

EXPERIMENT 1

AIM – To prepare the problem statement of Hotel Management system.

Minimum Hardware Specifications:

1. Processor (CPU): A basic dual-core processor (e.g., Intel Core i3 or AMD Ryzen 3) or equivalent is sufficient for basic tasks.
2. Memory (RAM): 4 GB of RAM is the minimum for a smooth experience. 8 GB or more is recommended for better multitasking and responsiveness.
3. Storage: At least 16 GB of storage is required for the operating system and essential software. A solid-state drive (SSD) is recommended for faster system performance and responsiveness.

Minimum Software Specifications:

1. Operating System: You'll need a compatible operating system, such as Windows, macOS, iOS or Android OS. Older hardware may work well with lightweight Linux distributions.
2. Web Browser: A modern web browser like Google Chrome, Mozilla Firefox, or Microsoft Edge for web browsing and online tasks.

Theory:

Traditionally, a hotel property management system was defined as a system that enabled a hotel or group of hotels to manage front-office capabilities, such as booking reservations, guest check-in/checkout, room assignment, managing room rates, and billing.

Problem Statement: The existing hotel management system faces several issues and challenges that need to be addressed through modernization. The key problems include:

SCOPE OF SOFTWARE PROJECTS:-

1)Reservation System:

Online booking for guests through the hotel's website.

Integration with third-party booking platforms.

Real-time updates on room availability.

Front Desk Operations:

2)Efficient check-in and check-out processes.

Guest profiling to store personal preferences and information.

Room assignment and management.

Billing and Invoicing:

3)Automated billing for room charges, additional services, and taxes.

Integration with point-of-sale systems for accurate invoicing.

Point of Sale (POS) Integration

4)Managing sales transactions for on-site services such as restaurants, bars, and spa.

Integration with inventory management for stock control.

Housekeeping Management:

5)Tracking and managing room cleaning status.

Task assignment for housekeeping staff.

Reporting and Analytics:

6)Generation of reports on occupancy rates, revenue, and other key performance indicators.

Business intelligence for informed decision-making.

Guest Relationship Management (CRM):

7)Building and maintaining a database of guest information.

Personalization of guest experiences based on historical data.

Channel Management:

8)Synchronization of room availability and rates across various online travel agencies (OTAs).

Management of reservations from multiple booking channels.

Security and Authentication:

9)Implementation of secure access controls to sensitive information.

Monitoring and logging of user activities for security purposes.

Mobile Accessibility:

10)Development of a mobile app for guests to facilitate check-in, check-out, and access to information.

Mobile alerts and notifications for reservation updates.

Integration with Third-Party Systems:

11)Integration with accounting software for financial management.

Integration with property management systems (PMS) and customer relationship management (CRM) tools.

Multi-Property Support:

12)Ability to manage multiple properties for hotel chains or groups.

Centralized control and reporting across all properties.

Feedback and Review Management:

13)Collection and analysis of guest feedback.

Integration with online review platforms.

Training and Support:

14)Provision of training for hotel staff on system usage.

Ongoing technical support and updates.

Compliance and Regulations:

15)Adherence to industry regulations and standards.

Data privacy and security compliance.

Scalability:

16)Design the system to accommodate the future growth of the hotel or hotel chain.

Objectives: To address these challenges, the modernized railway reservation system should aim to achieve the following objectives:

1. Enhance User Experience:

- Develop an intuitive and user-friendly interface for easy booking and ticket management.

2. Scalability:

- Ensure that the system can accommodate a growing number of passengers without performance degradation.

3. Security:

- Implement robust security measures to protect passenger data and financial transactions.

4. Integration:

- Enable seamless integration with other transportation systems and third-party services.

5. Real-time Information:

- Provide real-time updates on train schedules and relevant travel information to passengers.

6. Accessibility:

- Ensure the system is accessible to all passengers, regardless of their abilities.

7. Maintenance and Support:

- Establish a proactive maintenance strategy and responsive support system.

8. Resource Allocation:

- Implement efficient seat allocation and pricing strategies.

9. Data Analytics:

- Utilize data analytics for demand prediction and informed decision-making.

The modernization of the railway reservation system aims to provide an efficient, secure, and user-friendly platform that meets the evolving needs of passengers while improving the overall quality of service and operational efficiency.

EXPERIMENT 2

AIM – To do the requirement analysis and develop the Software Requirement Specification sheet (SRS) for suggested system.

SOFTWARE REQUIREMENTS SPECIFICATION

HOTEL MANAGEMENT SYSTEM

VERSION 1.0

SEPTEMBER 12,2023.

TABLE OF CONTENT

1. INTRODUCTION

1.1. PURPOSE

1.2. DEFINATION, ACRONYMS AND ABBREVIATION

1.3. SCOPE

2. OVERALL DESCRIPTION

2.1. PRODUCT PERSPECTIVE

2.1.1 USER INTERFACE

2.1.2 HARDWARE INTERFACE

2.1.3 SOFTWARE INTERFACE

2.1.4 COMMUNICATING INTERFACE

2.1.5 OPERATING ENVIRONMENT

2.1.6 OPERATIONS

2.2. PRODUCT FUNCTION

2.3. USER CHARACTERSITICS

2.4. CONSTRAINTS

3. SPECIFIC REQUIREMENT

3.1. EXTERNAL INTERFACE

3.2. PERFORMANCE REQUIREMENTS

4. DOCUMENT APPROVAL

5. OTHER NON-FUNCTIONAL REQUIREMENT

5.1. PERFORMANCE

5.2. USABILITY

5.3. RELIABILITY

5.4. SECURITY

5.5. COMPATIBLITY

5.6. DATA BACKUP AND RECOVERY

5.7. COMPLANCE

1. INTRODUCTION:

A Software Requirements Specification (SRS) document for a Hotel Management System serves as a blueprint outlining the functional and non-functional requirements of the system. It provides a comprehensive overview of what the system is expected to accomplish, its features, constraints, and the criteria for testing its functionality.

