

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



Lab Record

Software Engineering and Object-Oriented Modeling

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SANKETH M HANASI

1BM22CS242

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 Analysis and Design(22CS6PCSEO) laboratory has been carried out by **SANKETH M HANASI**(1BM22CS242) during the 5th Semester Oct24-Jan2025.

Signature of the Faculty Incharge:

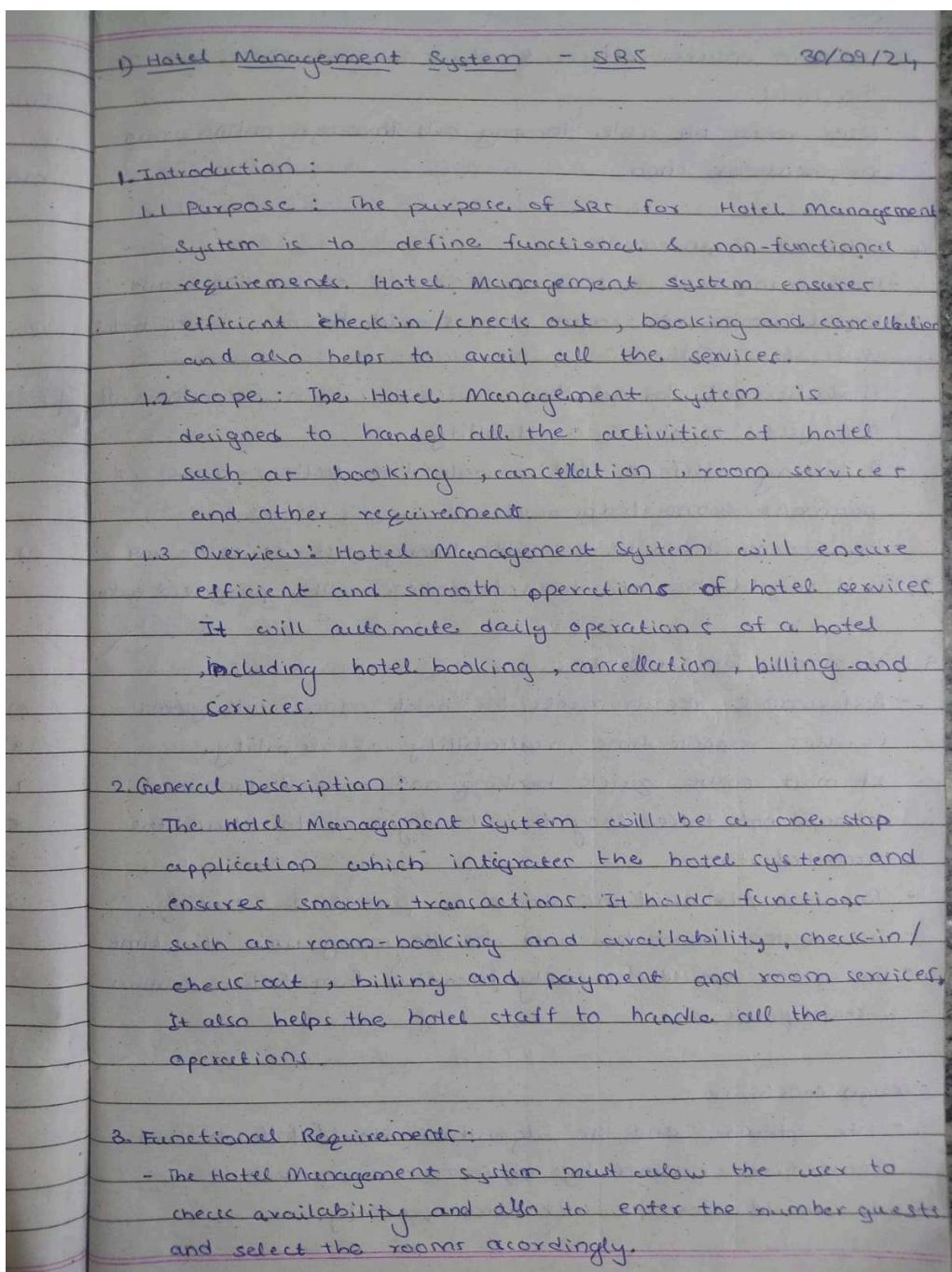
M Lakshmi Neelima
Assistant Professor
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

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



- It must ensure efficient check-in and check-out at the hotel. 1)
- User must be able to pay bill through online, card payment or cash. 7)

4. Interface Requirements:

- Hotel management system must ensure secure login and logout procedure. Login page should be for both hotel staff & guests.
- A dashboard for all the services which help the staff and guest to check availability.
- A billing interface for smooth transactions and payment acknowledgement.
- The system should be accessible through all devices and 8) it should be securely integrated with payment gateway.

5 Performance Requirements:

- Performance requirements for hotel management system includes response time, reliability, scalability. It must ensure quick booking and cancellation and should not take long time to login or to return the searched item.
- Payment / transaction should be processed quickly.
- The system must be able handle many users at a time.
- The system should be easily recovered from failure, within a short period. 10

6 Design Constraints:

- The software and the algorithm should follow standard procedure designed by requirements management.
- It should store the user data securely.

1) Credit Card Management System:

7) Non Functional Requirements:

- security should be provided to user data and only authorized entities can access them.
- The interface should be simple and intuitive for guests and hotel staff.
- The system should be reliable ie recoverable from failure.
- It must be portable, that can be accessed through any device.

8. Preliminary Schedule and Budget:

- Schedule: Requirement analysis → 1 week;
software development - 3 weeks, Validation - 1 week
testing - 1 week
- Budget - 25 weeks

