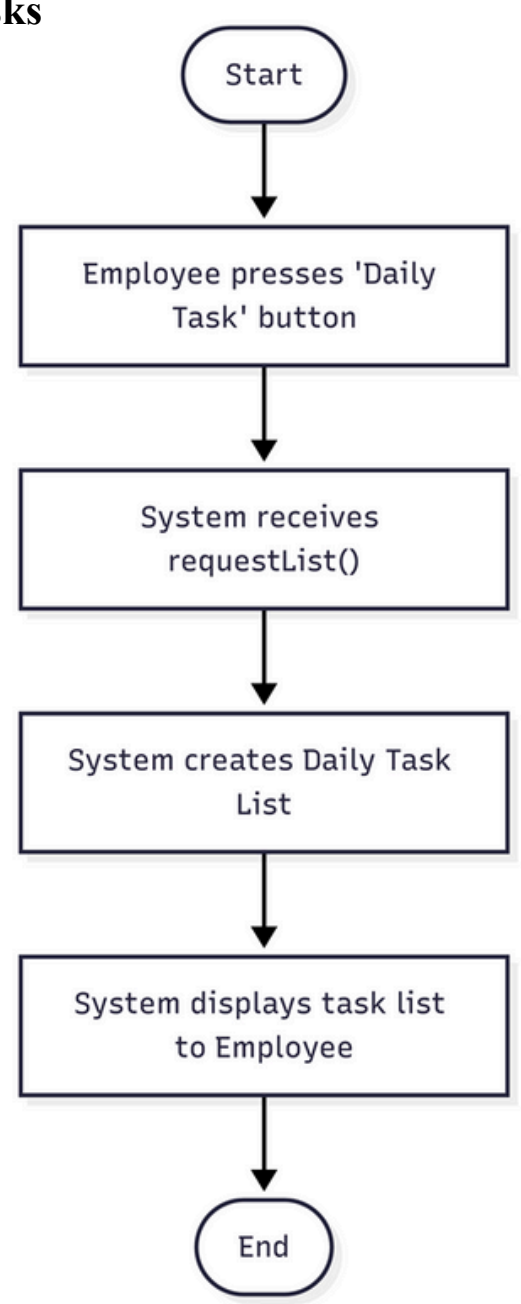
2.6 ACTIVITY DIAGRAMS

Show guest info



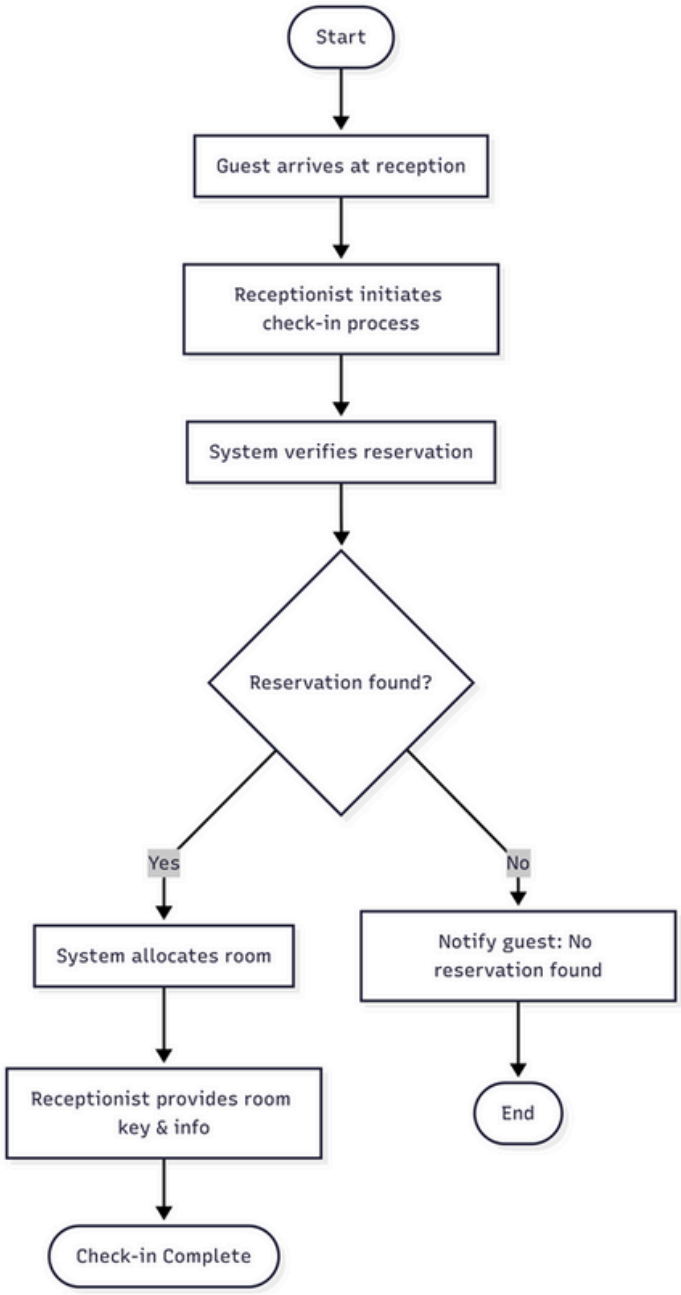
Activity Diagram

Display Daily tasks



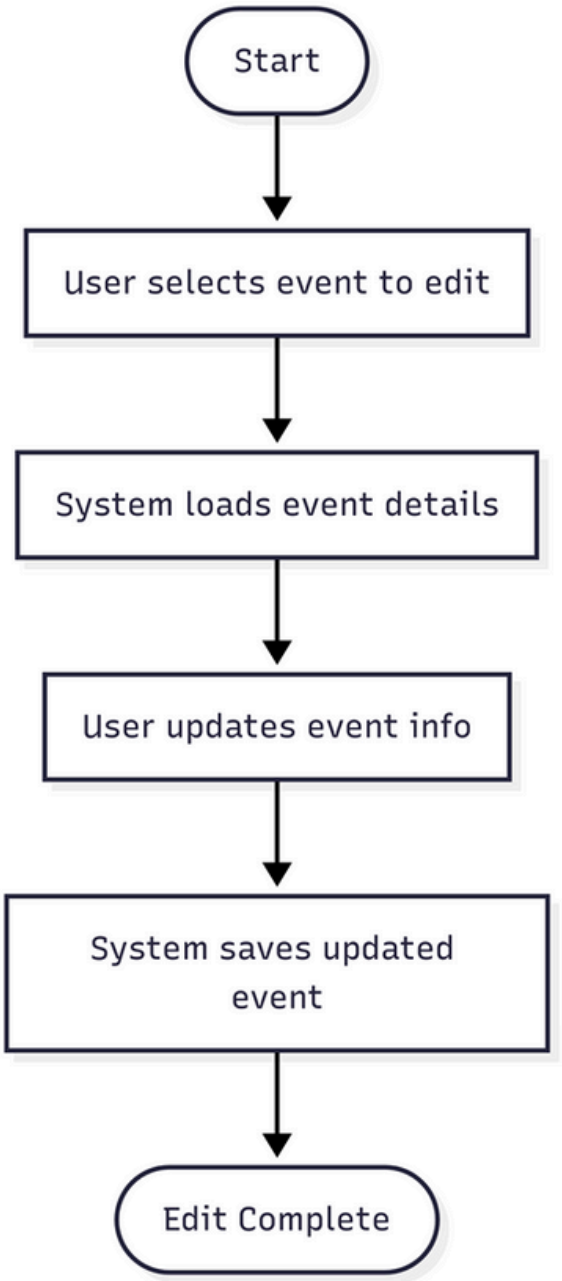
Activity Diagram

Check-in



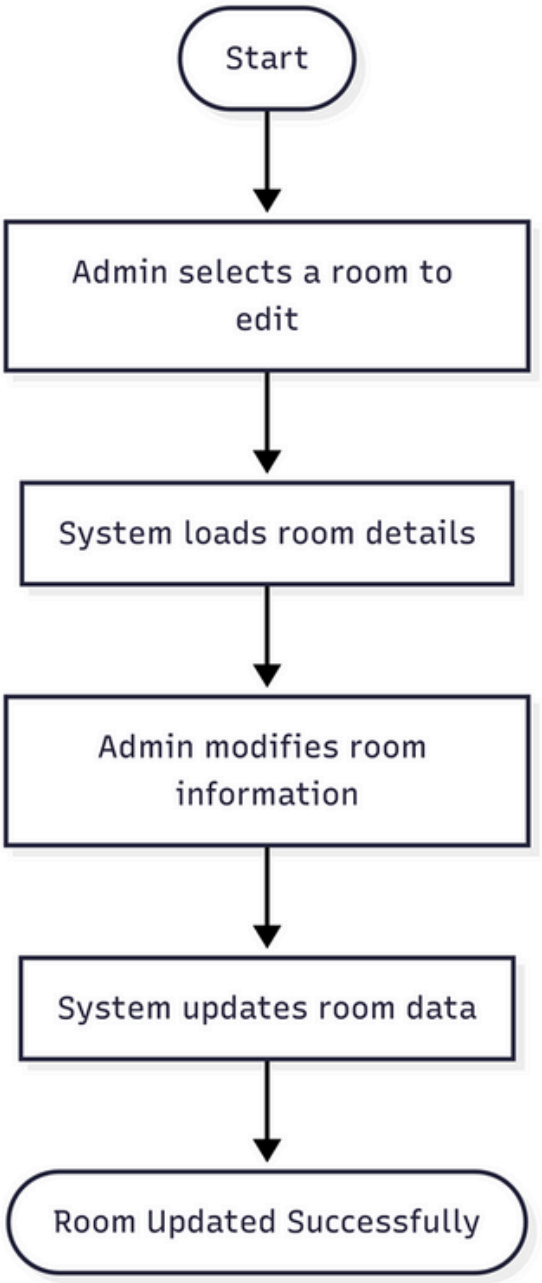
Activity Diagram

Edit event details



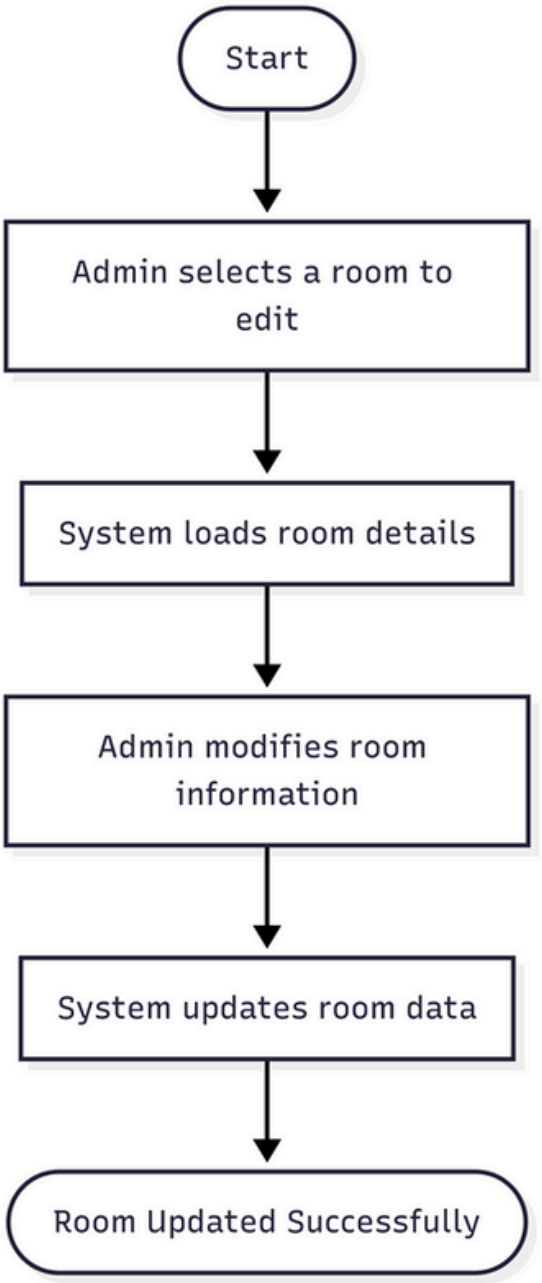
Activity Diagram

Update room data



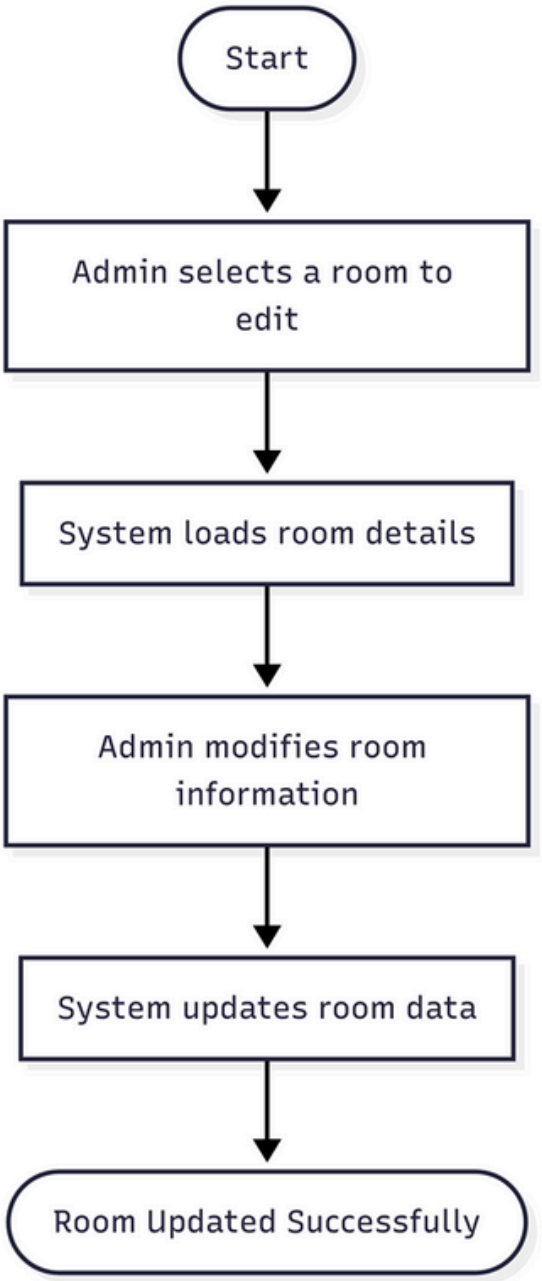
Activity Diagram

Edit tasks



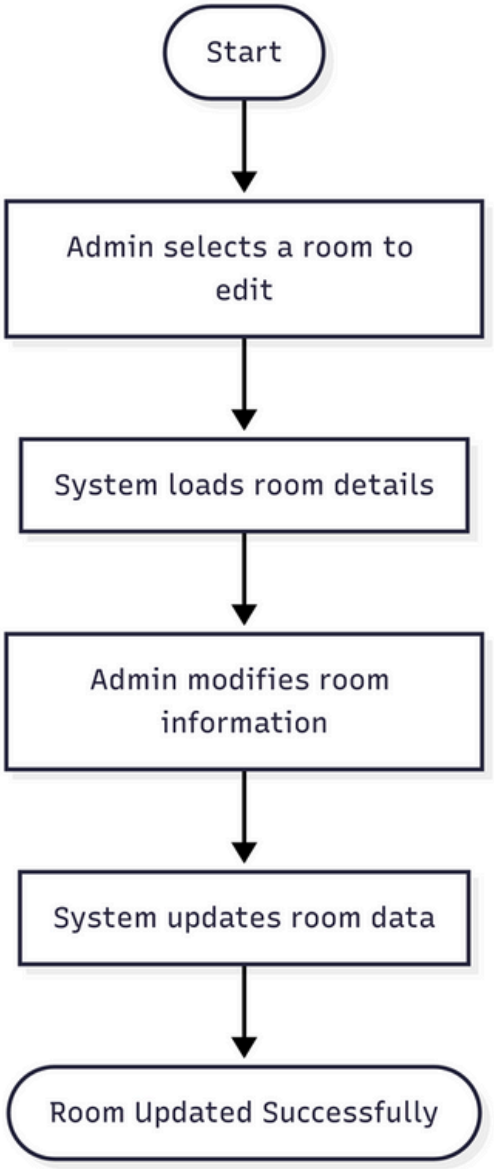
Activity Diagram

login



