

HOTEL MANAGEMENT

SYSTEM

Software Engineering

Project Report

Submitted by:

Himangi Aneja (21525)
Priyam Gupta (21546)
Yana Aggarwal (21569)

Supervisor:

Ms. Kavita Rastogi
(Course Instructor)



2023
Department of Computer Science
Shaheed Sukhdev College of Business Studies
University of Delhi

Process Model.....	6
1. Software Requirement Specification.....	7
1.1 Introduction.....	7
1.1.1 Purpose.....	7
1.1.2 Scope.....	7
1.1.3 Objective.....	7
1.1.4 Overview.....	8
1.2 Overall Description.....	9
1.2.1 Product Perspective.....	9
1.2.2 System Interfaces.....	9
1.2.3 Interfaces.....	9
1.2.4 Hardware Interfaces.....	9
1.2.5 Software Interfaces.....	10
1.2.6 Communication Interface.....	10
1.2.7 Memory Constraints.....	10
1.2.8 Product Function.....	10
1.2.9 User Characteristics.....	10
1.3 Constraints.....	11
1.4 Assumptions and Dependencies.....	11
1.5 Specific Requirements.....	12
1.5.1 Functional Requirements.....	12
1.5.2 Non- Functional Requirements.....	14

1.6 Performance Requirement.....	15
1.7 Design Constraints.....	15
1.7.1 Software System Attributes.....	13
1.8 Data Flow Diagram.....	16
1.9 Data Dictionary.....	21
1.10 Use Case.....	22
1.11 User Interfaces.....	28
2. Estimations.....	30
2.1 Function Points.....	30
2.2 Efforts.....	33
3. Scheduling.....	34
4. Risk Management.....	35
5. Design.....	39
5.1 Data Design.....	39
5.2 System Design.....	40
6. Coding.....	41
7. Testing.....	44
8. References.....	47

Problem Statement

A website for the management of a chain of hotels aims to provide a centralized platform for managing all the properties in the chain. The website will allow hotel staff and management to access important information and perform critical tasks related to the daily operations of each hotel while providing an exceptional user experience to customers.

On visiting the website, the person will have two options whether he/she is accessing the website as a user or an admin.

The User Standpoint:

As a user, the person can login using the registered email id and password or can register as a new user.

After successful login, the user will be presented with the following options.

1. Room Booking: Customers can easily search and book rooms in any of the chain's hotels, view room availability, and manage their reservations.
2. Online Payment: Customers can securely make payments for their bookings through the website, using multiple payment methods.
3. Customer Reviews: Customers can share their experiences and rate their stays, helping future customers make informed decisions about where to stay.
4. Hotel Information: Customers can access comprehensive information about each hotel, including location, amenities, and room details.
5. Loyalty Program: The website will also have a loyalty program that rewards customers for their stays, providing incentives to return.
6. Customer Support: Customers can easily contact the hotel for support or questions, either through a dedicated support page or by phone.

The user will first be presented with all the **Hotel information** through which they can choose the hotel they like by assessing the location, amenities, **Customer Reviews** etc. The customer then will go ahead with **Room Booking** according to the number of adults, and children, date of check-in and check-out etc. and use the **Online Payment** feature to pay for its booking. Here they will be able to use their **Loyalty Program** points and benefits such as free nights, free spa etc. After the booking the customer can contact **Customer Support** as needed which will be presented as a chat option with a customer service executive.

The Admin standpoint:

After successful login as the **admin**, he/she will be presented with the following options.

Features:-

1. Dashboard: A user-friendly dashboard to provide an overview of all hotel operations and performance.
2. Reservation Management: A system for managing room reservations, including real-time availability updates, and reservation confirmations.
3. Customer Management: A system for managing customer data, including personal information, booking history, and preferences.
4. Employee Management: A system for managing staff information, including schedules, roles, and payroll.
5. Revenue Management: A system for tracking revenue and occupancy, including daily, weekly, and monthly reports.
6. Multi-Hotel Support: The ability to manage multiple hotels within a single system, including centralized data management and reporting.
7. Secure Login: A secure login system with role-based access controls to ensure the security of sensitive data.

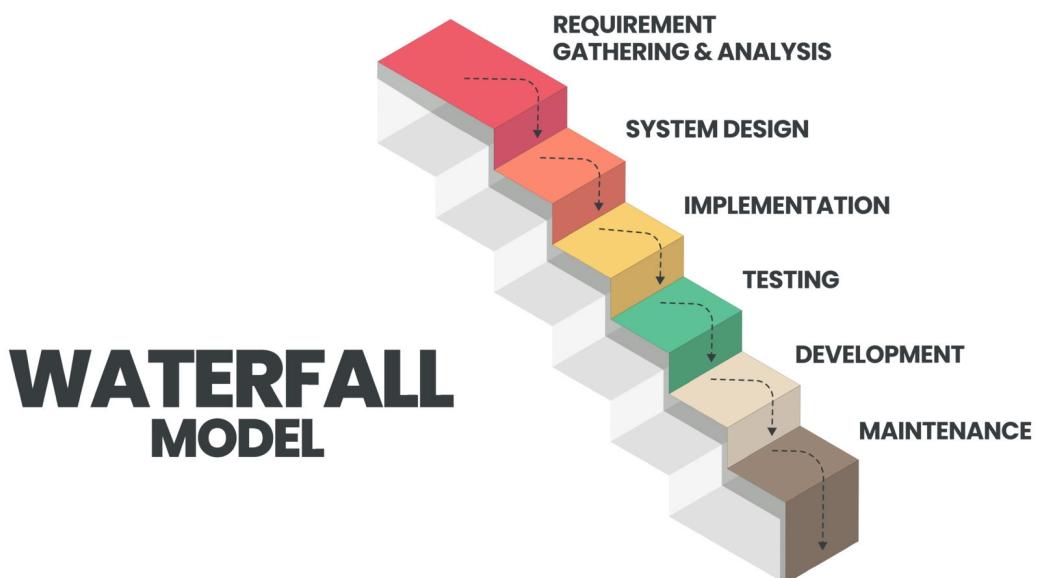
The admin can therefore update(add, delete, modify, view) his database regarding **Reservations, Customer** check-ins and check-outs, **Revenue generated, Employee Management** etc. by choosing any of the options available above.

In conclusion, the website for the management of a chain of hotels will provide a seamless and convenient experience for both customers and hotel staff, enabling efficient and effective hotel management.

PROCESS MODEL:

The **Waterfall Model** is the software architecture that best fits this project. The following are the reasons why we chose the waterfall methodology for our project:

- The needs were clearly outlined and understood before we began.
- There is no overlap between the phases; each step must be finished before the next one can start.
- The customer does not need to check on the progress of the programme while it is being implemented.
- The needs are fixed, and work can progress to completion in a linear sequential flow, hence this model does not cause a difficulty.



1. Software Requirement Specification

1.1 Introduction

1.1.1 Purpose:

The Software Requirement Specification (SRS) for a Hotel Management System project in software engineering outlines the functional and non-functional requirements of the system. It specifies the scope, features, and constraints of the project, and serves as a blueprint for the development team to follow. The SRS document ensures that all stakeholders have a clear understanding of what the system is supposed to do, and helps to avoid misunderstandings and miscommunication during the development process.

1.1.2 Scope:

The scope of Software Requirement Specification (SRS) for a Hotel Management System project includes defining the system's boundaries, identifying the target audience, outlining the system's functionalities and limitations, and specifying its performance metrics. It also provides a roadmap for the development team and serves as a communication tool between stakeholders and developers.

1.1.3 Objective:

The objective of this software engineering project is to develop a comprehensive website for managing a chain of hotels that will offer a user-friendly and efficient platform for both customers and hotel staff.

The website should enable customers to easily search and book rooms, make online payments, place room service requests, and access hotel information.

Additionally, the website should provide hotel staff with tools for managing reservations, rooms, customers, staff, revenue, sales and marketing, and reporting, while maintaining a high level of security and mobile optimization.

Ultimately, the project aims to streamline hotel operations and improve customer satisfaction, leading to increased revenue and growth for the chain of hotels.

1.1.4 Overview:

- The project involves developing a website for managing a chain of hotels.
- The website offers a centralized platform for managing various hotel operations.
- Users can access the website as either a customer or an admin.
- Customers can search and book rooms, make online payments, place room service requests, and access hotel information.
- The website offers a loyalty program for customers, providing incentives to return.
- Customers can easily contact the hotel for support or questions.
- Admins can manage reservations, rooms, customers, staff, revenue, sales and marketing, and reporting.
- The website offers support for multiple hotels within a single system.
- The website has a secure login system with role-based access controls.
- The website is optimized for mobile devices.
- The project aims to streamline hotel operations and improve customer satisfaction, leading to increased revenue and growth for the chain of hotels.

1.2 The Overall Description

1.2.1 Product Perspective:

The project aims to develop a website that provides a comprehensive set of features for customers and hotel management, enabling efficient and effective hotel management while also providing a seamless and convenient experience for customers. The website's impact on users and stakeholders will be significant, including a positive impact on the hotel chain's bottom line.

