

**B.M.S. COLLEGE OF ENGINEERING BENGALURU**  
Autonomous Institute, Affiliated to VTU



Lab Record

**Object-Oriented Modeling**

*Submitted in partial fulfillment for the 5<sup>th</sup> Semester Laboratory*

Bachelor of Engineering  
in  
Computer Science and Engineering

*Submitted by:*

**SANTOSH H JAMBAGI**

**1BM22CS244**

Department of Computer Science and Engineering  
B.M.S. College of Engineering  
Bull Temple Road, Basavanagudi, Bangalore 560 019  
Mar-June 2024

**B.M.S. COLLEGE OF ENGINEERING**  
**DEPARTMENT OF COMPUTER SCIENCE AND**  
**ENGINEERING**



***CERTIFICATE***

This is to certify that the Object-Oriented Modelling (23CS5PCOOM) laboratory has been carried out by **SANTOSH H JAMBAGI(1BM22CS244)** during the 5<sup>th</sup> Semester Oct24-Jan2025.

Signature of the Faculty Incharge:

M Lakshmi Neelima  
Assistant Professor  
Department of Computer Science and Engineering  
B.M.S. College of Engineering, Bangalore

## Table of Contents

1. Hotel Management System
2. Credit Card Processing
3. Library Management System
4. Stock Maintenance System
5. Passport Automation System

# 1. Hotel Management System

## Software Requirement Specification

SRS Document for  
Hotel Management System.

### Introduction:

#### 1.1 Purpose of this document:

The main purpose of this document is to provide a detailed description of functional, non-functional and interface requirements. This is intended to use by developers, testers and stakeholders involved in system's development.

#### 1.2 Scope

The Hotel Management system includes hotel-related activities such as booking rooms, cancel rooms, order food, check-in's and check-outs.

#### 1.3. Overview

The remainder of this document is organized into several sections. It begins with the purpose of the document followed by scope, functional and non-functional requirements, performance and interface requirements followed by preliminary budget and schedule.

### General Description:

The Hotel Management System provides several features such as reservation management, Room management, check-in / check-out features, automatic billing and invoicing. The HMS is integrated with an external payment gateway. It is designed to work on standard browser and across many devices such as (PC, tablet & mobile).

### Functional Requirements:

- Book or Cancel rooms:

The user must be able to book and cancel the rooms through the website, based on availability of rooms.

- CheckIn or CheckOut:

The system shall allow user to checkin / checkout and update the room status.

- Order food:

The system shall allow the users to order the food to their respective rooms and generate bill accordingly.

- Billing & Invoicing:

The system shall generate bills based on room rates and add taxes. The payment can be accepted through multiple payment methods (card, UPI, cash).

### Interface Requirements:

User Interface: The system shall provide a user-friendly interface for all hotel staff. The system shall be supportive to various devices like laptops, tablets and mobile phones.

### Design Constraints:

#### Security Constraints:

The system should use user authentication for access.

The system shall encrypt sensitive data such as customer info and payment details.

### Performance Requirements:

The system shall handle at least 200 users without performance issues.

The system shall respond to user actions within 3-7 seconds.

### Non-Functional Requirements:

- Reliability:

The system should be continuously available for hotel operations.

- Security:

The system should use secured data storage for storing sensitive data of users.

## Class Diagram

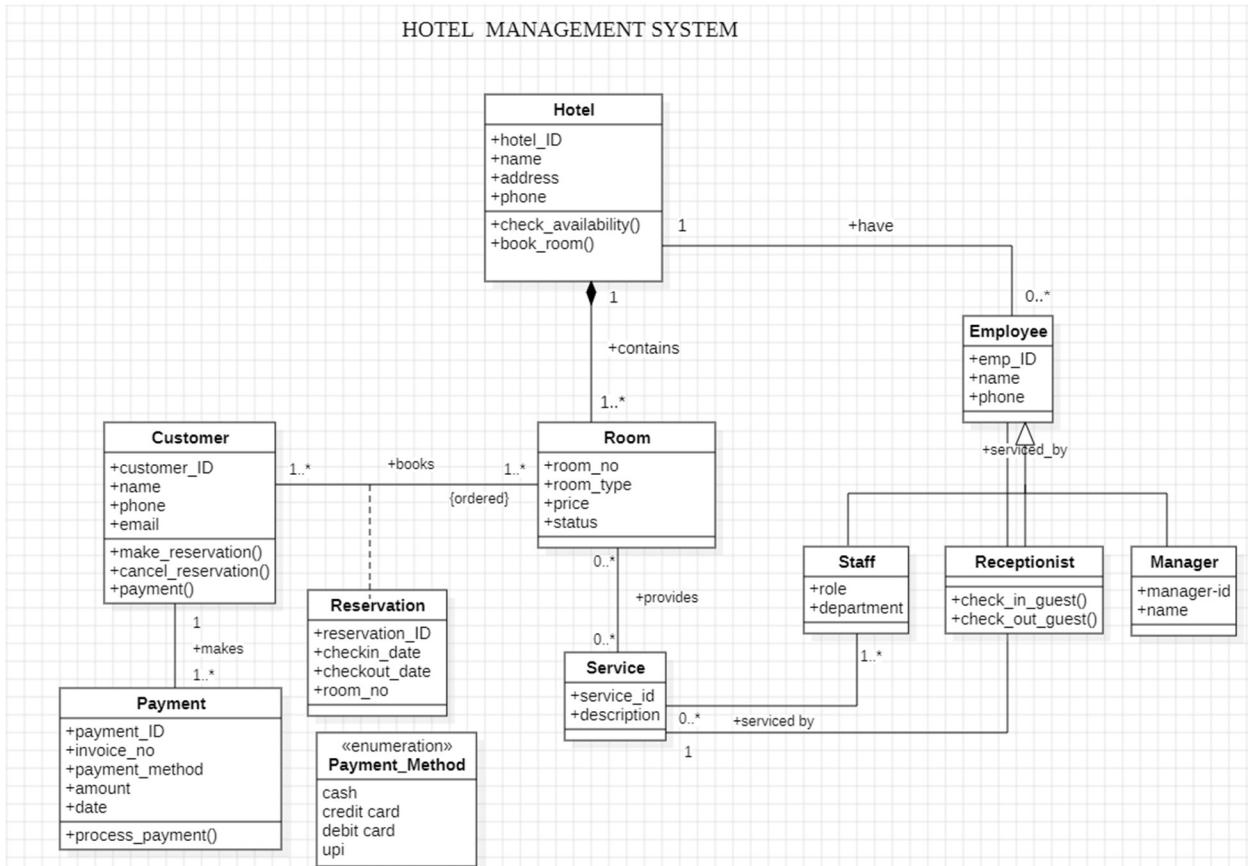


Fig1.1 Hotel Management System - Class Diagram

The diagram represents a hotel management system. It showcases the relationships between various entities such as Customer, Hotel, Room, Reservation, Payment, Service, and Staff. The diagram defines the attributes and operations associated with each entity, such as making a reservation, checking in/out guests, processing payments, etc. It also depicts the relationships between these entities, including one-to-one, one-to-many, and many-to-many relationships. For example, a customer can make multiple reservations, each reservation is associated with a specific room, and different types of staff members can be involved in various services. The diagram provides a comprehensive overview of the system's structure and interactions.

## State Diagram

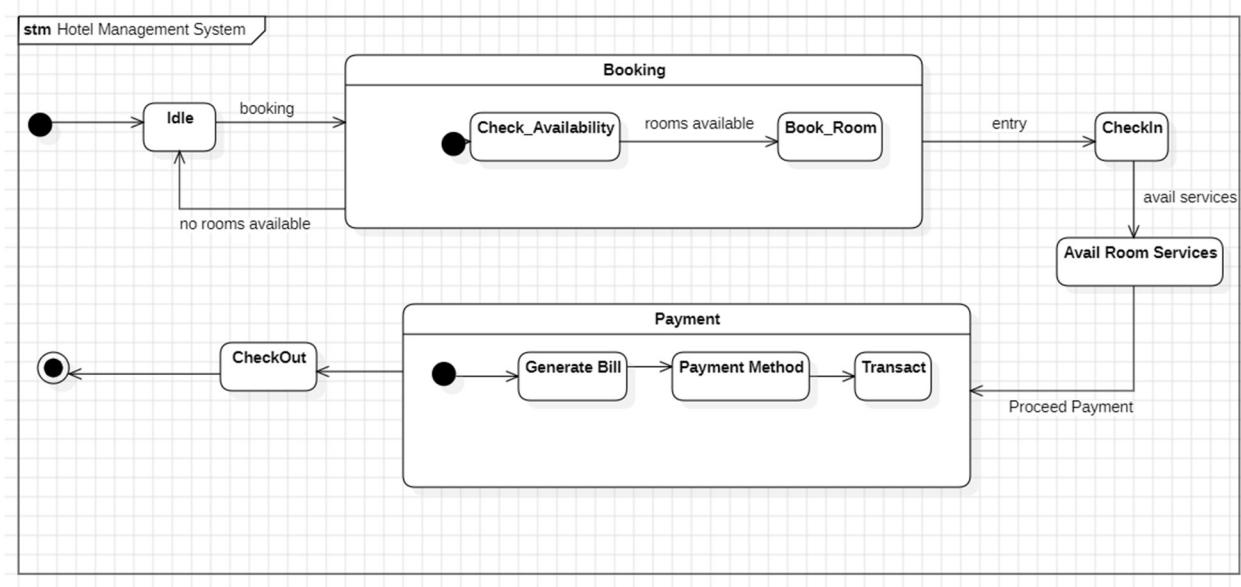


Fig1.2 Hotel Management System - State Diagram

The hotel management system state machine models the operational workflow of a hotel, transitioning through distinct states triggered by specific events. It begins in an **Idle** state, waiting for booking requests. Upon receiving a request, it transitions to **Check\_Availability** to verify room availability. If rooms are available, it moves to **Book\_Room**, confirming the booking; otherwise, it returns to **Idle**. Once booked, the customer proceeds to **Checkin**, after which they can avail services in **Avail Room Services**. The **CheckOut** state initiates upon the customer's departure, followed by **Generate Bill** to prepare their bill. In **Payment Method**, the customer selects how to pay, leading to **Transact**, where the payment is processed. Each state and transition ensures smooth and sequential operation of the system, ensuring efficiency and clarity in hotel management.

## Use Case Diagram

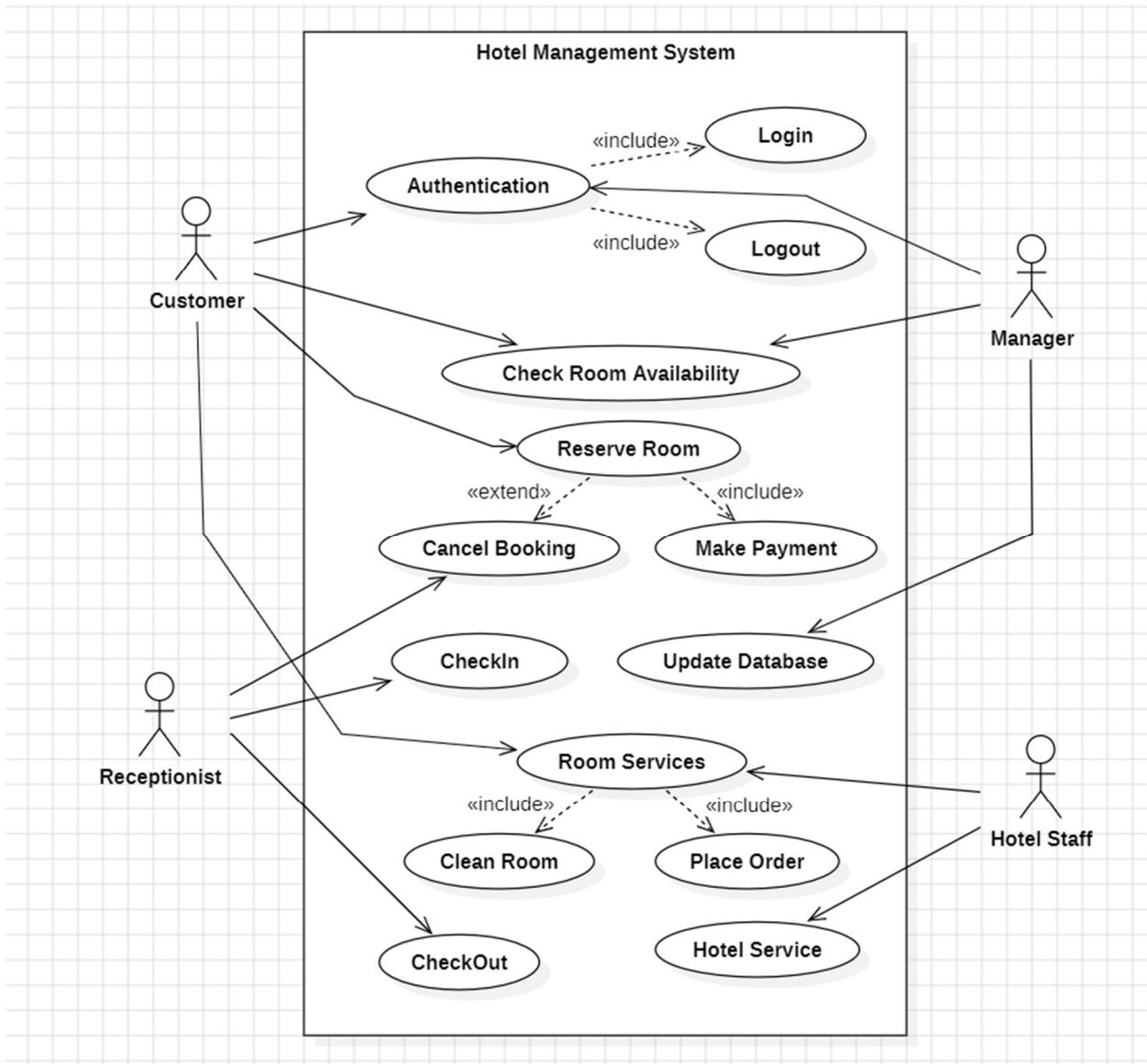


Fig1.3 Hotel Management System - Use Case Diagram

The diagram represents a Use Case Diagram for a Hotel Management System, showcasing various interactions between users (actors) and system functionalities. The primary actors include Customer, Manager, Receptionist, and Hotel Staff. Key use cases are grouped under the system, such as Authentication (which includes login and logout), Check Room Availability, Reserve Room (extended by Cancel Booking and including Make Payment), Check-In, and Room Services (further including cleaning, placing orders, and other hotel services). The diagram emphasizes the relationships and interactions among actors and system processes, demonstrating how each user

contributes to the system's operations. For example, the Manager and Receptionist oversee updates and reservations, while Hotel Staff handle room services.