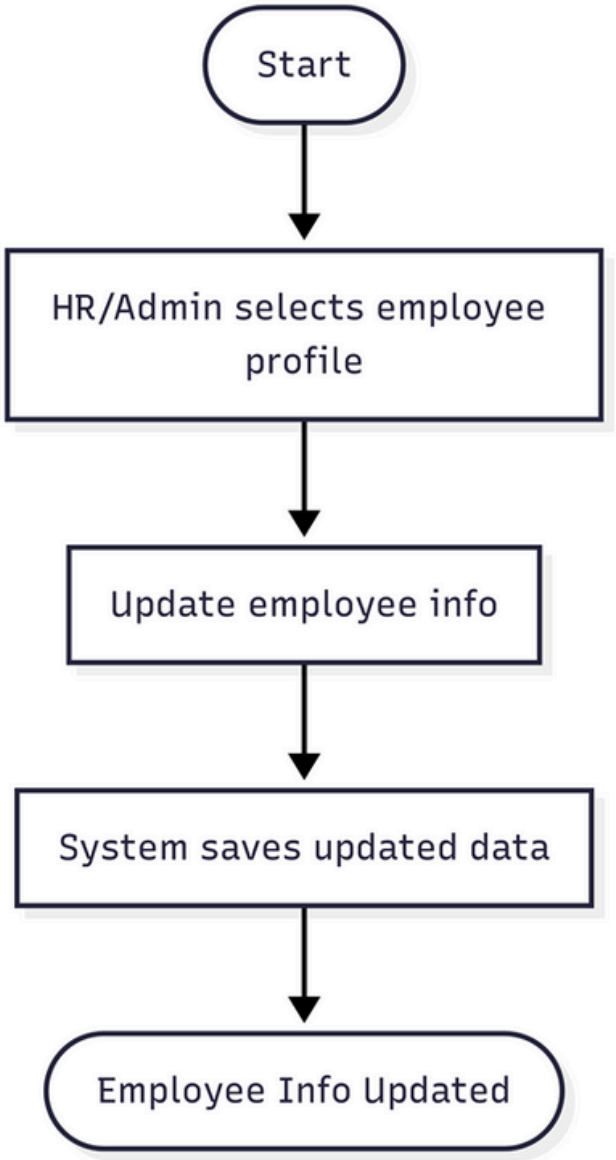
Activity Diagram

Display reports



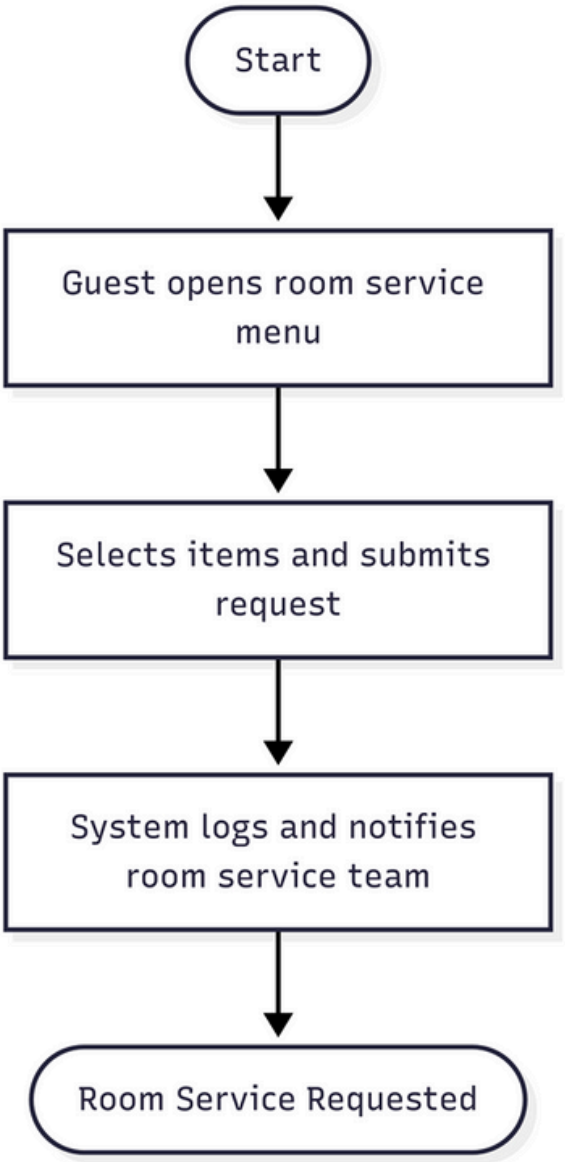
Activity Diagram

Edit Employee data

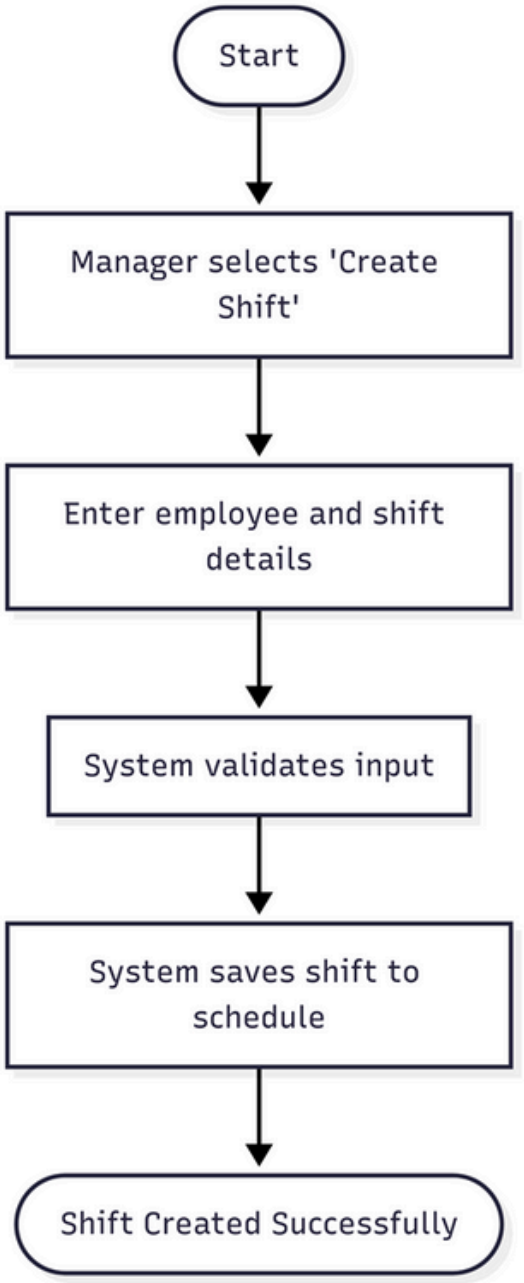


Activity Diagram

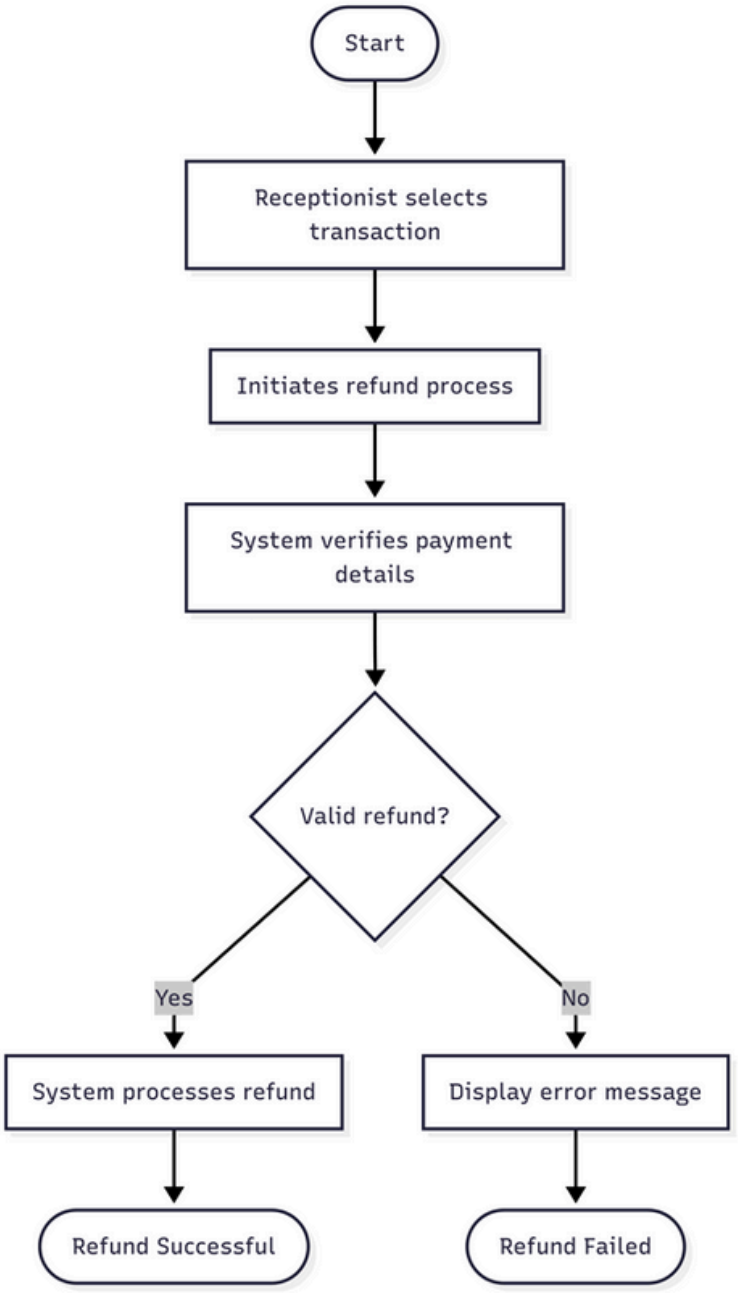
Refund



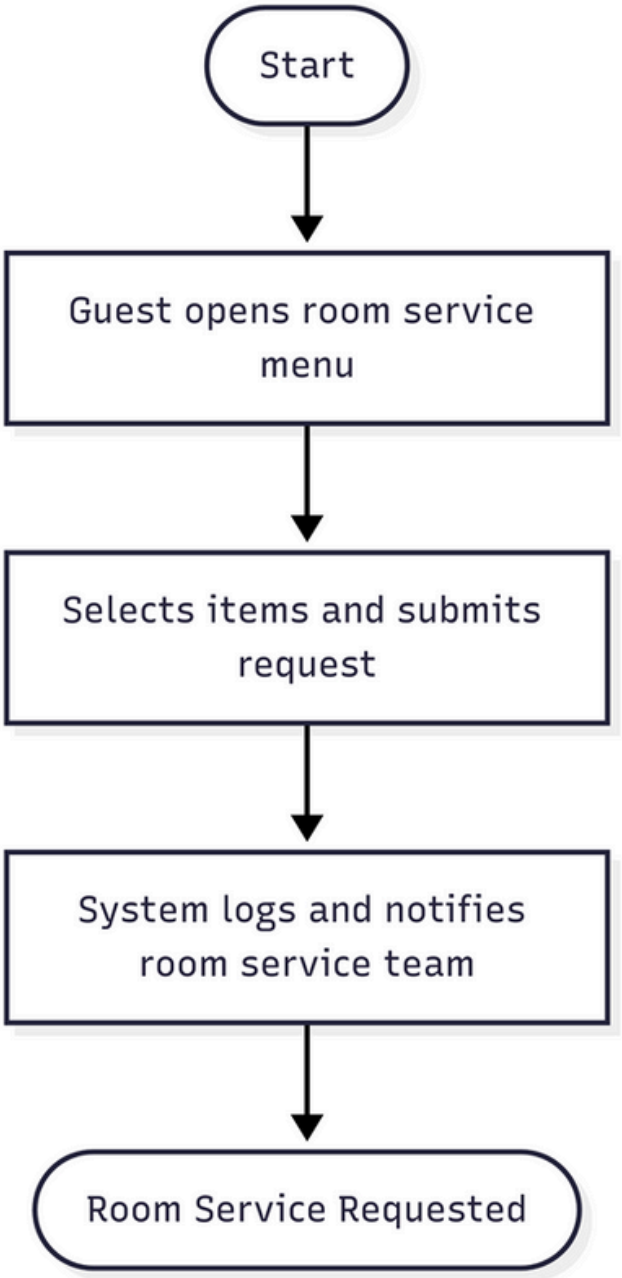
Create shift



Refund

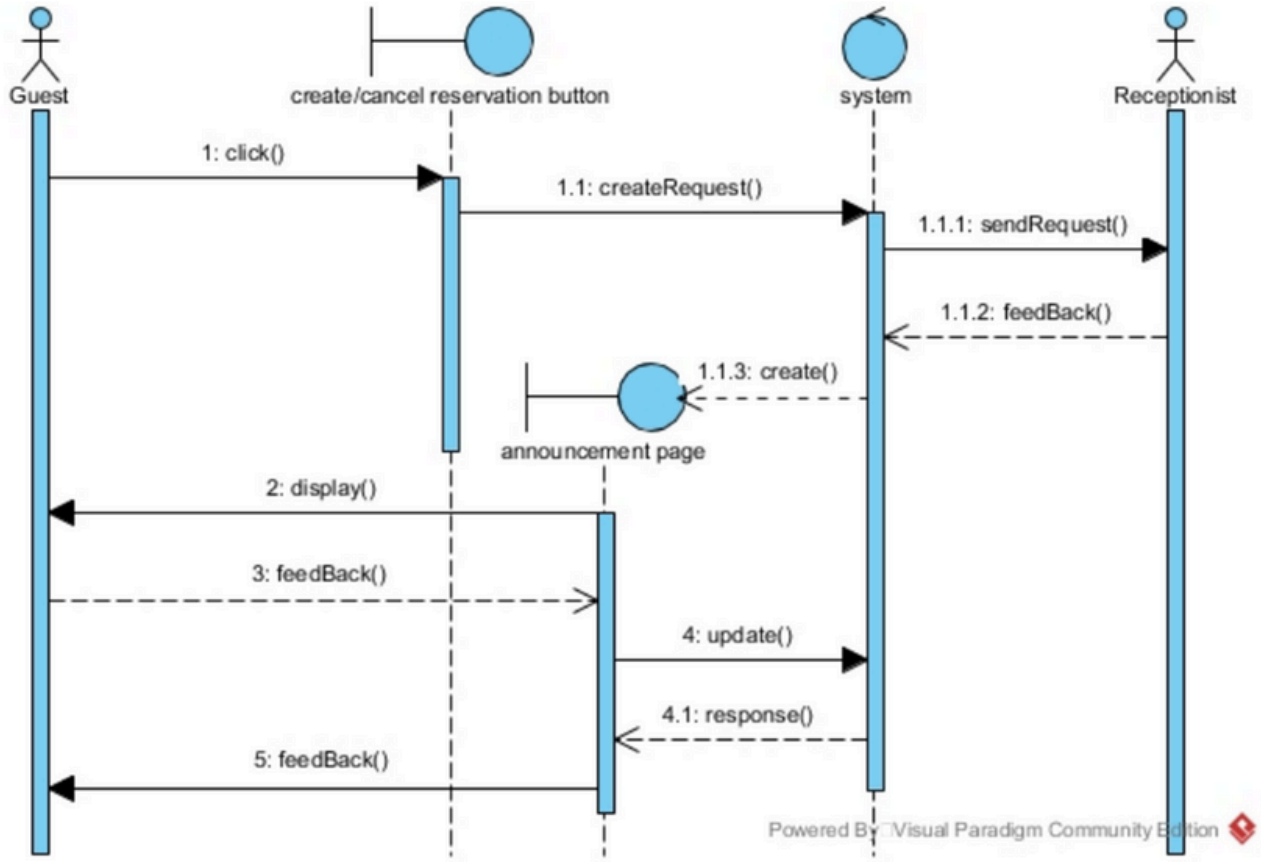

Start
Receptionist selects transaction
Initiates refund process
System verifies payment details
Valid refund?
Yes
No
System processes refund
Display error message
Refund Successful
Refund Failed