NP
20/9/20

Class Diagram

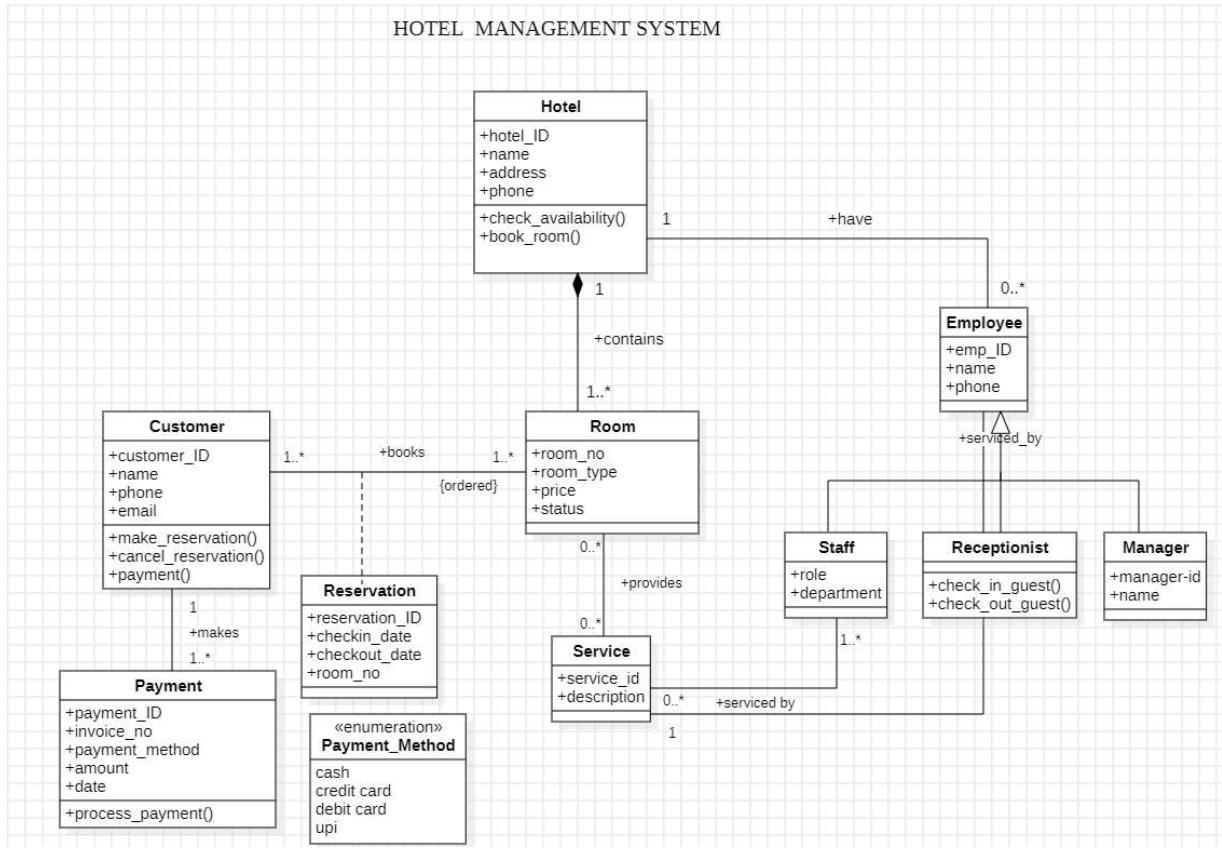


Fig1.1 Hotel Management System - Class Diagram

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

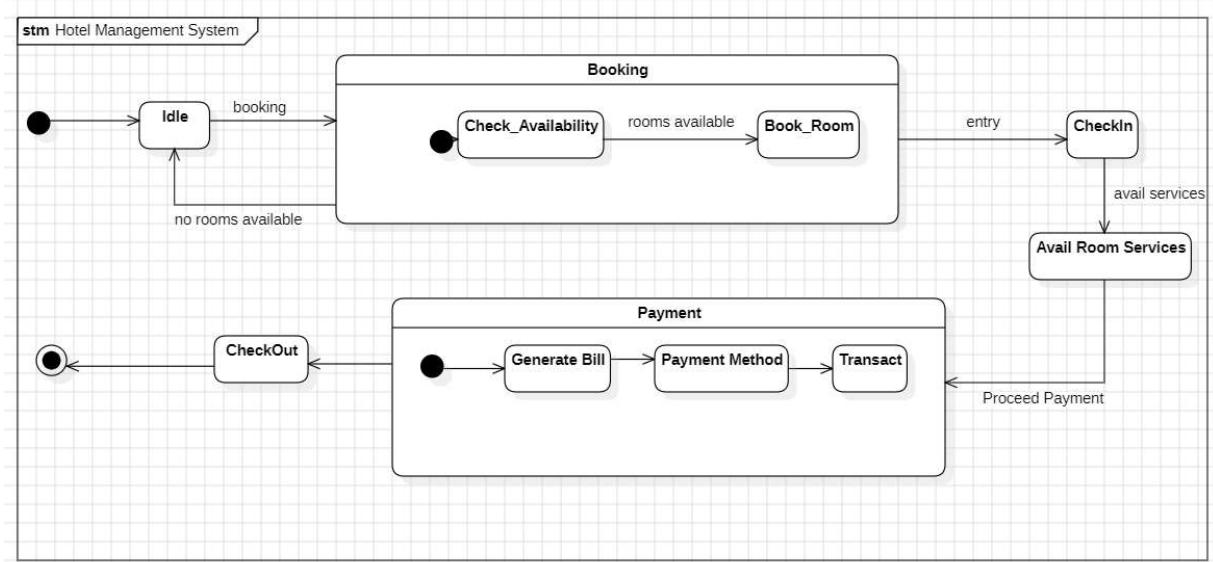


Fig1.2 Hotel Management System - State Diagram

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

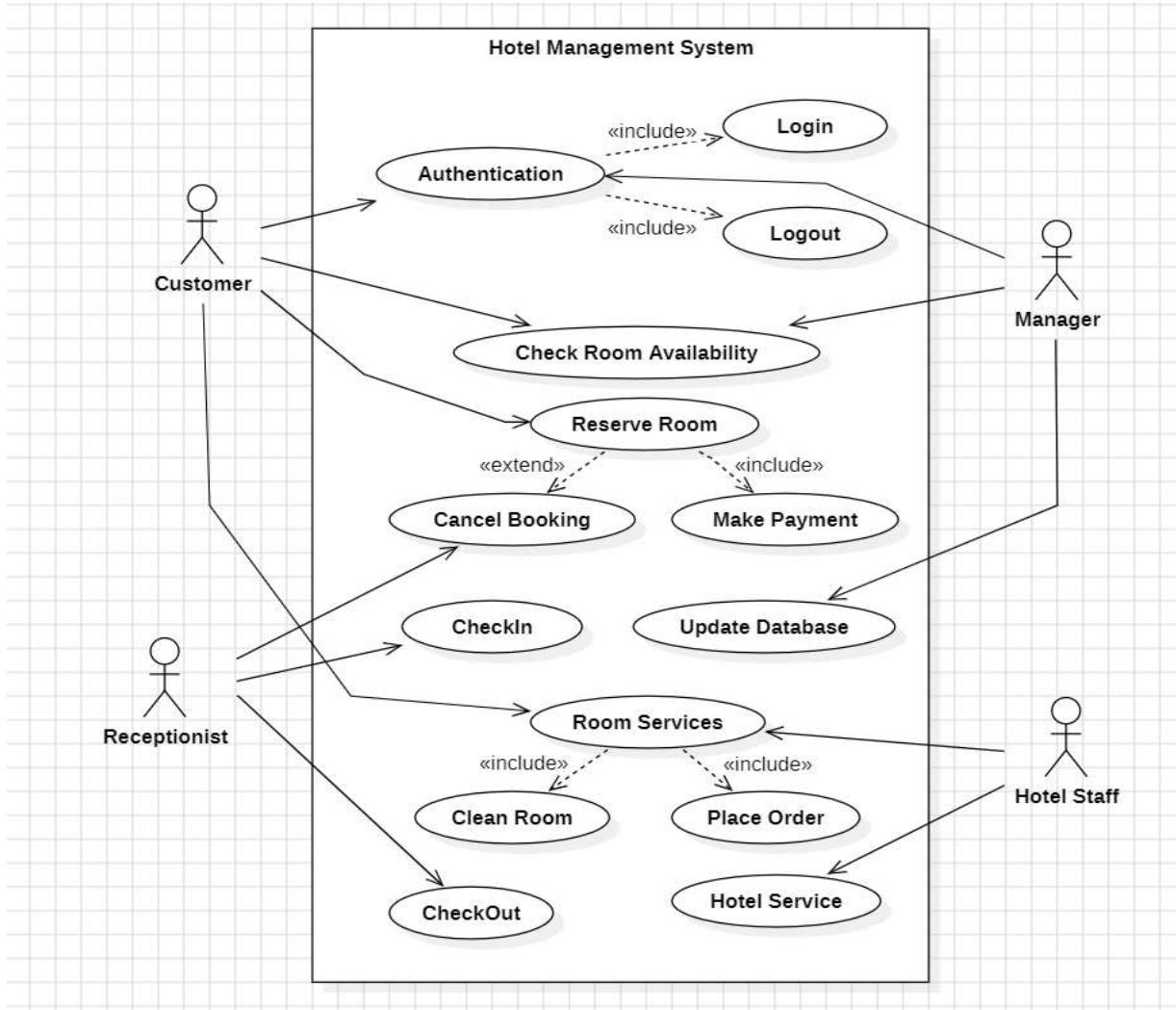


Fig1.3 Hotel Management System - Use Case Diagram

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