### Sequence Diagram

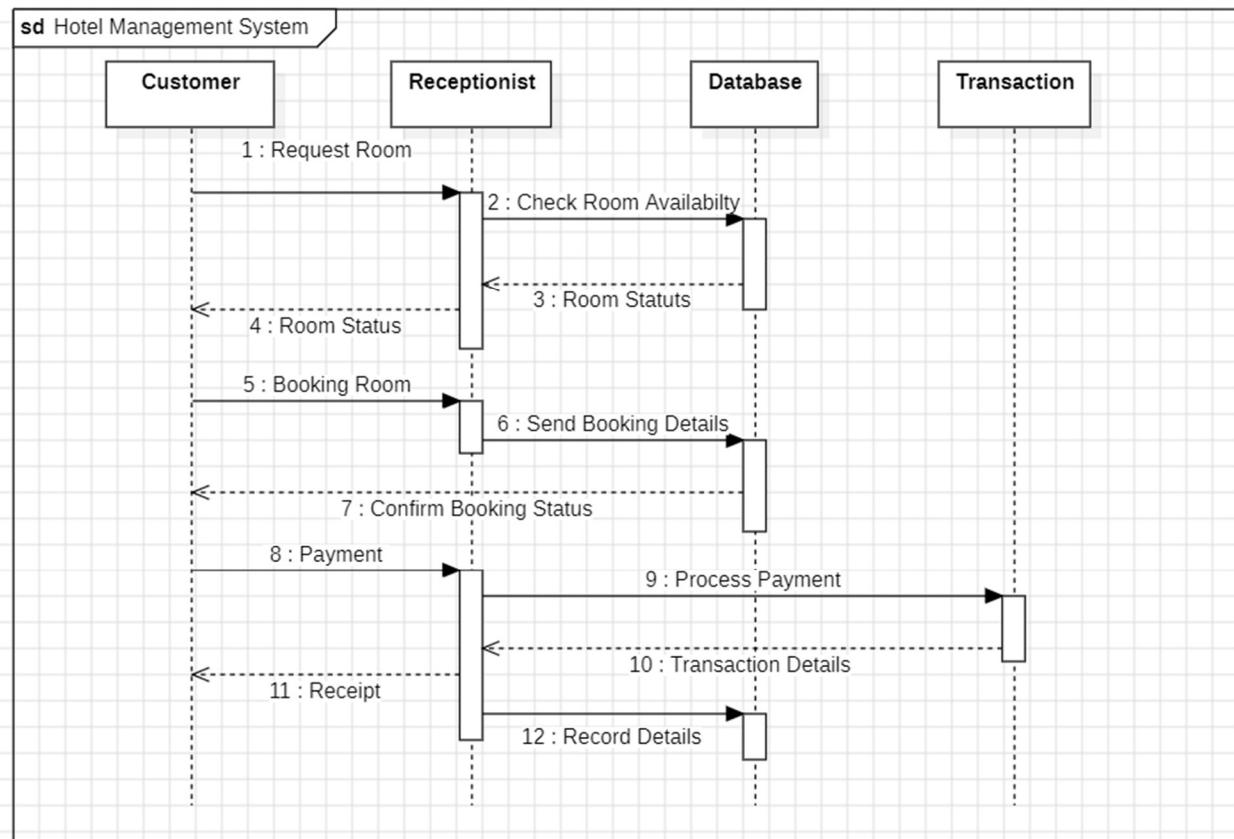


Fig1.4 Hotel Management System - Sequence Diagram

The sequence diagram illustrates the process of booking a room at a hotel. The customer initiates the process by requesting a room. The receptionist then checks the availability of the room in the database and returns the status to the customer. If the room is available, the customer can book the room. The receptionist sends the booking details to the database and confirms the booking status to the customer. The customer then makes the payment, and the transaction is processed by the database. Finally, the customer receives a receipt, and the database records the transaction details.

## Activity Diagram

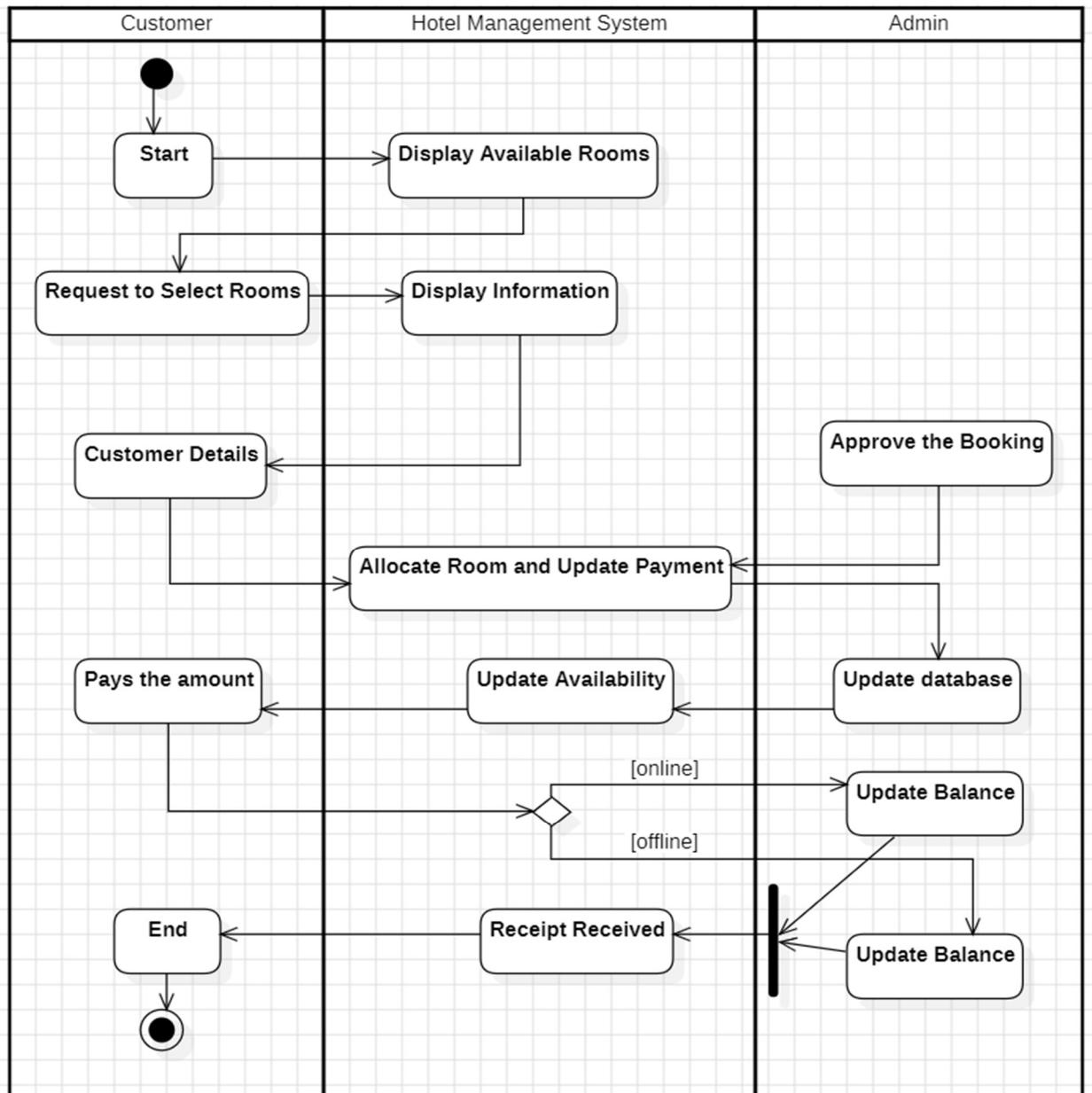


Fig1.5 Hotel Management System - Activity Diagram

The activity diagram illustrates the process of booking a room at a hotel. The customer starts by requesting to select rooms, and the system displays available rooms and their information. The customer then provides their details and selects a room. The admin approves the booking, and the system allocates the room and updates the payment. The customer pays the amount, and the system updates the room availability and balance. Finally, the customer receives a receipt, and the system updates the balance.

## 2.Credit Card Processing System

### Software Requirement Specification

SRS Document For  
Hotel Management System.  
Credit Card System

#### Introduction:

##### 1.1 Purpose of this document:

The main purpose of this document is to outline the services of a credit card system. This system aims to provide seamless transaction processing, user management, security etc.

##### 1.2 Scope

It includes user management features such as registration and login, transaction processing capabilities like authorization and refunds.

##### 1.3 Overview

The remainder of this document is organized into several sections. It begins with system's purpose and scope followed by functional and non-functional requirements and interface requirements.

#### General Description:

The credit card system provides user with efficient banking transactions such as online bill payments, EMI's payments, user management such as login/logout, secure transactions through user authentication, notifications through online e-mail's and to mobile number through which the user can benefit.

#### Functional Requirements:

##### • Notifications:

All the transactions are notified to the user by mail-id or mobile number.

##### • User Services:

→ Login/Logout: The user can login or logout of the internet banking website

→ Registration:

• User authorization through user's profile & profile password.

• Account management:

- Account statement generation.
- credit card limit notification.

Interface Requirements:

User Interface /

→ Online portal: through which a user can login or logout to his profile.

→ Able to perform all necessary banking transactions in the portal.

→ Able to manage his profile.

Design Constraints:

→ Data storage:

Use of secure data storage solutions that ensure data redundancy and backup of data.

→ Regulatory Compliance:

Must adhere to local and international financial regulations

Performance Requirements:

• Response Time:

The system must be able to authorize transactions within a maximum of 2 seconds from time of submission.

• System Availability:

Availability of the system continuously is essential.

Non-Functional Requirements:

• Portability:

A credit card can be used anywhere within the country or globally.

• Security:

→ DTF based transaction authorization provides

secure Transaction processing (CVV)

→ User authorization through profile password.

## Class Diagram

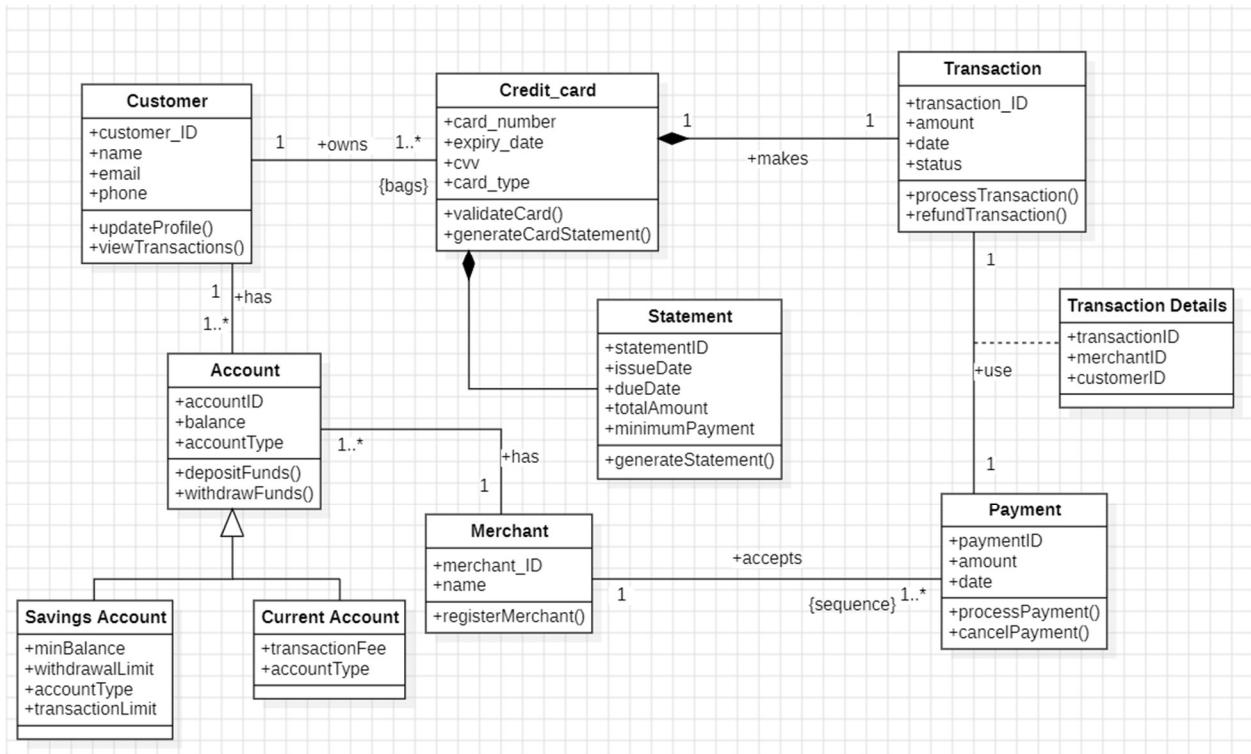


Fig 2.1 Credit Card Processing System - Class Diagram

The class diagram represents a credit card processing system. Customers own one or more Credit Cards, which are used to perform Transactions. Each credit card is validated and associated with a Statement that includes payment details like total amount and due date. Accounts (Savings or Current) store the customer's funds and enable deposits and withdrawals. Merchants register to accept payments, and payments are linked to Transaction Details, specifying the customer and merchant involved. Key functionalities include processing and refunding transactions, validating credit cards, generating statements, and updating customer profiles.

