Hotel Automation Project

Software Requirements Specification

Version <1.0>

Submitted by:

122CS0062 : Albert Antony
122CS0082 : Shadan Hussain
122CS0102 : Anshuman Sahoo
122CS0302 : Swastik Behera

Revision History

Date	Version	Description	Author
<04/02/25>	<1.0>	SRS 1.0	Group-Modulo 2

Table of Contents

Software Requirements Specification

1. Introduction

1.1 Purpose:

The Hotel Management System is a tool for booking hotel rooms online for the customer. The purpose of this document is to present a detailed description of the Hotel Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, and the constraints under which it will operate. We shall predict and sort out how we hope this product will be used to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

A clear understanding of the system and its functionality will allow for the correct software to be developed and will be used for the development of the future stage of the project. This documentation provides the foundation for the development of the software. It provides a blueprint to upcoming new developers and the maintenance team to assist in maintaining and modifying the project as per required changeability.

1.2 Scope of the project

The Hotel Automation project is intended for reservations for rooms that can be made online. It will be able to automate the various operations of the hotel. Our Hotel Management System will have three end users: Customer, Receptionist, and Hotel Manager. The Hotel Management System will consist of a Booking Management System, DBMS Server, and Report Generator. Customers will be able to check for room availability, select the rooms, and pay for the room. The receptionist will have access to update or modify booking details. Managers will be able to view the financial report and be able to update room information such as cost and category. The main goal of this automated HMS software is to simplify the everyday process of hotels. Day to day Hotels are increasing and they need to automate to provide customers ease of access. It will be able to take care of services to customers in a quick manner. This automation will be able to replace the drawbacks of large customer information physical files which were difficult to handle. Secure Transactions, quick retrieval of information, ease of use, quick recovery of errors, and fault tolerance are some of the benefits that the development team will be working on to achieve end-user satisfaction.

1.3 Definitions, Acronyms, and Abbreviations

Configuration	It means a product that is available / Selected from a catalog can be customized.
FAQ	Frequently Asked Questions
CRM	Customer Relationship Management
RAID 5	Redundant Array of Inexpensive Disk/Drives

1.4 References

The references are:

- ✔ Hostel Automation Structural Model
- ✔ Hostel Automation Behavioral Model
- ✔ Hostel Automation NFR Model
- ✔ Vision Draft 5

1.5 Overview

The remaining sections of this document provide a general description, including the characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. A general description of the project is discussed in section 2 of this document. Section 3 gives the functional requirements, data requirements, constraints, and assumptions made while designing the Hotel Automation Software. It also gives the user a viewpoint of the product. Section 3 also gives the specific requirements of the product. Section 3 also discusses the external interface requirements and gives a detailed description of functional requirements. Section 4 is for supporting information.

2. Overall Description

This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the company. It further contains a list of the stakeholders and users of the proposed solution. It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercise as part of the requirements workshop. It further lists and briefly describes the major features and a brief description of each of the proposed systems.

The following SRS contains the detailed product perspective from different stakeholders. It provides the detailed product functions of Hotel Automation Software with user characteristics permitted constraints, assumptions and dependencies, and requirements subsets.

3. Specific Requirements

3.1 Functional Requirements

Introduction

This section details the functional requirements for the Hotel Automation Software system, which aims to streamline and automate hotel operations, providing enhanced customer service, efficient management, and a smooth guest experience. These requirements are organized by features, and the system will be designed to meet the needs outlined in the vision document.

3.1.1 Login

- **Description**: The system should allow users to securely log in with a unique username and password.
- **Functionality**:
  - Users should be able to log in using credentials (username/email and password).
  - Support for different user roles (admin, staff, guest) with different permissions.

3.1.2 Provide Comprehensive Room Details

- **Description**: The system should display detailed information about each available room.
- **Functionality**:
  - Display room type (single, double, suite, etc.).
  - Show images and videos of the rooms.
  - Include pricing details, amenities, and availability.
  - Show room size, bed types, and occupancy limits.
  - Room features such as Wi-Fi, air conditioning, etc.

3.1.3 Detailed Product Categorizations

- **Description**: The system should categorize all available products and services offered by the hotel.
- **Functionality**:
  - Categorize rooms, food & beverages, spa services, and other hotel amenities.

- Show availability and pricing for each product category.
- Allow filtering by category, price, and features.

3.1.4 Provide Search Facility

- **Description**: The system should provide a search feature to help customers find rooms or services based on specific criteria.
- **Functionality**:
  - Customers can search by room type, price range, dates, amenities, etc.
  - Allow for advanced search options (e.g., by location, specific services).
  - Show search results with available rooms and details.

3.1.5 Maintain Customer Profile

- **Description**: The system should maintain a profile for each customer that stores personal information, preferences, and history.
- **Functionality**:
  - Allow customers to create and update their profile with details like name, contact info, preferences, etc.
  - Maintain a history of past bookings, feedback, and services used.
  - Provide a secure profile management interface.

3.1.6 Provide Personalized Profile

- **Description**: The system should allow customers to have a personalized profile based on their preferences and previous interactions.
- **Functionality**:
  - Display personalized recommendations (room types, services) based on user history.
  - Offer tailored promotions and discounts.
  - Allow users to save preferred room types and amenities for future bookings.

3.1.7 Provide Customer Support

- **Description**: The system should provide robust customer support features.
- **Functionality**:
  - Offer live chat, email, or phone support.
  - Track support tickets and resolution status.
  - Provide a knowledge base with FAQs and troubleshooting guides.

3.1.8 Room Booking

- **Description**: The system should allow customers to book rooms online.
- **Functionality**:
  - Select check-in and check-out dates, room type, and number of guests.
  - Display availability and pricing in real time.
  - Confirm the booking after payment.

3.1.9 Email Confirmation

- **Description**: The system should send an email confirmation after a room is booked.
- **Functionality**:
  - Include booking details like dates, room type, price, and guest name.
  - Provide booking reference number.
  - Optionally include a cancellation policy.

3.1.10 Detailed Invoice for Customer

- **Description**: The system should generate a detailed invoice for the customer after booking.
- **Functionality**:
  - Include breakdown of charges (room cost, taxes, extra services).
  - Provide a payment summary and due amount.
  - Optionally include payment history.

3.1.11 Room Services

- **Description**: The system should allow guests to request room services during their stay.
- **Functionality**:
  - Guests can order food, towels, cleaning, etc., from their room via the system.
  - Display a list of available services.
  - Track the status of service requests.

3.1.12 Conference Hall Booking

- **Description**: The system should allow guests to book conference halls or meeting rooms.
- **Functionality**:
    - Display available conference hall options with seating capacity and amenities.
    - Allow users to book based on their event requirements (e.g., time, number of attendees).
    - Option to add catering or audio-visual support to the booking.

3.1.13 Provide Online Tax Calculations

- **Description**: The system should automatically calculate applicable taxes on bookings and services.
- **Functionality**:
    - Calculate taxes based on local tax rates and regulations.
    - Include tax details in invoices.
    - Display applicable tax rates during booking confirmation.

3.1.14 Check-in and Check-out

- **Description**: The system should allow guests to check in and check out digitally.
- **Functionality**:
    - Allow guests to check-in before arrival via the system.
    - Send notifications about check-in times and procedures.
    - Facilitate self-check-out and generate final invoices automatically.
    - Include an option for extending stays and modifying bookings.

3.1.15 Allow Multiple Payment Methods

- **Description**: The system should support multiple payment methods for bookings and services.
- **Functionality**:
    - Support credit/debit cards, online wallets, bank transfers, and possibly cash payments.
    - Allow customers to save payment details for future transactions.
    - Provide secure payment processing.

3.1.16 Allow Online Room Reviews and Ratings

- **Description**: The system should allow customers to leave feedback on rooms and services.
- **Functionality**:
  - Customers can rate rooms, amenities, and services on a scale (e.g., 1-5 stars).
  - Display reviews publicly to help future customers in their decision-making.
  - Allow for moderation of reviews (e.g., report inappropriate content).