1.2.2 System Interface:

- Database interface: A database management system to store user information, reservation details, and loyalty program data.
- Payment gateway interface: An API to securely process payment transactions.
- Chat interface: An API for chat support.

1.2.3 Interfaces

The system will have the following interfaces:

- **User Interface:** The user interface will be the primary interface for users to interact with the system. It will be designed to provide a simple and intuitive user experience.
- **Admin Interface:** The admin interface will be used by system administrators to manage the system, including reservations, revenue and payments.

1.2.4 Hardware Interfaces

Client Side:

- Processor: Intel Pentium Dual Core or higher.
- RAM: 2 GB or higher.
- Display: 32-bit color display.
- Network Interface Card: Ethernet or Wi-Fi interface.

Server Side:

- Processor: Intel 3.00GHz processor or higher.
- RAM: 2 GB or higher.
- Display: 32-bit color display.

- Network Interface Card: Ethernet interface.

1.2.5 Software Interfaces:

- React.js - JavaScript library used for website.
- Oracle Server - For managing various database requirements.
- Server: Apache Tomcat Server
- Database JDBC Driver: MySQL JConnector

1.2.6 Communication Interfaces:

- HTTPS: Secure communication protocol between client and server.
- SMTP: Protocol for sending email confirmation messages to users.
- SMS: Protocol for sending booking confirmation messages to users.

1.2.7 Memory Constraints:

- A minimum of 2 GB of RAM (for handheld devices) and a minimum of 4 GB of RAM (for desktop devices) is required

1.2.8 Product Function

1. **Login/Registration:** Allow users/admin to register and log in to the website using their email address and password.
2. **Room Booking:** Allow users to search and book rooms in any of the chain's hotels, view room availability, and manage their reservations.
3. **Online Payment:** Provide a secure and easy-to-use online payment system for customers to pay for their bookings using multiple payment methods.
4. **Customer Reviews:** Allow customers to share their experiences and rate their stays, helping future customers make informed decisions about where to stay.
5. **Hotel Information:** Provide comprehensive information about each hotel, including location, amenities, and room details.
6. **Loyalty Program:** Offer a loyalty program that rewards customers for their stays, providing incentives to return.
7. **Customer Support:** Allow customers to easily contact the hotel for support or questions, either through a dedicated support page or by phone.
8. **Reservation Management:** Offer a system for managing room reservations, including real-time availability updates, and reservation confirmations.

1.2.9 User characteristics:

User Characteristics for the Hotel Management Website:

1. Customers who are looking for a place to stay, whether for leisure or business purposes.
2. Customers who prefer to book and pay for their hotel reservations online.
3. Customers who value convenience and efficiency in their hotel bookings and transactions.
4. Hotel staff who need a centralized platform to manage daily hotel operations.
5. Hotel management who need real-time access to hotel data and reports for decision-making purposes.

1.3 Constraints

- The complete Software's functionality will be written and shown in English, and users are expected to be fluent in the language.
- Users must have a steady Internet connection(Minimum internet speed \geq 75 Kbps) to access the Software; otherwise, the Software's features will be unavailable.
- **Software Requirements**:- this software will work only on those devices with at least Android 7, iOS 10, Windows XP, macOS 7 or the latest version of respective operating systems.
- **Hardware Requirements**:- a minimum of 2 GB of RAM (for handheld devices) and a minimum of 4 GB of RAM (for desktop devices) is required, failing which the software may produce lag issues.
- It is a multi-user programme that can currently support up to 500 concurrent users. This figure will rise in the future as the number of visitors to our resort grows.

1.4 Assumptions and Dependencies:

Assumptions:

- The users have basic computer skills and know how to use a web browser.
- The users have a stable internet connection to access the website.
- The website will be hosted on a reliable and secure server to ensure maximum uptime and data security.
- The website will be compatible with modern web browsers such as Google Chrome, Mozilla Firefox, and Safari.
- The hotel chain has a defined structure and processes that can be integrated into the website.

Dependencies:

- The website will depend on external payment gateways to process online payments securely.
- The website will depend on third-party APIs to fetch location and weather information.
- The website will depend on the hotel chain's existing database to fetch and update information about reservations, customers, employees, and revenue.
- The website will depend on the hotel chain's existing branding guidelines for the design and layout of the website.

1.5 Specific Requirements

1.5.1 Functional Requirements

Guest Login and Registration

- Users should be able to login using a registered email id and password.
- Users should be able to register as a new user by providing personal information such as name, email, and password.

Room Booking

- Customers should be able to search and book rooms in any of the chain's hotels.
- Customers should be able to view room availability and manage their reservations.
- Customers should be able to select the number of adults and children, date of check-in and check-out, and room type.
- The website should provide real-time updates on room availability.

Online Payment

- Customers should be able to securely make payments for their bookings through the website.
- The website should support multiple payment methods such as credit card, debit card, and net banking.
- The website should provide a payment confirmation page.

Customer Reviews

- Customers should be able to share their experiences and rate their stays.
- Customers should be able to view the reviews and ratings of other customers.

Hotel Information

- Customers should be able to access comprehensive information about each hotel, including location, amenities, and room details.
- The website should provide real-time updates on room availability.

Weddings



Meetings and events:



Amenities:





Loyalty Program

- The website should have a loyalty program that rewards customers for their stays.
- Customers should be able to use their loyalty program points and benefits such as free nights, free spa, etc.

Customer Support

- Customers should be able to contact the hotel for support or questions.
- The website should provide a dedicated support page or a chat option with a customer service executive.

Admin Login

- Admins should be able to login using a registered email id and password.

Reservation Management

- Admins should be able to manage room reservations.
- The website should provide real-time availability updates and reservation confirmations.

1.5.2 Non Functional Requirements:

1. **Security:** The website should implement robust security measures to prevent unauthorized access and protect sensitive data, such as user login information, payment details, and personal information.
2. **Scalability:** The website should be designed to handle a large number of users and transactions, especially during peak periods, without compromising performance or user experience.
3. **Availability:** The website should be available 24/7, with minimal downtime or maintenance windows, to ensure customers can make reservations and access information at any time.

4. **Performance:** The website should be fast and responsive, with minimal loading times and smooth navigation, to ensure customers can easily find and book rooms.

1.6 Performance Requirements

- Fast page load time: The website should load quickly to ensure that customers can easily find and book rooms. The target page load time should be less than 3 seconds.
- High website availability: The website should be available to users at all times, with a target uptime of 99.9%.
- Search engine optimized website: The website should be optimized for search engines, with keyword-optimized content, metadata, and alt tags to improve search engine rankings and attract potential customers.

1.7 Design Constraints

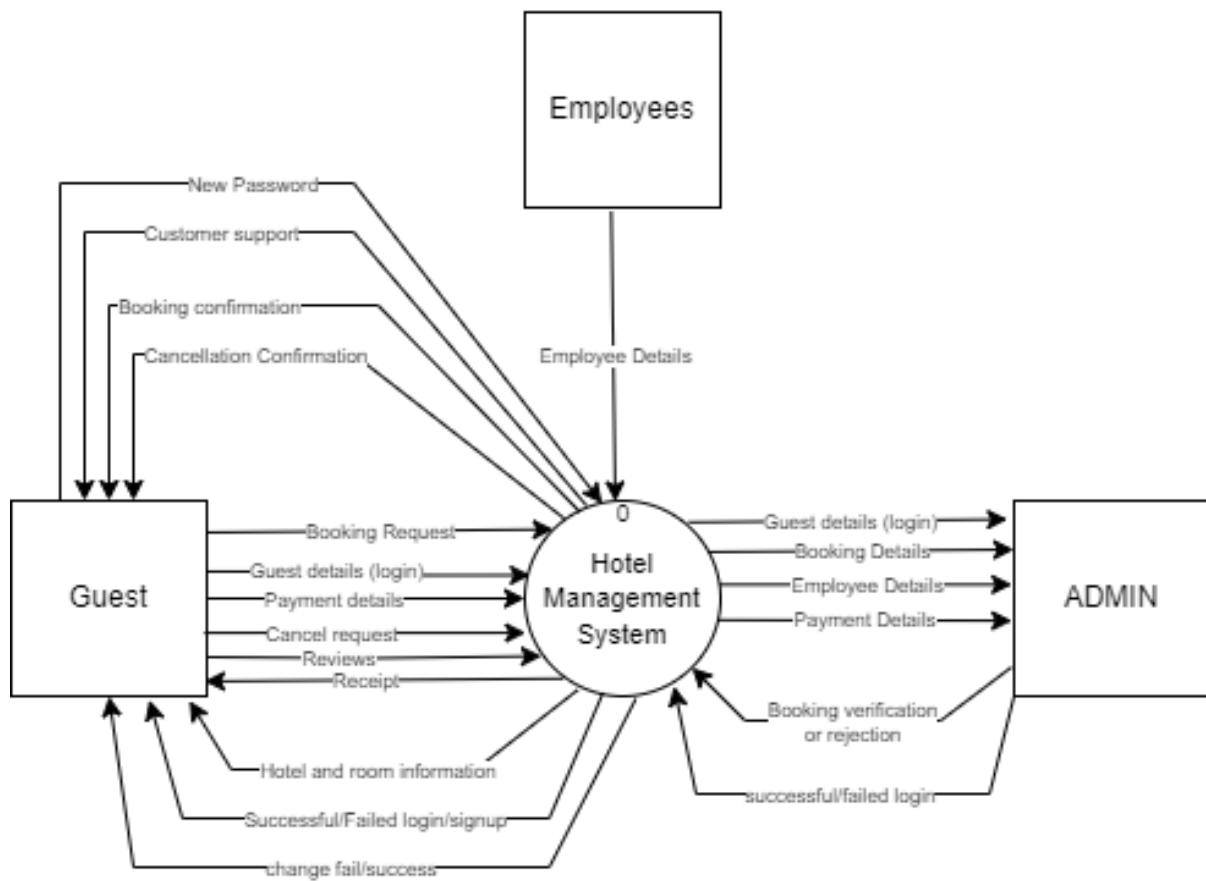
Standard Compliance: All the languages and database formats must conform to the software requirements listed below.