## State Diagram

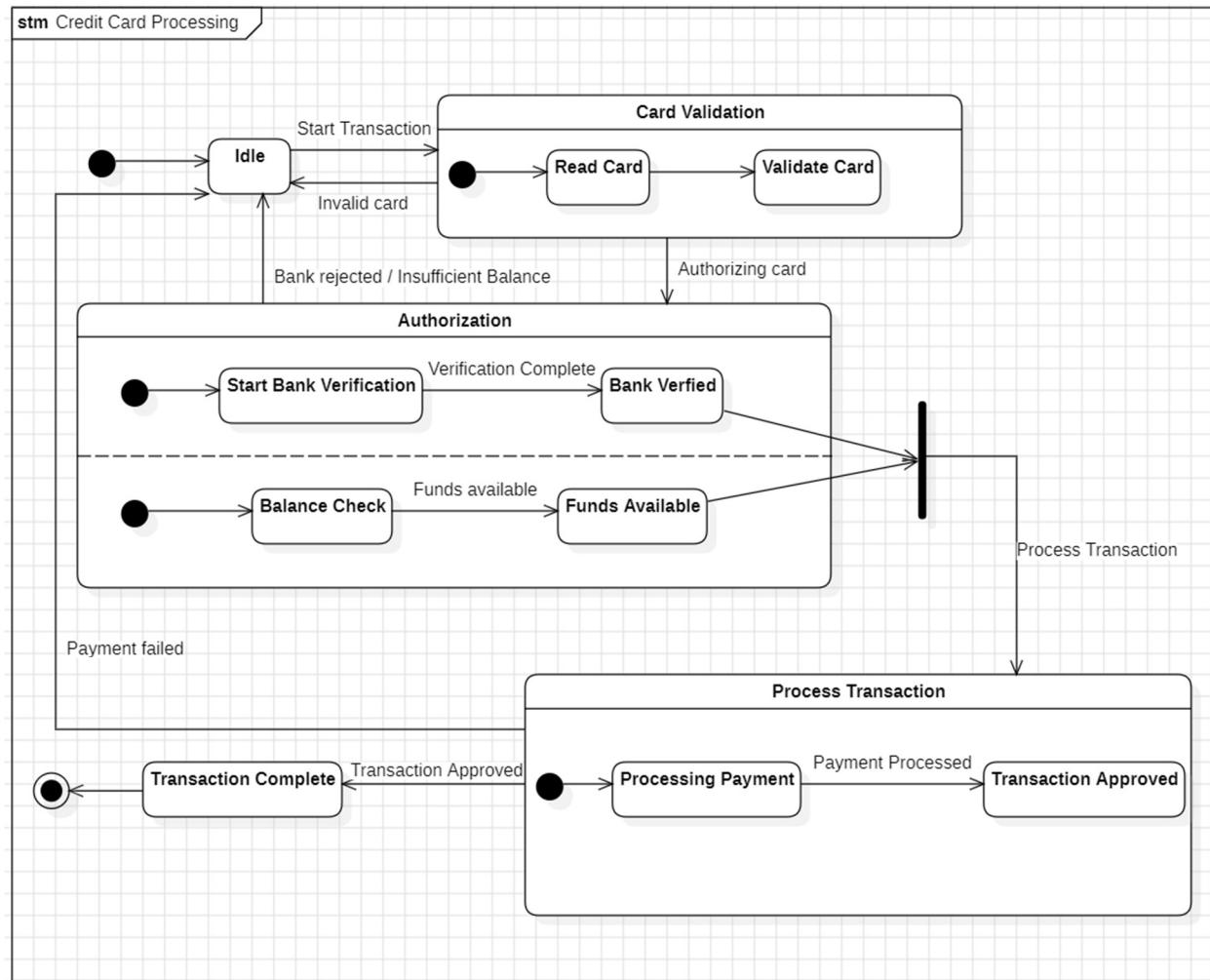


Fig 2.1 Credit Card Processing System - State Diagram

The state diagram illustrates the process of a credit card transaction. The system starts in an idle state and transitions to the "Read Card" state when a transaction is initiated. The card is then validated, and if it is invalid, the transaction is rejected. If the card is valid, the system moves to the "Authorization" state and verifies the card with the bank. If the card is verified and the funds are available, the system proceeds to the "Process Transaction" state and completes the transaction. If the card is not verified or there are insufficient funds, the transaction fails.

## Use Case Diagram

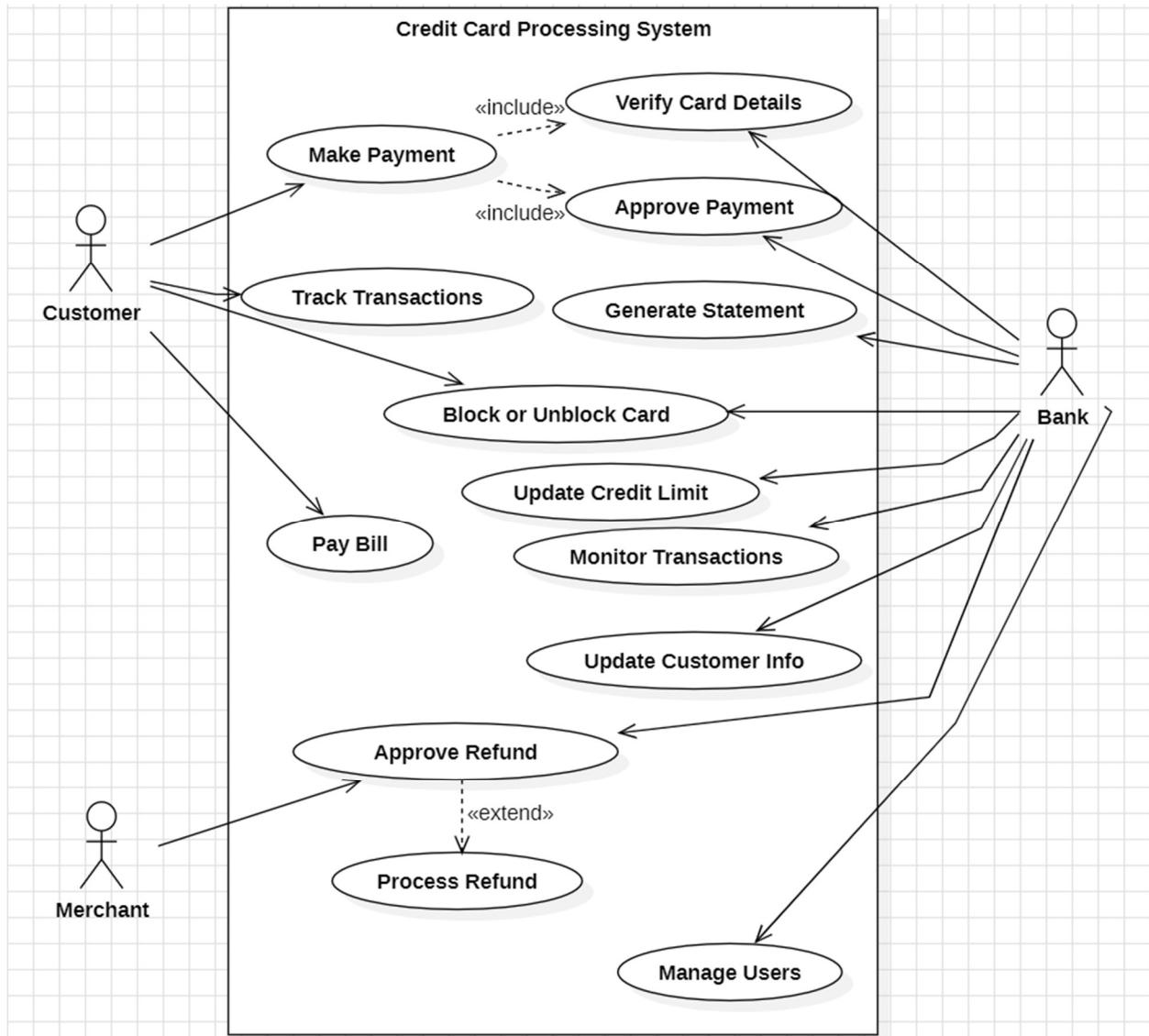


Fig 2.1 Credit Card Processing System - Use Case Diagram

The diagram depicts a Use Case Diagram for a Credit Card Processing System, highlighting the roles of the primary actors: Customer, Bank, and Merchant. The Customer interacts with the system to make payments (which includes verifying card details and approving payments), track transactions, generate statements, block or unblock cards, and pay bills. The Bank is responsible for approving payments, monitoring transactions, updating customer information, adjusting credit limits, and managing users. Additionally, merchants can request refunds, which involve approval and subsequent processing by the system. This diagram effectively illustrates the interactions and responsibilities within the credit card processing workflow.

## Sequence Diagram

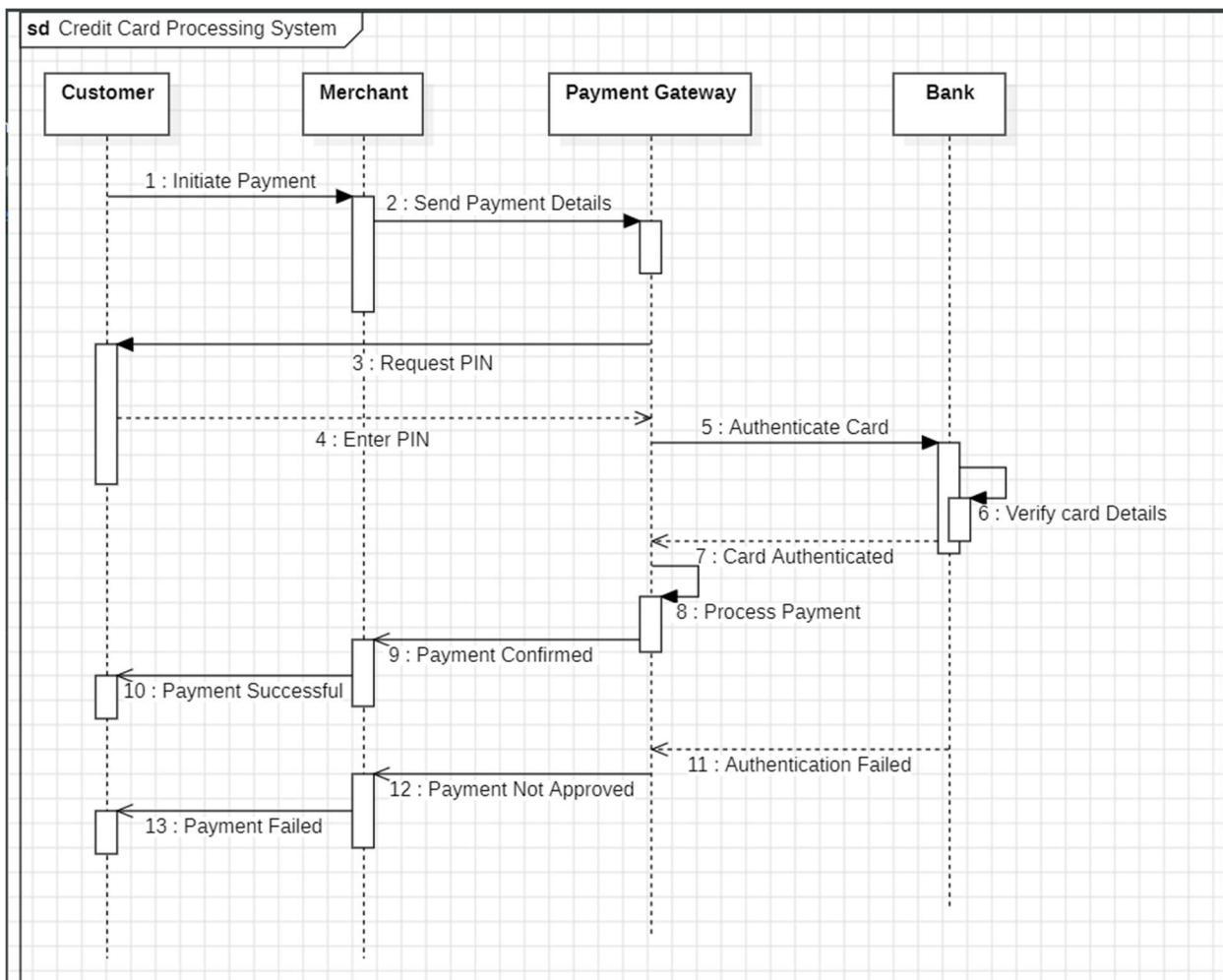


Fig 2.1 Credit Card Processing System - Sequence Diagram

The sequence diagram illustrates the process of a credit card transaction. The customer initiates the payment, and the merchant sends the payment details to the payment gateway. The payment gateway requests the customer to enter their PIN for authentication. Once the PIN is entered, the gateway authenticates the card with the bank. If the card is authenticated, the payment gateway processes the payment and confirms it to the merchant. Finally, the customer receives a notification of successful payment. If the card authentication fails, the payment is not approved.