Request room service

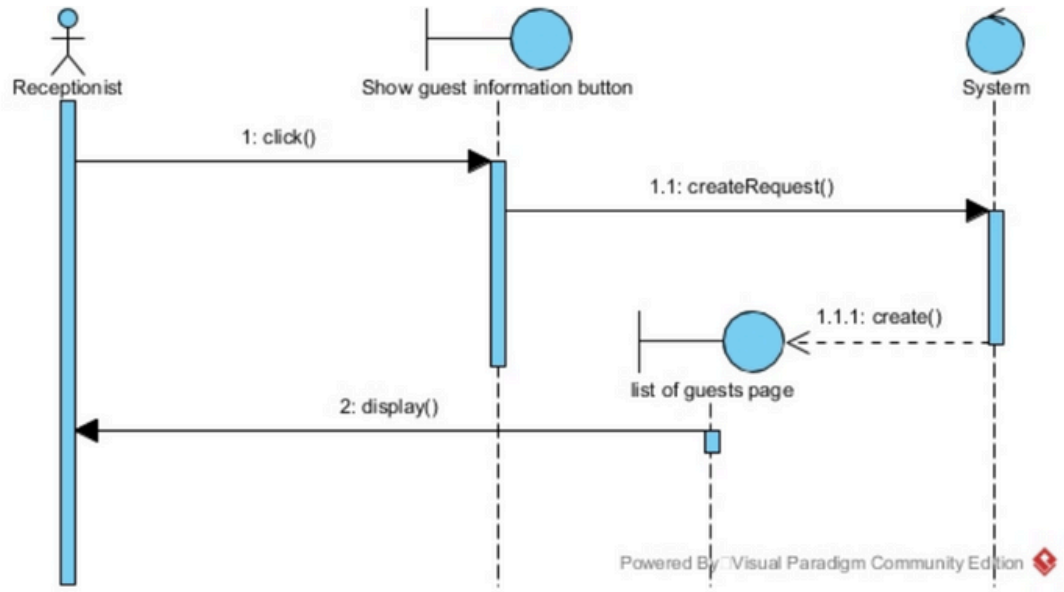


2.7 SEQUENCE DIAGRAMS

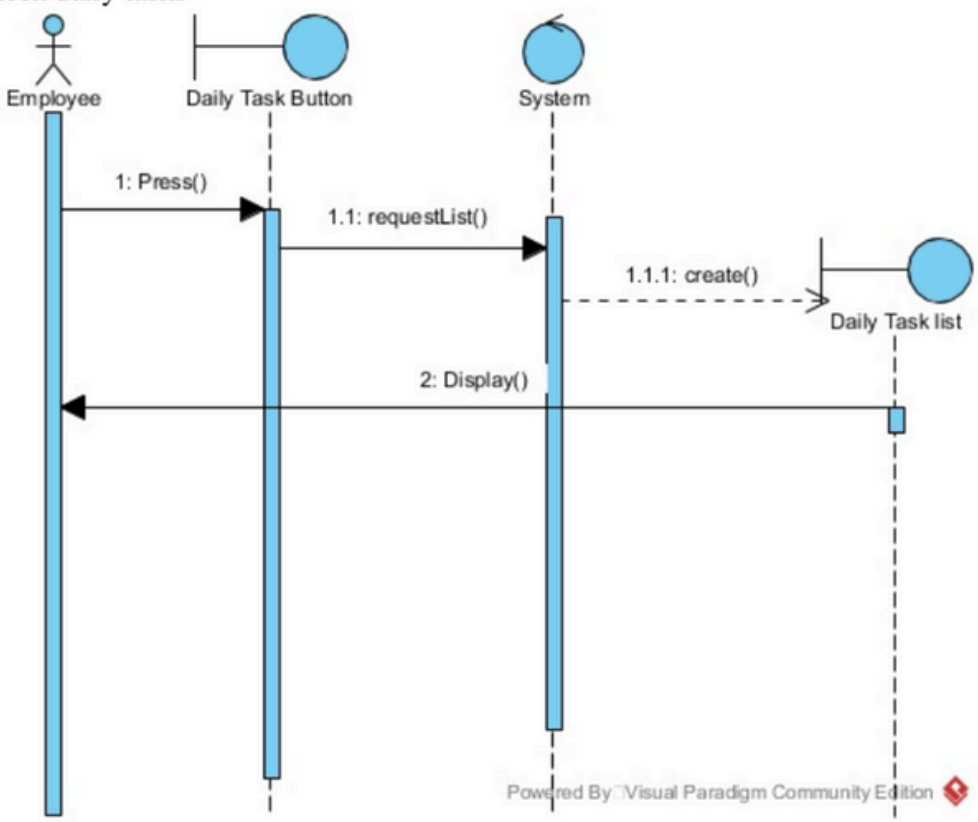
Book Room

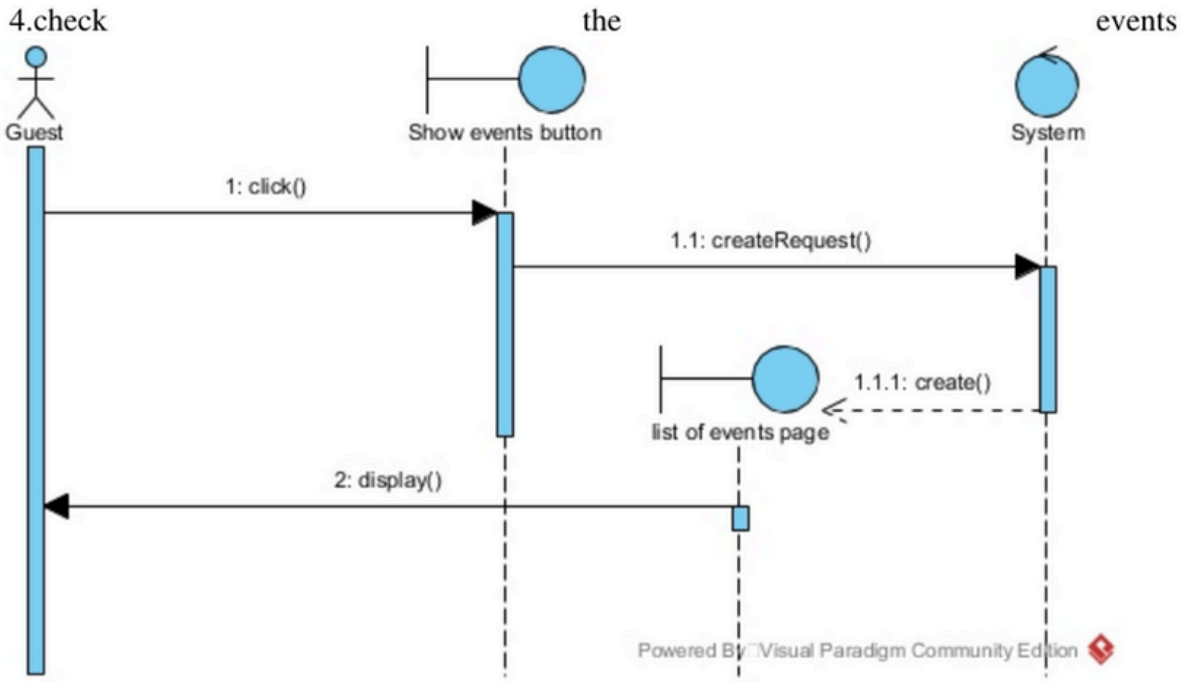


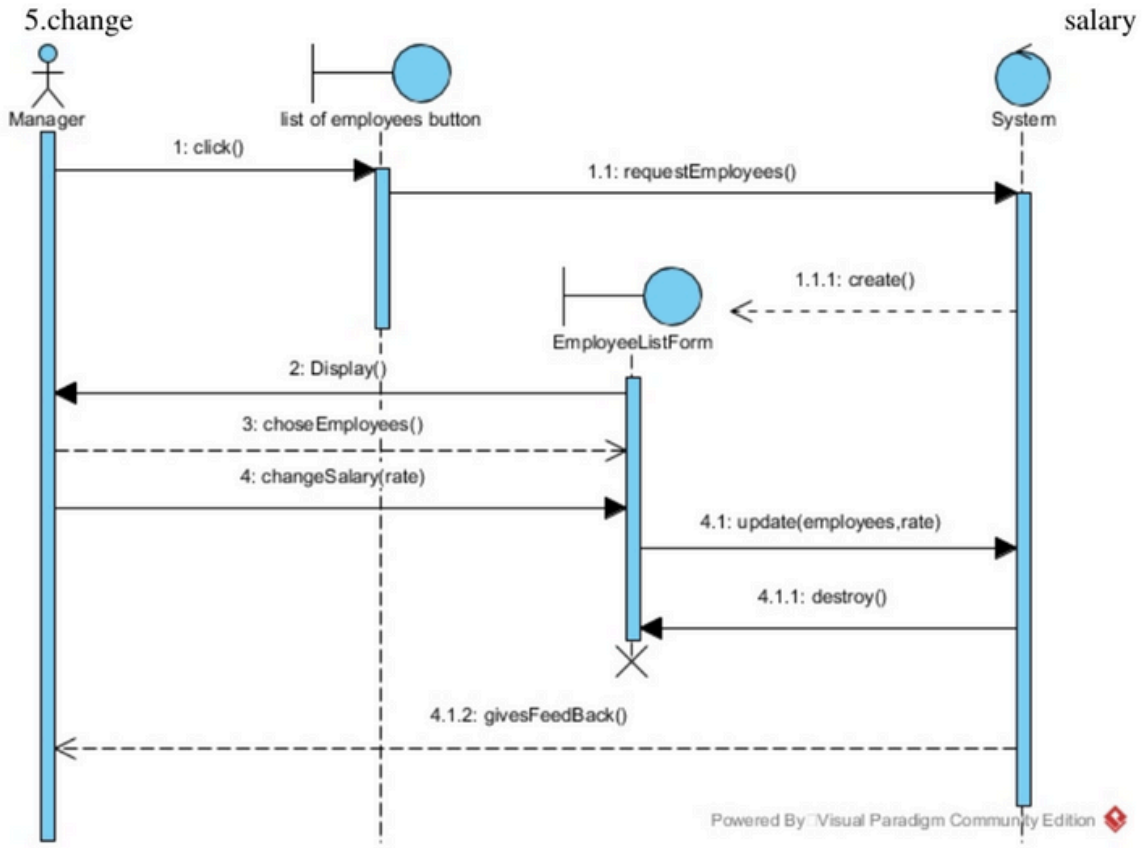
See guest info



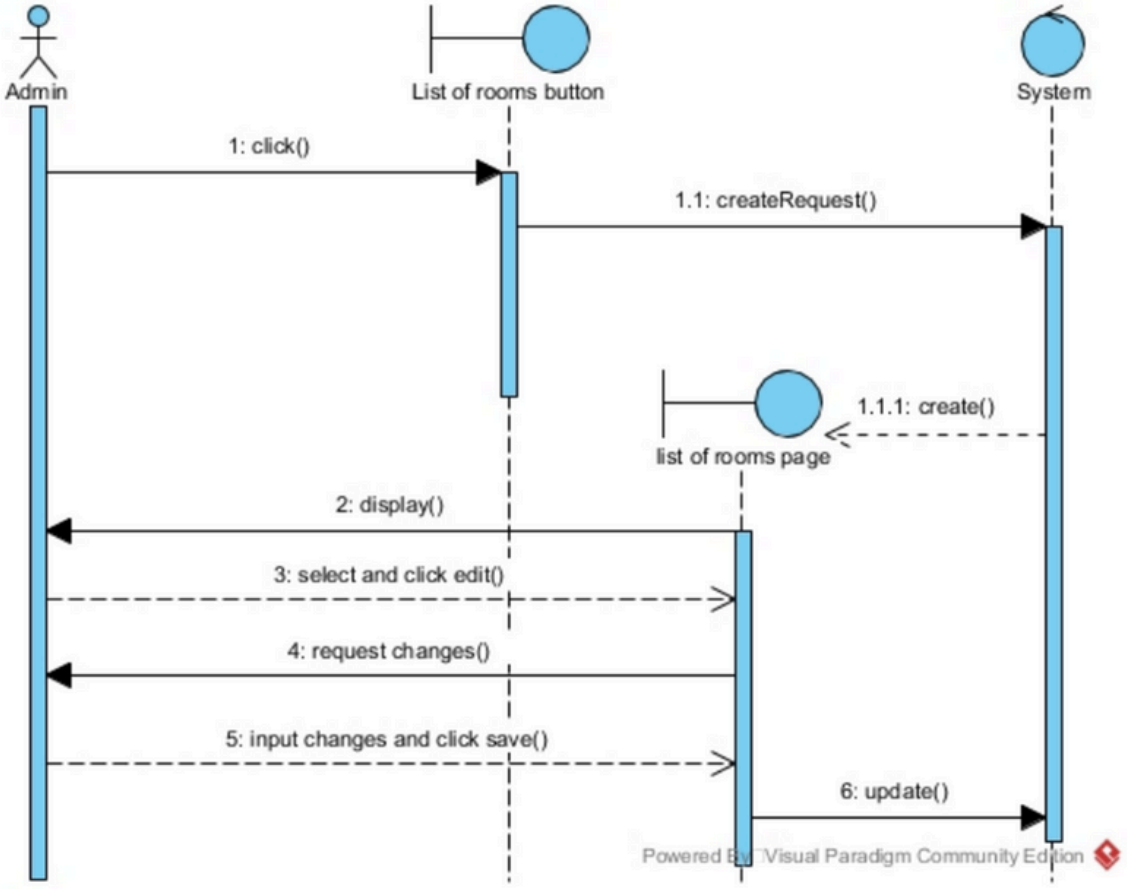
3.check daily tasks

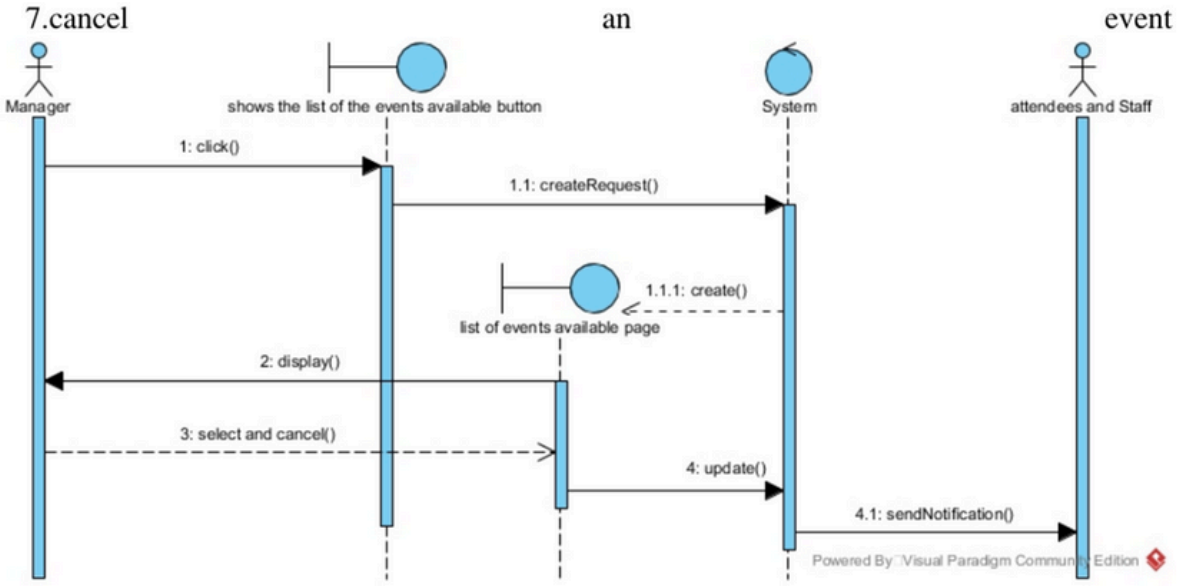



4.check
the
events
Guest
Show events button
System
1: click()
1.1: createRequest()