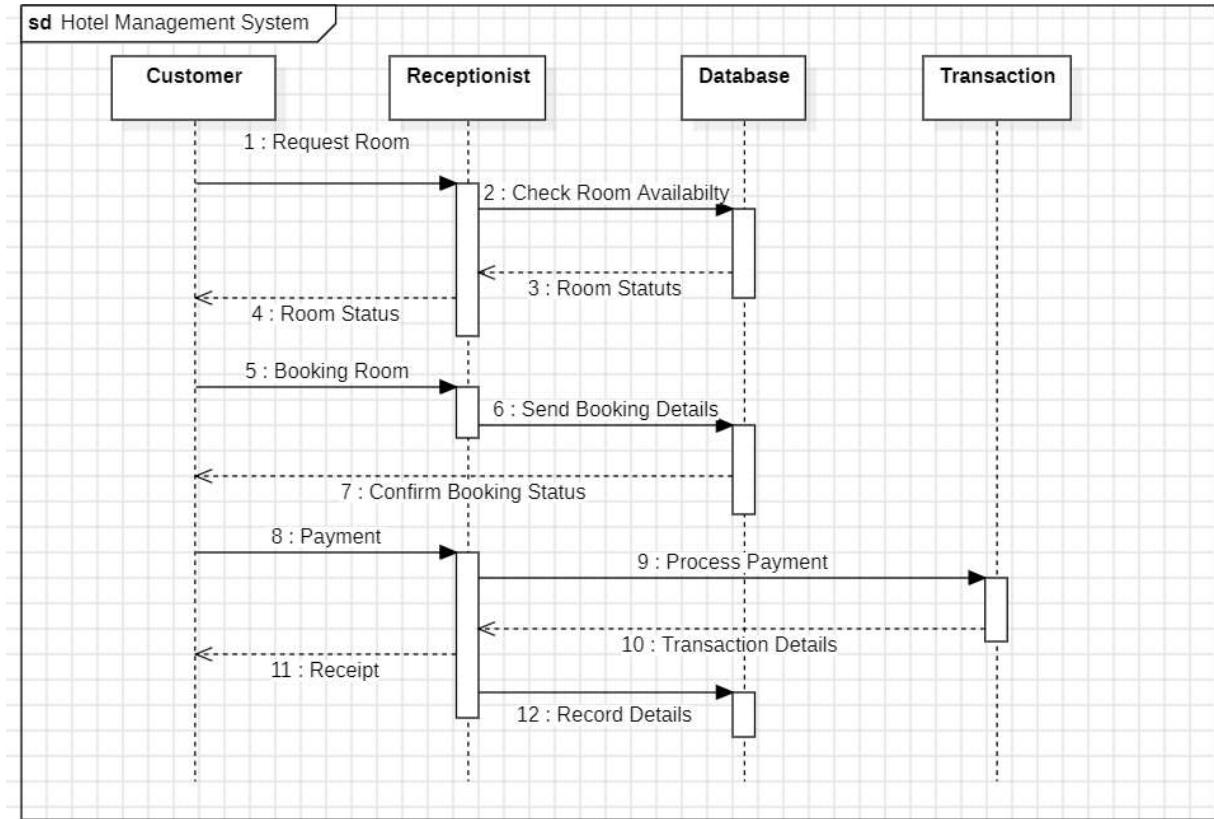


Fig1.4 Hotel Management System - Sequence Diagram

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

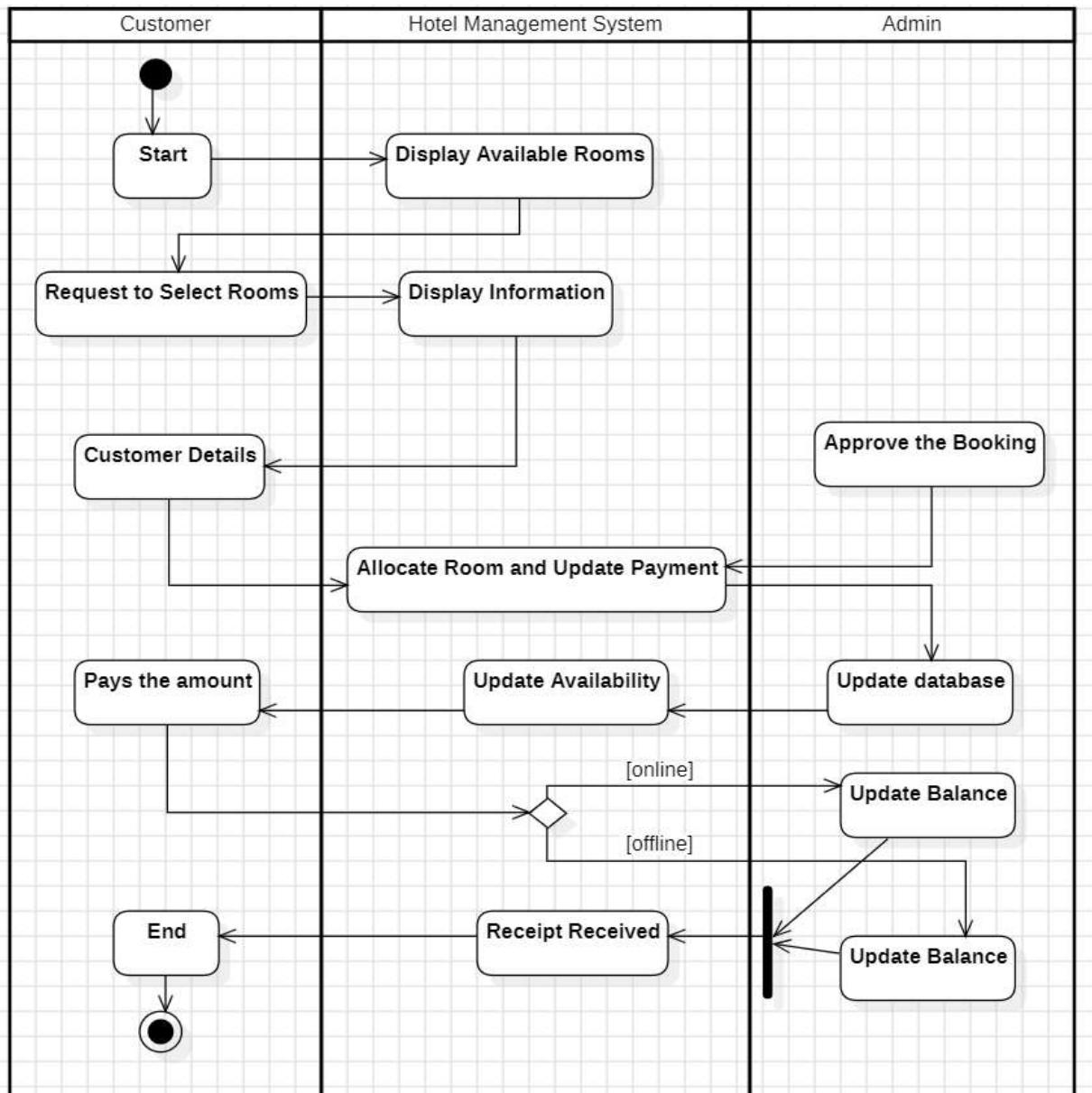


Fig1.5 Hotel Management System - Activity Diagram

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

2) Credit Card Management System:

1. Introduction:

Purpose: The credit card management system aims to provide a secure & efficient system for managing credit services.