## Activity Diagram

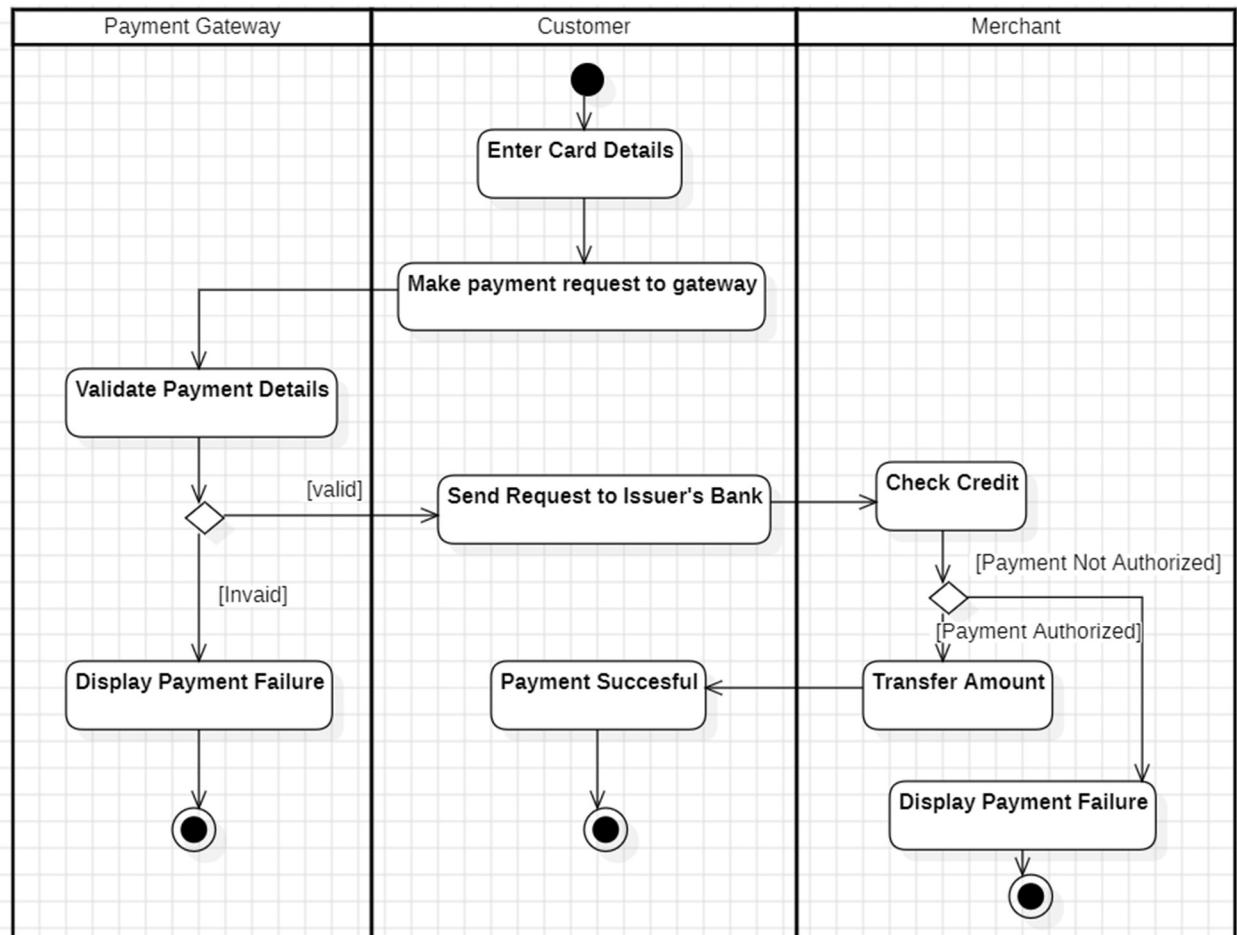


Fig 2.5 Credit Card Processing System - Activity Diagram

The activity diagram illustrates the process of a credit card transaction. The customer starts by entering their card details and making a payment request to the gateway. The gateway validates the payment details. If the details are valid, the gateway sends a request to the issuer's bank to check the credit. If the credit check is successful, the bank authorizes the payment and the gateway transfers the amount. The customer then receives a notification of successful payment. If the payment details are invalid or the credit check fails, the transaction is rejected, and the customer receives a notification of payment failure.

### 3. Library Management System

#### Software Requirement Specification

##### Introduction:

###### 1.1 Purpose of this document:

The main purpose of this document is to provide a detailed description of the functional & Non-functional requirements, interface requirements and performance requirements. This is intended to use by developers, testers and stakeholders involved in system's development.

###### 1.2 Scope of this document:

The Library management System includes the features, functionalities and technical requirements. It covers the features like book management, user registrations, borrowing and returning of books.

###### 1.3 Overview:

The remainder of this document is organized into different sections, which includes functional & non-functional requirements. Such as Book management, Borrowing & returning of books, Staff Management.

##### General Description:

The Library management system provides several features such as Book management, staff Management, Real-time update on books available in the library, borrowing of books and returning of books. Users include librarians, administrators etc.

##### Functional Requirements:

###### • Book Management:

The system must allow librarians to add, update and remove books.

###### • Borrow / Return of books:

The system must allows the users to borrow the books from the library and return the books within a certain time interval preferably 1-3 months.

The system must be able process the books on return date. If the return of books exceed the time interval, it should calculate the fine accordingly.

#### Interface Requirements:

##### User Interface:

The system shall provide a user-friendly interface for all Library staff. It should handle member registrations, profile updates, and member categorization. The system shall be supportive to all devices like laptops, tablets and mobile phones etc.

##### Design Constraints:

The system must work on both windows and macOs for desktop uses. The system shall use secure data storage solutions such as SQL.

#### Performance Requirements:

The system must handle multiple users simultaneously without any degradation in performance.

The system should respond to all user actions such as payment status, book status within 2-3 seconds.

#### Non-Functional Requirements:

\* Scalability: The system must support upto 150 concurrent users simultaneously.

\* Reliability: The system must be available continuously for library operations.

\* Security: The system should encrypt sensitive user information such as user name, payment info etc.

## Class Diagram

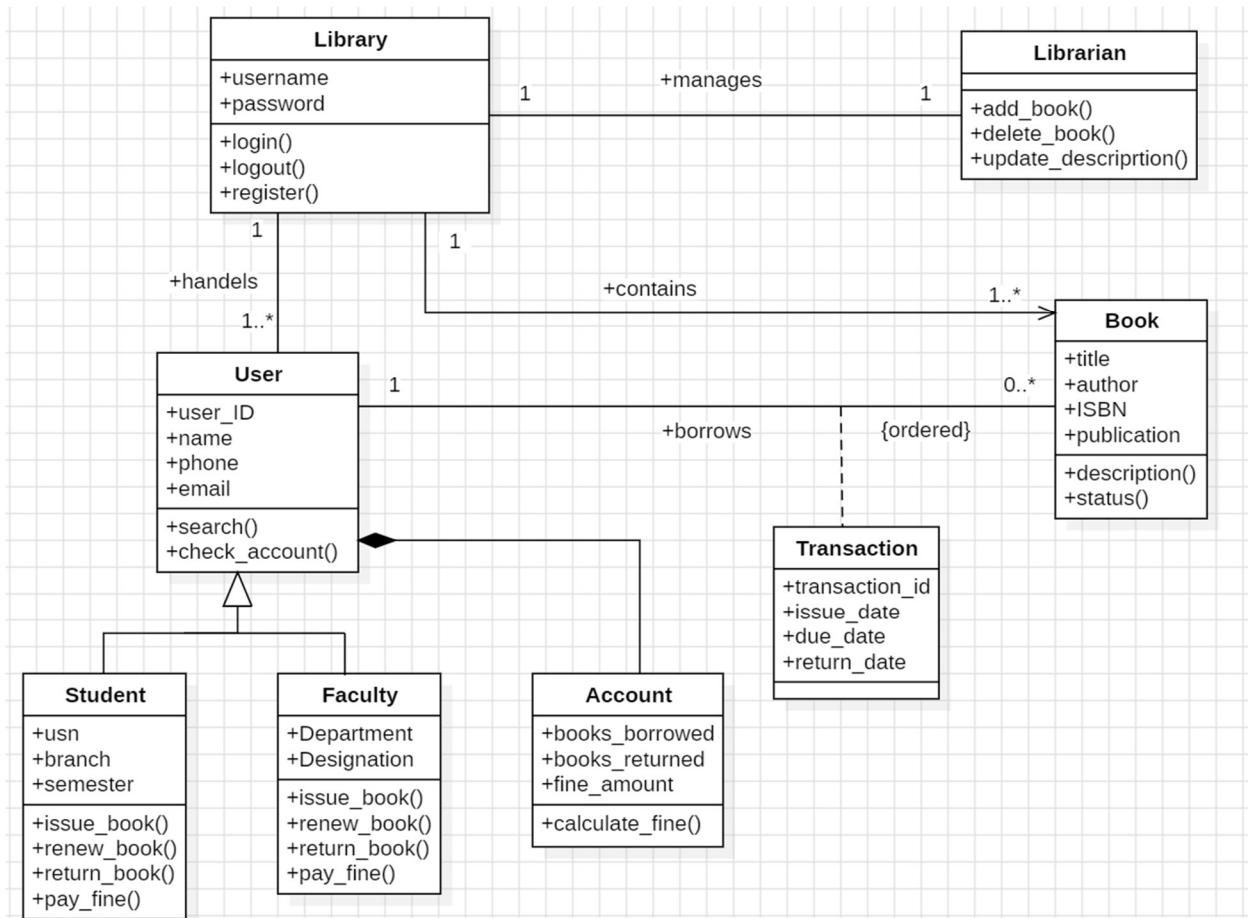


Fig 3.1 Library Management System -Class Diagram

The class diagram represents a library management system, showcasing entities like Library, Librarian, User, Book, Account, and Transaction. The Library handles the system's operations, managed by a Librarian who adds, updates, and deletes books. Users are divided into Students and Faculty, each with functionalities like issuing, renewing, and returning books, managed via their respective Accounts that track borrowed books and fines. Books store details like title, author, and status, while Transactions record borrowing and returning activities. The relationships between these entities ensure seamless management of books, users, and transactions.

## State Diagram

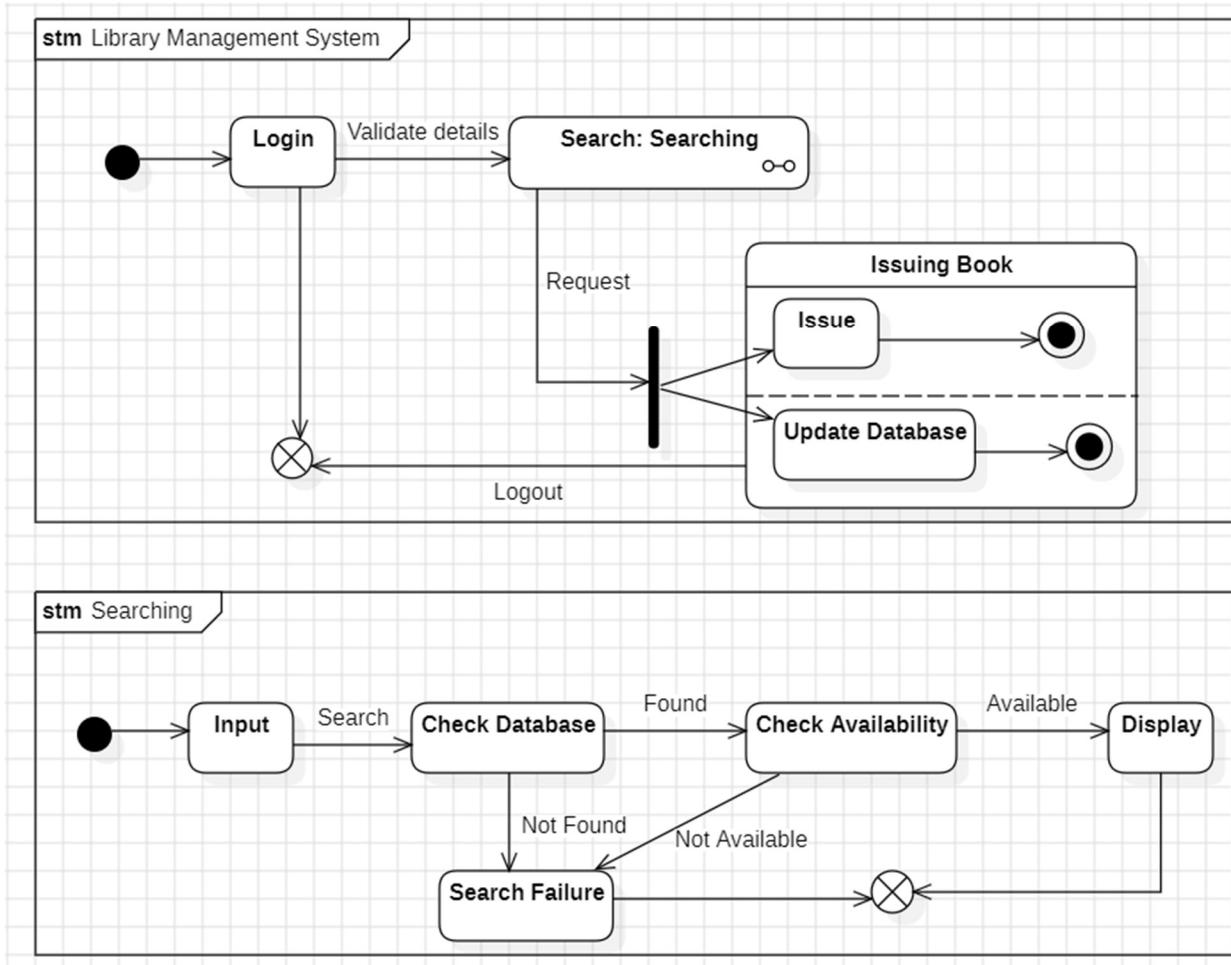


Fig 3.2 Library Management System - State Diagram

The state diagram illustrates the workflow of a library management system. It begins with user login, followed by credential validation. Upon successful login, the system enters the "Searching" state, where the user can search for books. The search process involves checking the database for matches and then checking availability. If a book is available, its details are displayed to the user, who can then request to issue it. The system updates its database accordingly. If the search yields no results or the book is unavailable, the system transitions to the "Search Failure" state. At any point, the user can log out of the system.