This introduction sets the stage for the Hotel Management System project by outlining its purpose, scope, intended audience, references, overview, assumptions, dependencies, and constraints. Subsequent sections of the SRS document would delve into more specific details about the functional and non-functional requirements, system features, and design considerations.

1.1. PURPOSE:

The Software Requirements Specification (SRS) document for a Hotel Management System (HMS) serves several important purposes throughout the software development lifecycle. Here are the key purposes:

1)Communication:

Clarity for Stakeholders: The SRS document acts as a communication tool that ensures a common understanding among all stakeholders (developers, testers, project managers, clients, etc.) regarding the objectives and functionalities of the Hotel Management System.

2)Guidance for Development:

Blueprint for Developers: It provides a detailed blueprint or roadmap for developers, guiding them on what needs to be implemented. This includes functional requirements, system behavior, and the overall architecture of the HMS.

3)Basis for Testing:

Test Planning: Testers use the SRS document as a basis for creating test plans and test cases. It defines the expected behavior of the system, helping testers verify whether the implemented system meets these expectations.

4)Project Management:

Scope Definition: The SRS document helps in clearly defining the scope of the project. It outlines what features and functionalities are in and out of scope, preventing scope creep during the development process.

5)Decision Support:

Requirements Prioritization: It assists project managers and stakeholders in prioritizing requirements based on their importance and impact. This helps in making informed decisions during the development process.

6)Change Management:

Change Control: The SRS document serves as a reference for change control. If there are changes in requirements during the development process, the document can be updated, and changes can be tracked and managed effectively.

7)Basis for Validation:

Validation Criteria: It provides the criteria against which the client or end-users can validate the developed system. This ensures that the final product meets the originally specified requirements.

8)Legal and Contractual Documentation:

Contractual Agreements: In some cases, the SRS may be a part of the contractual agreement between the development team and the client. It serves as a basis for legal and contractual documentation.

9)Cost Estimation:

Estimation Basis: The SRS document is used as a basis for estimating the costs and resources required for the development of the Hotel Management System. It helps in budgeting and resource allocation

1.2 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

- NTES – National Train Enquiry System
- IVRS – Interactive Voice Response system
- PRS – passenger reservation system

1.3. SCOPE

- Freight Revenue enhancement.
- Passenger Revenue enhancement.
- Improved & optimized service

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE:

It enables us to maintain the railway train details like their timings, number of seat available and reservation billing and cancelling the tickets.

Online railway ticket reservation is an online ticket booking website, which is capable of booking ticket and search the train availability. This website is mainly created to fulfil the following requirements, it comprises of the following properties: -

1. A central database that will store all information.
 2. An online website that will provide real- time information about the availability of tickets their prices.
 3. Every registered user is able to view his booking id that has been made in his/her name.
 4. Every registered user can change his password any time he wants to change.
 5. Every guest user can search train availability, price of the ticket, arrival and departure time, distance between source and destination etc.
 6. Every registered user has the facilities to print his ticket any time he wishes.
- Administration login - In admin mode the administrator can make changes in train details. He can also view all booking that has been made by different users.
7. The booking window contains all the facilities at one place, the user can simply login to his account and can book his ticket.

2.1.1 USER INTERFACE:

Keyboard and Mouse.

2.1.2 HARDWARE INTERFACE:

Printer and Normal PC

2.1.3. SOFTWARE INTERFACE:

- 1.Front end -> Visual Basic
- 2.Back end -> MS-Access

2.1.4 COMMUNICATION INTERFACES

- Indian Railway's web-site, www.indianrail.gov.in offers PRS enquiries on the internet Berth/Seat availability, Passenger Status, Fare, Train Schedule etc.
- National Train Enquiry System (NTES) website, www.trainenquiry.com gives dynamic information about the running status of any train and its expected arrival/departure at any given station.
- Mobile telephone-based SMS enquiry service. A new mobile phone-based facility for rail users.

2.1.5 OPERATING ENVIRONMENT

The OS types are:-

- 1.Windows NT
- 2.Windows XP
- 3.Linux

2.1.6 OPERATIONS

1. Any Reservation counter from 8 am to 8pm.
2. Prior to 30 days of journey.
3. One form for 4 persons only.
4. Reserved tickets done through predefined logic.
5. To save time and enhance customer experience.

2.2 PRODUCT FUNCTIONS:

It tells the short note about the product.

2.2.1 Reservation Management

Online Booking: The system should allow guests to make reservations through the hotel's website, providing real-time availability information.

Reservation Modification: Users should be able to modify or update their reservations, subject to availability.

Reservation Cancellation: Guests should have the ability to cancel their reservations with appropriate cancellation policies.

2.2.2 Front Desk Operations

Efficient Check-In/Check-Out: The system should facilitate a streamlined check-in and check-out process for guests.