Scope: The credit card management system will automate the processes such as card issuance, transactions, billing and payment management.

Overview: The credit card management system will be a application that allows users to manage credit card accounts & perform various transactions. It will provide a user friendly interface & robust security measures to protect sensitive customer data.

2 General description:

The credit card management system will integrate with existing banking infrastructure to support real-time credit card processing & management. It will ensure card issuance, account management, transaction, feedback, reporting and other services.

3 Functional Requirements:

- Enable user to manage credit card applications & approvals.
- Setting the credit limit based on customer preference.
- Transaction history view for users.
- Enable secure online payments for cardholders & payment acknowledgement.
- Reporting sections to report financial activities.

4. Interface Requirements:

- Enabling card holders to have secure logging & logout procedure, also allowing them to view account information, payment & check transaction history.

monitoring transactions & customer support tools.

5. Design Constraints:

- The system must be in the design standards for credit card data security.
- Must stick on PCI data protection regulations & to ensure user privacy & data security.
- The system must support variety of web browsers & devices.

6. Performance Requirements:

- The system should process transactions within seconds under normal conditions.
- Scalability - system must be able handle to large number of transactions.
- Reliability - The system must maintain higher uptime and must be able to recover from failure within in small time.

7. Non-functional Requirements:

- Security - implementing strong encryption for data transmission & storage.
- Usability - the interface must be minimal & user friendly.
- Performance - the system should handle concurrent access by many user without degrading the performance.

8. Preliminary Schedule & Budget:

- Schedule - Requirement specification & analysis - 4 weeks, system design & software development - 12 weeks, Deployment & testing - 4 weeks
- Budget - 60 lakhs