3.1.17 Offer Online Promotions and Rewards

- **Description**: The system should offer promotions, discounts, and rewards for customers.
- **Functionality**:
  - Provide time-limited promotions and discounts based on seasons, events, or loyalty programs.
  - Offer rewards points for frequent customers (e.g., redeemable for discounts).
  - Allow users to redeem promo codes during checkout.

3.2 Non-Functional Requirements

Non-Functional Requirements (NFRs) define the quality attributes, constraints, and operational characteristics that the system must exhibit to ensure optimal performance, security, and user satisfaction. While functional requirements focus on the specific tasks the software must perform—such as booking rooms, processing payments, or generating invoices—non-functional requirements outline how well the software should perform those tasks under varying conditions.

3.2.1 Graphical User Interface (GUI)

- **Description**: The system should provide a visually appealing, user-friendly, and intuitive graphical interface for all users (admin, staff, guests).
- **Functionality**:
  - The user interface should be easy to navigate, with clear labels and icons.
  - It should be visually consistent across all pages and sections, with a professional and modern design.
  - The system should support responsive design, adjusting the layout and elements according to the device's screen size (desktop, tablet, mobile).

- The interface should provide feedback to users for actions (e.g., confirmation of a booking, error messages).
- The UI should allow users to easily interact with all available features (room search, booking, profile management, etc.) through simple, clean, and effective interactions.

3.2.2 Accessibility

- **Description**: The system should comply with accessibility standards to ensure it is usable by people with disabilities, following best practices in web accessibility.
- **Functionality**:
  - The system should meet the Web Content Accessibility Guidelines (WCAG) 2.1, Level AA standards.
  - Ensure compatibility with screen readers and other assistive technologies.
  - Provide keyboard navigation support for all functionality, so users can operate the system without a mouse.

3.2.3 Performance Requirements

- **Description**: The system should meet performance standards to ensure it operates efficiently under various conditions, including high traffic volumes.
- **Functionality**:
  - The system should load within **3 seconds** under normal conditions (page load time).
  - Response times for all user interactions (e.g., booking a room, searching) should be less than **2 seconds**.
  - The system should support **at least 10,000 simultaneous users** during peak hours.
  - The backend system (e.g., database) should be optimized for high performance, reducing query times and preventing bottlenecks.
  - The system should be capable of handling spikes in traffic during promotional periods, large events, or holidays.
  - The system should provide real-time availability updates for rooms, conference halls, and services without significant delays.

3.2.4 Safety Requirements

- **Description**: The system should ensure that operations are safe and protect users and hotel staff from risks associated with system malfunction or data corruption.

- **Functionality**:
    - **Backup and Recovery**: The system should include regular data backups and a disaster recovery plan, ensuring data can be restored quickly in case of failure.
    - **Redundancy**: The system should have redundancy for critical components like databases, payment gateways, and servers to prevent service interruptions.
    - **Failover Mechanism**: In case of a system crash or failure, the system should failover seamlessly to a secondary system without user disruption.
    - **Monitoring**: The system should include health monitoring tools that notify administrators of system issues like server overloads, slowdowns, or outages.
    - **User Error Prevention**: The system should incorporate features to reduce user errors, such as validation checks and confirmation dialogues for actions (e.g., cancellations, modifications).

3.2.5 Security Requirements

- **Description**: The system should be secure from threats such as unauthorized access, data breaches, and cyberattacks.
- **Functionality**:
    - **Authentication**: Implement strong user authentication (e.g., multi-factor authentication) for admin and staff users.
    - **Data Encryption**: All sensitive data (e.g., customer personal data, payment information) should be encrypted using **SSL/TLS** protocols during transmission and encryption at rest.
    - **Access Control**: Implement role-based access control (RBAC) to ensure that users (e.g., guests, staff, administrators) can only access the information and features relevant to their role.
    - **Session Management**: Implement session timeouts for idle users to prevent unauthorized access in case of forgotten logouts.
    - **Audit Logging**: The system should log all user actions (e.g., bookings, cancellations, changes to profiles) for accountability and security audits.
    - **Vulnerability Management**: Regularly test and patch the system for security vulnerabilities, and conduct security audits.
    - **Payment Security**: Compliance with the **Payment Card Industry Data Security Standard (PCI DSS)** for safe processing of card transactions.

3.2.6 Software Quality Attributes

- **Description**: The system should follow established software engineering principles to ensure high-quality development.

- **Functionality**:
    - **Reliability**: The system should have high availability, with uptime of at least **99.9%**.
    - **Scalability**: The system should be able to scale horizontally (e.g., adding more servers) or vertically (e.g., upgrading server hardware) to accommodate growing user demands.
    - **Maintainability**: The codebase should be modular, well-documented, and easy to maintain and upgrade with minimal downtime.
    - **Portability**: The system should be platform-agnostic and able to function across different operating systems (e.g., Windows, Linux, macOS) and devices (smartphones, tablets, desktops).
    - **Extensibility**: The system should be designed to accommodate future feature additions, such as new payment methods, integrations with third-party software, and additional hotel services.
    - **Interoperability**: The system should be compatible with external systems like accounting software, email servers, and payment gateways for seamless integrations.
    - **Testability**: The system should be easily testable with automated testing tools to ensure functionality, performance, and security are consistently met.

3.2.7 Business Rules

- **Description**: The system should adhere to specific business rules that define how hotel management processes are executed.
- **Functionality**:
    - **Pricing Rules**: Implement dynamic pricing strategies based on demand, season, or promotional campaigns (e.g., discounts, last-minute offers).
    - **Booking Policies**: The system should enforce hotel-specific booking policies, such as minimum stay requirements, cancellation policies, and refundable/non-refundable options.
    - **Room Allocation**: Automatically allocate rooms based on guest preferences and availability while avoiding double bookings.
    - **Tax Calculation**: Apply appropriate local, regional, and national taxes based on the guest's location and the services provided.
    - **Promotions and Discounts**: Enforce rules regarding promotions, such as eligibility criteria (e.g., loyalty program members), limits on usage, and expiration dates.
    - **Payment Terms**: Define and enforce rules around payment methods, deposits, and payment due dates.
    - **Rewards Program**: Implement the rules for customer loyalty rewards, including points accumulation, redemption methods, and tier benefits.

4. Data Dictionary

1. Customer Table

Field Name	Data Type	Description	Constraints
customer_id	Integer	Unique identifier for each customer	Primary Key, Auto-increment
first_name	Varchar(50)	Customer's first name	Not Null
last_name	Varchar(50)	Customer's last name	Not Null
email	Varchar(100)	Customer's email address	Unique, Not Null
phone_number	Varchar(15)	Customer's phone number	Optional
address	Varchar(255)	Customer’s home address	Optional
date_of_birth	Date	Customer’s date of birth	Optional
loyalty_points	Integer	Points accumulated by the customer through bookings and services	Default : 0

2. Room Table

Field Name	Data Type	Description	Constraints
room_id	Integer	Unique identifier for each room	Primary Key, Auto-increment
room_type	Varchar(50)	Type of room (e.g., Single, Double, Suite)	Not Null
price	Decimal(10,2)	Price per night for the room	Not Null
capacity	Integer	Maximum number of guests allowed in the room	Not Null
status	Varchar(20)	Current status of the room (e.g., Available, Booked, Maintenance)	Not Null
description	Varchar(255)	Additional description of the room (e.g., bed type, amenities)	Optional
amenities	Varchar(255)	List of amenities available in the room (e.g., Wi-Fi, TV, minibar)	Optional
room_size	Integer	Size of the room in square feet	Optional