Guest Profiles: Maintain detailed profiles for guests, including personal information, preferences, and previous stay history.

Room Assignment: Automate the assignment of rooms based on guest preferences and availability.

2.2.3 Billing and Invoicing

Automated Billing: Generate accurate and detailed bills for room charges, additional services, and taxes.

Integration with POS Systems: Integrate with point-of-sale systems to include charges from on-site services (restaurants, bars, spa) in the final invoice

2.2.4 Point of Sale (POS) Integration

Transaction Processing: Enable the processing of sales transactions for on-site services.

Inventory Management: Integrate with inventory systems to manage stock levels for on-site services.

2.2.5 Housekeeping Management

Room Cleaning Status: Provide a dashboard for housekeeping staff to view and update the status of rooms (clean, dirty, in progress).

Task Assignment: Assign and manage housekeeping tasks efficiently.

2.2.6 Reporting and Analytics

Occupancy Reports: Generate reports on room occupancy rates over specific periods.

Revenue Reports: Provide insights into revenue generated through room bookings and additional services.

2.3 USER CHARACTERISTICS:

1. Knowledgeable user

2. No voice user

3. Expert user

2.4 CONSTRAINTS

- Less than 1 sec for local transactions.
- 3 sec for network transaction.
- Capable for providing transaction for 22 hours per day.

SOFTWARE CONSTRAINTS:

- Designing -> Rational Rose
- Developing -> Visual Basic.

3. SPECIFIC REQUIREMENTS

3.1.1 Reservation Management

Online Booking (FR-RES-001):

The system shall provide a user-friendly interface for guests to make online reservations.

The system shall display real-time availability information.

Confirmation emails shall be sent to guests upon successful reservation.

Reservation Modification (FR-RES-002):

Users shall be able to modify their reservations online within 24 hours of check-in.

Any modification that affects the cost shall be recalculated and reflected in the reservation.

Reservation Cancellation (FR-RES-003):

Guests shall have the option to cancel reservations through the online system.

Cancellation policies shall be applied, and refunds processed accordingly.

3.1.2 Front Desk Operations

Efficient Check-In/Check-Out (FR-FDO-001):

The system shall support a quick and efficient check-in process, requiring minimal guest information.

Express check-out options shall be provided.

Guest Profiles (FR-FDO-002):

The system shall maintain detailed guest profiles, including personal information, preferences, and historical stays.

Profiles shall be accessible by authorized hotel staff.

3.1.3 Billing and Invoicing

Automated Billing (FR-BI-001):

The system shall automatically generate itemized bills for room charges, additional services, and taxes.

Bills shall be made available electronically and in print.

Integration with POS Systems (FR-BI-002):

The system shall integrate with point-of-sale systems to include charges from on-site services in the final invoice.

Inventory levels for on-site services shall be updated in real-time.

3.1.4 Point of Sale (POS) Integration

Transaction Processing (FR-POS-001):

The system shall process sales transactions for on-site services in real-time.

Transaction details shall be recorded for reporting purposes.

3.1.5 Housekeeping Management

Room Cleaning Status (FR-HKM-001):

The system shall provide a visual dashboard for housekeeping staff to view and update room cleaning status.

Cleaning status changes shall trigger notifications to relevant staff.

Task Assignment (FR-HKM-002):

Housekeeping managers shall be able to assign cleaning tasks to specific staff members.

Staff members shall receive notifications of assigned tasks.

SOFTWARE SYSTEM ATTRIBUTES:

Reliable

Available

Secure

4. DOCUMENT APPROVAL

The bill passed on any proposals related to railway management needs approval of Ministry of railway department.

5. Other Non-Functional Requirements

5.1 Performance

5.1.1 Response Time

Description: The system should provide timely responses to user interactions, ensuring a smooth and responsive user experience.

Requirement:

Critical functions (e.g., user authentication, ticket allocation) should have a response time of less than 2 seconds.

Non-critical functions (e.g., generating reports) should provide responses within 5 seconds.

5.1.2 Scalability

Description: The system should be scalable to accommodate potential growth in the number of users and data volume.

Requirement:

The system should handle a minimum of 100 concurrent users without significant performance degradation.

Scalability planning should allow for future expansion of the database and infrastructure to support increased load.

5.2 Usability

5.2.1 User Interface

Description: The user interface (UI) should be intuitive and user-friendly to minimize user training requirements.

Requirement:

The UI should follow standard design conventions, making it easy for users to navigate and perform tasks.

User training materials, including documentation and tutorials, should be provided to aid users in using the system effectively.

5.3 Reliability

5.3.1 Availability

Description: The system should be available to users with minimal downtime for maintenance or unexpected failures.

Requirement:

The system should aim for 99.9% uptime, excluding scheduled maintenance windows.

Maintenance windows, if required, should be scheduled during off-peak hours to minimize disruption.

5.4 Security

5.4.1 Data Security

Description: The system should ensure the confidentiality, integrity, and availability of user data.

Requirement:

User data should be encrypted during transmission and storage using industry-standard encryption protocols.

Passwords should be securely hashed and stored.

Access to sensitive data should be restricted based on user roles, following the principle of least privilege.

5.5 Compatibility

5.5.1 Browser Compatibility

Description: The system should be compatible with a range of web browsers to accommodate user preferences.

Requirement:

The system should support the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge.

Browser-specific issues should be addressed promptly to ensure consistent functionality.