## Use Case Diagram

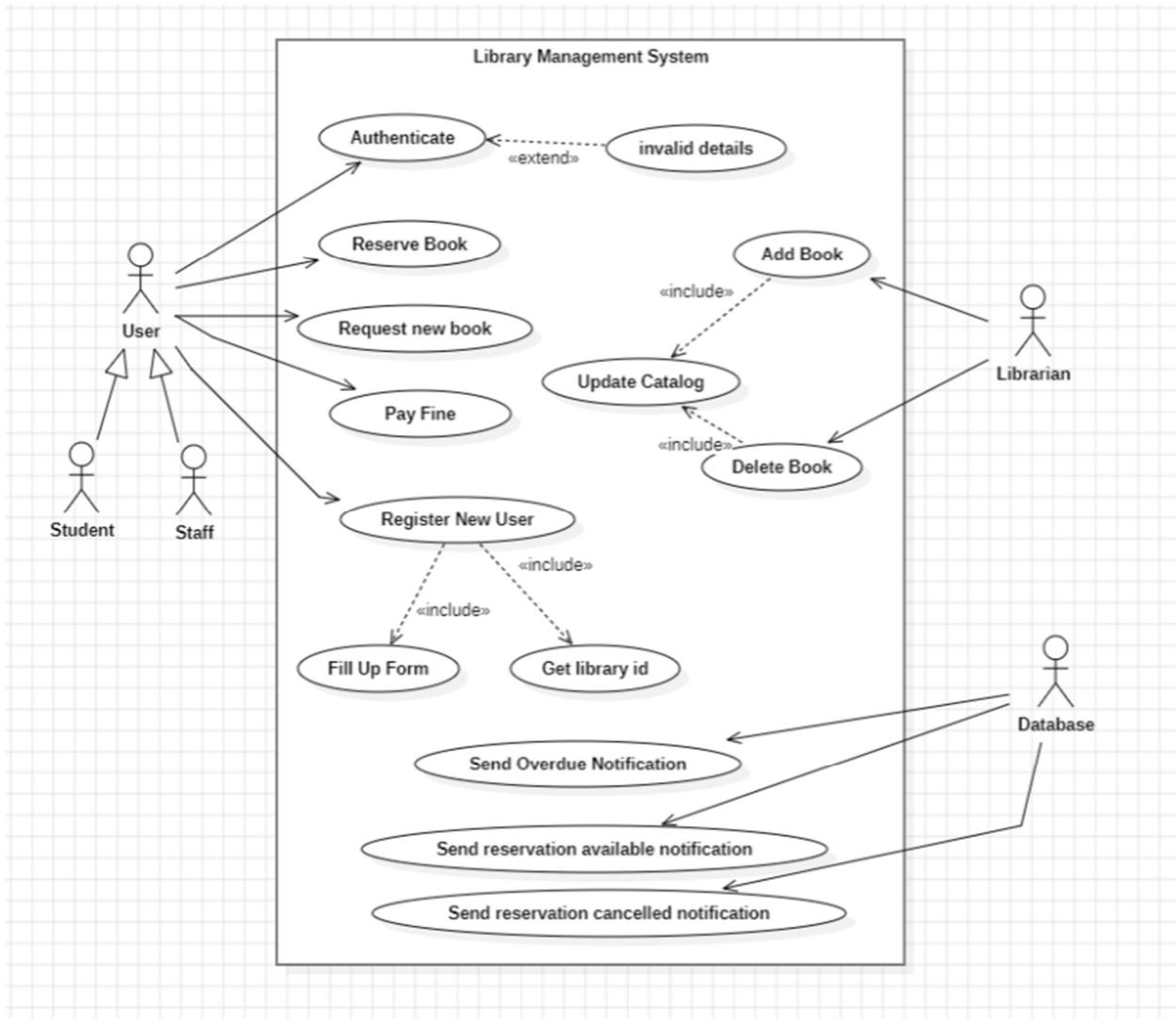


Fig 3.3 Library Management System - Use Case Diagram

The Library Management System is designed to manage the library's resources and user interactions. The system has three main actors: User, Librarian, and Database. The User can reserve books, request new books, pay fines, and register as a new user. The Librarian can add books to the catalog, update the catalog, delete books, and send overdue notifications. The Database stores and manages all the information related to the library, users, and books. The system includes use cases for authentication, filling up forms, and getting library IDs, which are further elaborated by the "include" relationships. This system aims to streamline library operations and provide a convenient experience for users.

## Sequence Diagram

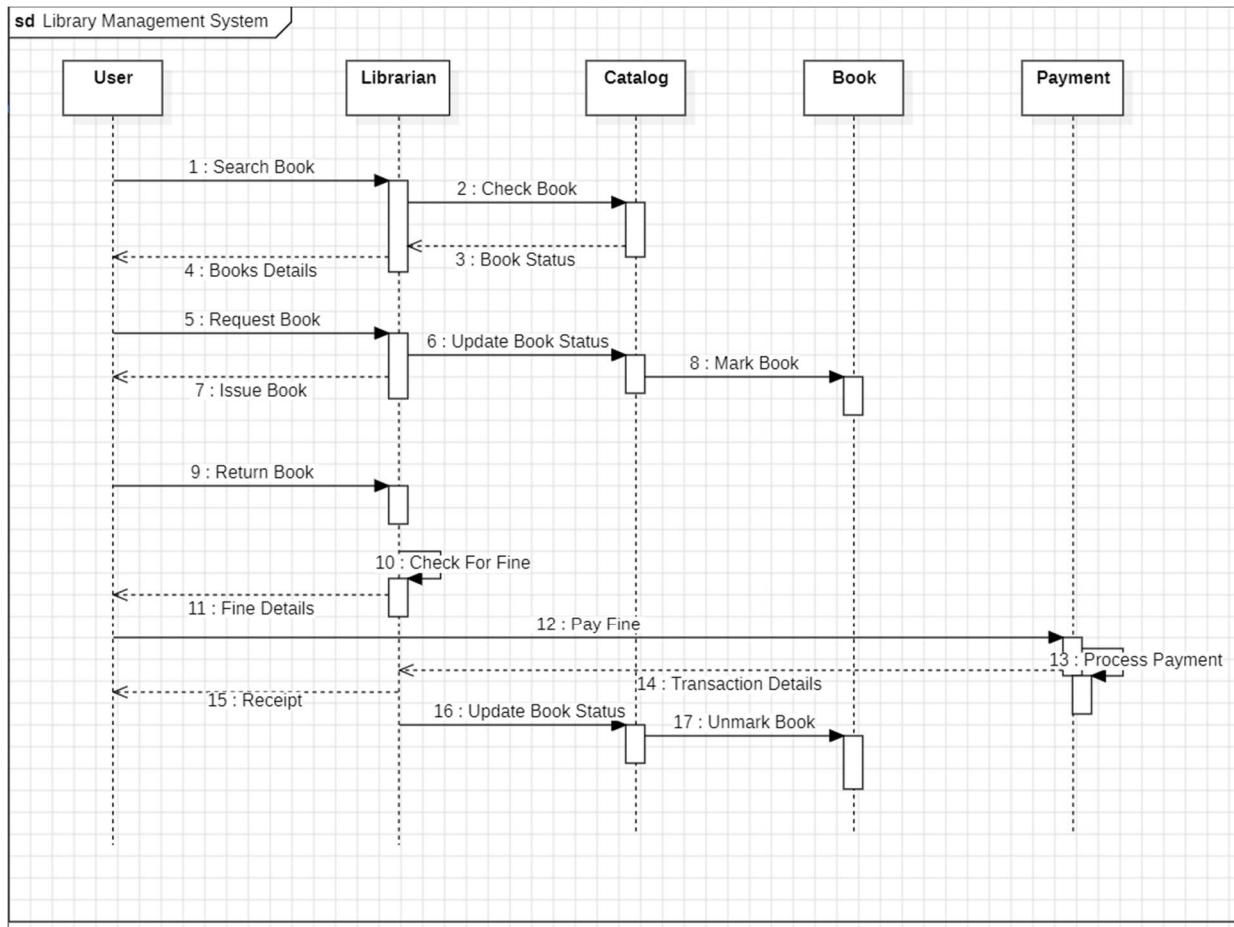


Fig 3.4 Library Management System - Sequence Diagram

The sequence diagram illustrates the process of a user borrowing a book from the library. The user begins by searching for a book in the library catalog. The catalog then searches for the book and returns the results to the user. The user then requests to borrow the book, and the library system checks its availability. If the book is available, the system issues the book to the user and updates its records. Finally, the user receives a receipt confirming the checkout. This diagram highlights the automated steps involved in the process and the interactions between the user and the library system.

## Activity Diagram

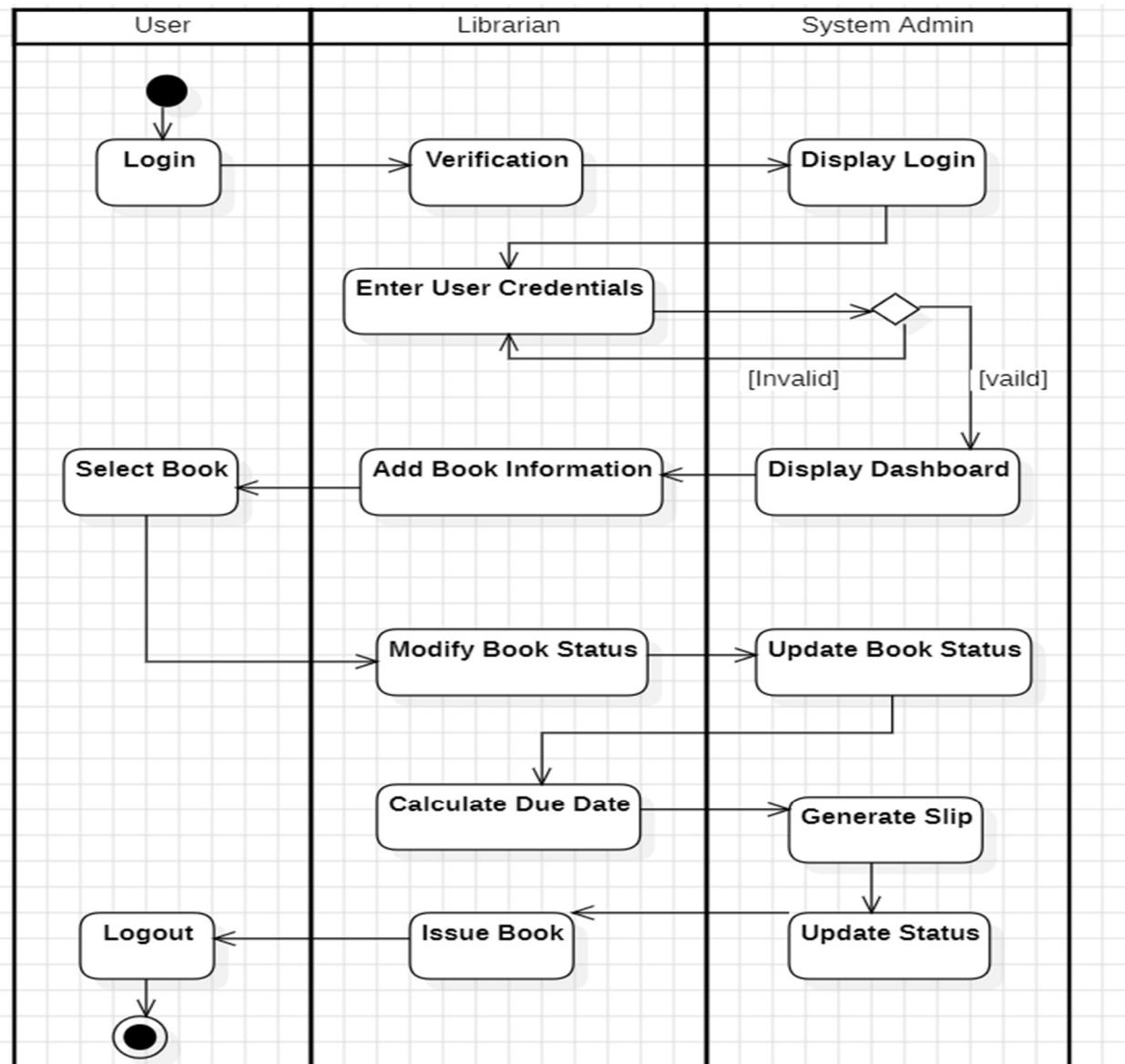


Fig 3.5 Library Management System - Activity Diagram

The activity diagram outlines the workflow of a library management system. It starts with a user logging in, followed by credential verification. Successful login grants access to book selection for the user. Simultaneously, librarians can add new books or modify existing book information. System administrators possess the authority to update book statuses and generate slips related to book transactions. Upon book selection, the system calculates the due date and issues the book to the user, updating the database accordingly.

## 4. Stock Maintenance System

### Software Requirement Specification

#### Stock Maintenance System

##### Introduction:

###### 1.1 Purpose of this document:

The main purpose of this document is to provide a detailed description of Functional, Non-functional, and system design and performance. This is intended to use by system developers and stakeholders involved in system's development.

###### 1.2 Scope of this document:

It covers essential features such as stock level management, order processing, and inventory maintenance. It also allows to purchase and sell stocks.

###### 1.3 Overview:

The document gives primary functions including inventory tracking, sales management and purchase monitoring. It also gives functional, non-functional requirements.

##### General Description:

The Stock Maintenance System provides several features such as stock maintenance, inventory monitoring, sell stock or buy stock, analyze sales data, real-time updates about stock-price etc.

- Inventory Updations:

The system must allow users to add, update and delete stock items including quantity of stock and buy prices.

- Stock Monitoring:

It should monitor stock level in real-time, alerting users when items fall below predefined thresholds.

### Interface Requirements:

User Interface: The system must provide a web based interface for users to manage stock & inventory easily.

Data Storage: It should include RDBMS to store & retrieve Data..

### Performance Requirements:

The system must handle concurrent users simultaneously without any degradation in the performance and the actions like searching and updating stock must be done quickly.

### Design Constraints:

The system must operate on Windows and Linux platforms. The system must comply with data protection, ensuring secure handling of user and stock data.

### Non-Functional Requirements:

\* Performance: The system shall respond to user actions within 2-3 seconds.

\* Security: The system must encrypt sensitive data.

\* Reliability: The system must be available continuously for stock maintenance operations.