1.1.1: create()
list of events page
2: display()
Powered By Visual Paradigm Community Edition

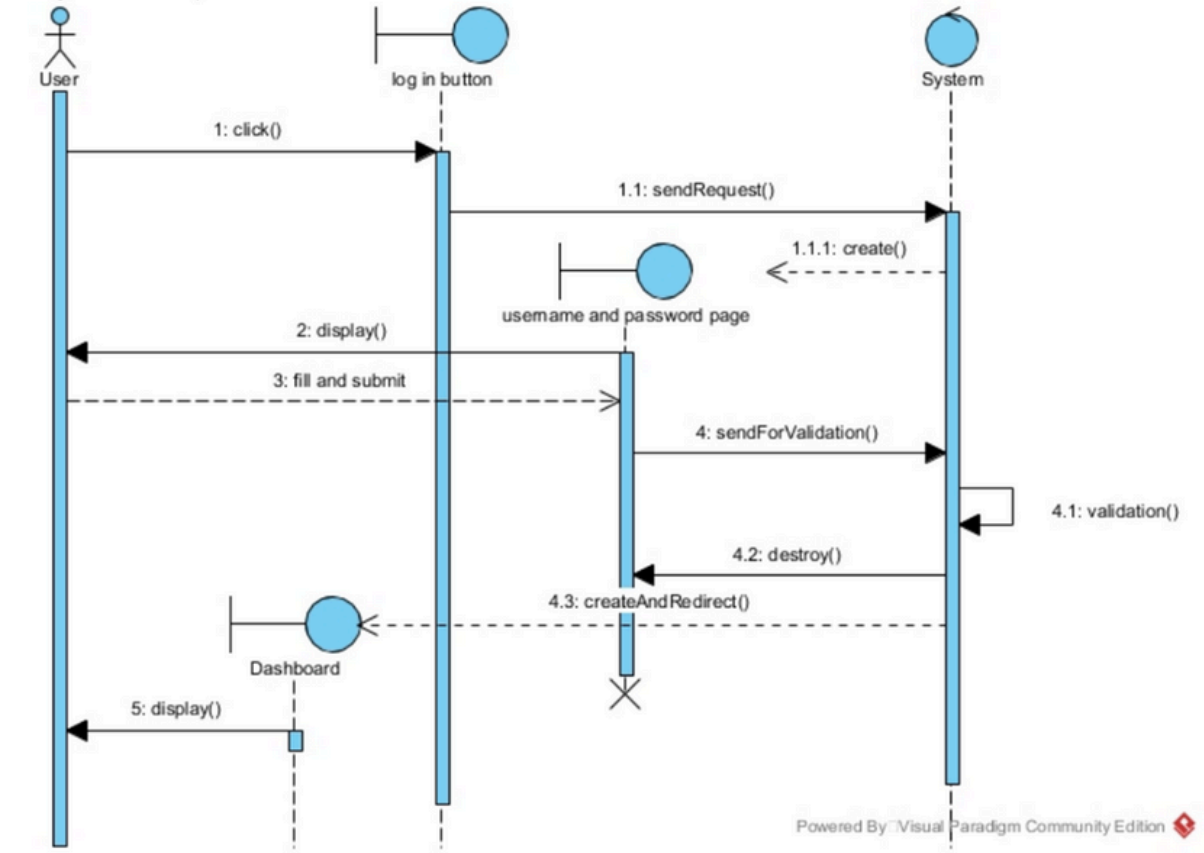

5.change
salary
Manager
list of employees button
System
1: click()
1.1: requestEmployees()
1.1.1: create()
EmployeeListForm
2: Display()
3: choseEmployees()
4: changeSalary(rate)
4.1: update(employees,rate)
4.1.1: destroy()
4.1.2: givesFeedBack()
Powered By Visual Paradigm Community Edition