Class Diagram

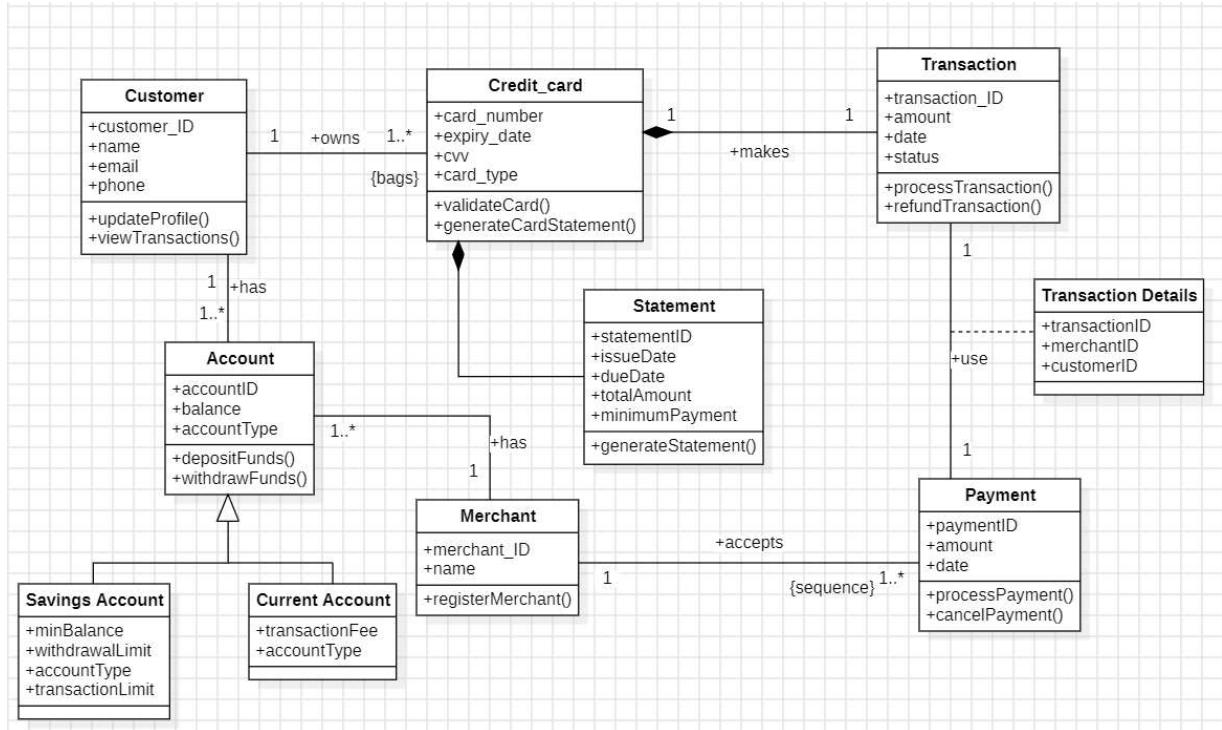


Fig 2.1 Credit Card Processing System - Class Diagram

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

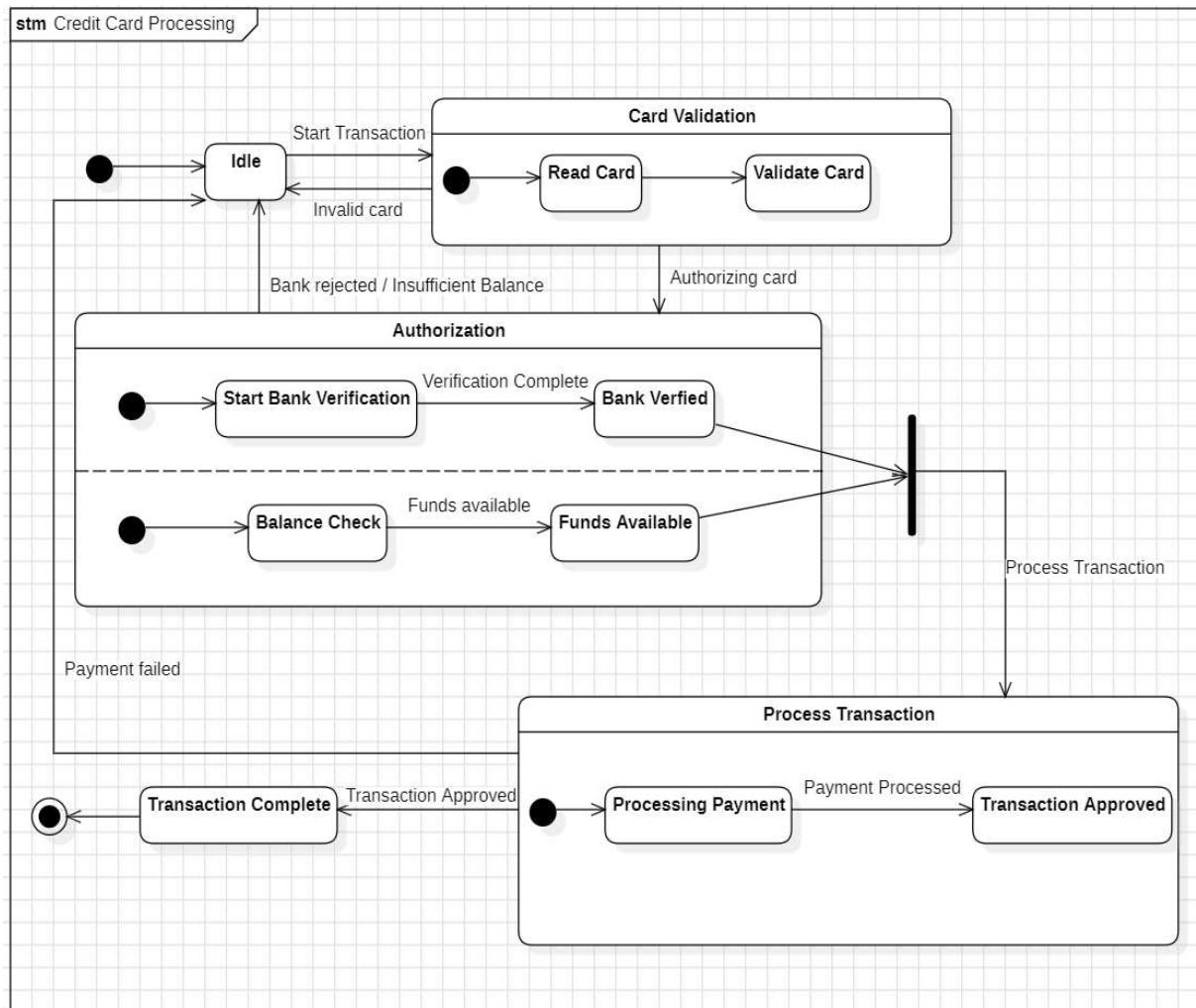


Fig 2.1 Credit Card Processing System - State Diagram

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

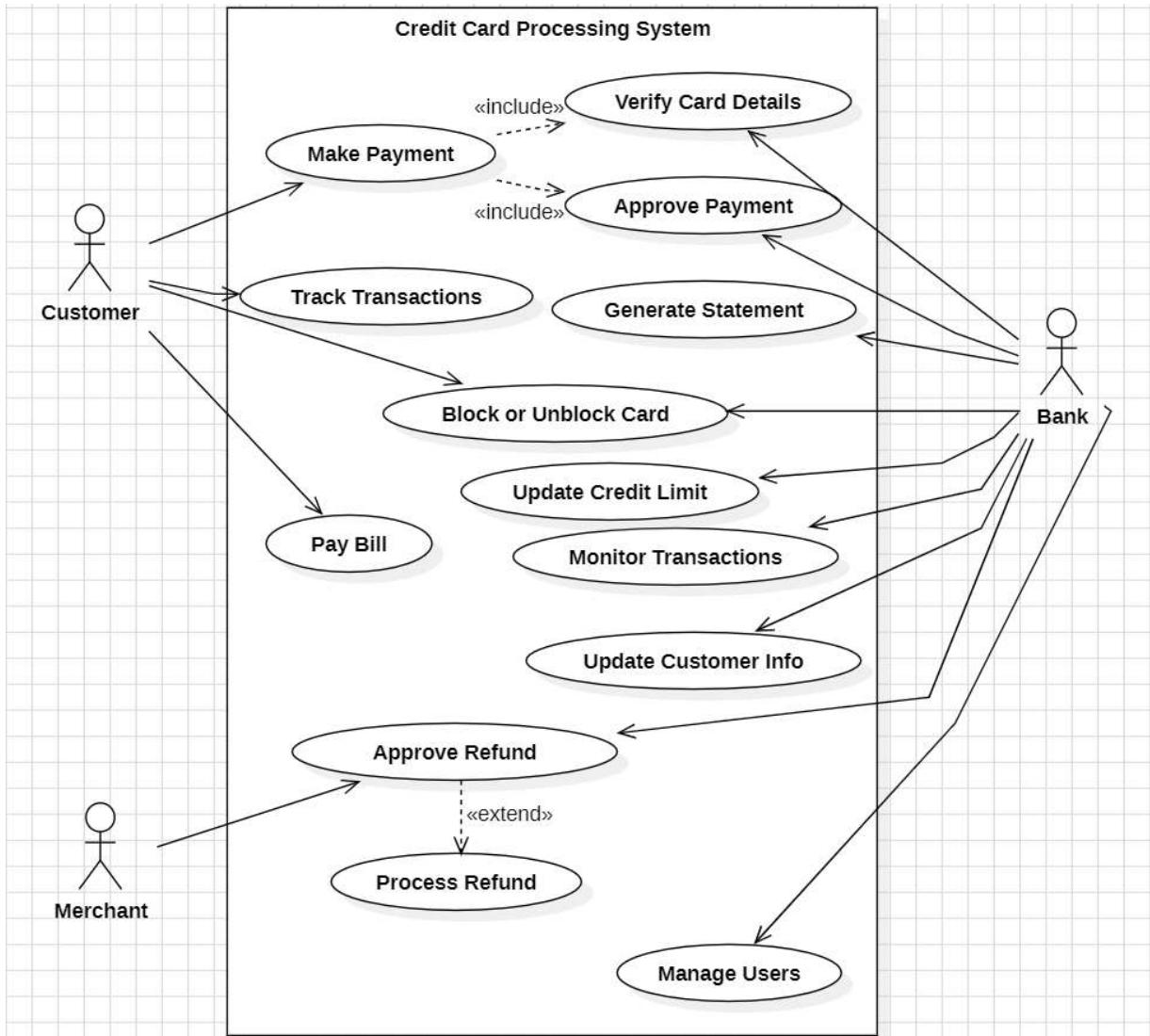


Fig 2.1 Credit Card Processing System - Use Case Diagram

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

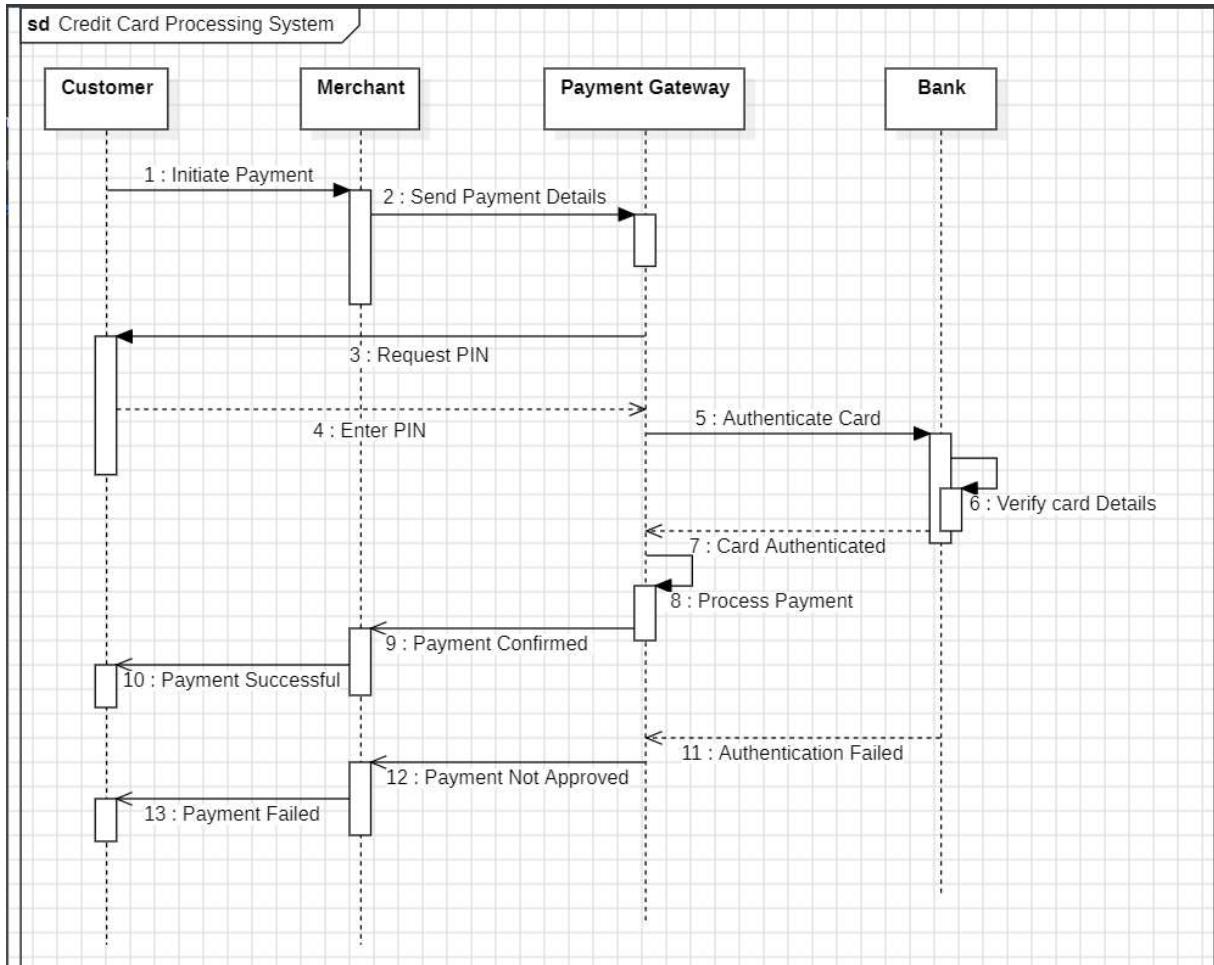


Fig 2.1 Credit Card Processing System - Sequence Diagram

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

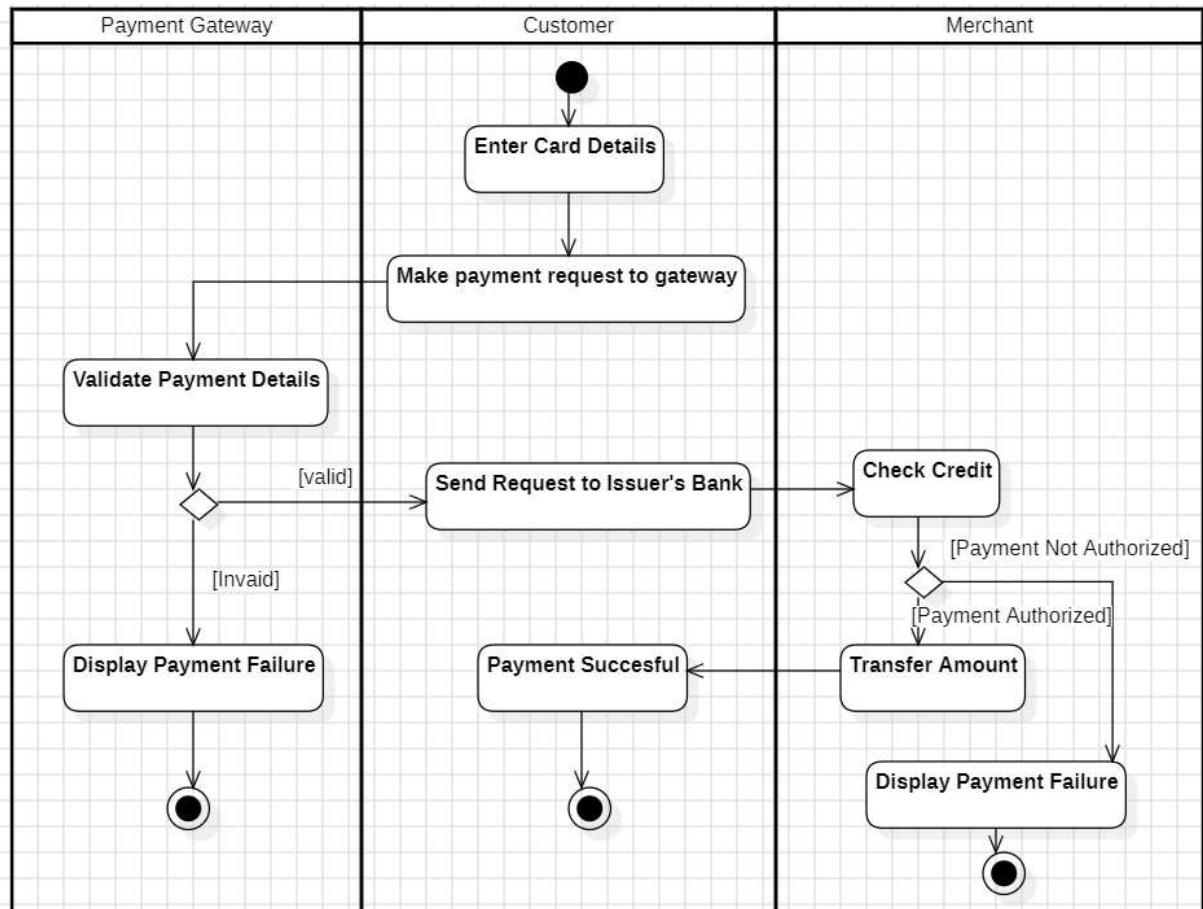


Fig 2.5 Credit Card Processing System - Activity Diagram

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

details & monitor transactions.

- Interface must support the payment gateways.

5) Design Constraints:

- The user data must be protected under protection regulations.
- The system should be accessible from all the systems (Windows, MacOs, Linux) & also using mobiles.

6) Performance Requirements:

- The system must be capable of handling large number of users under normal conditions without affecting user experience.
- The payment transactions must be completed within seconds.

7) Nonfunctional requirements:

- Security - only the authorized user must be able to use the services & role-based authentication must be provided.
- Usability - the interface must be minimal & userfriendly.
- Reliability - the system must be recoverable & must have low failure rates.

8) Preliminary Schedule & Budget:

- Schedule : Requirement specification & analysis - 3 weeks, Software development - 6 weeks, testing - 2 weeks, Evaluation - 2 weeks, Deployment - 3 weeks.
- Budget : 10 lakhs.