## Class Diagram

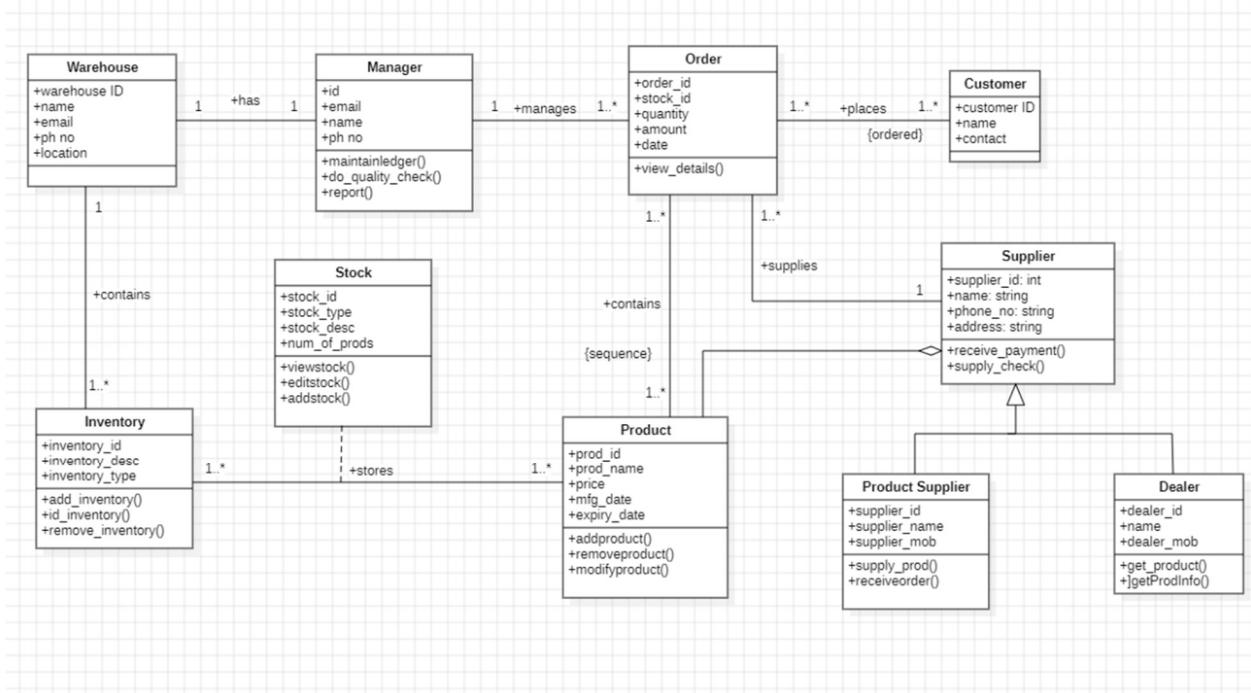


Fig 4.1 Stock Maintenance System - Class Diagram

The class diagram illustrates a warehouse inventory and order management system. The Warehouse contains multiple Inventory items, managed by a Manager who oversees operations like reporting and quality checks. Stock stores details about products, which are managed with functionalities like addition and modification. Orders placed by Customers link products to quantities and amounts, while Suppliers, including Product Suppliers and Dealers, handle the supply of products to the warehouse. The diagram highlights the interactions between inventory, stock, orders, and suppliers within the system.

## State Diagram

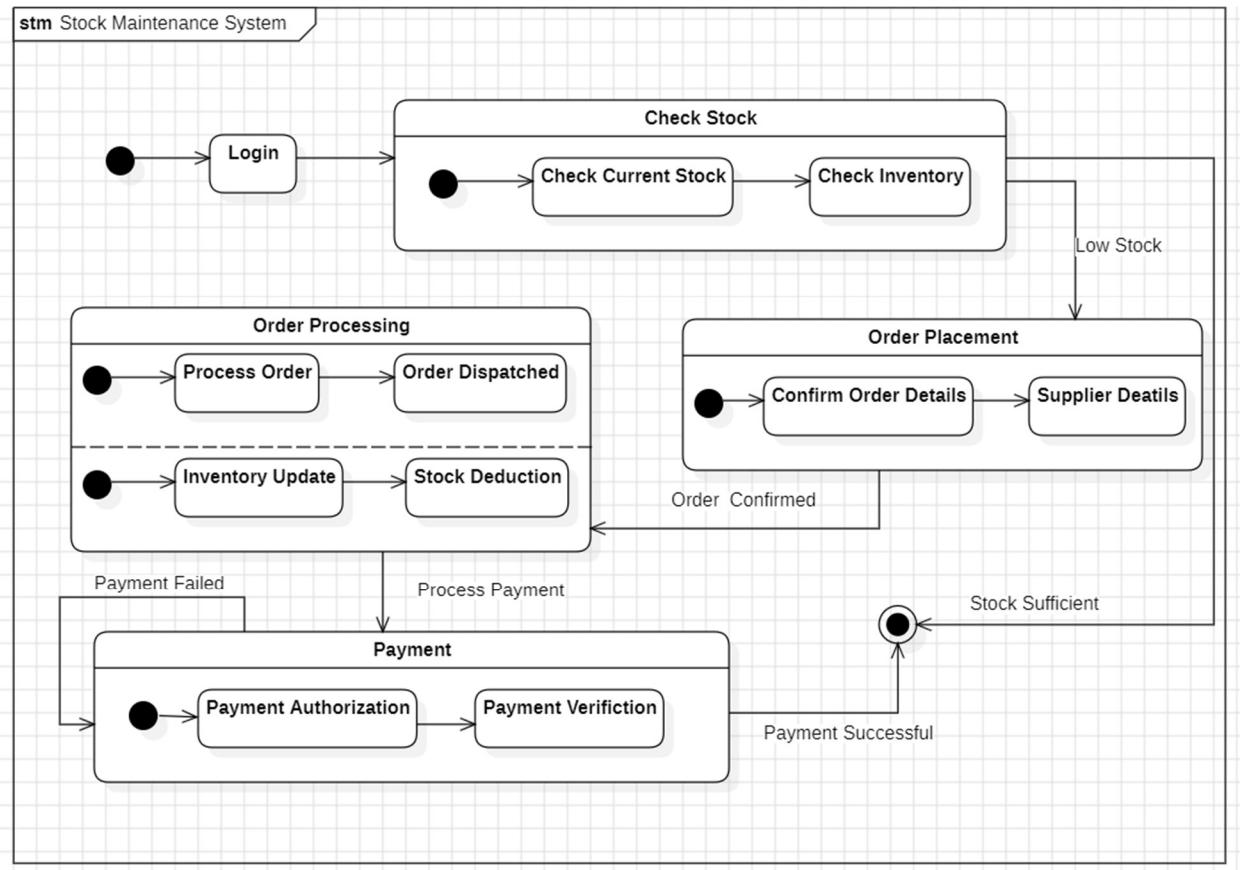


Fig 4.2 Stock Maintenance System - State Diagram

The state diagram illustrates the stock maintenance system's workflow. It starts with a user logging in. The system then checks current stock and inventory levels. If stock is low, the system transitions to the "Order Placement" state, where order details are confirmed and supplier details are obtained. After the order is confirmed, the system moves to the "Order Processing" state, where the order is processed and dispatched. During order processing, stock is deducted and inventory is updated. Finally, the system transitions to the "Payment" state, where payment is authorized and verified. Upon successful payment, the system returns to the "Check Stock" state to monitor inventory levels.

## Use Case Diagram

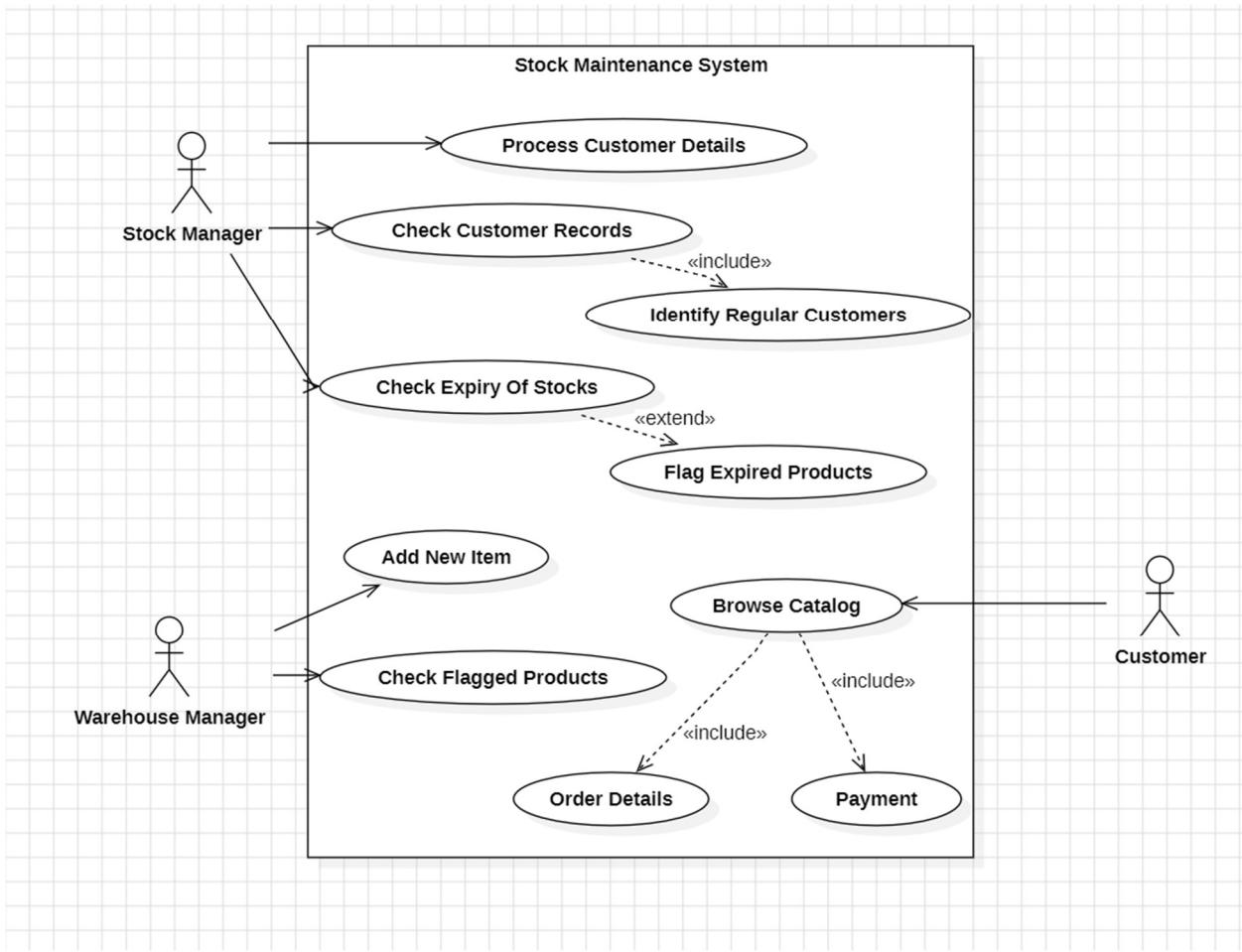


Fig 4.3 Stock Maintenance System - Use Case Diagram

The Stock Maintenance System is designed to manage inventory and customer interactions for a business. The system has three main actors: Stock Manager, Warehouse Manager, and Customer. The Stock Manager can process customer details, check customer records, and identify regular customers. They can also check the expiry of stocks and flag expired products. The Warehouse Manager can add new items to the inventory and check flagged products. The Customer can browse the catalog, place orders, and make payments. The system includes use cases for order details and payment, which are further elaborated by the "include" relationships. This system aims to streamline inventory management and provide a seamless experience for customers.

## Sequence Diagram

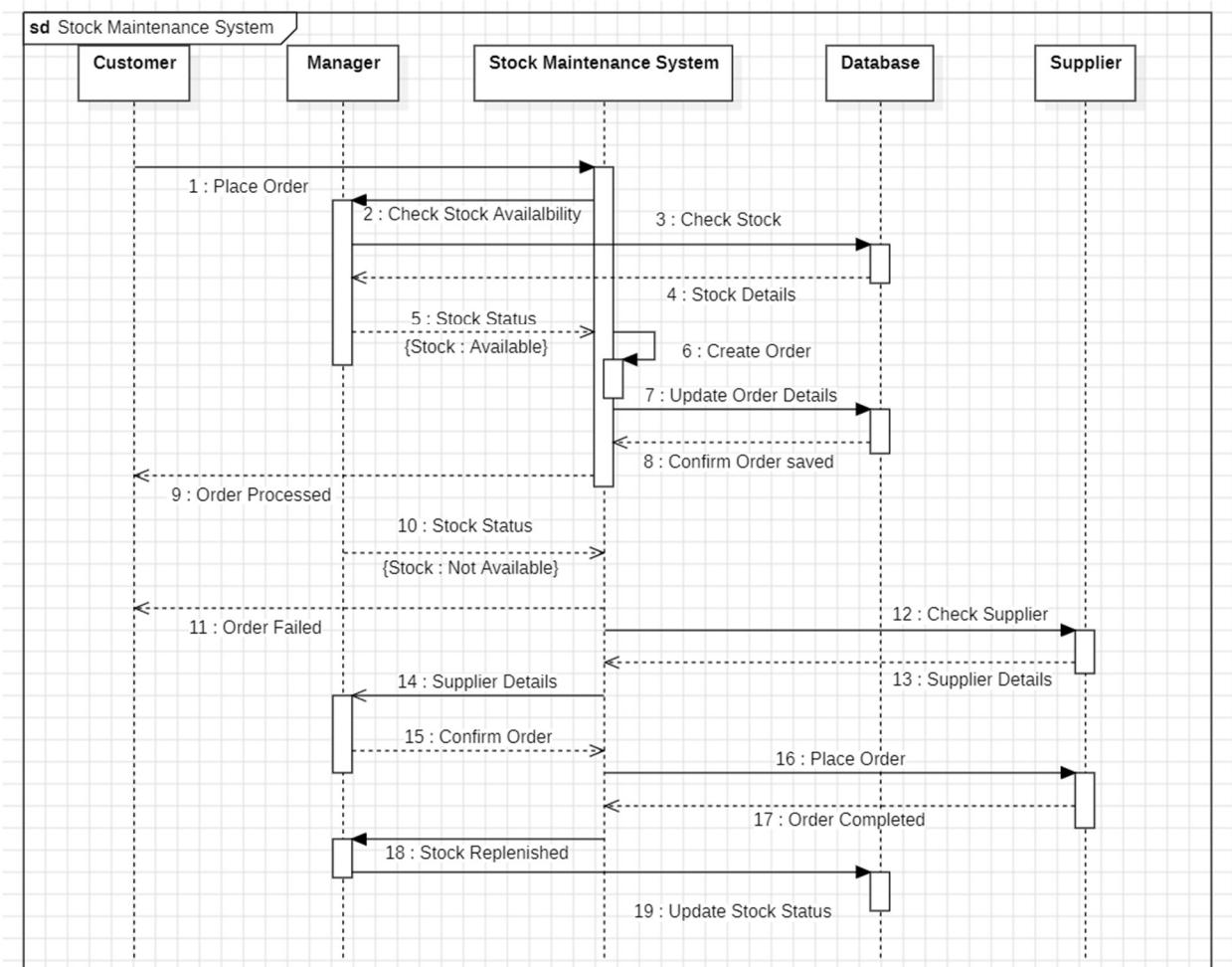


Fig 4.4 Stock Maintenance System - Sequence Diagram

The sequence diagram outlines the order fulfillment process in a stock maintenance system. It begins with the customer placing an order. The manager then checks stock availability, and the system verifies stock levels in the database. If stock is sufficient, the order is created and processed. If stock is insufficient, the system checks with suppliers, places orders, and updates stock levels once the replenishment is complete. The system communicates order status updates to the customer throughout the process. This diagram illustrates the interactions between the customer, manager, database, and suppliers, highlighting the steps involved in fulfilling an order effectively.