1.7.1 Software System Attributes

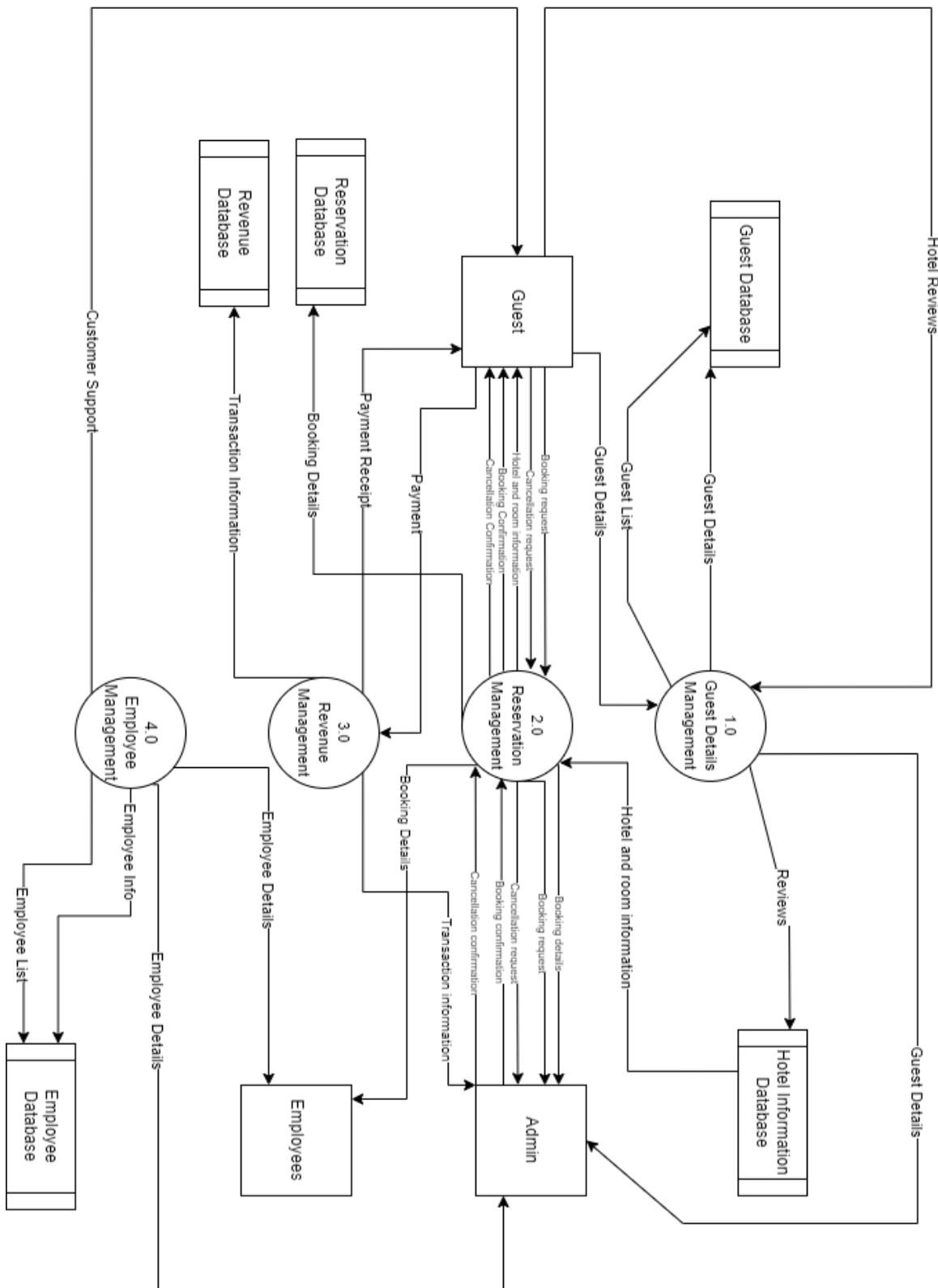
- **Reliability:** The system functions continuously without downtime, allowing hotel staff to access customer data and book rooms with minimal inconvenience.
- **Maintainability:** The system is designed and developed in a way that makes it easy to maintain and update, reducing the likelihood of system failure and downtime.
- **Portability:** The system is portable and runs on multiple platforms, making it accessible to a wide range of users and reducing the risk of platform-specific issues.
- **Usability:** The system is user-friendly and intuitive, making it easy for both hotel staff and customers to navigate and perform tasks efficiently.