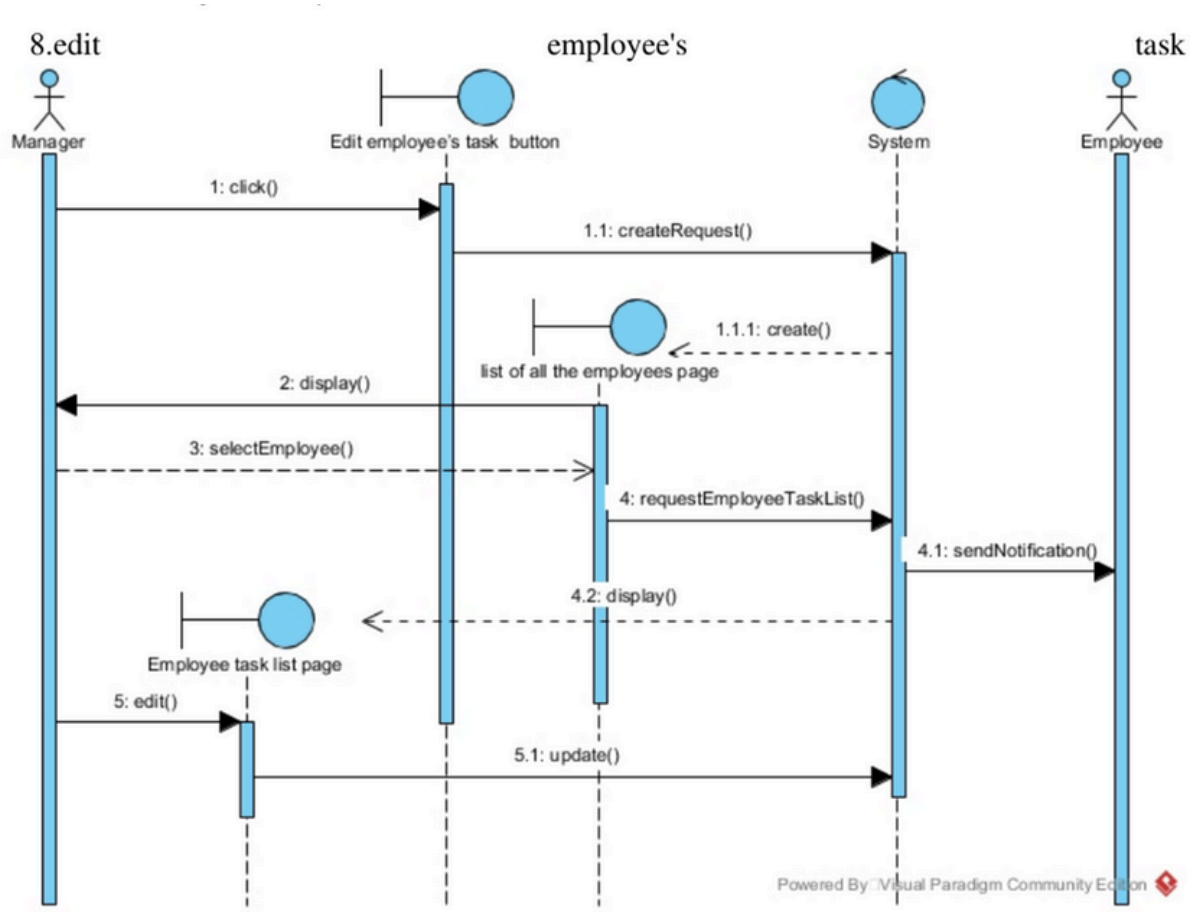
6.edit room details



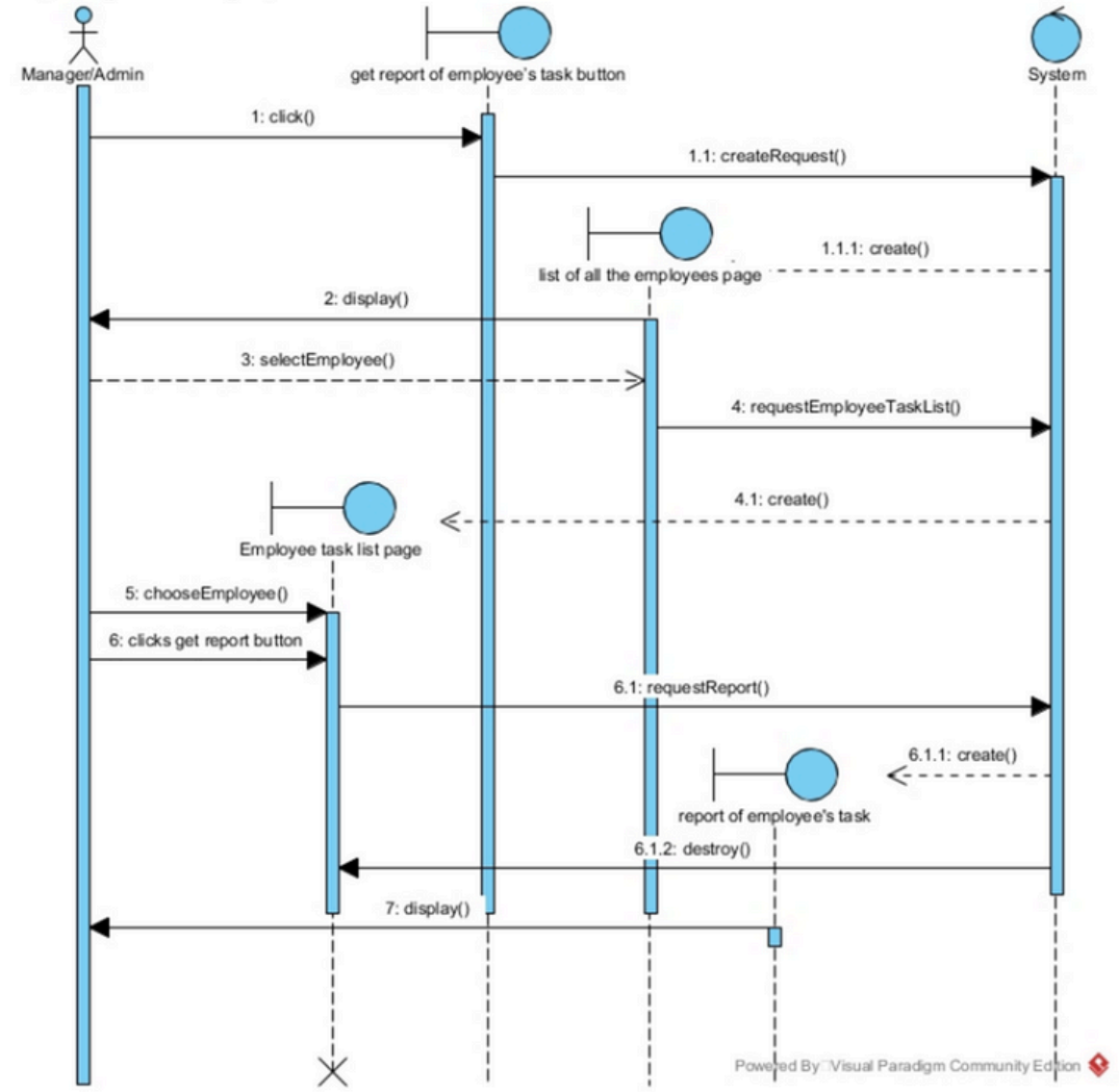

7.cancel an event
Manager
shows the list of the events available button
System
attendees and Staff
1: click()
1.1: createRequest()
1.1.1: create()
list of events available page
2: display()
3: select and cancel()
4: update()
4.1: sendNotification()
Powered By Visual Paradigm Community Edition

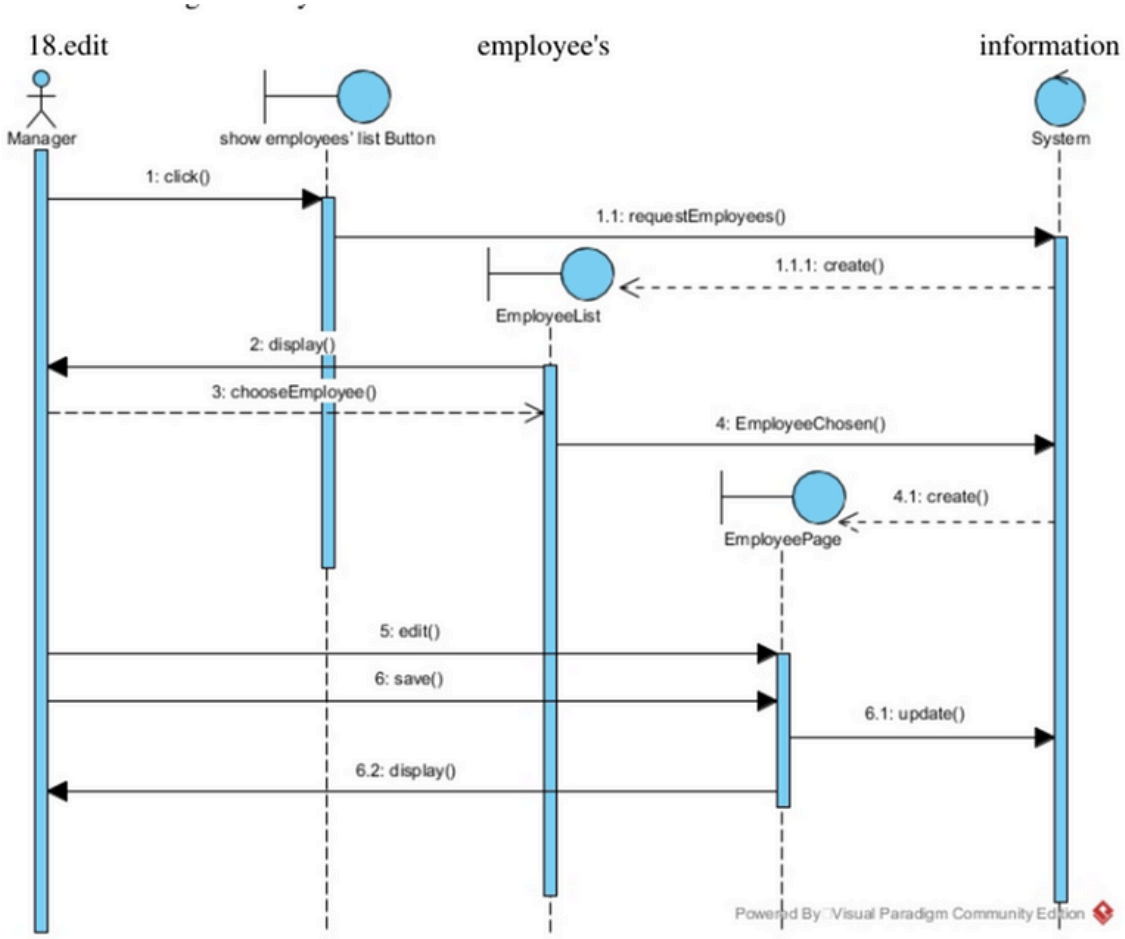
9.log into the system



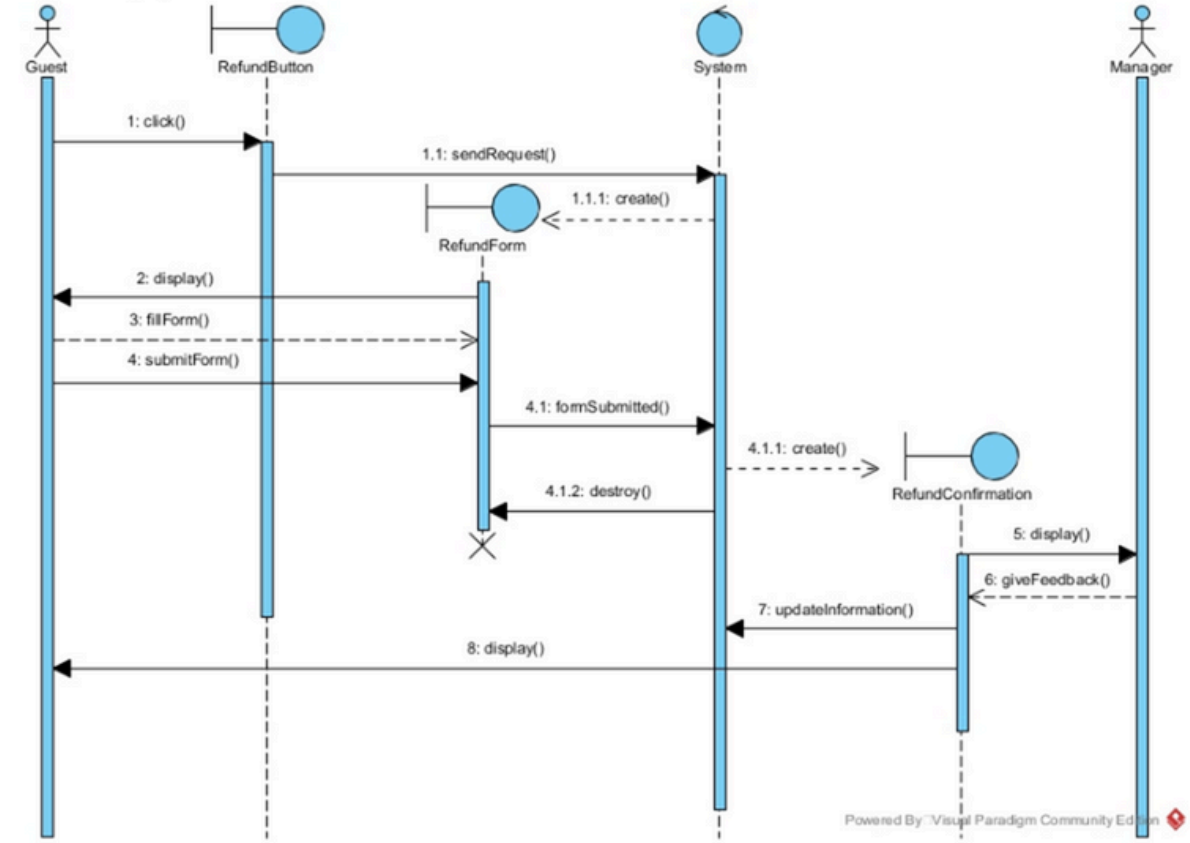

8.edit
employee's
task
Manager
Edit employee's task button
System
Employee
1: click()
1.1: createRequest()
1.1.1: create()
list of all the employees page
2: display()
3: selectEmployee()
4: requestEmployeeTaskList()
4.1: sendNotification()
4.2: display()
Employee task list page
5: edit()
5.1: update()
Powered By Visual Paradigm Community Edition