## Activity Diagram

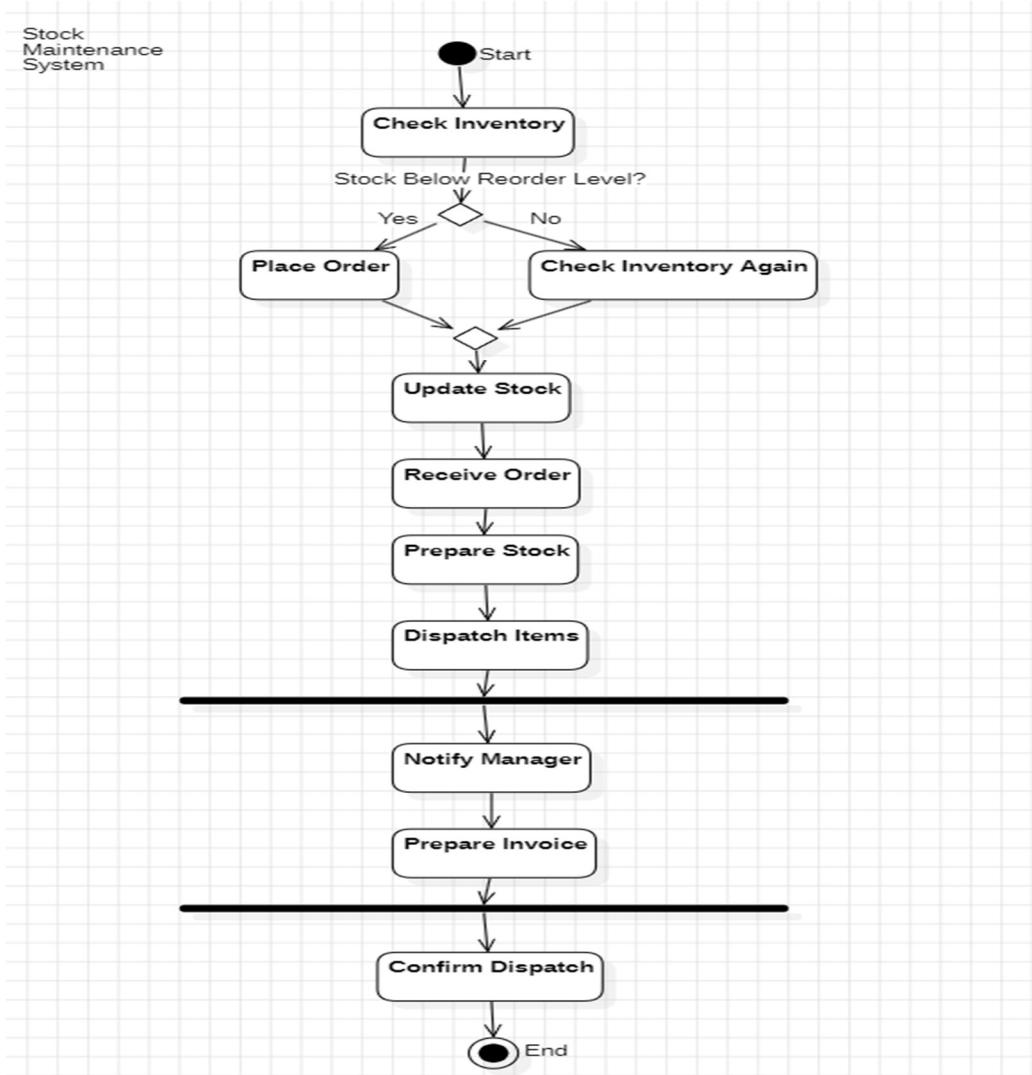


Fig 4.5 Stock Maintenance System-Activity Diagram

This activity diagram represents the workflow of a stock maintenance system. It begins with checking inventory levels, followed by a decision point to assess if stock is below the reorder level. If stock is low, an order is placed, otherwise, inventory is rechecked. Upon receiving the order, the stock is updated, prepared, and dispatched. The process then involves notifying the manager, preparing the invoice, and confirming the dispatch, concluding the workflow.

## 5. Passport Automation System

### Software Requirement Specification

#### Introduction:

##### 1.1 Purpose of this document:

The main purpose of this document is to provide a detailed description of railway reservation system. It is intended to be used by the system's developers, testers and stakeholders involved in system development.

##### 1.2. Scope of this document:

The document covers the functional and technical aspects of the Railway Reservation System. It includes booking, cancellation, seat availability checking etc.

##### 1.3. Overview:

The remainder of this document gives information about system Design, performance, User interface, functional and non-functional requirements.

#### General Description:

The system is designed to automate the process of booking, cancelling and managing train tickets. It will provide an easy-to-use platform for passengers to check seat availability, book tickets, make online payments.

### Functional Requirements:

- \* Ticket booking : The system must allow users to search for trains, check seat availability and book tickets.
- \* Payment Processing : The system must support online payments through multiple payment gateways for ticket purchases.

### Interface Requirements:

- \* User Interface : The system must provide a web-based and mobile interface for users to book/ cancel tickets etc.
- \* Database Interface : The system should interface with RDBMS to store train schedules, bookings and users.

### Performance Requirements:

- \* The system should complete actions such as searching for trains and booking tickets.
- \* The system shall process bookings and cancellations under 2 seconds.

### Non-Functional Requirements:

- \* Performance : The system should respond to user actions like ticket booking or seat availability checking.
- \* Security : The system must implement secure authentication and encryption methods to protect sensitive user data.

### Preliminary Budget and Schedule:

Requirement specification - 2 weeks

Design & development - 4 weeks

Evolution - 4 weeks

Budget - ₹ 10,000/-

## Class Diagram

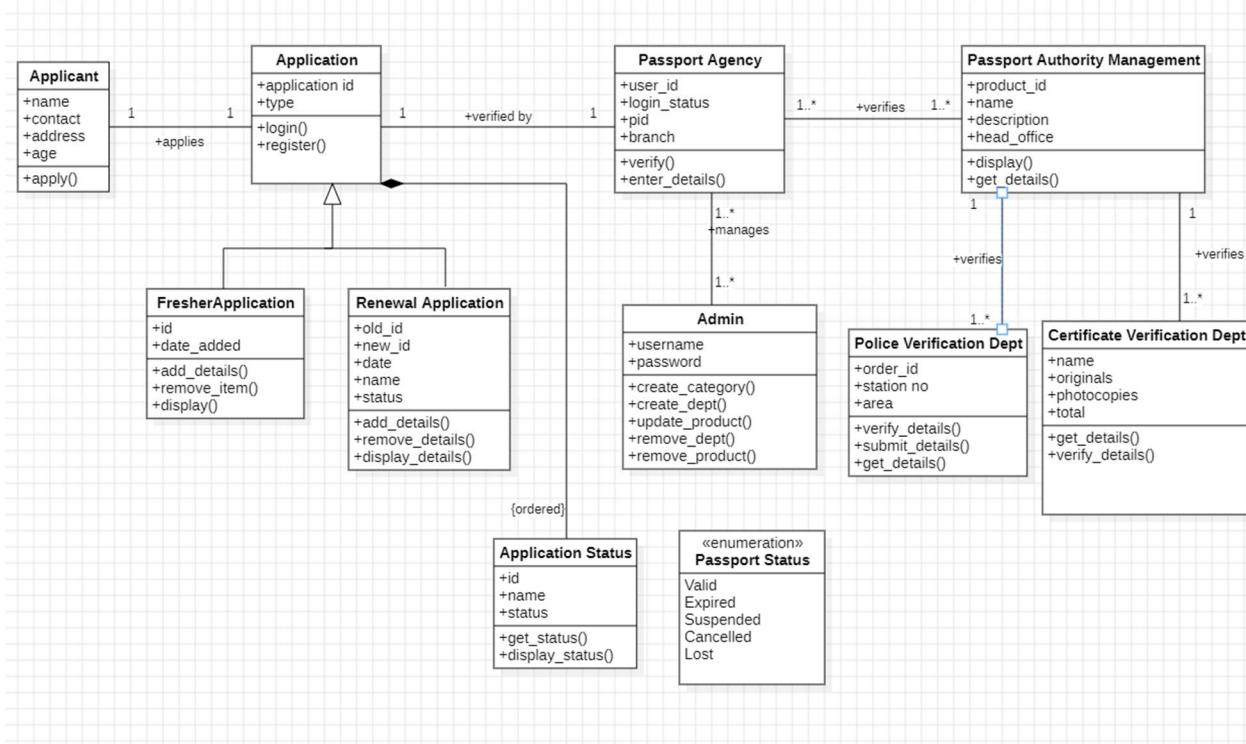


Fig 5.1 Passport Automation System - Class Diagram

The class diagram depicts the structure of a passport application and verification system. It illustrates various entities, such as **Applicant**, **Application**, and its specialized forms: **FresherApplication** and **RenewalApplication**. The **Application** class is associated with **Applicant**, who can apply and register for passport services. The system includes a **Passport Agency** and its management under **Passport Authority Management**, which oversees verification processes through departments like **Police Verification** and **Certificate Verification**. The diagram also involves an **Admin** class responsible for managing categories, departments, and products. Key features include status tracking through **Application Status** and **Passport Status** enumeration. Relationships between classes are depicted with multiplicity, inheritance, and composition, highlighting functionalities like verifying details, managing applications, and updating statuses.

## State Diagram

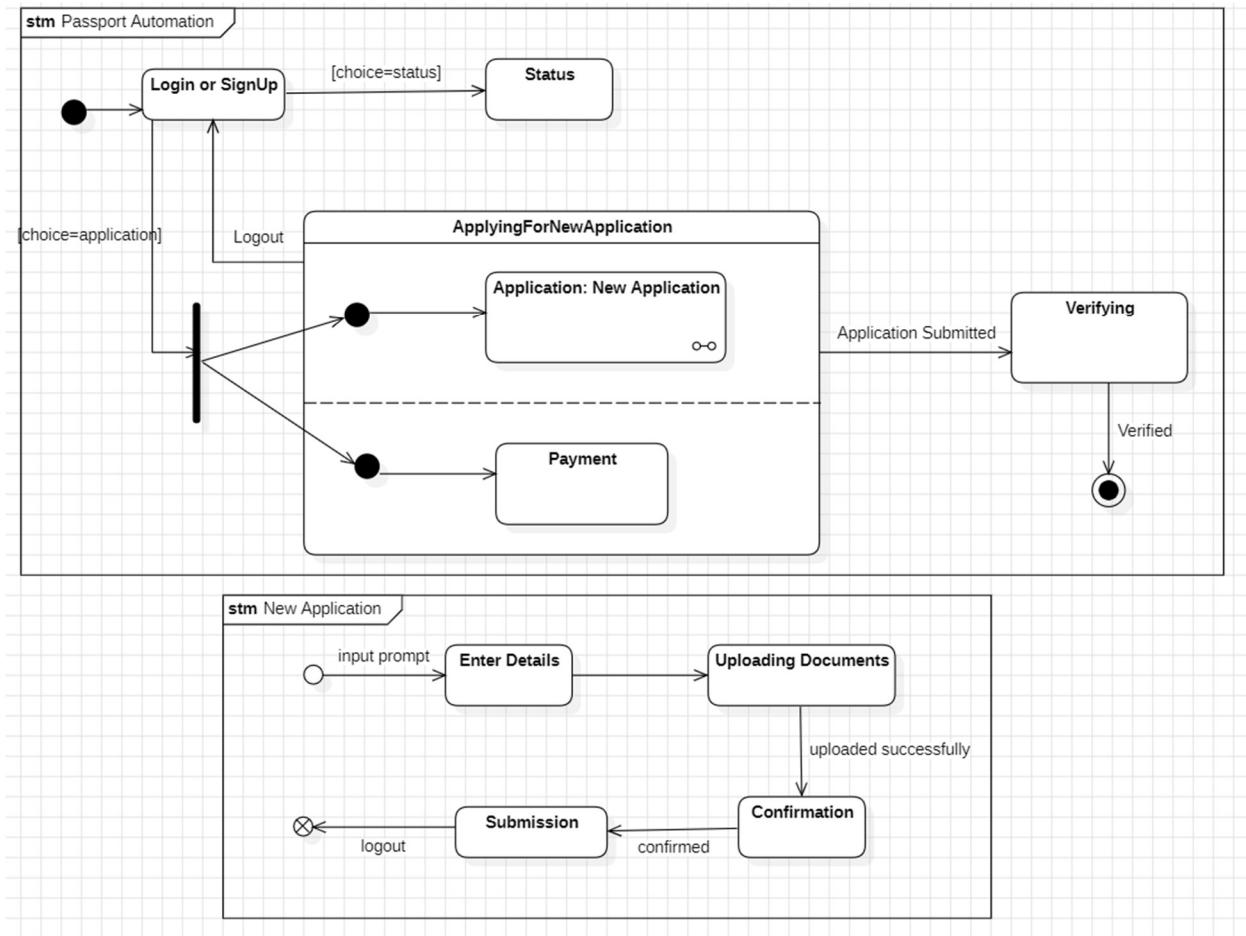


Fig 5.2 Passport Automation System - State Diagram

The state diagram illustrates the passport automation system. The system starts with the user logging in or signing up. After login, the user can choose to check the status of their application or apply for a new one. If the user chooses to apply, they enter the "ApplyingForNewApplication" state. Within this state, the user fills out the application form, uploads documents, and submits the application. Once submitted, the application enters the "Verifying" state. If the application is verified successfully, the user receives a confirmation. The user can also log out at any point during the process.