1.8 Data Flow Diagram

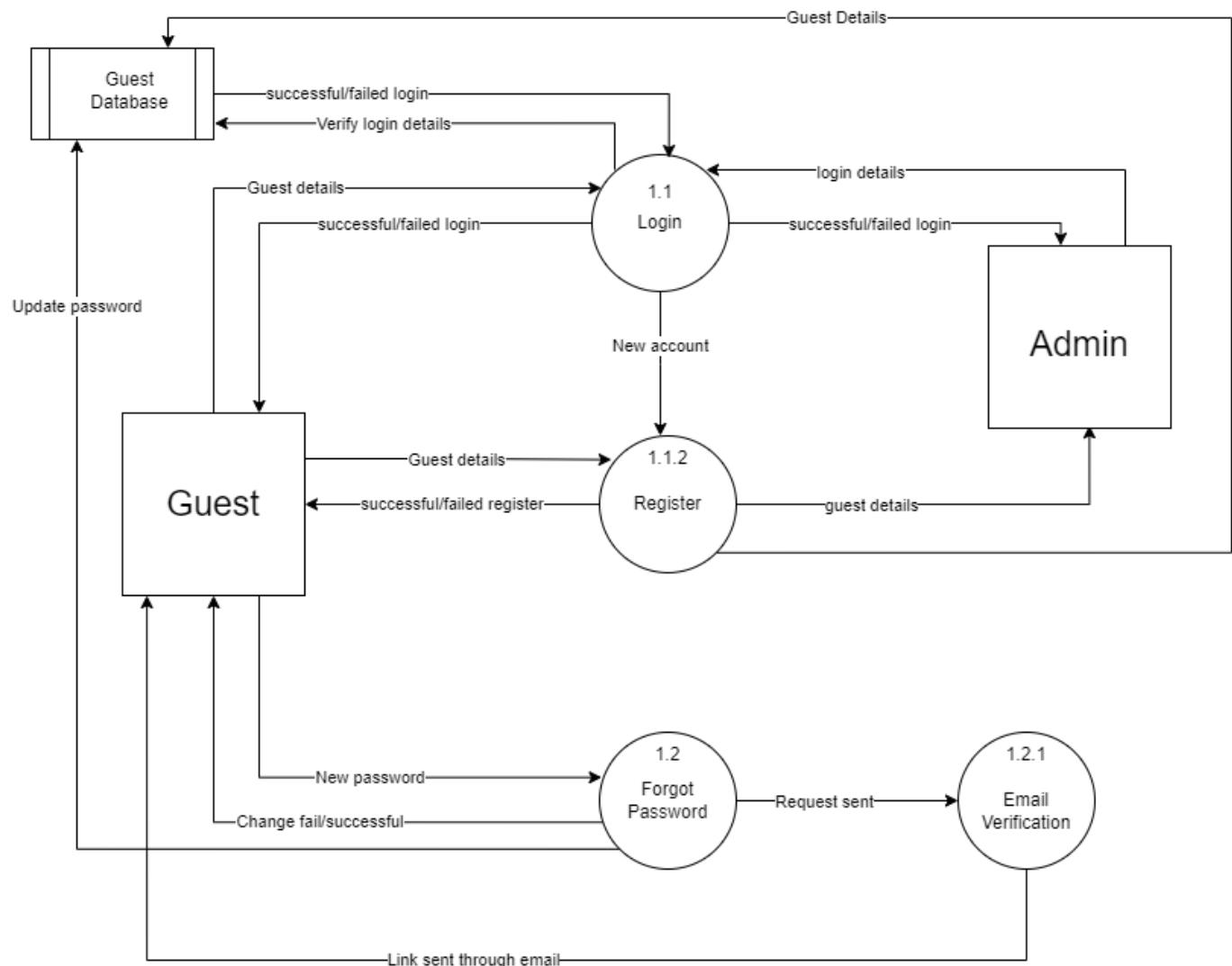
LEVEL 0



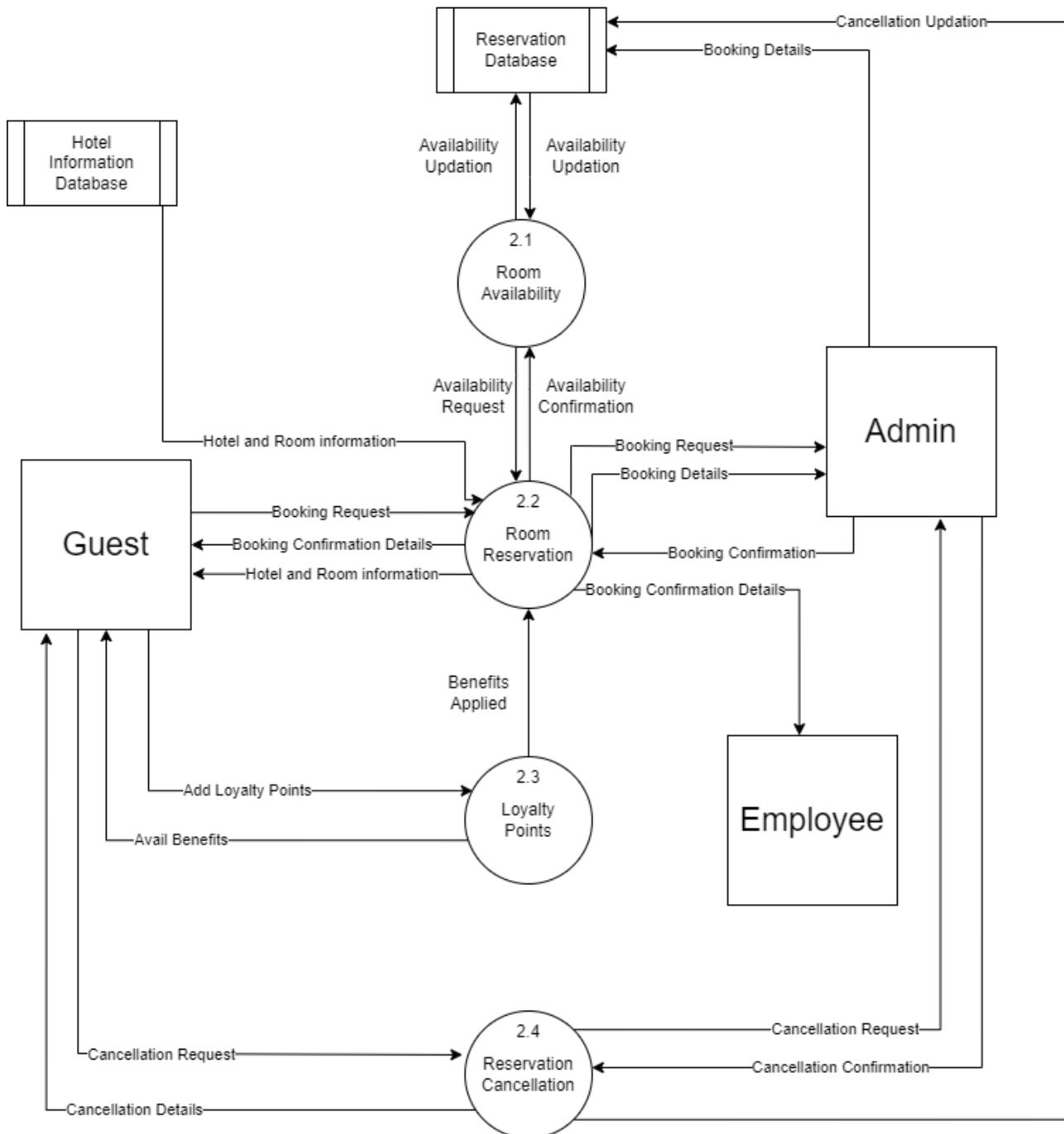
LEVEL 1



LEVEL 2 - GUEST DETAILS MANAGEMENT



LEVEL 2 - RESERVATION MANAGEMENT



1.9 DATA DICTIONARY

Data Flow	Description
Booking details	No_of_rooms + room_type + no_of_adults + no_of_children
Employee details	emp_id + emp_name + emp_address + emp_phone + emp_salary + emp_department + emp_email
Guest details	Guest_id + name + contact_email + contact_phone + guest_address + login_password + loyalty_points
Hotel and room information	hotel_id + hotel_name + location + room_type + price + number_of_rooms + available_rooms + number_of_reviews + other_facilities
Login details	guest_id + guest_password
Payment details	guest_id + transaction_amount
Reviews	any message by the guest
Transaction information	Transaction_id + transaction_date + transaction_amount

1.10 Use - Cases

1.10.1 Login

Description:

Customers and administrators may quickly log in using this use case.

Actors:

Admin and user

Flow of Events:

Basic Flow: The actor is prompted to provide their username and password by the system. The system checks the actor's credentials after they are entered.

Alternative Flow: The system provides an error notice if the actor inputs an invalid username or password. The performer has the option of canceling or trying again.

Pre-Condition:

The customer actor must be a registered user on the hotel administration portal as a prerequisite.

Post-Condition:

The customer actor can view their account information after successfully login in.

1.10.2 Register

Description:

This use case is for customers who want to register for a new account on the resort portal but do not already have one.

Actors:

User

Flow of events:

Basic Flow: The customer creates a new account by providing their email address and a new password.

Alternative Flow: The system displays an error notice if the user inputs an invalid email address or an illegal password. The customer has the option of cancelling the sign-up procedure or retrying the basic sequence.

Pre Condition:

No prerequisites

Post condition:

After successfully signing up, the consumer can log in to the system and view their information.

1.10.3 Book Rooms

Description:

Customers can reserve rooms or cottages at our resort using this use case.

Actors:

User

Flow of Events:

Basic Flow: Clients are asked for information such as check-in and check-out dates, the number of guests, their desired room or cottage type, and any discounts that may be available. The submitted data is processed by the system.

Alternative Flow: The system displays a message expressing regret for the inconvenience if rooms or cottages are not available owing to excessive demand or other problems. Error notifications could also be produced as a result of payment processing mistakes.

Pre Condition: Clients must have accounts registered on our resort portal as a prerequisite.

Post-Condition: Customers get an email with their itinerary information after making a successful reservation.

1.10.4 Booking cancellation

Description:

Customers may cancel their reserved hotel rooms or cottages using this use case.

Actors:

User

Flow of the events:

Basic Flow: When a customer requests to cancel a reservation, the system informs them of the refund policy in accordance with the set rules and guidelines.

Alternative Flow: The system displays an error message, indicating a failed attempt to cancel the reserved rooms, if the customer violates the terms and conditions of the return policy during the cancellation process.

Pre Condition:

Reservations for rooms are required.

Post-condition:

The consumer will receive a confirmation email along with the reimbursement after a successful cancellation.

1.10.5 Payment:

Description:

This use case describes how payments are made using the system.

Actors:

User

Flow of events:

Basic flow: After receiving an invoice, the user goes to the payment dashboard to choose their preferred method of payment and finish the transaction.

Alternate flow: N/A

Pre-Condition:

Users must have an account that is registered.

The user needs a bank account.

Post-Condition:

The reservation for the room has been finalized.

The user can download the receipt for payment.

1.10.6 Customer Support

Description:

This enables the user to specify their problems or questions related to the hotel reservation management system.

Actors:

User

Flow of Events:**Basic Flow:**

1. The client gets in touch with the authorities by phone, chat, or email.
2. By requesting the client's account information, the support agent can confirm the client's identity.
3. The customer service professional listens to their complaint and notes the specifics.

4. The consumer is given a solution after the authorities make an effort to diagnose the problem.
5. If the support agent is unable to handle the problem, it is escalated to the relevant department or higher-level support.

Alternate Flows:

- The support agent may request extra verification methods if the customer is unable to submit their account details.
- If it takes more time to fix the problem than expected, the support agent may provide the client an approximate time frame and update them on the situation.

Preconditions:

On the platform, the customer has registered.

The client is having a problem or has a query about the platform.

Postconditions:

The concern or question of the client has been satisfactorily addressed.

For future reference, the authorities have entered the problem and its resolution into the system.

1.10.7 Reservation Management

Description:

The management of room reservations is covered in this use case, along with real-time availability updates and reservation confirmations. The admin can view, add, edit, or cancel reservations.

Actors:

Admin

Flow of Events:

Basic Flow: The administrator oversees reservations, manages them, updates hotel availability, and verifies them.

Alternative Flow: Not applicable

Preconditions:

The admin has logged in with his email id and password.

Postconditions:

The admin approved or canceled the reservation request as per the availability.

1.10.8 Review

Description:

The user's ability to provide input is described in this use case.