3. Booking Table

Field Name	Data Type	Description	Constraints
booking_id	Integer	Unique identifier for each booking	Primary Key, Auto-increment
customer_id	Integer	Foreign key linking to the Customer table	Not Null, Foreign Key
room_id	Integer	Foreign key linking to the Room table	Not Null, Foreign Key
check_in_date	Date	Date when the customer checks into the room	Not Null
check_out_date	Date	Date when the customer checks out from the room	Not Null
total_price	Decimal(10,2)	Total price for the entire stay, including taxes and additional services	Not Null
status	Varchar(20)	Booking status (e.g., Confirmed, Pending, Cancelled)	Not Null
booking_date	Date	Date when the booking was made	Not Null

4. Payment Table

Field Name	Data Type	Description	Constraints
payment_id	INT	Unique identifier for each payment	Primary Key, Auto-increment
reservation_id	INT	Link to the reservation associated with the payment	Foreign Key (reservation_id)
payment_date	DATE	Date of payment	Not Null
amount	DECIMAL(10, 2)	Total amount paid	Not Null
payment_method	VARCHAR(50)	Method used for payment (e.g., Credit Card, Cash, Online)	Not Null
payment_status	VARCHAR(50)	Status of payment (e.g., completed, failed)	Not Null
transaction_id	VARCHAR(100)	Unique identifier for the transaction	Nullable

5. Staff Table

Field Name	Data Type	Description	Constraints
staff_id	INT	Unique identifier for each staff member	Primary Key, Auto-increment
first_name	VARCHAR(100)	Staff member's first name	Not Null
last_name	VARCHAR(100)	Staff member's last name	Not Null
position	VARCHAR(100)	Staff member's position (e.g., Receptionist, Housekeeper)	Not Null
email	VARCHAR(150)	Staff member's email address	Not Null, Unique
phone_number	VARCHAR(15)	Staff member's phone number	Not Null
hire_date	DATE	Date the staff member was hired	Not Null
salary	DECIMAL(10, 2)	Salary of the staff member	Not Null

6. Service Table

Field Name	Data Type	Description	Constraints
service_id	INT	Unique identifier for each service	Primary Key, Auto-increment
service_name	VARCHAR(100)	Name of the service (e.g., Spa, Room Service)	Not Null
description	TEXT	Description of the service offered	Nullable
price	DECIMAL(10, 2)	Price of the service	Not Null
availability	VARCHAR(50)	Availability status (e.g., available, unavailable)	Not Null, Default: 'available'

7. Room Service Table

Field Name	Data Type	Description	Constraints
room_service_id	INT	Unique identifier for each room service order	Primary Key, Auto-increment
reservation_id	INT	Link to the reservation for which service is ordered	Foreign Key (reservation_id)
service_id	INT	Link to the service ordered	Foreign Key (service_id)
order_date	DATETIME	Date and time of the service order	Not Null
quantity	INT	Quantity of service items ordered	Not Null
status	VARCHAR(50)	Status of the order (e.g., pending, completed)	Default: 'pending'

8. Maintenance Table

Field Name	Data Type	Description	Constraints
maintenance_id	INT	Unique identifier for each maintenance request	Primary Key, Auto-increment
room_id	INT	Room that needs maintenance	Foreign Key (room_id)
staff_id	INT	Staff responsible for the maintenance	Foreign Key (staff_id)
request_date	DATETIME	Date and time when maintenance was requested	Not Null
completion_date	DATETIME	Date and time when maintenance was completed	Nullable
issue_description	TEXT	Description of the maintenance issue	Not Null
status	VARCHAR(50)	Status of the maintenance (e.g., pending, completed)	Default: 'pending'

5. Data Flow Diagram (DFD)

5. 1 Levels of the DFD:

1. Level 0 (Context Diagram):

- Level 0 DFD shows the entire system as a single process with interaction from external entities.

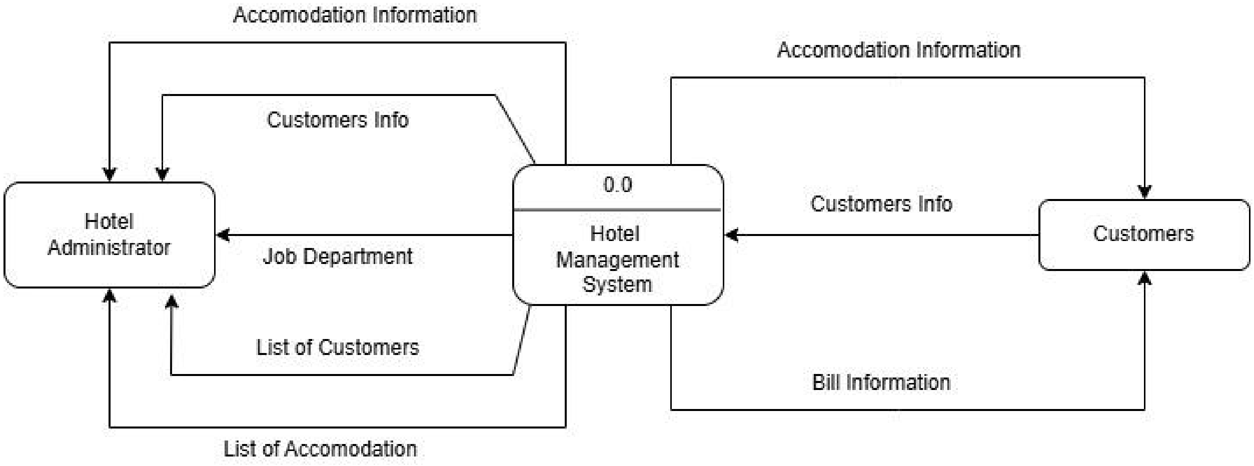
2. Level 1:

- Level 1 DFD decomposes the main system into key processes like reservation management, room management, staff management, and payment processing.

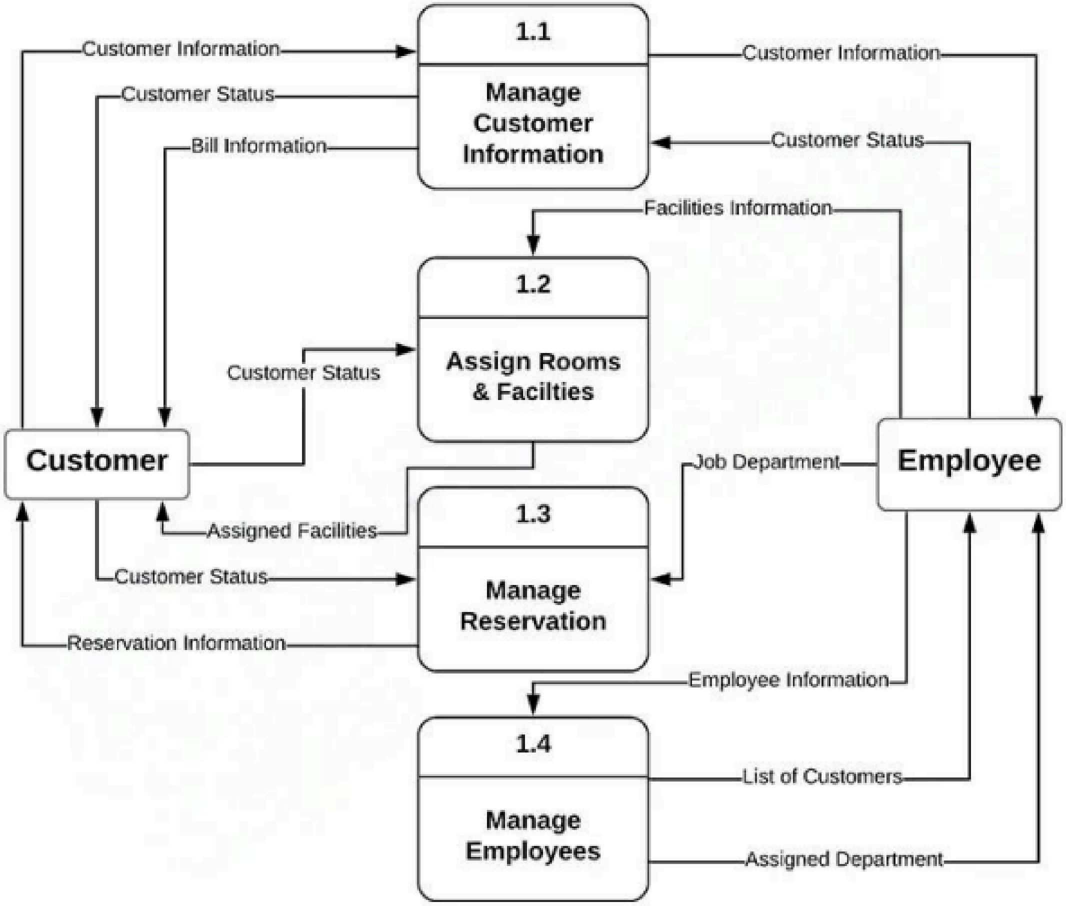
3. Level 2:

- Level 2 DFD further breaks down each key process into smaller, specific sub-processes that handle detailed operations.