5.6 Data Backup and Recovery

5.6.1 Regular Backups

Description: Regular backups of the system data should be performed to prevent data loss in case of hardware failures or other emergencies.

Requirement:

Daily backups of the database should be scheduled, with backup data stored securely off-site.

A backup retention policy should be established to manage backup data.

5.7 Compliance

5.7.1 Regulatory Compliance

Description: The system should comply with relevant data protection and privacy regulations.

Requirement:

The system should adhere to data protection regulations such as GDPR, including providing mechanisms for user consent and data deletion requests.

Regular audits and compliance checks should be conducted to ensure ongoing adherence to regulations.

EXPERIMENT 3

AIM: To perform function-oriented diagram: Data Flow Diagram and Structured Chart.

REQUIREMENTS:

Hardware Interfaces

1. Pentium(R) 4 CPU 2.26 GHz, 128 MB RAM
2. Screen resolution of at least 800 x 600 required for proper and complete viewing of screens. Higher resolution would not be a problem.
3. CD ROM Driver

Software Interfaces

Any window-based operating system.

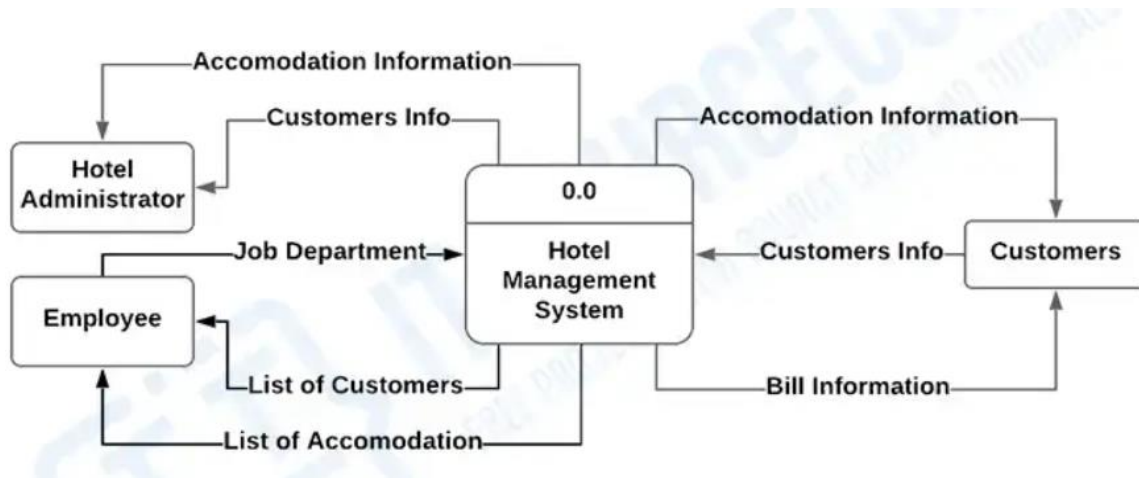
THEORY-

Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

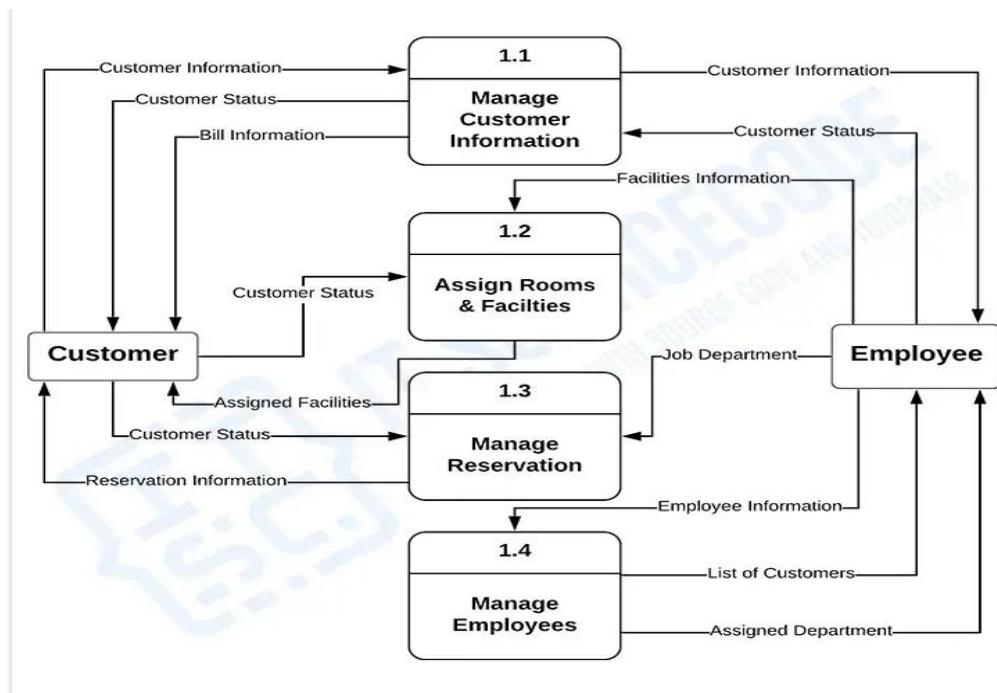
1. External entity: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.
2. Process: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as "Submit payment."
3. Data store: files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as "Orders."
4. Data flow: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name,

DATA FLOW DIAGRAM FOR HOTEL MANAGEMENT SYSTEM:

LEVEL 0 DFD :-



Level 1 DFD:



CONCLUSION –

The Data Flow Diagram was made successfully.