Actor:

User

Flow of events:

Basic Flow:

1. The system asks the user to provide input.
2. The user provides feedback.
3. The hotel database is where the admin keeps the ratings and feedback.

Alternative Flow: Not applicable

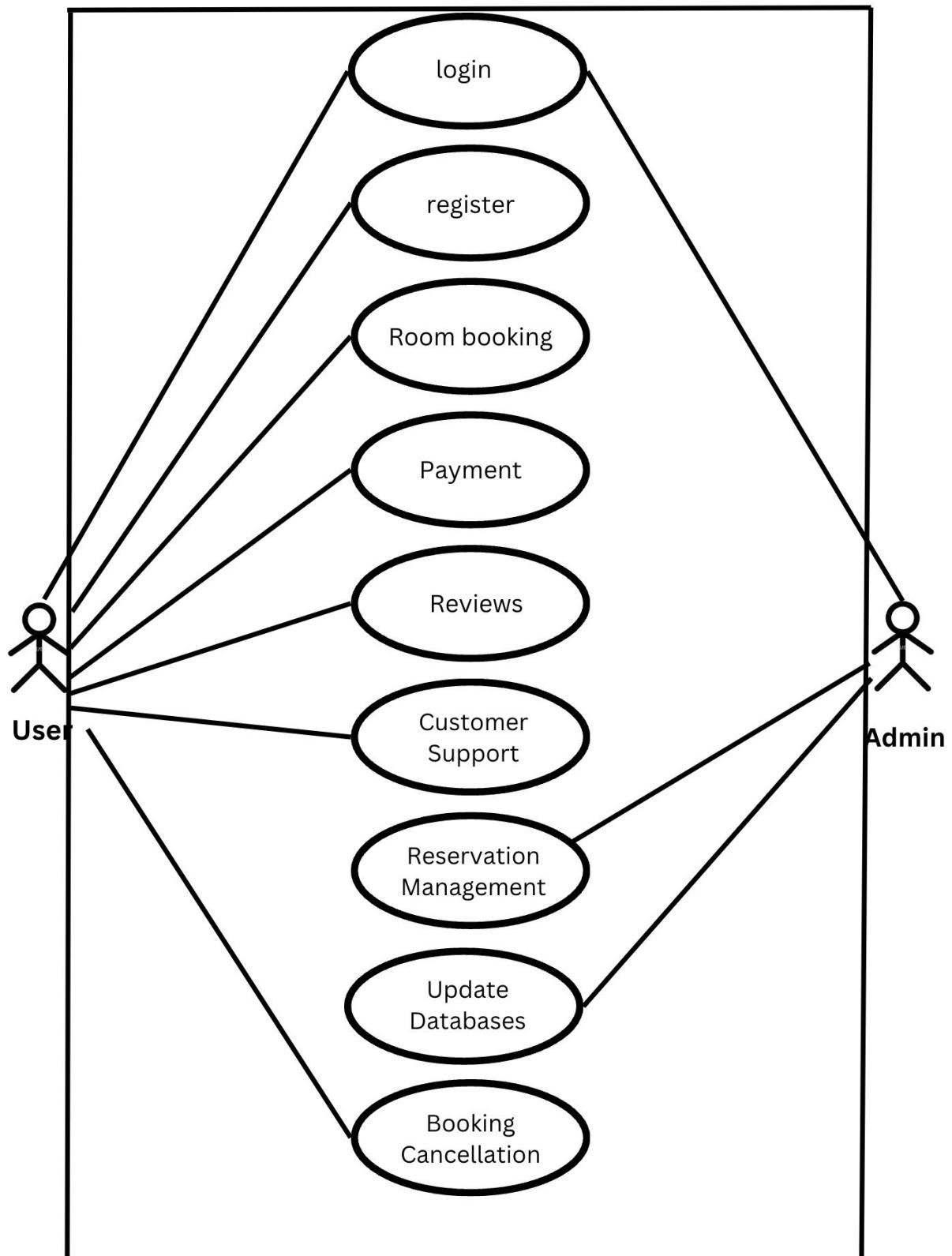
Pre condition:

Users must log into the system before proceeding.

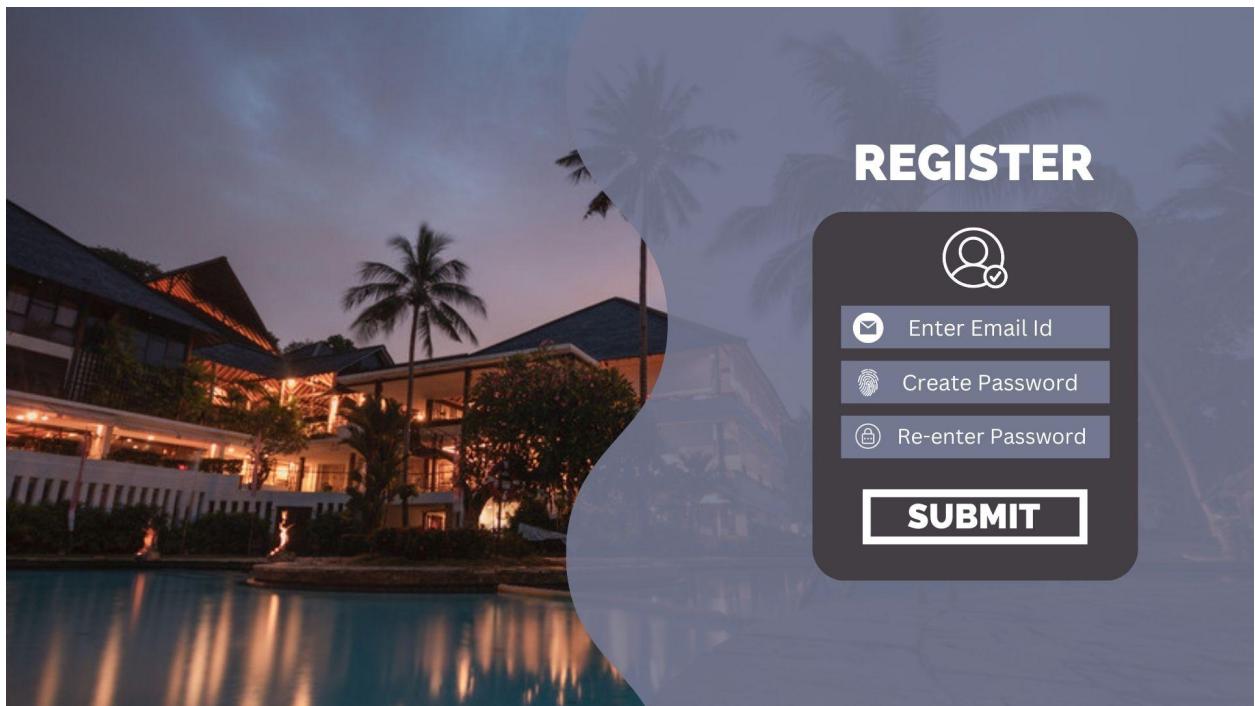
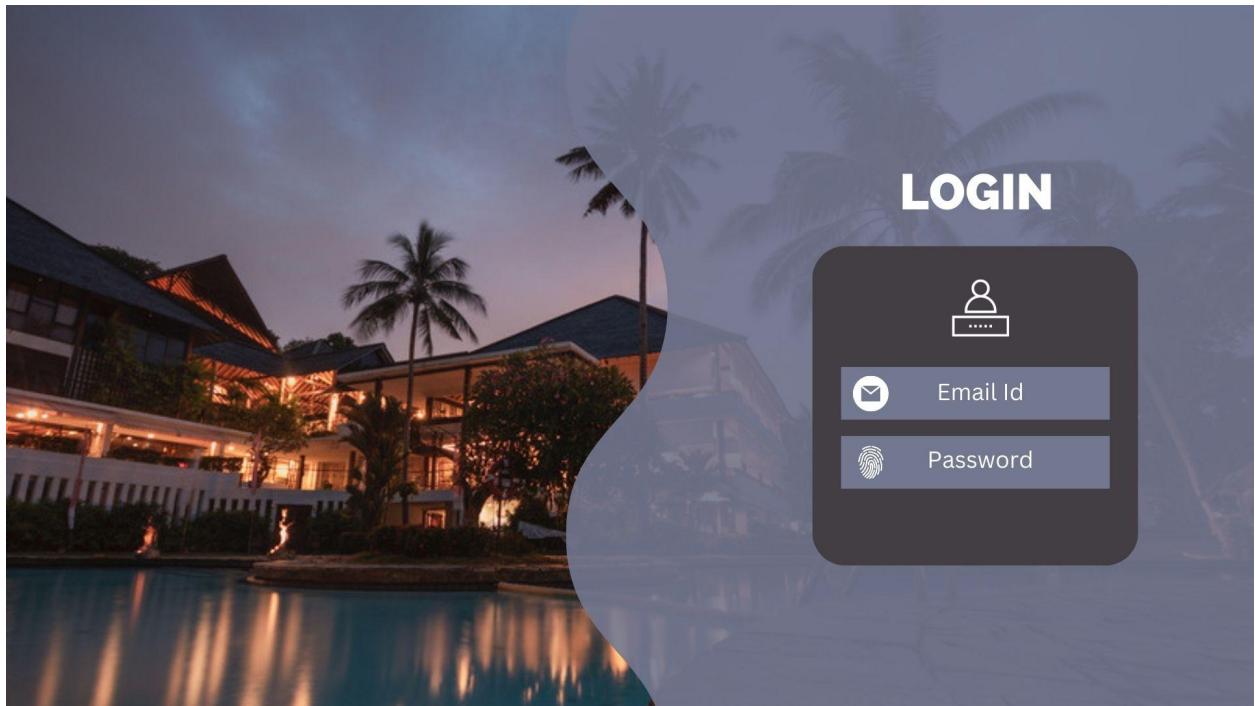
Post condition:

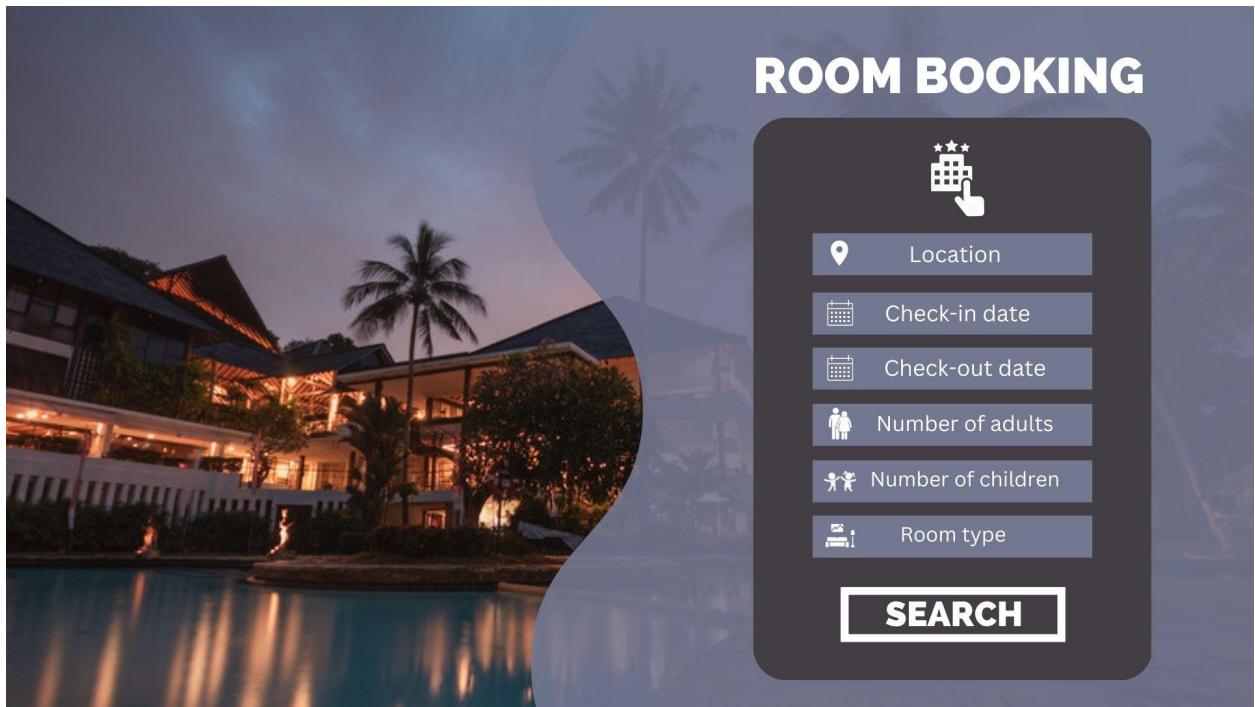
Your feedback has been successfully sent.

Use-Case Diagram :-



1.11 User Interfaces





2. Estimations

2.1 Function Points

Information Domain	Estimation Count	Weighing factor			Total
		Simple	Average	Complex	
External Input	9	3	4	6	36
External output	10	4	5	7	50
External enquiries	0	3	4	6	0
No. of Logical files	5	7	10	15	50
External interface files	0	5	7	10	0

Unadjusted Function Point(UFP): $36+50+0+50+0 = 136$

Complexity Adjustment Factor(CAF): $0.65+(0.01 \cdot \sum f_i)$
 $= 0.65+(0.01 \cdot 49) = 1.14$

Function Point Metric(FP): $UFP \cdot CAF = 136 \cdot 1.14$
 $= 155.04$

VALUE ADJUSTMENT FACTOR (VAF):

Sno	Questions	Grade Value
1	Does the system require reliable backup and recovery?	5
2	Are specialised data communication required to transfer information to or from the application?	4
3	Are there distributed processing functions?	2
4	Is performance critical?	4
5	Will the system run in an existing, heavily utilized operational environment?	4
6	Does the system require Online data entry?	5

7	Does the online data entry require the input transaction to be built over multiple screen or operation?	5
8	Are the ILFs(Internal Logical Files) updated online?	4
9	Are the inputs, outputs, files or enquiries complex?	2
10	Is the internal processing complex?	2
11	Is the code designed to be reusable?	3
12	Are conversion and installations included in design?	2
13	Is the System designed for multiple Installation in different organisations?	2
14	Is the application designed to facilitate change and ease of use by the user?	5

Value adjustment factor = $\Sigma f(i)$ = 49

Rate on each factor on scale 0-5:

- 0: No influence
- 1: Incidental
- 2: Moderate
- 3: Average
- 4: Significant
- 5: Essential

2.2 EFFORTS

We can estimate the effort put in to build this software system by dividing the total number of functional points with the average productivity of each individual employed to develop this software.

Hence,

Effort= Total FP/Average productivity

Using this, Net cost of building this software can be calculated as follows:

Total project cost = Effort × Labour rate

3. SCHEDULING

S.No.	WORK TASKS	JAN	FEB	MAR	APR	MAY
1	Identify needs and constraints Establish problem statement Analysis of each solution					
	Milestone: Detailed description of the project completed					
	Identifying scope of the proposed system Description of requirements Identify desired output/control/[Input (OCI)] Input output Interfaces Review OCI Revise OCI as required					
	Milestone: SRS phase completed					
2						
3	Designing DFD Architectural Design Database design					
	Milestone: Designing phase completed					
4	Defining Use Case Use case diagram					
	Milestone: Use case completed					
5						
6	Function point estimation Effort Calculation					
	Milestone: Estimation phase completed					
7	Identifying risk of system Developing Risk Table Developing RMM Table					
	Milestone: RMM phase completed					
8	Formulate system architecture Generate code Create sample database					
	Milestone: Development phase completed					
	Identifying testing strategy Developing Flow graph Resolve Derivations					
	Milestone: Testing phase completed					

4. RISK MANAGEMENT:

S.No.	RISKS	CATEGORY	PROBABILITY	IMPACT
1.	Delays in Transactions due to payment gateway server issues	TE	30%	2
2.	Technical Compatibility Issues (with different operating systems, architectures, devices etc.)	TE	30%	2
3.	Data Breach	TE	10%	2
4.	Website Downtime due to server congestion	TE	10%	2
5.	In later phases, requirements are modified.	PD	10%	2

LEGENDS:

TE - Technological Risk

BU - Business Impact Risk

PD - Product Definition Risk

Impact values:

- 1—catastrophic
- 2—critical
- 3—marginal
- 4—negligible

RISK MITIGATION, MONITORING AND MANAGEMENT

A. DELAY IN TRANSACTION:

Risk Mitigation:

- Proper payment gateway mechanisms for seamless transfers of money.
- Ensure the minimum speed of internet connection is 75 kbps

Risk Monitoring:

- Encouraging other modes of online payments like UPI transactions, or one can go to the nearest branch/office for room booking.

Management:

- Communication channels between the parties should be made hassle-free.

B. Technical Compatibility Issues:

Risk Mitigation:

- Developers should ensure that the website is tested on a variety of operating systems and browsers to ensure compatibility. They may also consider using cross-platform frameworks or technologies that can work on multiple operating systems.
- Developers should ensure that the website is designed with responsive design principles, allowing it to adapt to different screen sizes and resolutions. They may also consider developing native apps for different devices to ensure optimal performance.
- Developers should ensure that the website is designed with architecture-agnostic technologies or tools that can work on both architectures. They may also consider using cloud-based infrastructure that can scale to meet the needs of different architectures.

Risk Monitoring:

- Developers should monitor user feedback and error logs to identify any issues related to operating system compatibility.
- Developers should monitor user feedback and analytics data to identify any issues related to device compatibility.
- Developers should monitor user feedback and error logs to identify any issues related to architecture compatibility.

Risk Management:

- If issues are identified, developers should work quickly to address them, potentially releasing updates or patches to improve compatibility.
- If issues are identified, developers should work to improve compatibility by optimizing the website or releasing updates for native apps.
- If issues are identified, developers should work to improve compatibility by optimizing the website or releasing updates to support different architectures.