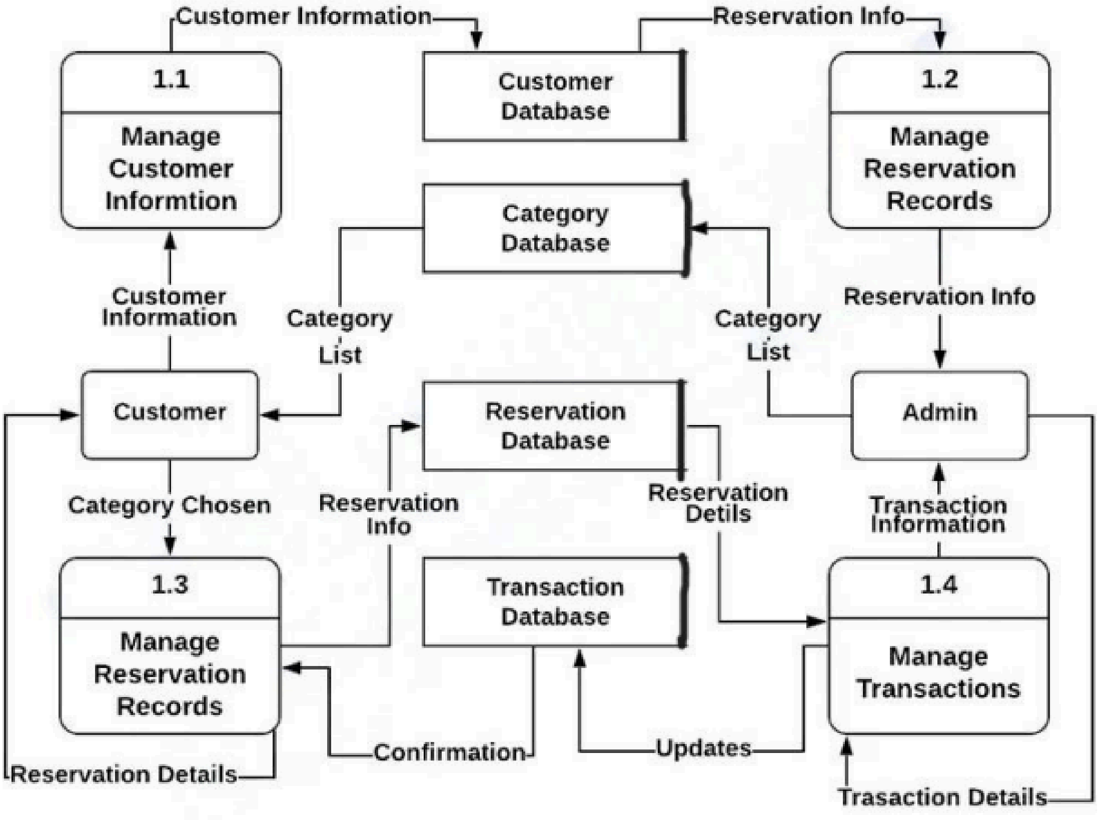
1. Level 0 DFD FOR HOTEL AUTOMATION SYSTEM



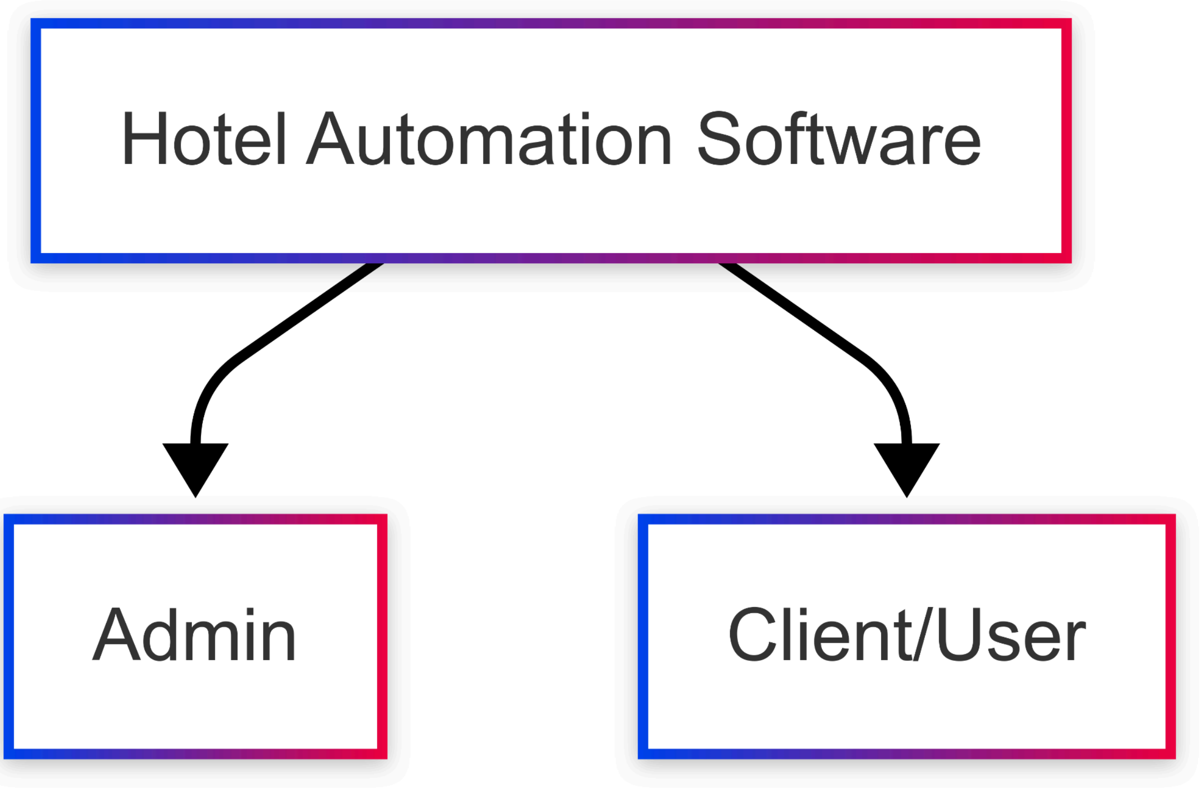
2. Level 1 DFD FOR HOTEL AUTOMATION SYSTEM