EXPERIMENT 4

AIM – To make the entity relationship diagram for the Hotel Management System.

Hardware Requirements:

Pentium 4 processor (2.4 GHz), 128 Mb RAM, Standard keyboard n mouse, colored monitor.

Software Requirements:

Rational Rose, Windows XP,

THEORY-

Entity Relationship Diagrams are a major data modelling tool and will help organize the data in your project into entities and define the relationships between the entities. This process has proved to enable the analyst to produce a good database structure so that the data can be stored and retrieved in a most efficient manner.

1. Entity: A data entity is anything real or abstract about which we want to store data. Entity types fall into five classes: roles, events, locations, tangible things or concepts.

E.g., employee, payment, campus, book. Specific examples of an entity are called instances.

2. Relationship: A data relationship is a natural association that exists between one or more entities. E.g., Employees process payments.

3. Cardinality defines the number of occurrences of one entity for a single occurrence of the related entity. E.g., an employee may process many payments but might not process any payments depending on the nature of her job.

4. Attribute: A data attribute is a characteristic common to all or most instances of a particular entity. Synonyms include property, data element, field. E.g., Name, address, Employee Number, pay rate are all attributes of the entity employee. An attribute or combination of attributes that uniquely identifies one and only one instance of an entity is called a primary key or identifier.

Entity Relationship Diagram for Hotel Management System:



CONCLUSION –

The ER diagram for Railway Reservation System was created successfully.

EXPERIMENT 5

AIM – To perform the user's view analysis for the suggested system: Use Case Diagram.

Hardware Requirements:

Pentium 4 processor (2.4 GHz), 128 Mb RAM, Standard keyboard n mouse, colored monitor.

Software Requirements:

Rational Rose, Windows XP,

THEORY -

According to the UML specification a use case diagram is —a diagram that shows the relationships among actors and use cases within a system. Use case diagrams are often used to:

1. Provide an overview of all or part of the usage requirements for a system or organization in the form of an essential model or a business model.
2. Communicate the scope of a development project.
3. Model your analysis of your usage requirements in the form of a system use case model.

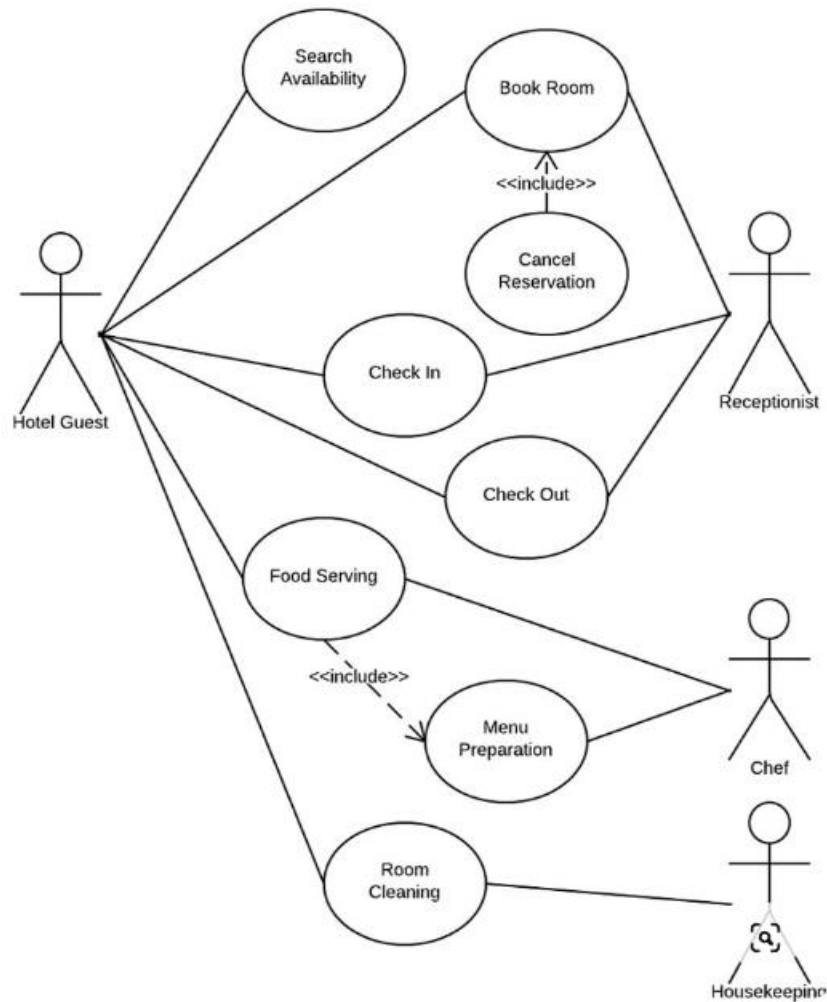
A use case describes a sequence of actions that provide a measurable value to an actor. A use case is drawn as a horizontal ellipse on a use case diagram.

1. Use Case Names Begin with a Strong Verb.
2. Name Use Cases Using Domain Terminology.
3. Place Your Primary Use Cases in The Top-Left Corner of the diagram.
4. Imply Timing Considerations by Stacking Use Cases.