C. DATA BREACH:

Risk Mitigation:

1. Data Encryption: All sensitive data such as customer details, payment information, and employee records will be encrypted and stored securely in the database.
2. Access Control: Role-based access control will be implemented to ensure that only authorized personnel can access sensitive data.

Risk Monitoring:

1. Security Checks: Regular security checks will be conducted to ensure the system is secure and protected against potential threats.
2. Monitoring and Logging: Monitoring and logging of system activities will be conducted to identify any suspicious behaviour or unauthorized access attempts.

Risk Management:

1. Risk Assessment: Regular risk assessments will be conducted to identify potential data breach risks and implement necessary measures to mitigate them.
2. Risk Response: Appropriate measures will be taken to respond to any identified risks and prevent them from causing damage to the system and its users.
3. Continuous Improvement: The risk management process will be continuously improved based on the feedback received and any new risks that may arise.

B. **WEBSITE DOWNTIME:**

Risk Mitigation:

1. Develop and implement a robust and scalable hosting solution to ensure high availability and minimize downtime.
2. Set up a monitoring system to continuously monitor the website's performance and detect any issues promptly.
3. Conduct regular maintenance and updates to prevent any technical issues that could lead to downtime.
4. Establish redundancy and backup systems to ensure that critical data and services are not lost during downtime.

Risk Monitoring:

1. Continuously monitor the website's performance and uptime through automated monitoring tools.
2. Conduct regular performance and security audits to identify potential vulnerabilities that could lead to downtime.
3. Implement a communication plan to notify stakeholders and customers of any downtime or maintenance schedules.

Risk Management:

1. Create a comprehensive disaster recovery plan to minimize the impact of website downtime.
2. Test the disaster recovery plan regularly to ensure that it is up-to-date and effective.
3. Establish a crisis management team to handle any downtime events and quickly restore the website to normal operation.

5. DESIGN

5.1 DATA DESIGN

GUEST

ATTRIBUTE NAME	DATA TYPE	DATA SIZE	CONSTRAINT
GUEST_ID	INT	8	PRIMARY KEY
NAME	VARCHAR	45	NOT NULL
CONTACT_EMAIL	VARCHAR	30	NOT NULL
CONTACT_PHONE	INT	10	NOT NULL
GUEST_ADDRESS	VARCHAR	100	NOT NULL
LOGIN_PASSWORD	VARCHAR	32	NOT NULL
LOYALTY_POINTS	INT	4	-

HOTEL INFORMATION

ATTRIBUTE NAME	DATA TYPE	DATA SIZE	CONSTRAINT
HOTEL_ID	INT	8	PRIMARY KEY
HOTEL_NAME	VARCHAR	60	NOT NULL
LOCATION	VARCHAR	30	NOT NULL
ROOM_TYPE	VARCHAR	15	NOT NULL
PRICE	INT	6	NOT NULL
NUMBER_OF_ROOMS	INT	2	NOT NULL
AVAILABLE_ROOMS	INT	2	NOT NULL
NUMBER_OF_REVIEWS	INT	6	-
OTHER_FACILITIES	VARCHAR	45	-

REVENUE

ATTRIBUTE NAME	DATA TYPE	DATA SIZE	CONSTRAINT
TRANSACTION_ID	INT	8	PRIMARY KEY
TRANSACTION_DATE	DATE	8	NOT NULL
TRANSACTION_AMOUNT	INT	6	NOT NULL
GUEST_ID	INT	8	FK (GUEST TABLE)

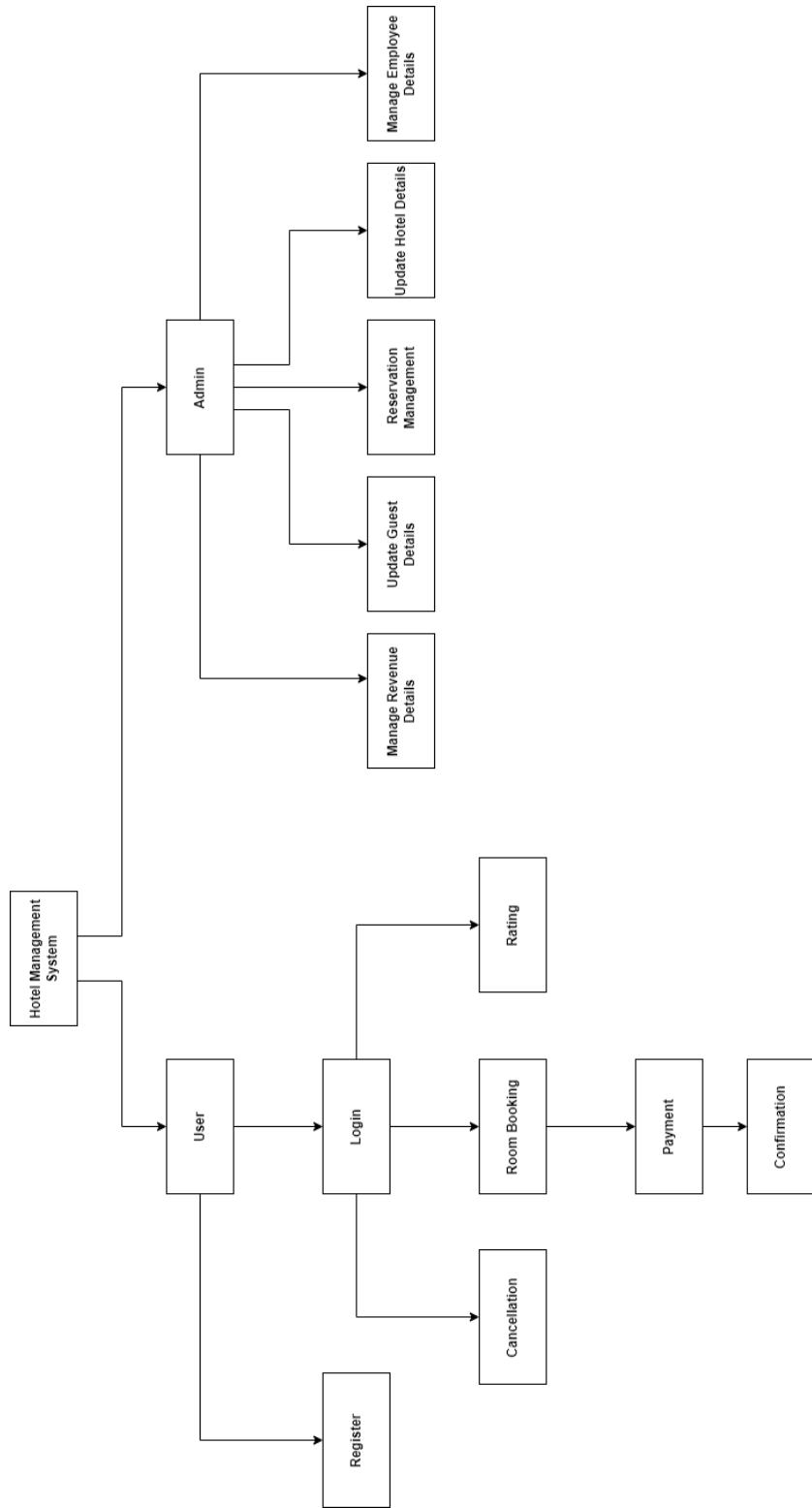
EMPLOYEE

ATTRIBUTE NAME	DATA TYPE	DATA SIZE	CONSTRAINT
EMP_ID	INT	8	PRIMARY KEY
EMP_NAME	VARCHAR	45	NOT NULL
EMP_ADDRESS	VARCHAR	100	NOT NULL
EMP_PHONE	INT	10	NOT NULL
EMP_SALARY	INT	6	NOT NULL
EMP_DEPARTMENT	VARCHAR	15	NOT NULL
EMP_EMAIL	VARCHAR	45	NOT NULL
HOTEL_ID	INT	8	FK (HOTEL INFORMATION)

RESERVATION

ATTRIBUTE NAME	DATA TYPE	DATA SIZE	CONSTRAINT
GUEST_ID	INT	8	FK(GUEST TABLE)
HOTEL_ID	INT	8	FK (HOTEL INFORMATION)
NO_OF_ROOMS	INT	2	NOT NULL
ROOM_TYPE	VARCHAR	15	NOT NULL
NO_OF_ADULTS	INT	2	-
NO_OF_CHILDREN	INT	2	-
TRANSACTION_ID	INT	8	FK(REVENUE TABLE)