Class Diagram

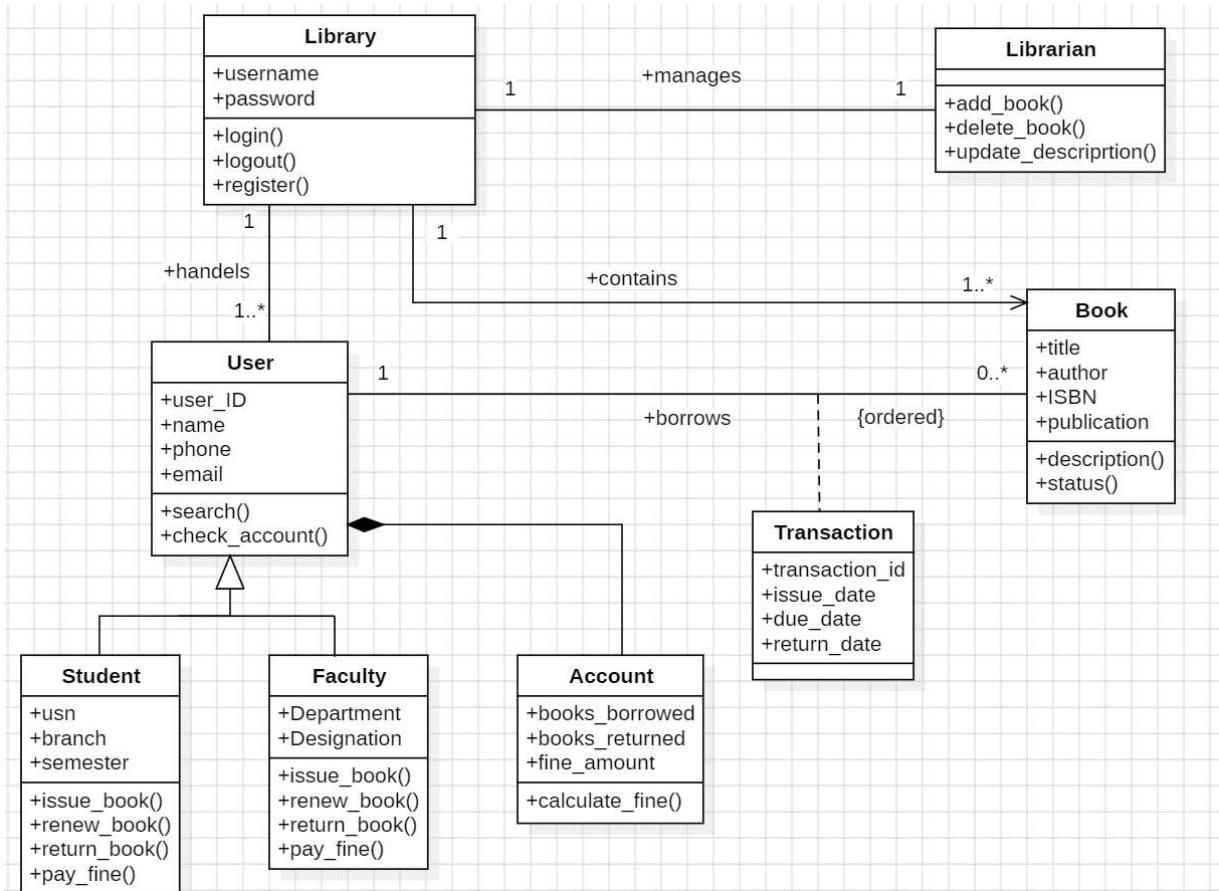


Fig 3.1 Library Management System -Class Diagram

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

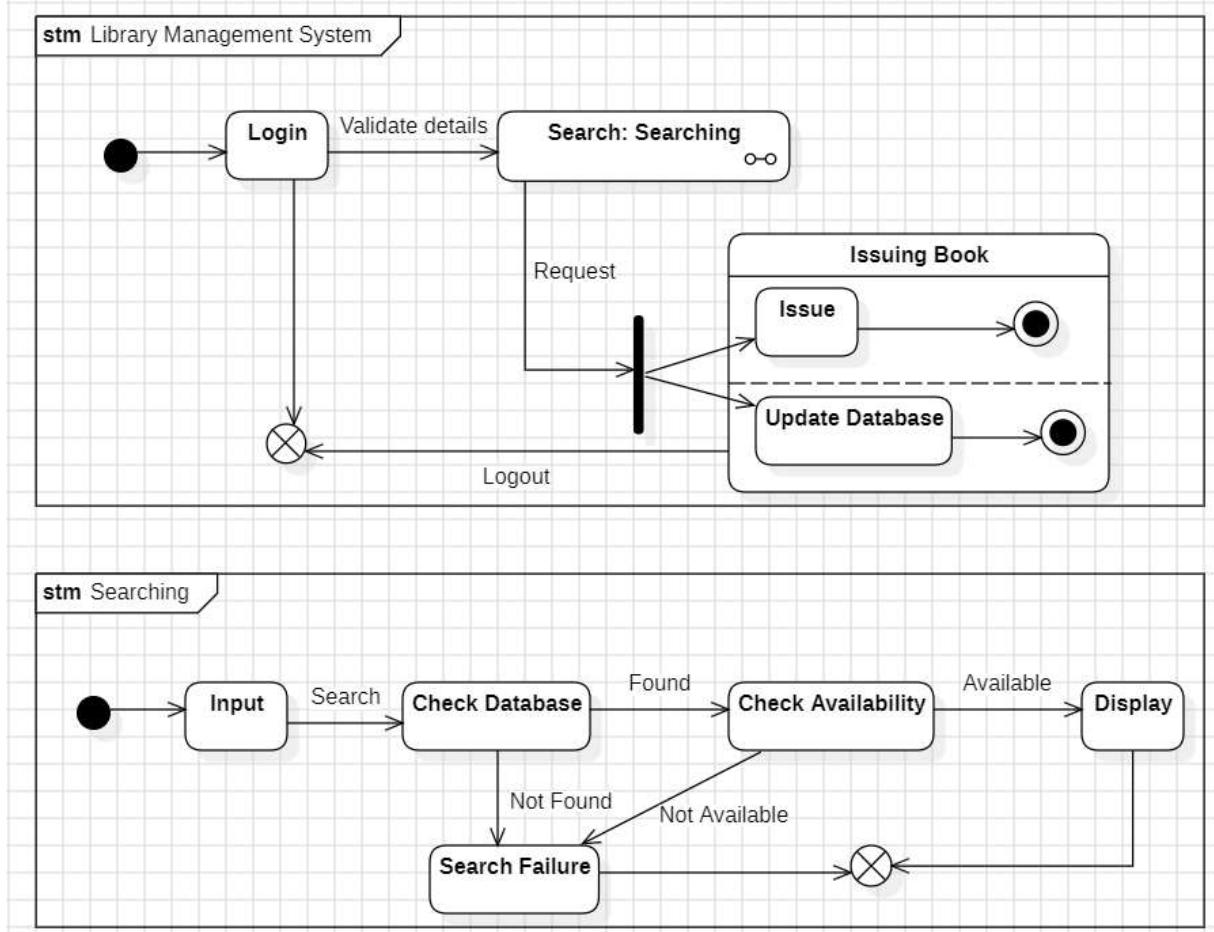


Fig 3.2 Library Management System - State Diagram

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