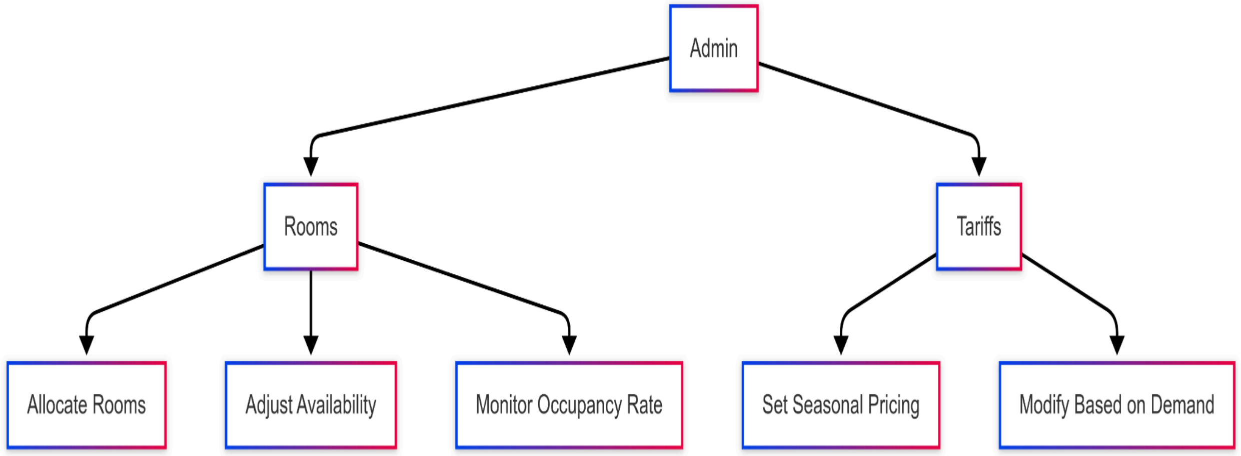


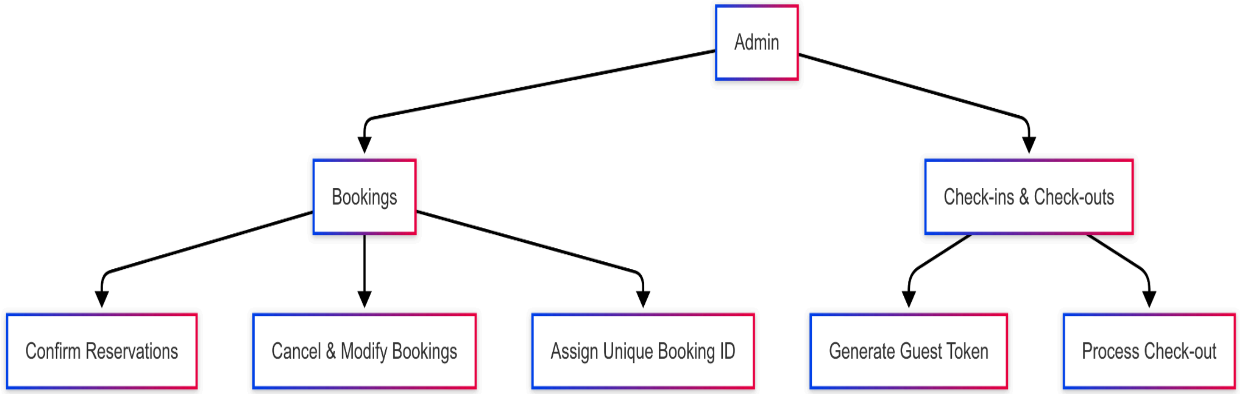
3. Level 2 DFD FOR HOTEL AUTOMATION SYSTEM