5.2 SYSTEM DESIGN



6. CODING

```
def guest_login():
    email = input("Enter email: ")
    password = input("Enter password: ")
    # Authenticate user
    # Code for user authentication goes here
    # returns true or false

def guest_registration():
    name = input("Enter name: ")
    email = input("Enter email: ")
    password = input("Enter password: ")
    # Store user information in the database
    # Code for user registration goes here

def room_booking():
    # Display available hotels and their information
    # Code to display hotels goes here
    # Prompt user to select a hotel
    hotel_choice = int(input("Enter hotel choice: "))

    # Display available rooms in the selected hotel
    # Code to display available rooms goes here

    # Prompt user to select a room
    room_choice = int(input("Enter room choice: "))

    # Prompt user to enter check-in and check-out dates
    check_in_date = input("Enter check-in date (YYYY-MM-DD): ")
    check_out_date = input("Enter check-out date (YYYY-MM-DD): ")

    # Prompt user to enter the number of adults and children
    num_adults = int(input("Enter number of adults: "))
    num_children = int(input("Enter number of children: "))

    # Confirm the reservation and process payment
    # Prints out the reservation number
    print("reservation number:")
    # Code for reservation confirmation and payment goes here

    online_payment()
```

```

def booking_cancellation():
    # Prompt user to input reservation number
    #if reservation number is valid, cancel the reservation
    print("reservation cancelled")
    #else
    print("enter valid reservation number")

def online_payment():
    # Prompt user to select a payment method
    payment_method = int(input("Select payment method:\n1. Credit Card\n2. Debit
Card\n3. Net Banking\n"))
    # Process the payment
    # Code for payment processing goes here

def customer_reviews():
    # Display reviews and ratings of the hotels
    # Code for displaying reviews goes here

def hotel_information():
    # Display comprehensive information about each hotel
    # Code for displaying hotel information goes here

def loyalty_program():
    # Display loyalty program benefits and rewards
    # Code for displaying loyalty program information goes here

def customer_support():
    # Display support page or chat option with a customer service executive
    # Code for displaying customer support options goes here

def admin_login():
    email = input("Enter email: ")
    password = input("Enter password: ")
    # Authenticate admin user
    # Code for admin authentication goes here
    #if admin login is successful, reservation management rights are given

def reservation_management():
    # Display list of room reservations
    # Code for displaying room reservations goes here

def guestfunctions():
    #code for calling for all the guest functions

```

```

def main():
    while True:                                (1)
        print("1. Guest Registration")          (2)
        print("2. Guest Login")
        print("3. Admin Login")
        print("0. Exit")

    choice = input("Enter choice: ")
    if choice == "1":                          (3)
        guest_registration()                  (4)

    elif choice == "2":                        (5)
        guest_login()                         (6)
        guest_functions()

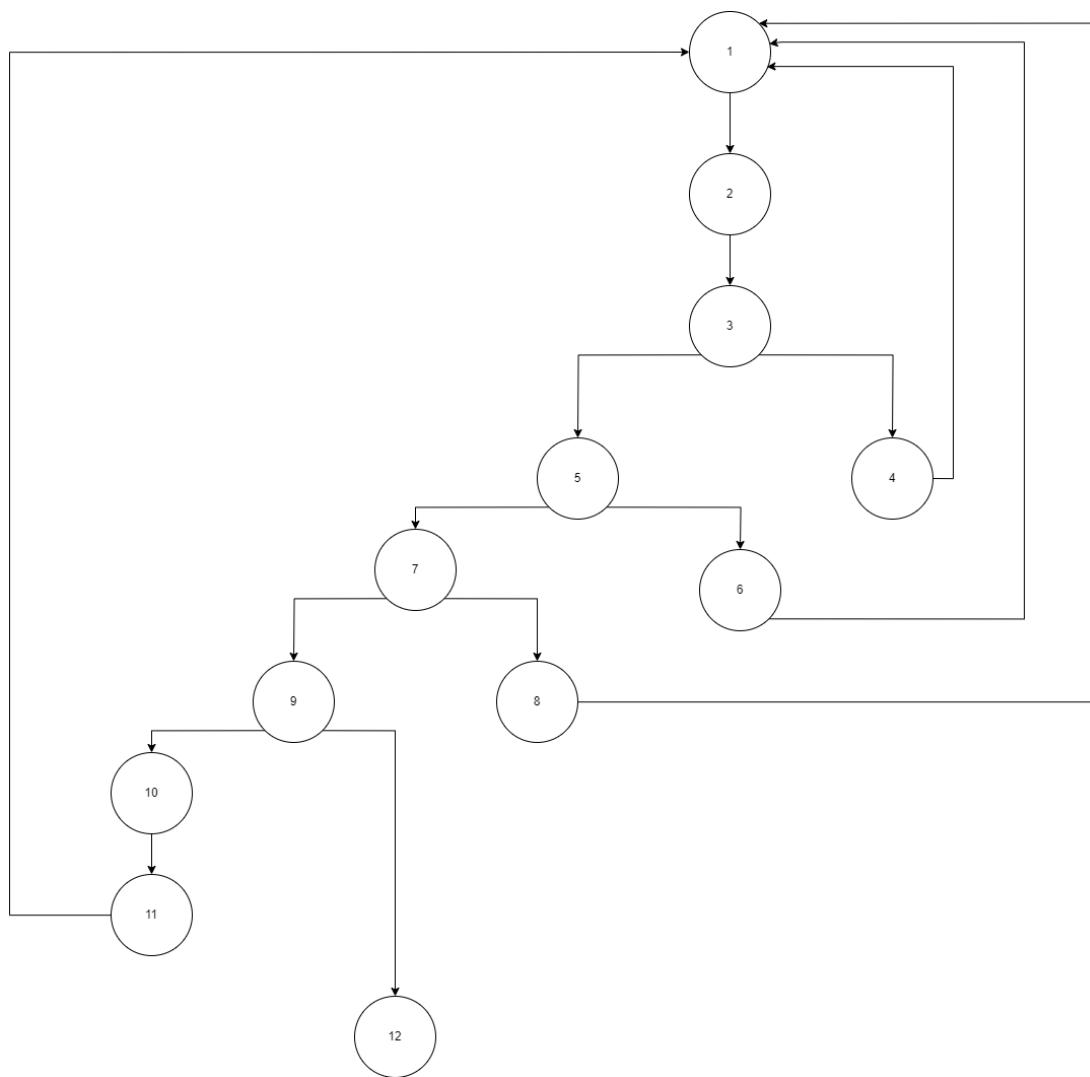
    elif choice == "3":                        (7)
        admin_login()                         (8)
    elif choice == "0":                        (9)
        break
    else:                                     (10)
        print("Invalid Choice")               (11)

#END                                         (12)

```

7. TESTING

CONTROL FLOW DIAGRAM



CYCLOMATIC COMPLEXITY:

Number Of Regions = 5

Number Of Predicate Nodes = 4

Complexity = $4+1= 5$

Complexity = $E-N+2 = 15-12+2 = 5$

INDEPENDENT PATHS:

- 1-2-3-4-1-2-3-5-7-9-12
- 1-2-3-5-6-1-2-3-5-7-9-12
- 1-2-3-5-7-8-1-2-3-5-7-9-12
- 1-2-3-5-7-9-10-11-1-2-3-5-7-9-12

BOUNDARY VALUE ANALYSIS

Module: Online Payment

Description: This module enables customers to securely make payments for their bookings through the website. The website should support multiple payment methods such as credit card, debit card, and net banking. The website should provide a payment confirmation page.

Inputs:

- Credit/Debit Card Number (16 digits)
- Expiration Date (MM/YY format)
- CVV Code (3 digits)
- Payment Amount (Rs.)

Input Range:

- Credit Card Number: 1000000000000000 to 9999999999999999
- Expiration Date: 04/23 to 12/99
- CVV Code: 001 to 999
- Payment Amount: Rs.1 to Rs.1,00000

Expected Output:

If all inputs are correct: Payment Successful

If any input is incorrect: Error

Test Cases:

Using the Single Valuation Theory, $4n+1$ test cases can be designed which, in our case, is equal to 17

Test Case #	Credit Card Number	Expiration Date	CVV Code	Payment Amount	Expected Output
1	1000000000000000	07/23	123	Rs.50000	Payment Successful
2	1000000000000001	07/23	123	Rs.50000	Payment Successful
3	9999999999999998	07/23	123	Rs.50000	Payment Successful
4	9999999999999999	07/23	123	Rs.50000	Payment Successful
5	5000000000000000	04/23	123	Rs.50000	Payment Successful
6	5000000000000000	05/23	123	Rs.50000	Payment Successful
7	5000000000000000	12/98	123	Rs.50000	Payment Successful
8	5000000000000000	12/99	123	Rs.50000	Payment Successful
9	5000000000000000	08/23	1	Rs.50000	Payment Successful
10	5000000000000000	08/23	2	Rs.50000	Payment Successful
11	5000000000000000	08/23	998	Rs.50000	Payment Successful
12	5000000000000000	08/23	999	Rs.50000	Payment Successful
13	5000000000000000	08/23	123	Rs.1	Payment Successful
14	5000000000000000	08/23	123	Rs.2	Payment Successful
15	5000000000000000	08/23	123	Rs.99999	Payment Successful
16	5000000000000000	08/23	123	Rs.100000	Payment Successful
17	5000000000000000	08/23	123	Rs.0	Error

8. REFERENCES

- Aggarwal, K. K., & Singh, Y. (2007). Software Engineering. 3rd edition. New Age International publishers
- Pressman, R. S., & Maxim, B. R. (2015). Software Engineering: A Practitioner's Approach.8th edition. McGraw-Hill.