## Use Case Diagram

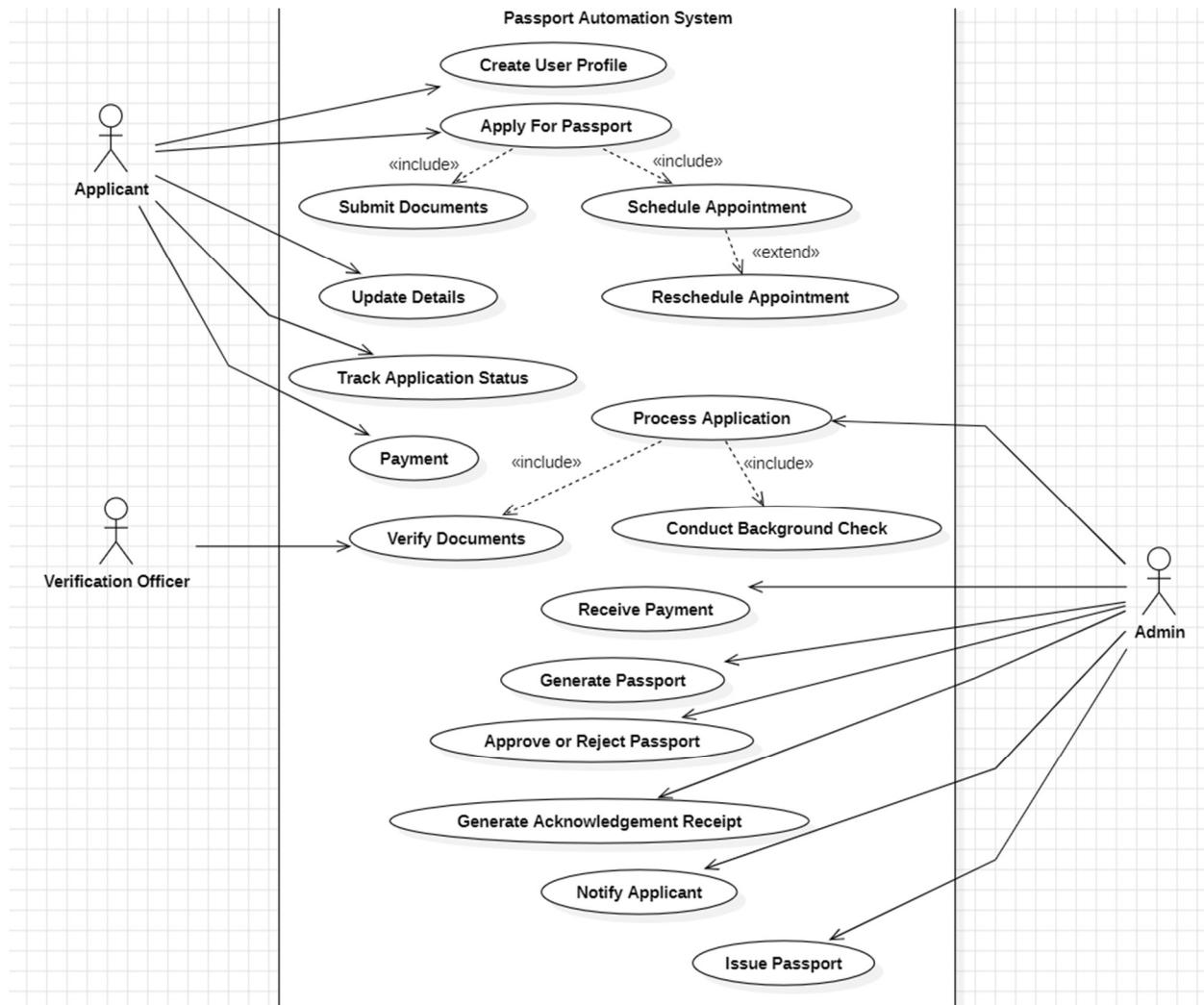


Fig 5.3 Passport Automation System - Use Case Diagram

The diagram illustrates a Use Case Diagram for a Passport Automation System, outlining the interactions between the system and its primary actors: Applicant, Verification Officer, and Admin. The Applicant begins by creating a user profile and applying for a passport, which includes submitting documents, scheduling (or rescheduling) appointments, making payments, and tracking application status. The Verification Officer is responsible for verifying documents and supporting the application processing. The Admin plays a key role in processing applications, conducting background checks, receiving payments, approving or rejecting passport requests, and issuing passports. Additional use cases include generating acknowledgments and notifying applicants of the application's status. This diagram effectively demonstrates the workflow and responsibilities of each actor in the passport issuance process.

## Sequence Diagram

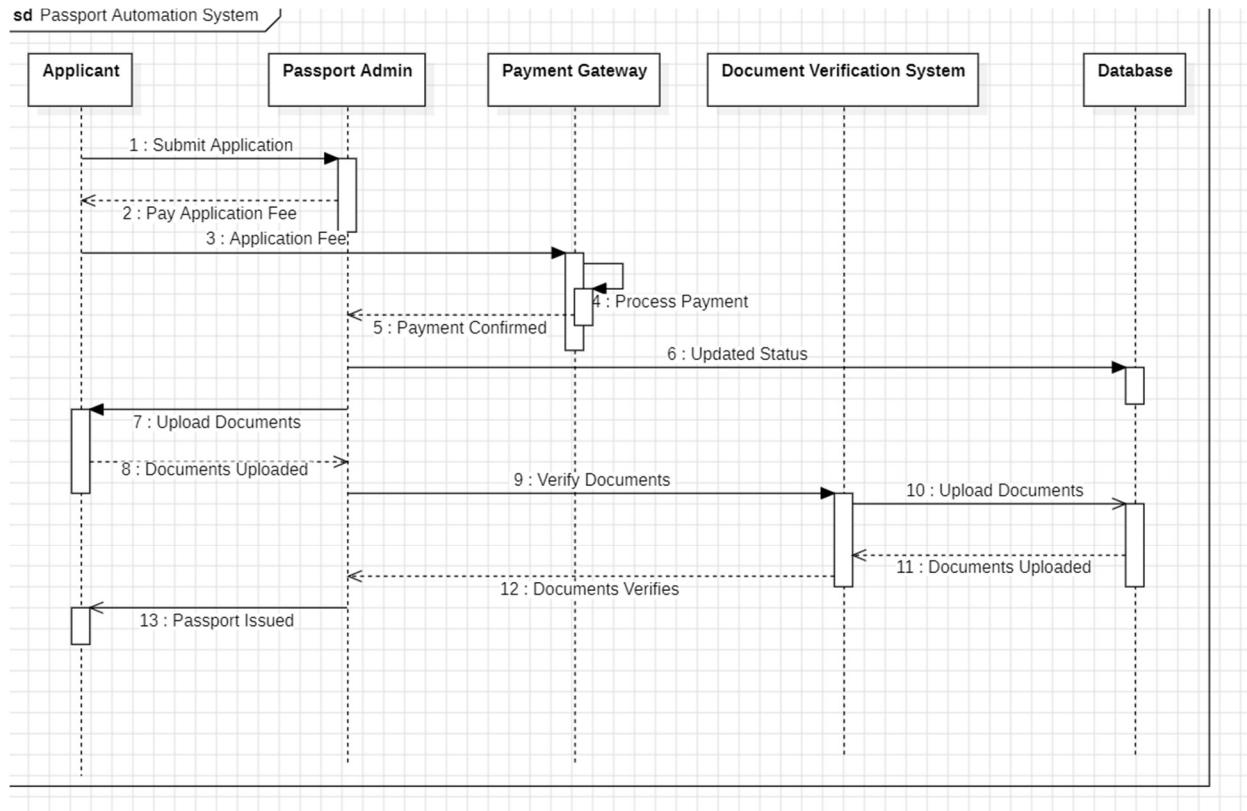


Fig 5.4 Passport Automation System - Sequence Diagram

The sequence diagram illustrates the process of applying for a passport. The applicant starts by submitting an application and then pays the application fee. The payment gateway processes the payment and updates the status. The applicant then uploads the required documents, which are verified by the document verification system. Once the documents are verified, the passport is issued to the applicant. This diagram shows the interactions between the applicant, passport admin, payment gateway, document verification system, and database throughout the passport application process.

## Activity Diagram

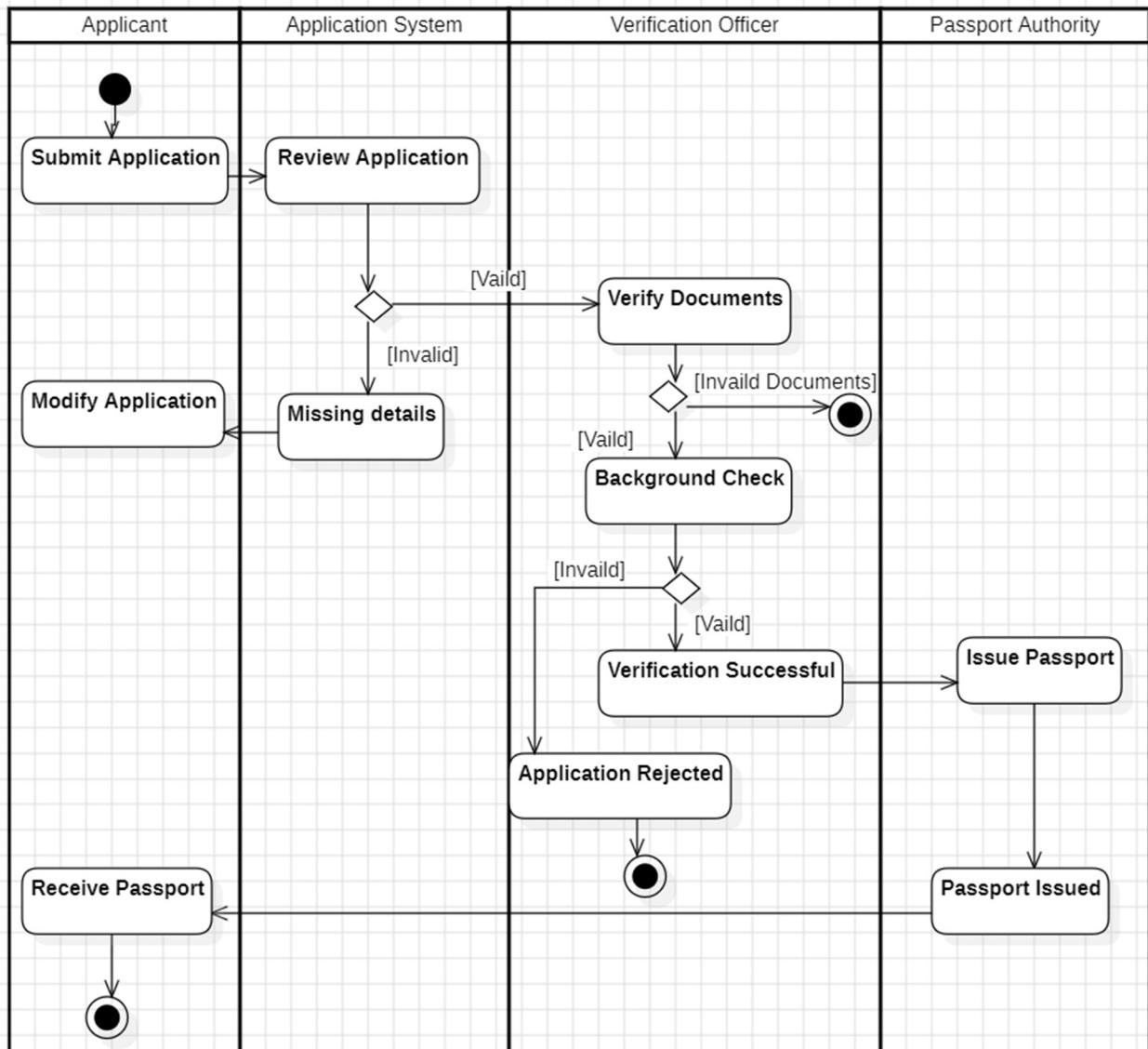


Fig 5.5 Passport Automation System - Activity Diagram

The activity diagram illustrates the passport application process. It starts with the applicant submitting an application. The application system reviews the application. If the application is complete, it proceeds to document verification. If invalid documents are found, the application is rejected. If valid, a background check is conducted. If the background check is clear, the verification is successful, and the passport is issued. If any stage fails, the application is rejected. The applicant can receive the passport once it's issued.