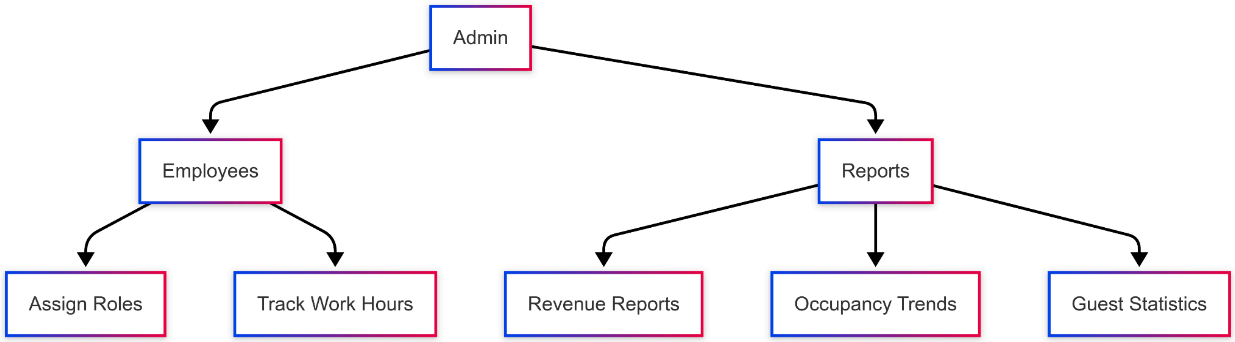


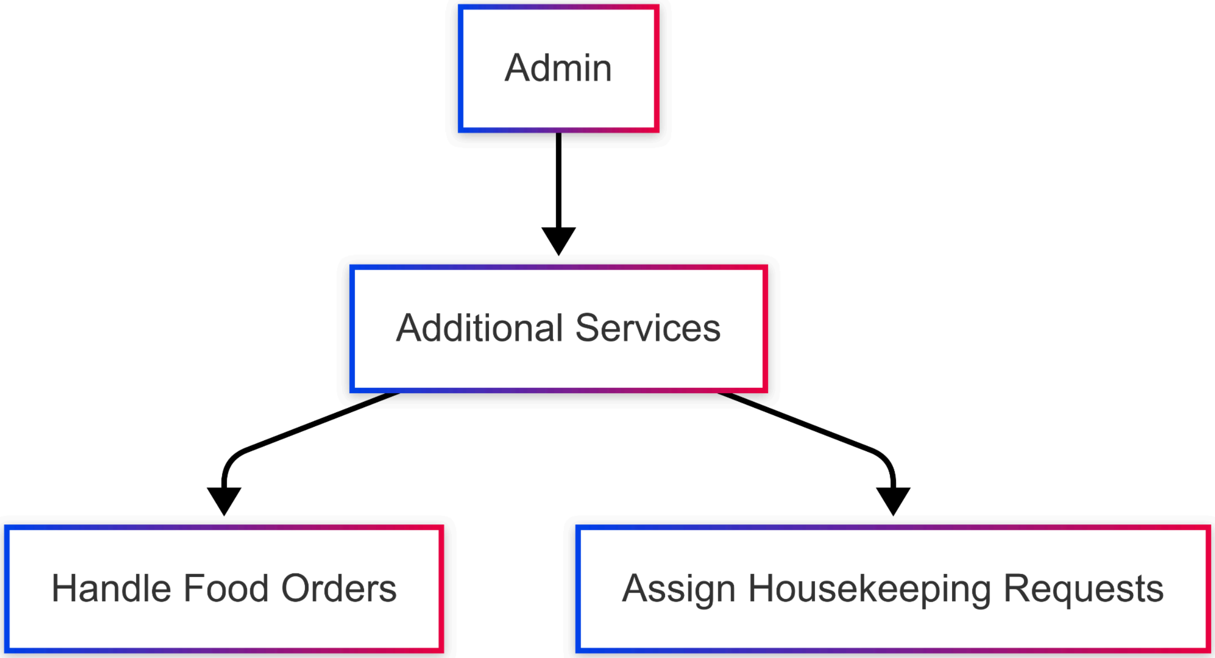
Structure Chart

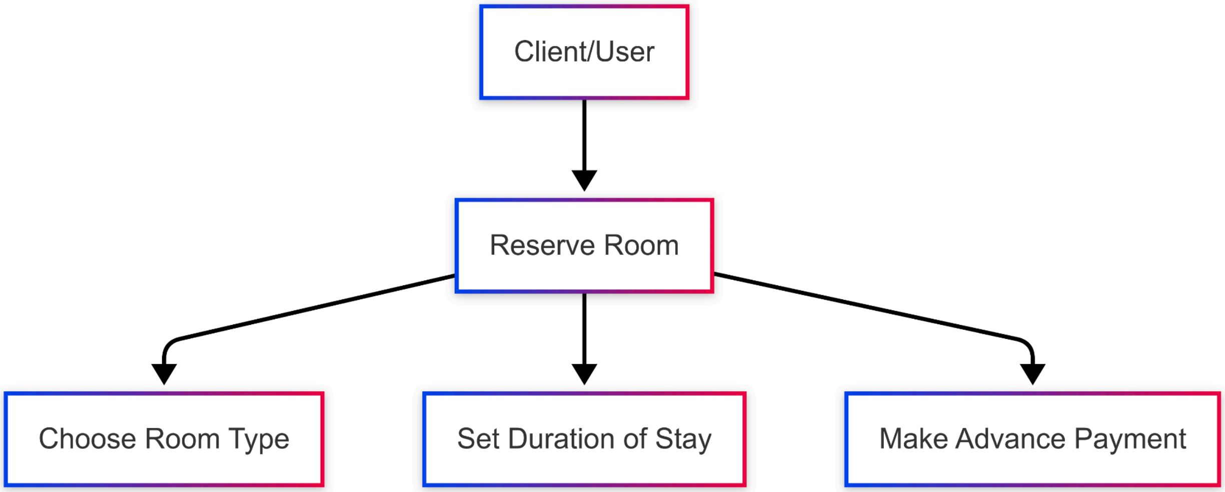


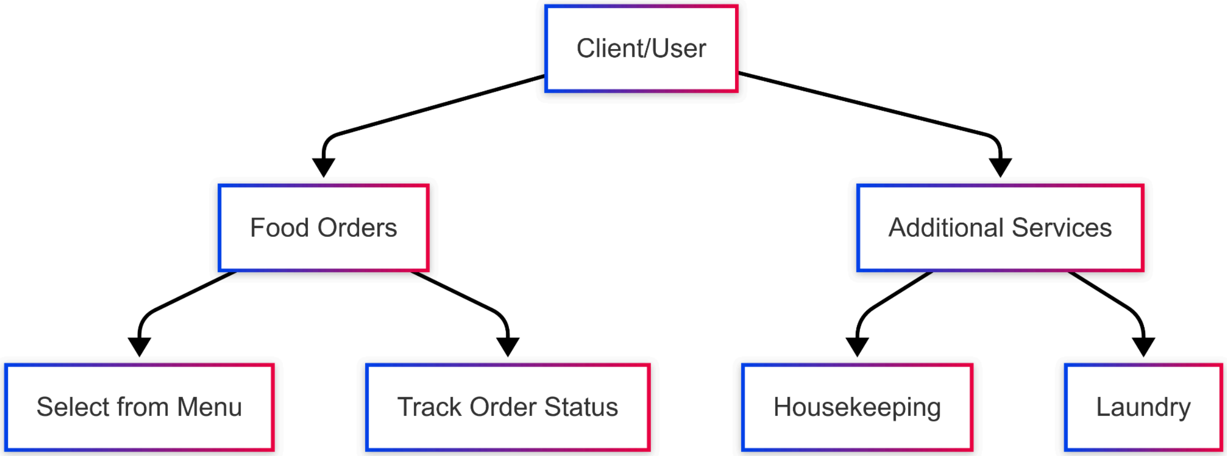


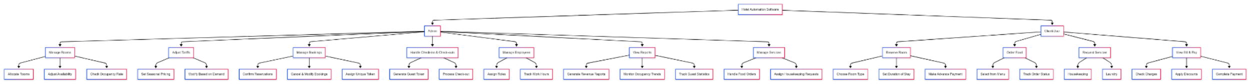

Admin
Bookings
Check-ins & Check-outs
Confirm Reservations
Cancel & Modify Bookings
Assign Unique Booking ID
Generate Guest Token
Process Check-out


Admin
Employees
Reports
Assign Roles
Track Work Hours
Revenue Reports
Occupancy Trends
Guest Statistics