11.get report of employee's tasks

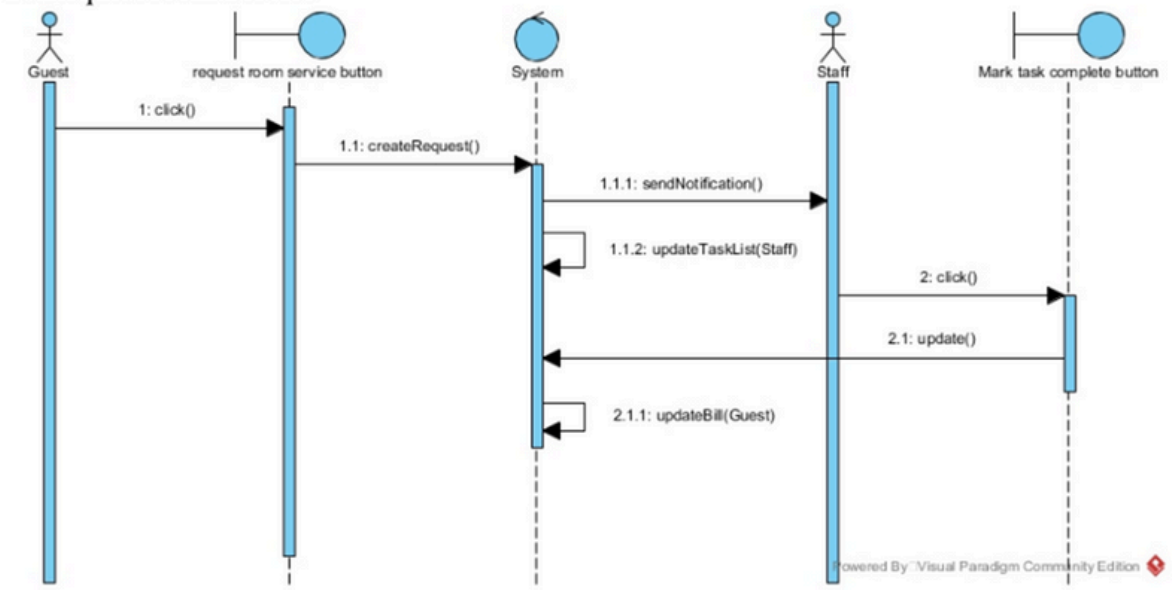



18.edit
employee's
information
Manager
show employees' list Button
System
1: click()
1.1: requestEmployees()
1.1.1: create()
EmployeeList
2: display()
3: chooseEmployee()
4: EmployeeChosen()
4.1: create()
EmployeePage
5: edit()
6: save()
6.1: update()
6.2: display()
Powered By Visual Paradigm Community Edition

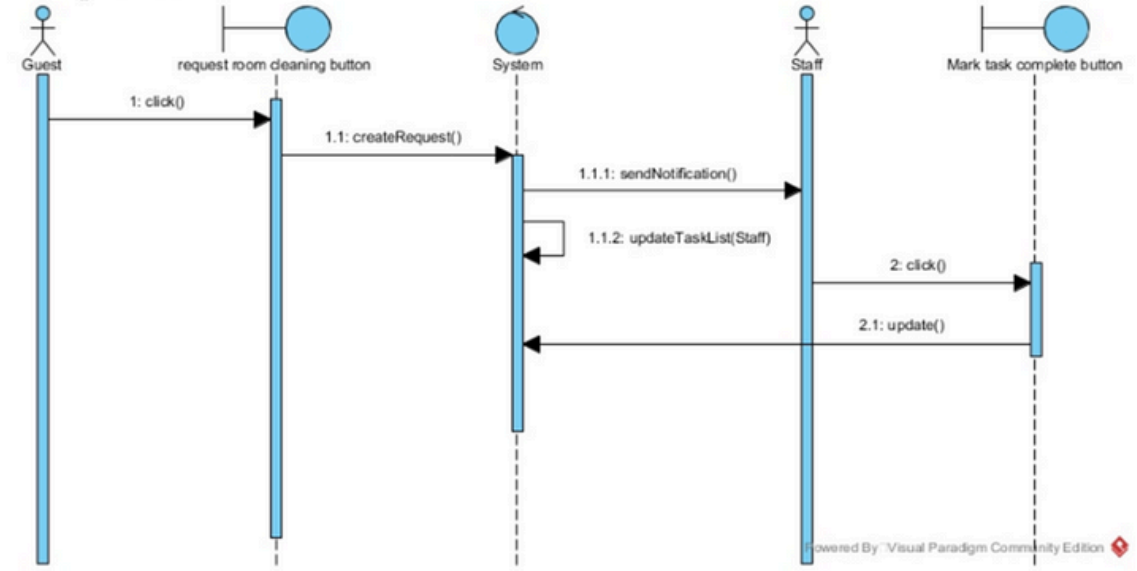
19.refund payment



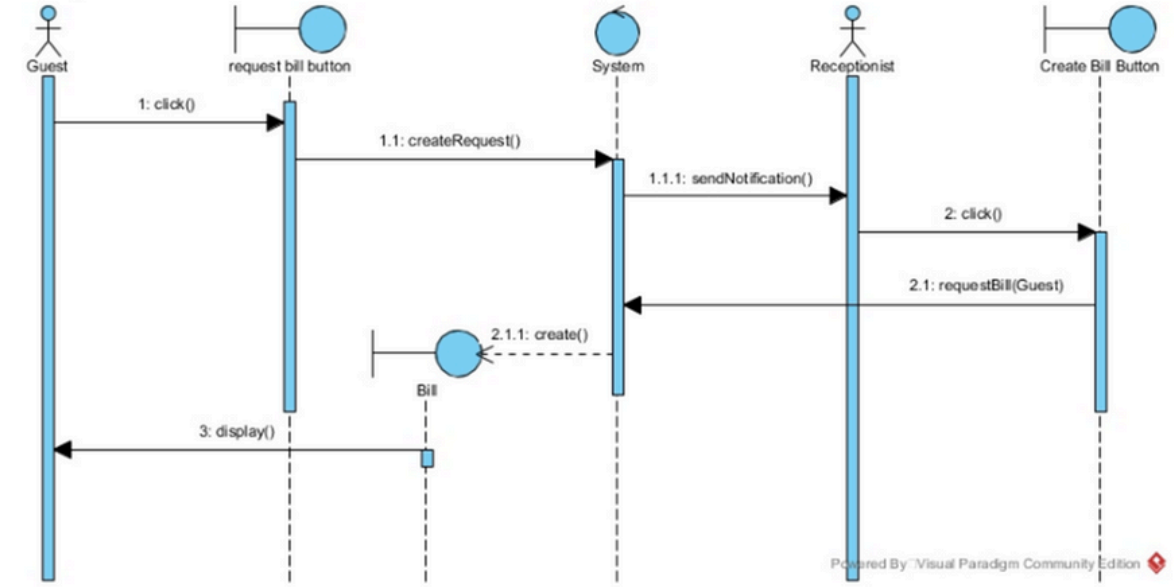
20.Request room Service



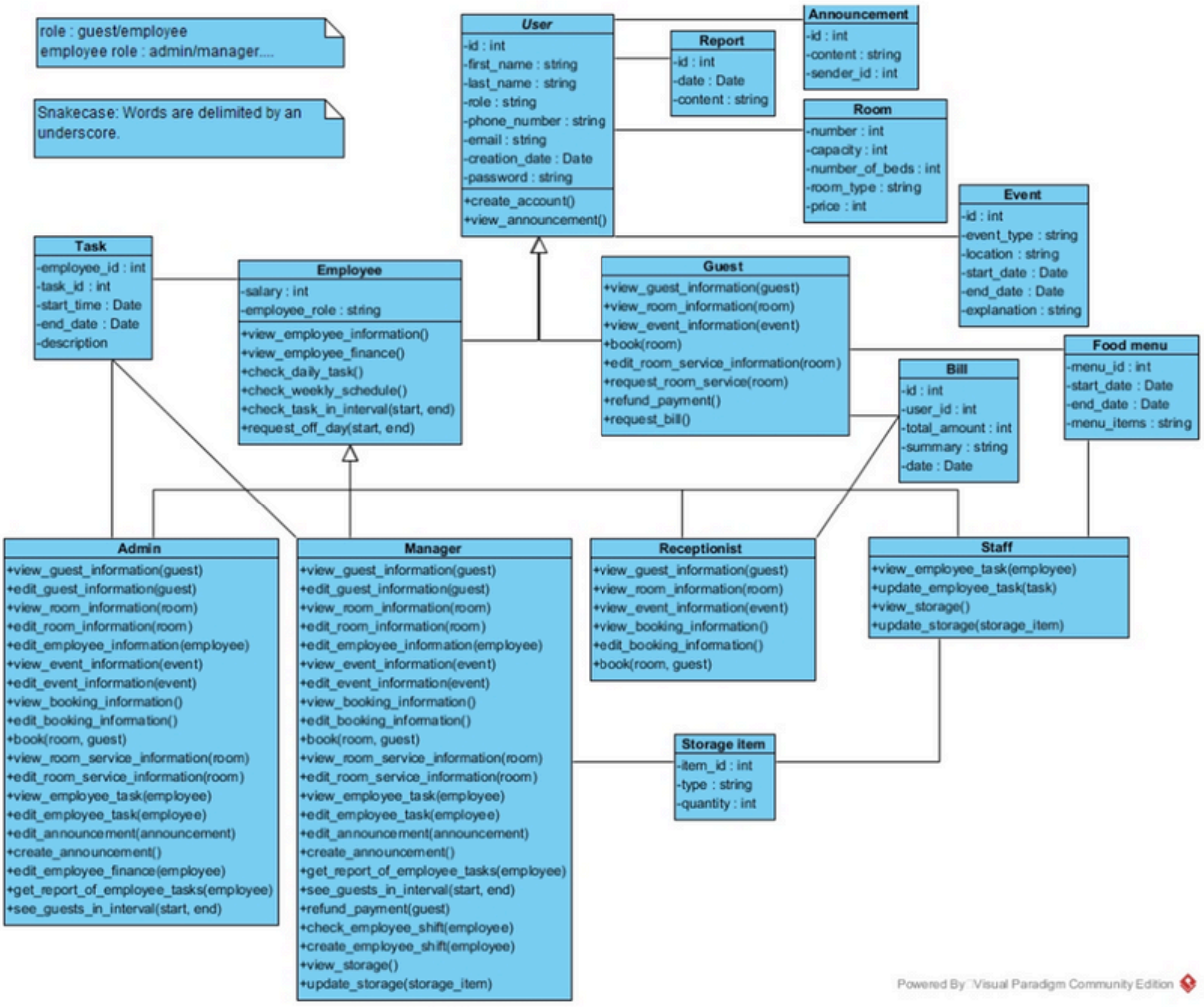
22.Request room cleaning



23.request bill



2.8 ER DIAGRAM



Chapter 3

Software Testing

3.1 Testing Features

Feature Area	Specific Functionality Tested
User Authentication	Creating user accounts, logging users in and out, verifying password validation, and checking role-based access control for admin and staff. Ensuring unauthorized users cannot access restricted pages.
Room Management	Adding new rooms, updating room details, deleting rooms, and verifying room availability status. Ensuring correct room type, price, and capacity are stored and displayed properly.
Reservation & Booking	Creating room reservations, checking room availability for selected dates, preventing double bookings, modifying booking details, and cancelling reservations.
Check-In & Check-Out	Verifying check-in process updates room status to Occupied, validating check-out updates room status to Available, and ensuring billing is generated correctly.
Billing & Payment	Calculating total bill based on stay duration and services, validating payment submission, generating payment receipts, and updating bill status to paid or unpaid.
Guest Management	Adding new guests, editing guest information, viewing history, and verifying correct data storage.
Staff/Admin Dashboard	Viewing system statistics, managing bookings, rooms, guests, and generating reports. Ensuring only authorized staff can access dashboard functions.
Report Generation	Generating occupancy, revenue, and booking reports. Ensuring correct filtering by date, month, or room type.
Notification & Alerts	Getting alerts for booking confirmations, cancellations, check-in reminders, and system updates. Ensuring alerts display correctly for admin and staff.

3.2 Testing Strategies

The testing strategies applied in order to ensure that Hotel Management System works properly were as follows, reliably, and securely.

1. Unit Testing

Individual tests were conducted in the form of unit tests on controllers, models, and utility functions.

This aided in the checking of room calculations, booking logic, date validation and user. authentication functions.

2. Integration Testing

Integration testing ensured the interactions of various modules

Examples include:

Room module + Booking module Booking module + Payment module. Authentication of users + Role-based access.

This guaranteed the free flow of data around the system.

3.3 System Testing

The whole system was put to test in its entirety to make sure that all required specifications are achieved.

System Test Specification (STS) 3.3.1.

Project ID: Smart Hotel Management System.
Prepared By: ZIAUL RASHID ELHAM
Date: 20/11/2025

1. Test Objectives

The primary objectives of testing Hotel Management System is to test that system is in operation. safe, user friendly and able to survive when put through the expected working conditions.The goals of the testing will be to identify defects, improve the quality of the system and guarantee.that all requirements were fulfilled before the deployment.

1.Test all the functionality that works such as the room management, the bookings, the check. in/check out, the billing and the guest operations.
2.Ensure that the modules interact in the correct manner and the information flows smoothly across the. system.
3.Authenticate role-based access, only authorized users are to be permitted to access.
Reliability of the check system, whether it does not crash and can work with an assortment. of operations
.**4.** Turn it into something to be useful and make the system user friendly to the hotel staff members.

2.Scope

In Scope

.User Authentication- login, log out, and role-based access.
Room Management- add, edit, delete room; room availability status.
Reservations / Booking- creation, update,
cancellation of bookings, avoiding duplication of booking.
Check-In / Check-Out- room status, the guest assignment, billing creation. Billing & Payments – calculation of prices, development of the invoice, payment update. Guest Management - add, edit, delete guest details. Reports- occupancy, revenue and booking reports.

3. Test Environment

Category	Details
Operating System	Windows 10 / Windows 11 / Ubuntu 22.04
Web Server	Apache (XAMPP) / Laravel Built-in Server
Programming Language	PHP 8.x (Laravel Framework)
Framework	Laravel 8/9 (based on project repo)
Database	MySQL 5.7 / MySQL 8.0
Browser for Testing	Google Chrome, Mozilla Firefox, Microsoft Edge
Hardware Requirements	Minimum 4 GB RAM, 2 GHz Processor, 10 GB Storage

4. Test Data

4.1 Sample User Accounts

User Type	Email (Username)	Password	Notes
Admin	admin@mail.com	123456	Used for full management and administrative access.
User	user@mail.com	123456	Used for standard user testing.