There are several types of relationships that may appear on a use case diagram:

1. An association between an actor and a use case
2. An association between two use cases
3. A generalization between two actors
4. A generalization between two use cases

USE CASE DIAGRAM FOR HOTEL MANAGEMENT SYSTEM:



CONCLUSION:

The use case diagram for Railway Reservation System was created successfully.

EXPERIMENT 6

AIM

To draw the structural view diagram for the system: Class diagram, object diagram

SOFTWARE USED

Star UML

THEORY

Class Diagram:

A Class Diagram is a static structural diagram that depicts the classes in a system, their attributes, relationships, and the constraints governing these elements. Key theoretical aspects include:

1. **Class:**
 - a. Represents a blueprint for creating objects, encapsulating data attributes and methods.
2. **Attributes:**
 - a. Characteristics or properties of a class, describing the data it holds.
3. **Methods:**
 - a. Functions or operations that can be performed by objects of a class.
4. **Association:**
 - a. Represents relationships between classes, indicating how they are connected.
5. **Multiplicity:**
 - a. Specifies the number of instances participating in a relationship between classes.
6. **Inheritance:**
 - a. Represents an "is-a" relationship, where a subclass inherits attributes and behaviors from a superclass.
7. **Abstract Class:**
 - a. A class that cannot be instantiated and may contain abstract methods to be implemented by its subclasses.

Object Diagram:

An Object Diagram is an instance of a Class Diagram, capturing a snapshot of the system at a specific point in time. The key theoretical concepts include:

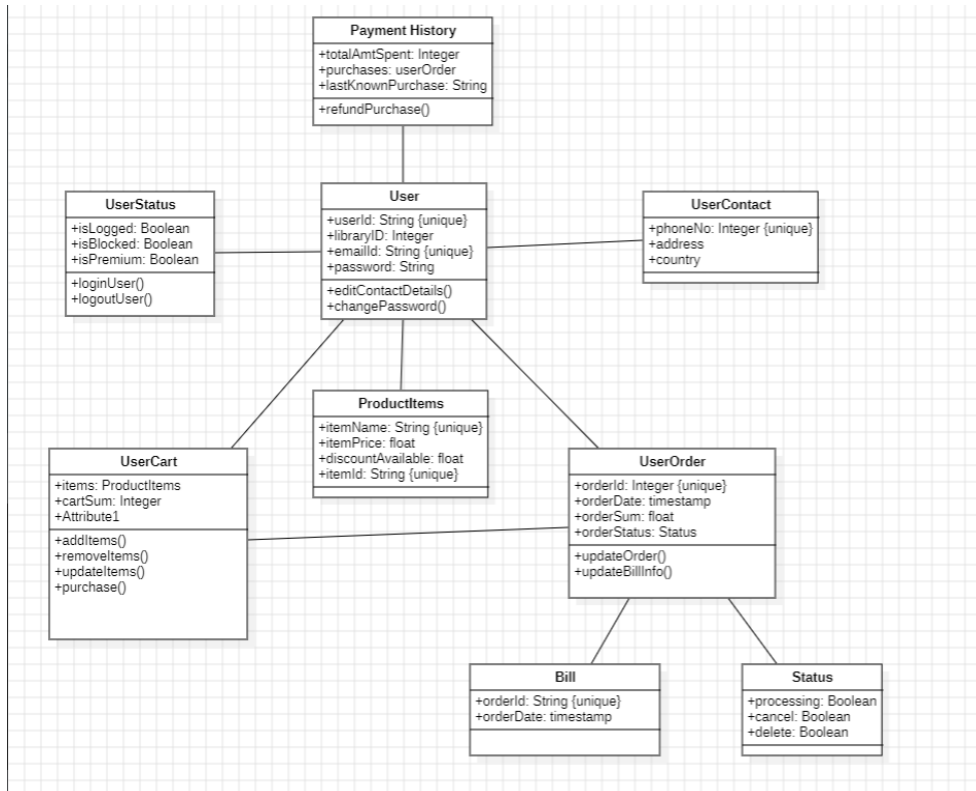
1. **Object:**
 - a. Represents an instance of a class, with specific values for its attributes.
2. **Link:**
 - a. Represents a connection between objects, showing relationships as they exist in a particular instance.
3. **Multiplicity:**
 - a. Indicates the number of instances participating in a link between objects.
4. **Role:**
 - a. Describes the way an object participates in a particular relationship.

Relationship Between Class and Object Diagrams:

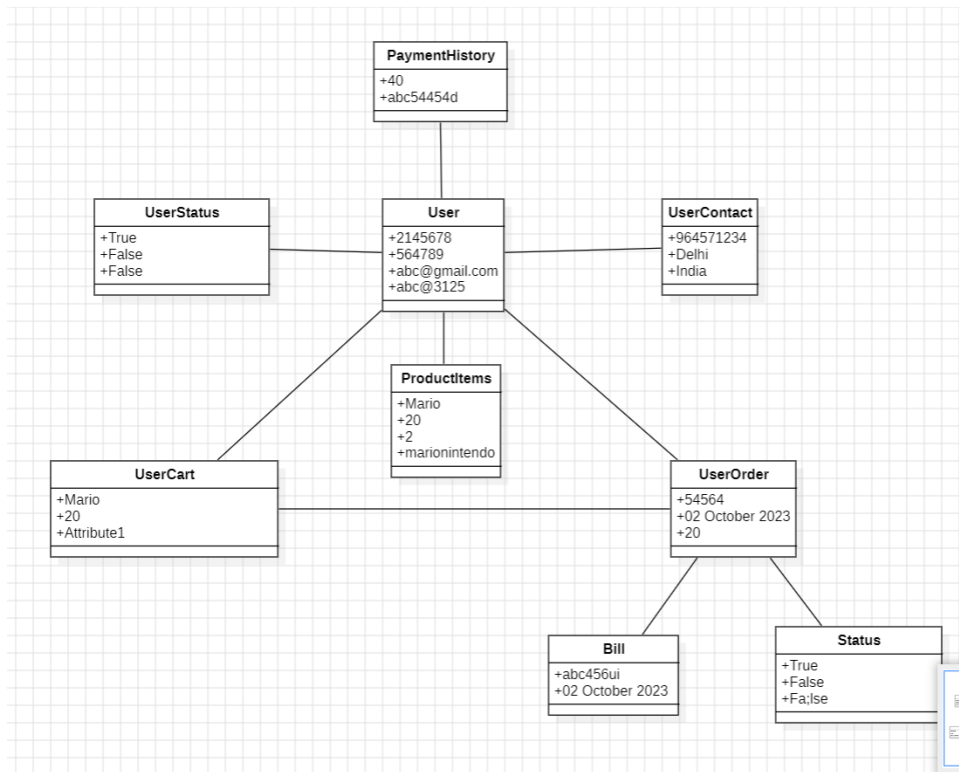
- **Abstraction:**
 - Class Diagrams provide a blueprint for the system's structure, while Object Diagrams instantiate these classes to represent specific instances.
- **Dynamic Aspect:**

- Class Diagrams focus on static aspects, depicting the structure, while Object Diagrams capture the dynamic relationships between objects at a specific moment.
- **Visualization:**
 - Class Diagrams are used for system design and planning, while Object Diagrams are beneficial for visualizing specific scenarios and instances during implementation.

Together, Class and Object Diagrams form a powerful duo in UML, facilitating the understanding and communication of a system's static structure and dynamic behavior throughout the software development lifecycle.



Class Diagram



Object Diagram

CONCLUSION

The Class Diagram and Object Diagram were successfully created.

EXPERIMENT 7

AIM

To draw the behavioral view diagram: State-chart diagram, Activity diagram.

SOFTWARE USED

Star UML

THEORY

STATE CHART DIAGRAM:

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system. A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. Purpose of state chart diagrams Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system. Following are the main purposes of using Statechart diagrams –

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.
-

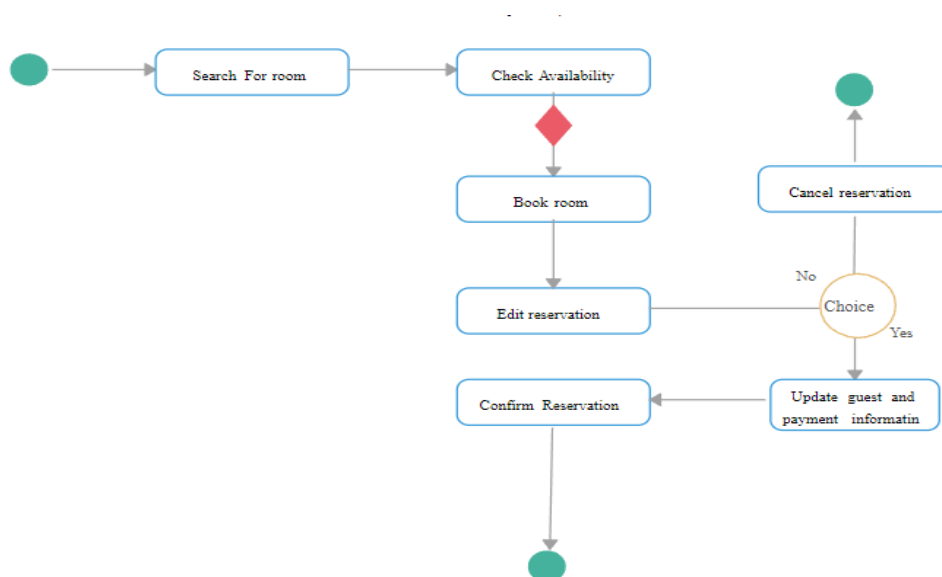


FIG 7.1: STATE CHART DIAGRAM

ACTIVITY DIAGRAM:

Activity diagrams describe the activities of a class. They are similar to state transition diagrams

and use similar conventions, but activity diagrams describe the behavior/states of a class in response to internal processing rather than external events. They contain the following elements:

1. Swimlanes , which delegate specific actions to objects within an overall activity
2. Action States , which represent uninterruptible actions of entities, or steps in the execution of an algorithm
3. Action Flows , which represent relationships between the different action states on an entity
4. Object Flows , which represent utilization of objects by action states, or influence of action states on objects