Admin
Additional Services
Handle Food Orders
Assign Housekeeping Requests

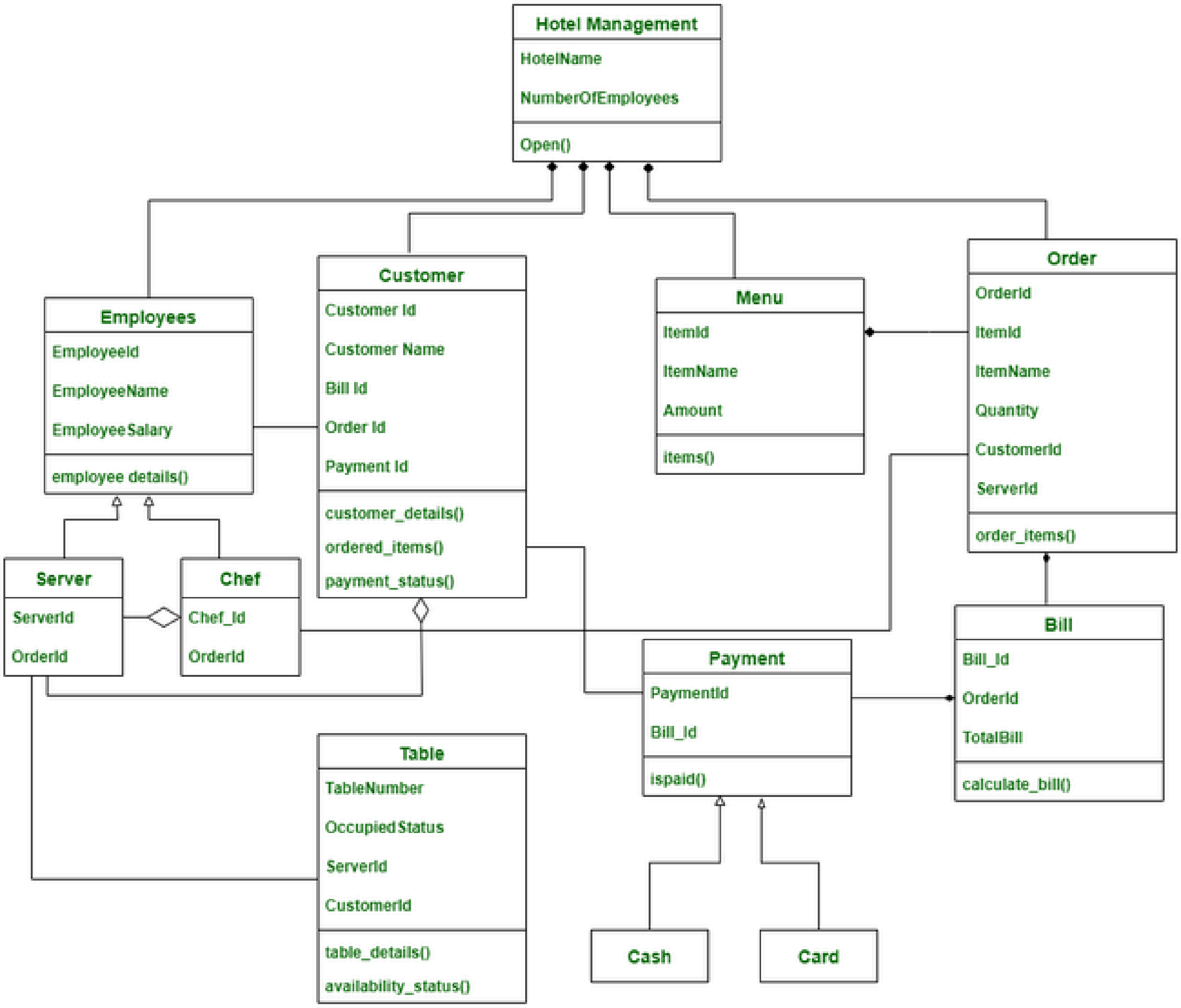




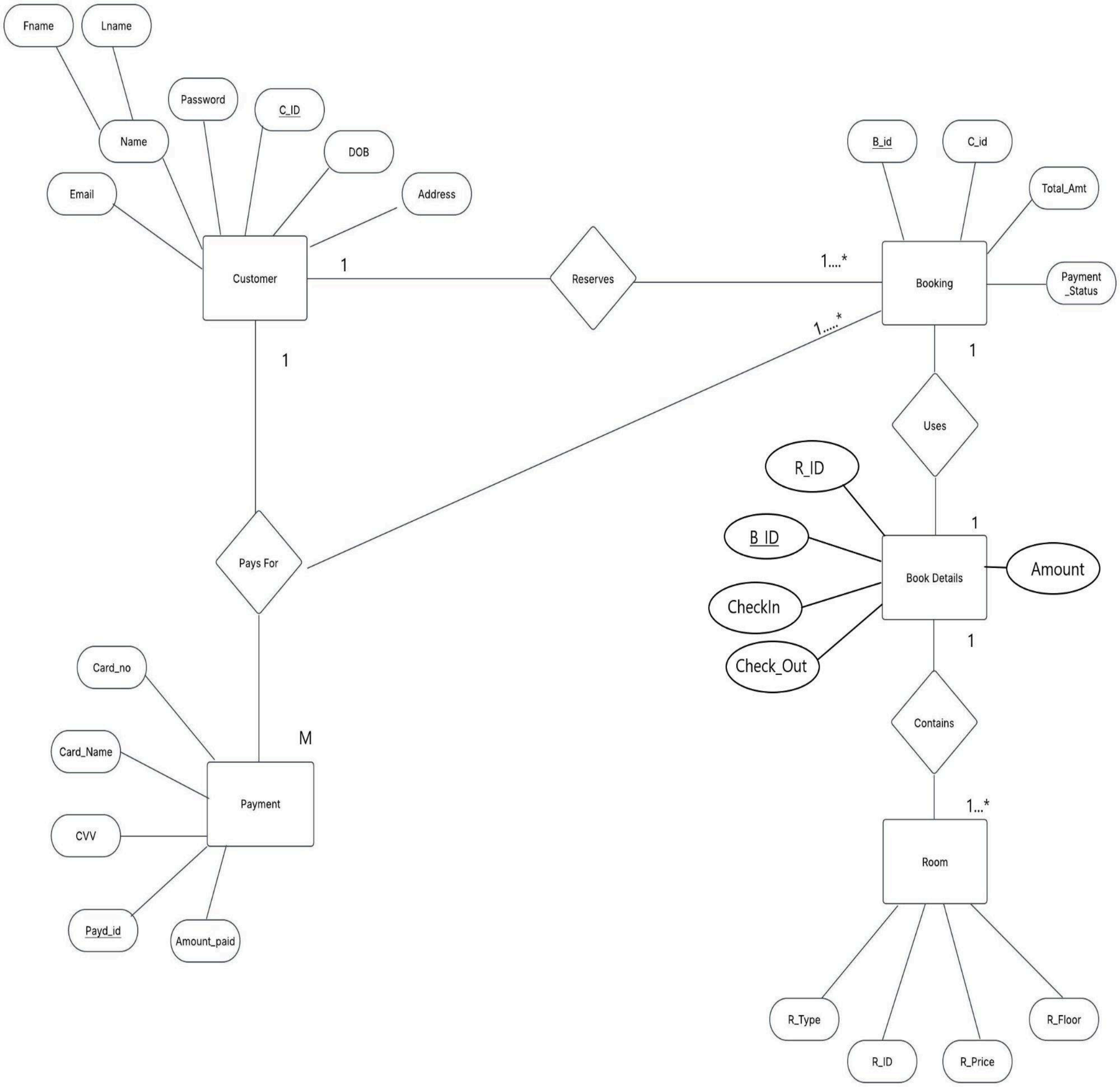


https://www.mermaidchart.com/raw/a5bc4eb9-fc6f-43de-8284-b44fe077e127?theme=light&version=v0.1&format=svg