5. Pass/ Fail Criteria

Condition	Pass Criteria	Fail Criteria
Functional Test	Output matches expected result exactly	Output deviates from the expected result
Non-Functional Test	Meets performance/security benchmarks	Fails benchmarks or causes errors
Negative Test	System rejects invalid input or unauthorized actions correctly	System accepts invalid input or grants unauthorized access
Reliability Test	System recovers from error conditions	System crashes or becomes unresponsive

6. Detailed Test Cases

1. User Registration Scenarios

Test Case ID	Scenario Type	Test Steps	Expected Result
TC-AUTH-001	Positive (Successful Registration)	1. Go to the registration page. 2. Enter necessary information. 3. Submit.	Email delivered, confirmation displayed, account setup.
TC-AUTH-002	Negative (Invalid Email)	1. Enter an incorrect email format. 2. Send it in.	Error: "Invalid email address" is the error.

2. User Login Scenarios

Test Case ID	Scenario Type	Pre-Condition	Test Steps	Expected Result
TC-AUTH-003	Positive (Successful Login)	There is a legitimate email account.	1. Put in your legitimate email address. 2. Login.	Redirect to correct dashboard.
TC-AUTH-004	Negative (Wrong Password)	Valid account exists.	1. You entered the incorrect password. 2. Submit.	"Invalid credentials."
TC-AUTH-005	Connectivity Issue	Internet disconnected.	1. Enter valid credentials. 2. Submit.	Error: "No internet connection."

3. Role-Based Access Scenarios

Test Case ID	Feature Area	User Role	Scenario Type	Test Steps	Expected Result
TC-AUTH-006	Role-Based Access	Student	Unauthorized	1. Access admin dashboard.	Error: "Unauthorized Access."
TC-AUTH-007	Role-Based Access	Organizer	Unauthorized	1. Access system settings.	Error: "Unauthorized"

4. Session Management Scenario

Test Case ID	Feature Area	User Role	Scenario Type	Pre-Condition	Test Steps	Expected Result
TC-AUTH-008	Session Timeout	All	Negative	Logged in for a session limit	1. Stay inactive for 10 min. 2. Try an action.	Forced logout, redirect to login page.

1. Event & Club Enrollment (TC-STU-001 to TC-STU-007)

These cases test the viewing, enrollment, and cancellation of events by a Student user.

Test Case ID	Feature Area	Scenario Type	Pre-Conditions	Test Steps	Expected Result
TC-STU-001	View Events	Positive	Logged in	1. Navigate to "Events."	All active events displayed with details
TC-STU-002	View Events	System Error	DB down	1. Navigate to "Events."	Error: "Unable to load events."
TC-STU-003	Enroll in Event	Positive	Logged in, event open for enrollment	1. Select event. 2. Click "Join."	Enrollment successful.
TC-STU-004	Enroll in Event	Negative – Already Enrolled	Already enrolled	1. Click "Join" again.	Error: "Already enrolled."
TC-STU-005	Enroll in Event	Connectivity Issue	Internet disconnected	1. Click "Join."	Error: "No internet connection."
TC-STU-006	Cancel Enrollment	Positive	Enrolled in event	1. Click "Cancel."	Enrollment removed successfully
TC-STU-007	Cancel Enrollment	Negative – Not Enrolled	Not enrolled	1. Click "Cancel."	Error: "You are not enrolled."

Test Case ID	Feature Area	Scenario Type	Pre-Conditions	Test Steps	Expected Result
TC-STU-008	View Notifications	Positive	Logged in	1. Open Notifications.	All recent notifications shown.
TC-STU-009	View Available Clubs	Positive	Logged in	1. Login as Student. 2. Open clubs list.	List of clubs displayed.
TC-STU-010	Enroll in Club – Valid	Positive	Logged in	1. Select club. 2. Click "Enroll." 3. Confirm enrollment.	Confirmation message displayed, status change
TC-STU-011	Enroll in Club	Negative – Already Enrolled	Already enrolled	1. Enroll in club again.	Message: "Already enrolled."

1. Admin - Event Approval & Management

Test Case ID	Feature Area	Scenario Type	Pre-Conditions	Test Steps	Expected Result
TC-ADM-001	Approve Event	Positive	Event pending	1. Open "Pending Events." 2. Click "Approve."	Event approved, organizer notified.
TC-ADM-002	Approve Event	Negative – Already Approved	Event already approved	1. Approve again.	Error: "Event already approved."
TC-ADM-003	Reject Event	Positive	Event pending	1. Open event. 2. Click "Reject" with feedback sent.	Event rejected, feedback sent.
TC-ADM-004	Reject Event	Negative – No Reason Given	Event pending	1. Click "Reject" without reason.	Error: "Reason is required."

B. Monitoring and Scheduling Scenarios (TC-ADM-005 to TC-ADM-008)

Test Case ID	Feature Area	Scenario Type	Pre-Conditions	Test Steps	Expected Result
TC-ADM-005	View Venue Schedule	Positive	Logged in	1. Open "Venue Schedule."	All bookings and free slots shown.
TC-ADM-006	Monitor Clubs	Positive	Logged in	1. Open "Club Management."	Club list & stats displayed.
TC-ADM-007	View Venue Schedule	System Error	DB down	1. Open "Venue Schedule."	Error: "Unable to load schedule."
TC-ADM-008	Monitor Clubs	Connectivity Issue	Internet disconnected	1. Open "Club Management."	Error: "No internet connection."

Chapter 4

Deployment and Maintenance

Hotel Management System: Implementation and Support Plan.

1.Deployment Strategy

1.1 Web Application

- Platform: Vercel/Railway/Netlify
- Technology Stack HTML, CSS, Bootstrap, Backend: DRF, Postgres
- Deployment Steps:

1. Build Optimization

Minimize the static files and activate the compression of Gzip/Brotli through Nginx. Index database queries and select-related

2. Testing & QA

1.Both Unit and UI tests on a variety of screen sizes and devices should be used.

2.To ensure that all the existing functionality is not broken, do some.

regression testing.

3. Deployment Credentials & Environment Security

- Create and store secrets about the environment
- making sure that no confidential information is revealed in logs or code base.
- Sign and encrypt all web with HTTPS and SSL certificates (ex: Let's Encrypt). traffic.
- Limit entry of production servers through SSH keys as well as firewall regulations.

4. Web Application Deployment

- Push the production to a safe cloud solution (e.g. Railway, Render or VPS)
- Incorporate branded items: favicon, logo, feature banners, and meta descriptions to be used in the SEO.
- Have a privacy policy and terms of service page with a face to it.
- Closed beta testing, internal QA Before going live, complete user role (Admin,) testing. Agent, Merchant, Customer)
- Versioned CI/CD Secrets Secret key (SECRET KEY) DB credentials. vault (e.g., GitHub Secrets

5. Post-Deployment Verification

Measure the performance and analytics, uptime; verify the backups and error. tracking are active.

2. Maintenance Plan

2.1 Routine Monitoring

- Tracking errors, performance, and logs with the help of such tools as Sentry, Prometheus, or New.
- Relic.
- Monitoring the health of the API and analytics user activity, using uptime monitoring. Performan of databases reviewed regularly.

2.2 Scheduled Maintenance

Monthly:

Complete full regression and security test.
Following performance measures and logs of errors.
Clear out unneeded information and records, temp files.
Address customer support and feedback.

Quarterly:

Carry out security tests and penetration testing.
Maximize database query and indexing.
Record past transaction and activity records.
Revise/rework privacy policy and terms of service.
CSS compatibility with new browser and operating systems.
Assess infrastructure usage and reassign resources (where necessary).

2.3 Incident Management

Bug Fix Protocol:

Recreate the bug in a staging environment and find the cause of the bug.
Test the fix internally, apply and deploy to the production after it is confirmed.
Record the resolution and observe after deployment to see whether there is a recurrence or not.

Downtime Protocol:

- Show a maintenance page and alert banner when there is a downtime and recovery should be monitored. through uptime tools.

2.4 User Feedback Loop

- Provide the consumers with the opportunity to leave comments in the site.
- Respond to important comments within a period of 48 hours.

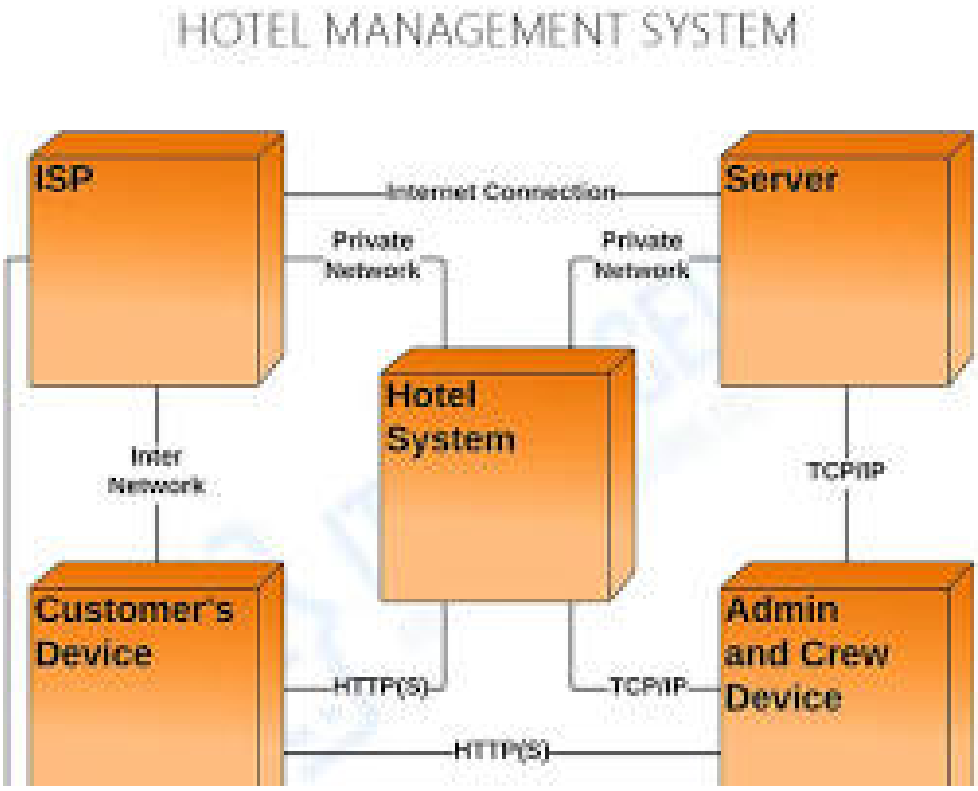
2.5 CI/CD Automation

In order to allow for quicker development and deployment processes the CI/CD tools will use GitHub Actions or Bitrise to automate the testing of code every time a change has been made through committing to the repository or creating a pull request. This will allow for quick detection of problems with changes to the code when it is run by comparing the new code with the older code and making sure only stable code is committed to the repository and the proper branch. Once the code has been approved through a pull request and merged into the main branch the application will automatically be added to either staging or production. Environment variables will also be securely managed so that sensitive data (e.g. API keys or database credentials) are kept safe during these processes. The CI/CD pipeline will also monitor the success/failure of each build in order to help you know if something fails to build and stop the process, and if needed roll back to the previously successful build. This means that your system can be kept fully operational with minimal downtime for your end users.

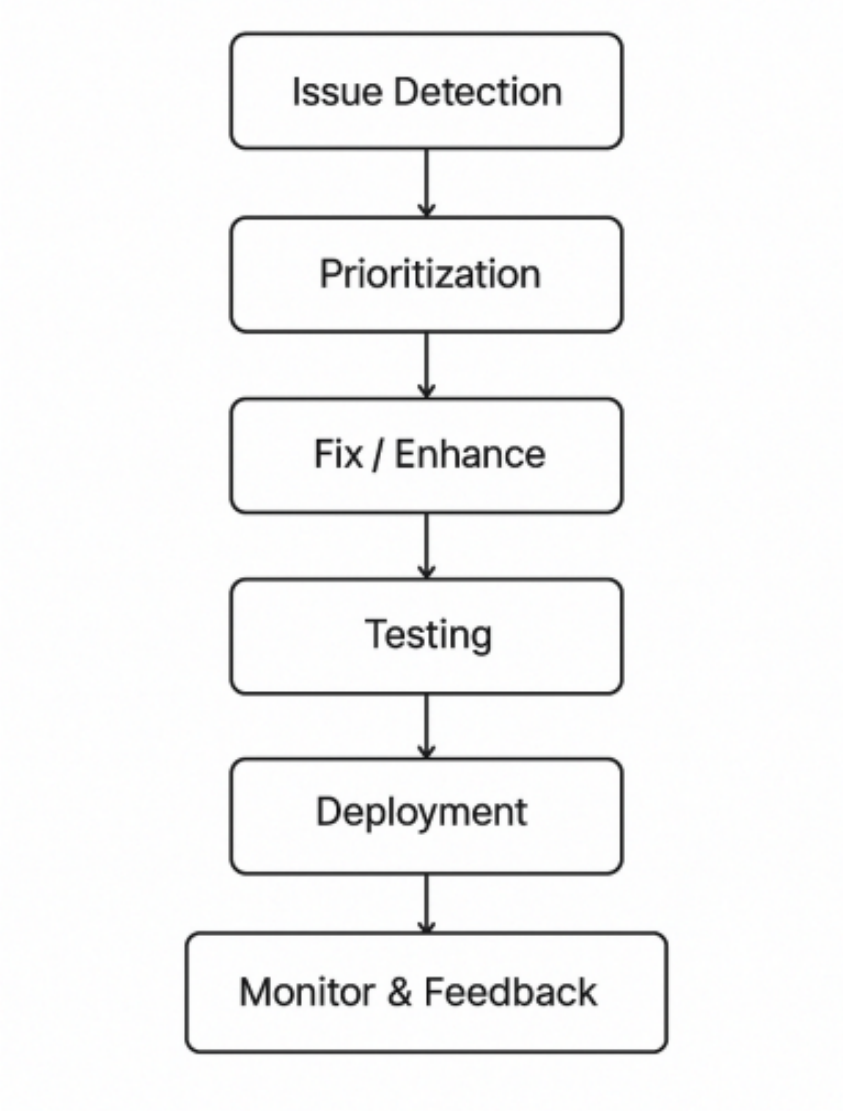
⁕ 3. Versioning and Release Policy

The project will be using Semantic Versioning (https://semver.org/) and the versions will follow the format of Major.Minor.Patch. An illustration would be: 2.1.0 Major Version - this is increased for major changes where backwards compatibility is not possible. Typically major releases will have new major pieces of functionality added as well as dramatic changes in the system structure which may or may not affect the operation of the system. Minor Version - the use of the Minor version is for adding functionality that does not break an existing system. Minor Version will increase as functionality is added to an existing system but remain compatible with earlier versions. Patch Version - Patch Version will be used for small incremental enhancements to performance, bug fixes, and minor stability enhancements to a system where the behaviour of the system does not change. Frequency of Releases • Patch and