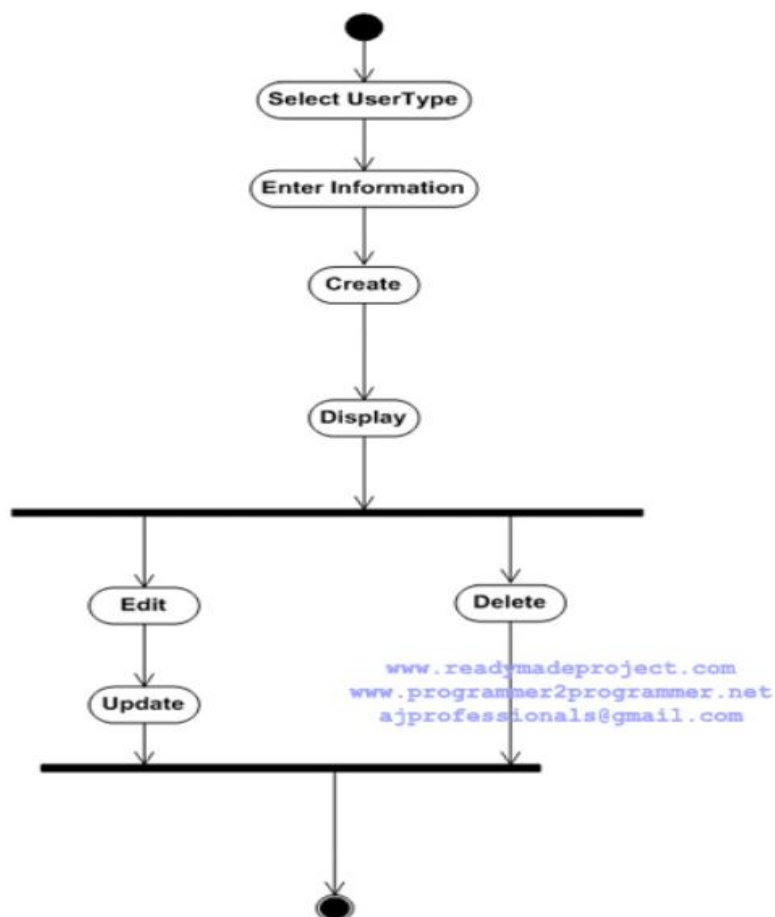


FIG 7.2: ACTIVITY DIAGRAM

CONCLUSION

The State Chart Diagram and Activity Diagram were successfully created.

EXPERIMENT 8

AIM

To perform the implementation view diagram: Component diagram for the system.

SOFTWARE USED

Star UML

THEORY

COMPONENT DIAGRAM:

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

PURPOSE OF COMPONENT DIAGRAM

Since it is a special kind of a UML diagram, it holds distinct purposes. It describes all the individual components that are used to make the functionalities, but not the functionalities of the system. It visualizes the physical components inside the system. The components can be a library, packages, files, etc.

1. The component diagram also describes the static view of a system, which includes the organization of components at a particular instant. The collection of component diagrams represents a whole system. The main purpose of the component diagram are enlisted below: It envisions each component of a system.
2. It constructs the executable by incorporating forward and reverse engineering.
3. It depicts the relationships and organization of components.

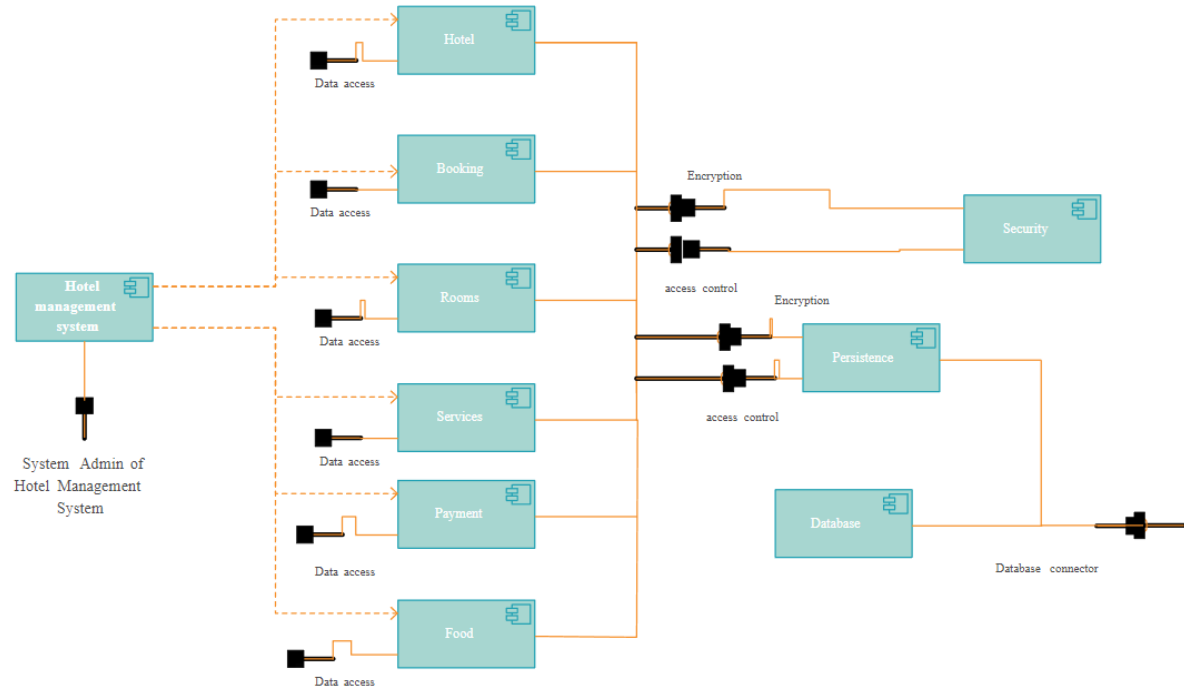


FIG 8.1: COMPONENT DIAGRAM

CONCLUSION

The Component Diagram was successfully created.