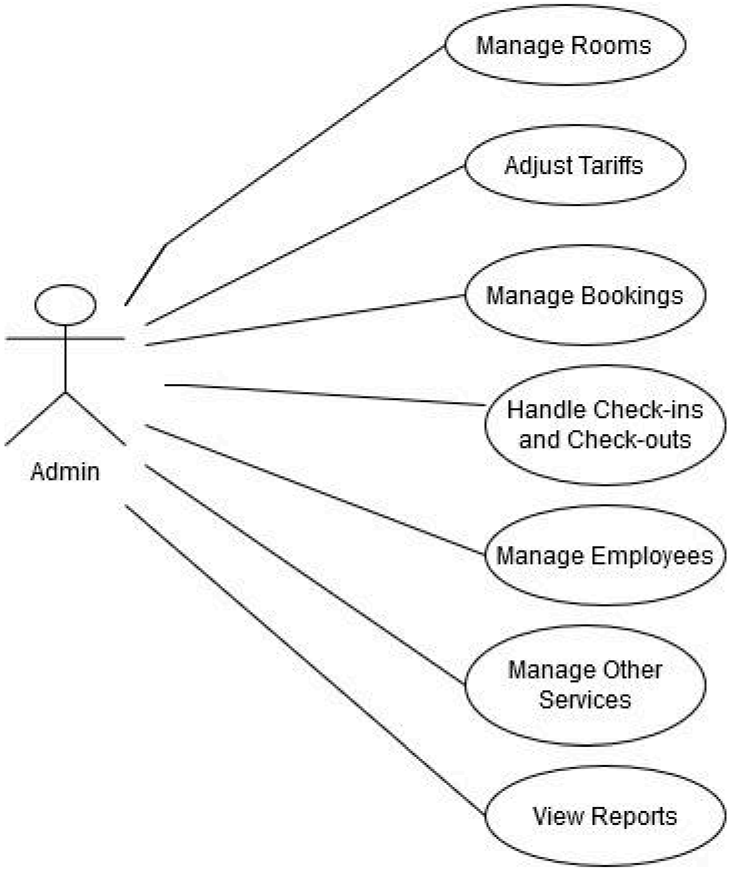
Class Diagram

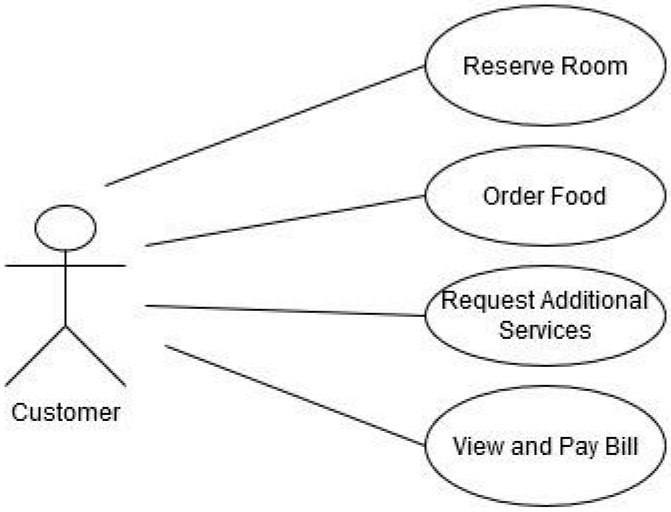


ER Diagram

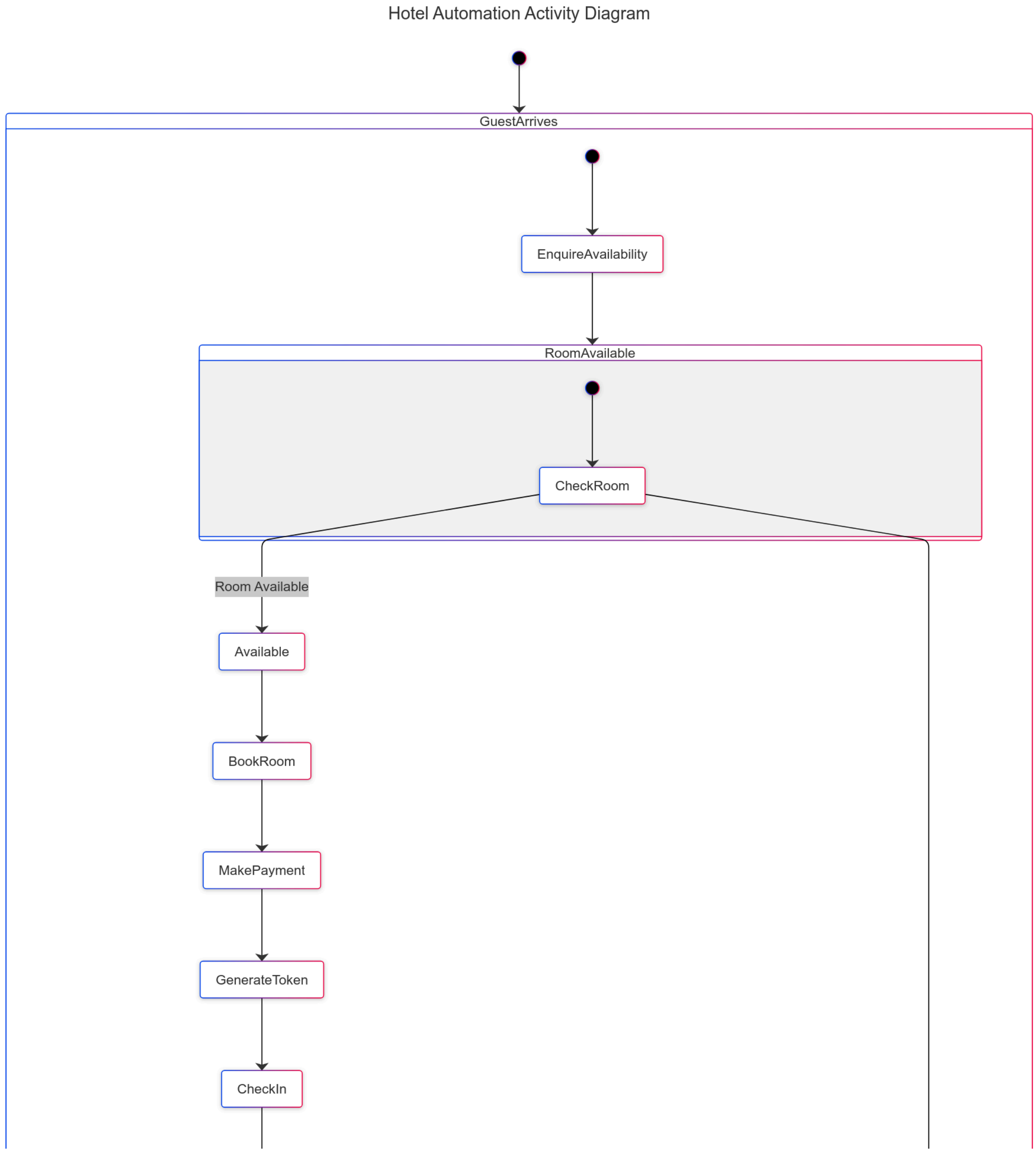

Fname
Lname
Password
C_ID
Name
DOB
Email
Address
Customer
1
Reserves
1....*
B_id
C_id
Total_Amt
Booking
Payment
_Status
1.....*
1
1
Uses
R_ID
B ID
1
Amount
Book Details
CheckIn
Check_Out
1
Pays For
Card_no
Card_Name
M
Payment
CVV
Contains
1...*
Room
Payd_id
Amount_paid
R_Type
R_Floor
R_ID
R_Price

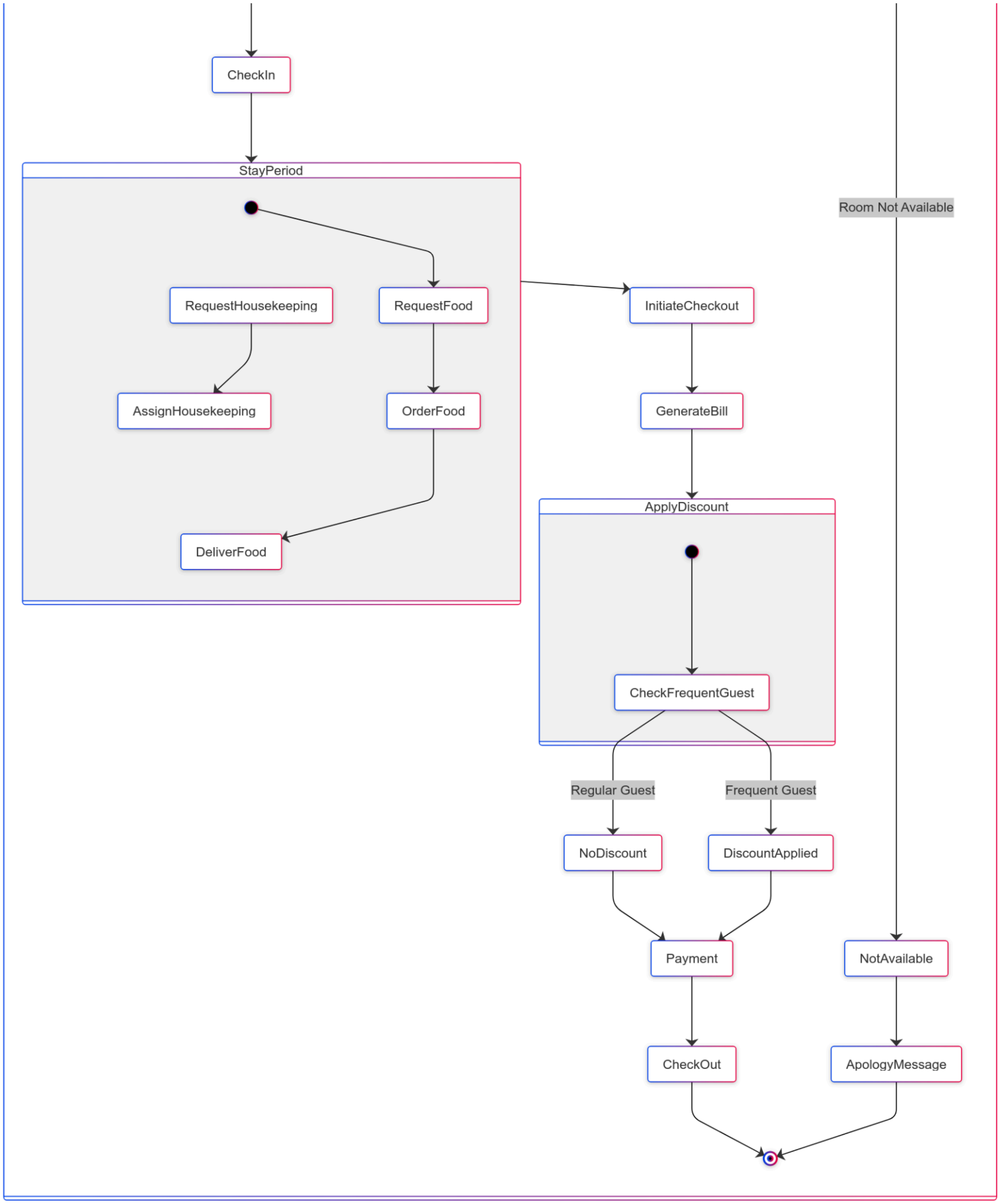
Use-Case Diagram





Activity Diagram





https://www.mermaidchart.com/raw/a50a68be-c57c-444b-9680-17a3f2f2d681?theme=light&version=v0.1&format=svg

Conclusion

The Hotel Automation Software (HAS) enhances operational efficiency by streamlining key hotel management processes, ensuring seamless guest experiences and optimized resource utilization. The system is highly scalable and adaptable, allowing hotels to handle fluctuating demands with ease.

With data-driven decision-making, historical booking and service data help administrators improve business strategies, optimize staffing, and enhance customer satisfaction. Real-time tracking of room availability, housekeeping, and guest requests ensures better service delivery and reduces delays.

The user-friendly interface makes adoption easy for hotel staff at all levels, from front desk clerks to managers and administrators. By automating essential hotel functions, HAS minimizes manual workload, reduces errors, and enhances overall service quality, making it a valuable asset for modern hospitality management.