3. Deployment Flow Diagram



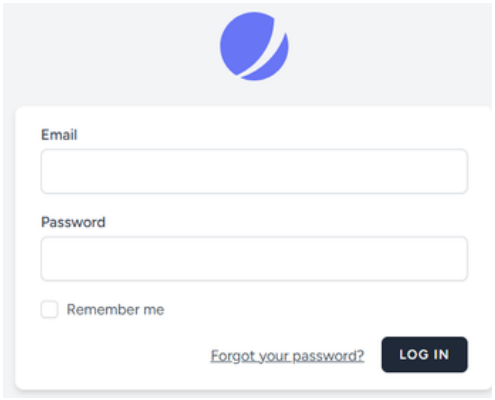
4. Maintenance Workflow Diagram



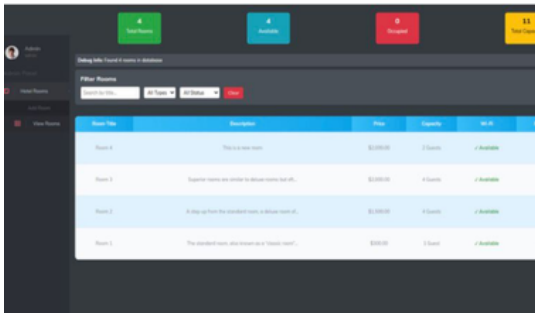
Chapter 5

User Manual

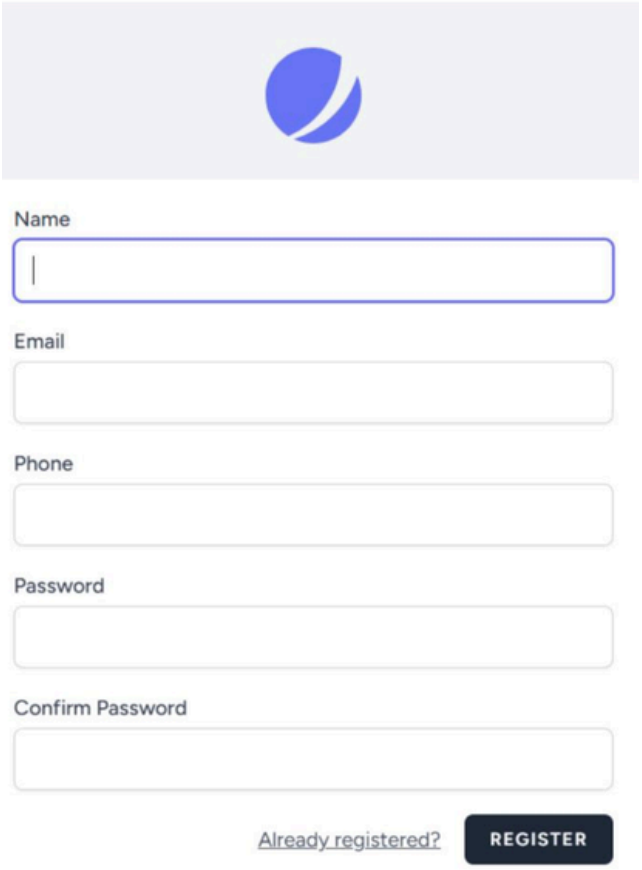
Login



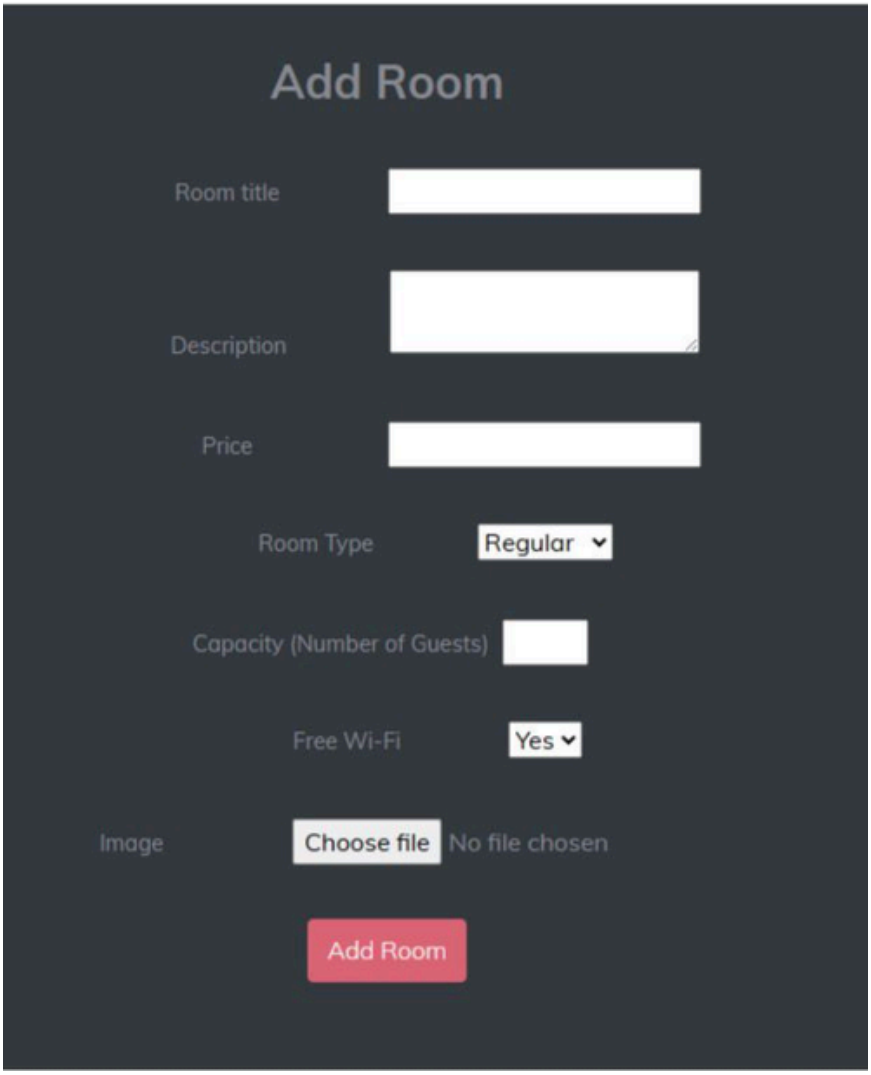
Admin View Room



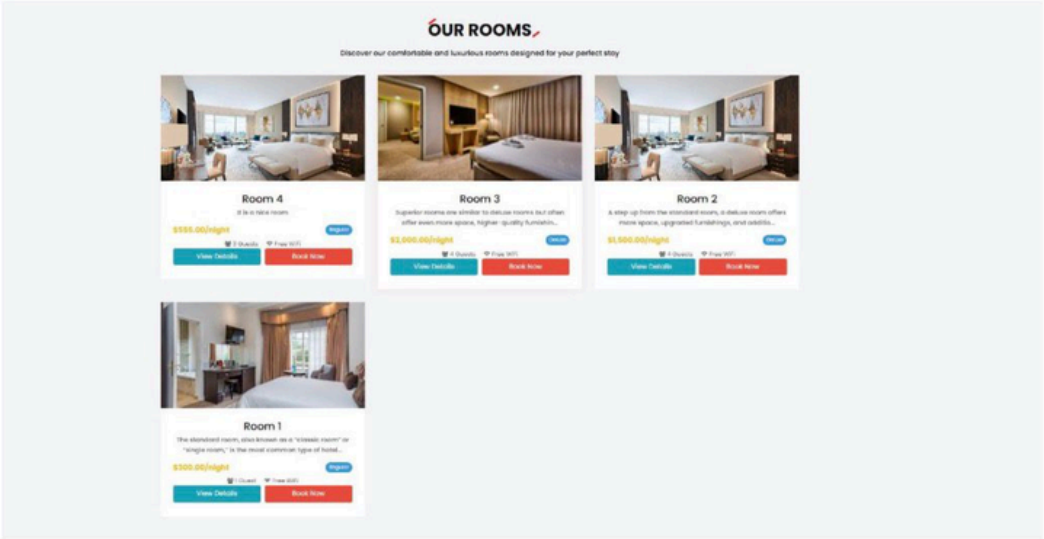
Registration



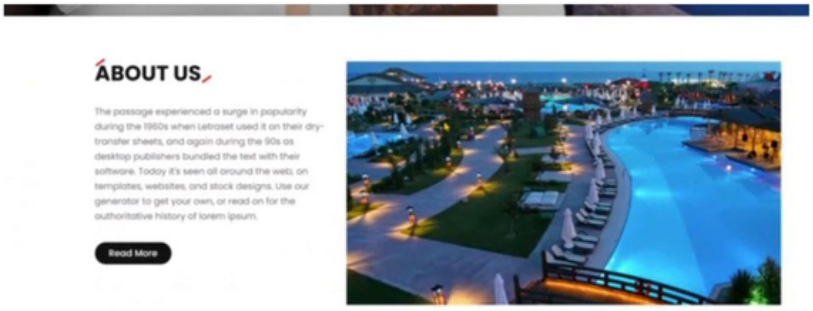
Admin Create Room



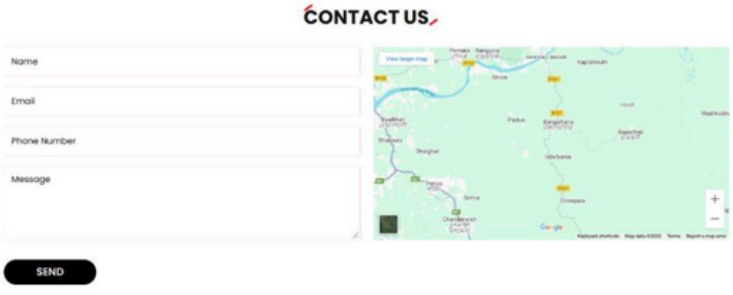
Our Room



About Us



Contact Us



Chapter 6

Project Summery

6.1 Project Overview

The main goal of this project is to develop a complete system that automates the core activities of a hotel or a similar organization, such as a university managing campus facilities.

6.2 Purpose

This Hotel Management System will work as a centralized and automated website that manages the main operations of the organization. It will especially focus on user access, event or club enrollment, and scheduling shared resources. The system will make it easier for students or guests to register for different activities, while also helping the admin monitor, control, and approve resource usage smoothly and without confusion.

6.3 Key Features

Category	Key Feature	Description
I. Access & Security	Role-Based Authentication	Secure login and registration for all users (Admin, Student, Organizer). Ensures different users only access their authorized modules.
II. User Activities	Event & Club Enrollment	Allows students/users to view active events and clubs, successfully enroll, and cancel their participation.
III. Administration	Event Approval Workflow	Gives the Admin the power to review, approve, or reject pending events, ensuring only valid activities are listed.
IV. Resource Control	Venue Scheduling & Monitoring	Enables the Admin to view the venue schedule (all bookings and free slots) and monitor club statistics and membership.
V. System Reliability	Error Handling	Includes specific handling for negative scenarios like incorrect login, connectivity loss, system database failure, and unauthorized access attempts.

6.4 Technology Stack

Layer	Technology	Role in the Project
Front-End	HTML5, CSS3, JavaScript	It is used to construct the user interface (UI), client side interactions, and make sure it has a responsive design to various devices.
Back-End (Server-Side)	PHP (Hypertext Preprocessor)	It is the main scripting language. It serves all server-side logic, user requests (e.g. sign in, registering), and interacts with the database.
Database	MySQL	The relational database management system (RDBMS) used to store all structured data, including user accounts, event details, club information, and venue schedules.
Web Server	Apache HTTP Server (or Nginx)	Serves the PHP application files and delivers the web pages to the end-user's browser.
Operating System	Linux (or Windows/WAMP)	The underlying environment where the server, database, and PHP runtime are hosted.
Version Control	Git / GitHub	Used for managing source code changes, collaboration, and project tracking (as seen by the repository link).

6.5 Expected Impact

The expected impact of this Hotel Management System can be viewed in three major areas: operational efficiency, security and data integrity, and overall user experience. The impact is likely to be significant because the system replaces many manual, time-consuming tasks with automated processes.

APPENDIX A: Source Code & Live Website Link

1. Source Code: https://github.com/rashid35/Smart-Hotel-Management-System-with-AI-Powered-Customer-Support