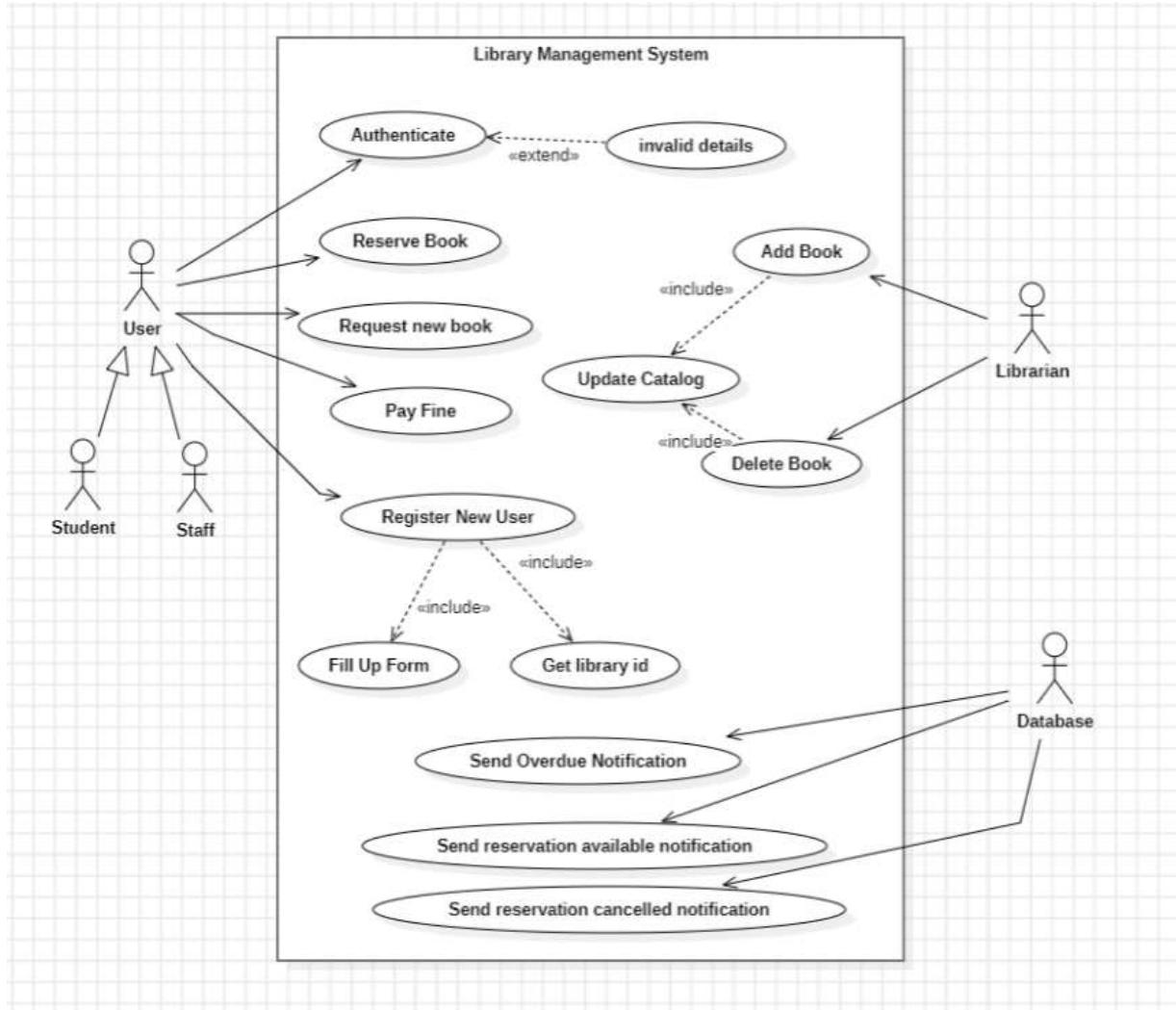


Fig 3.3 Library Management System - Use Case Diagram

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

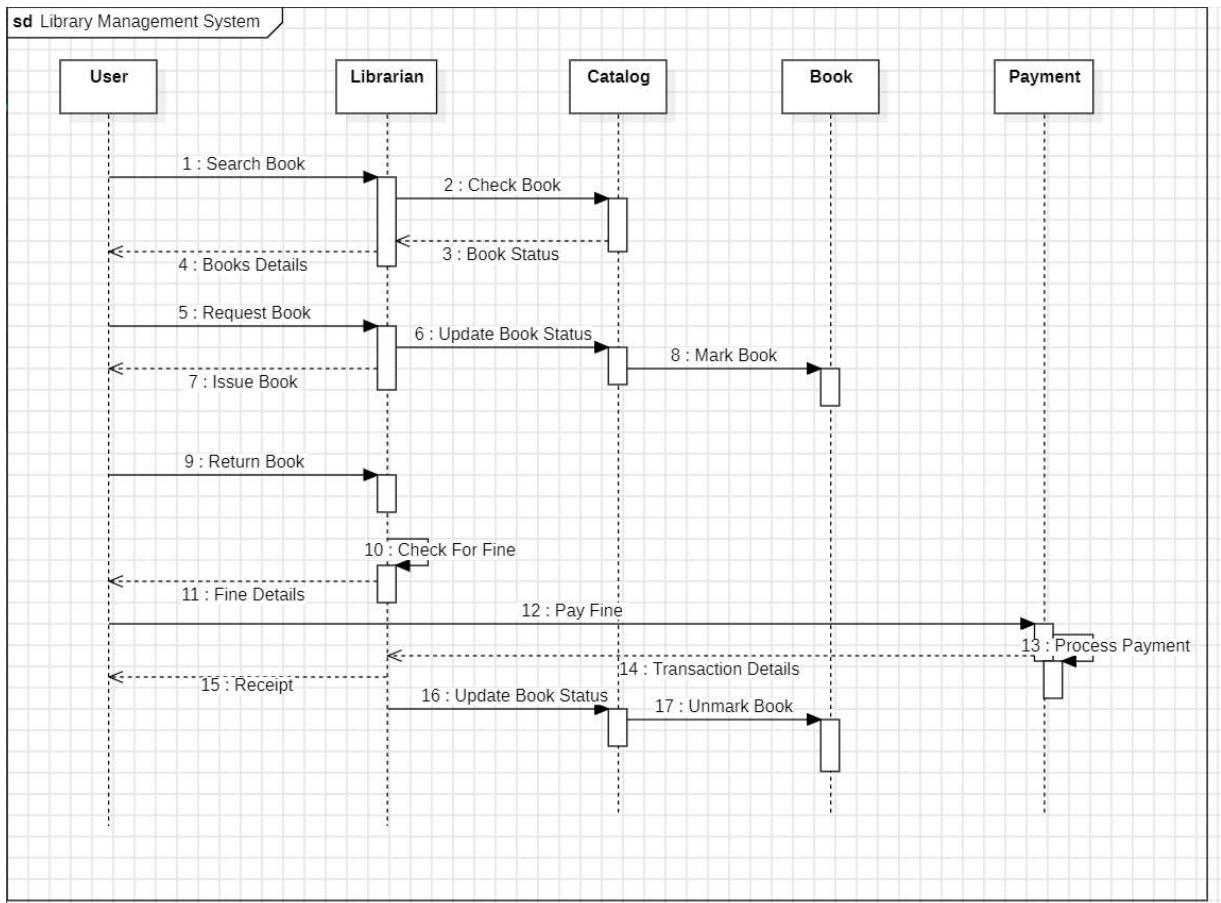


Fig 3.4 Library Management System - Sequence Diagram

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

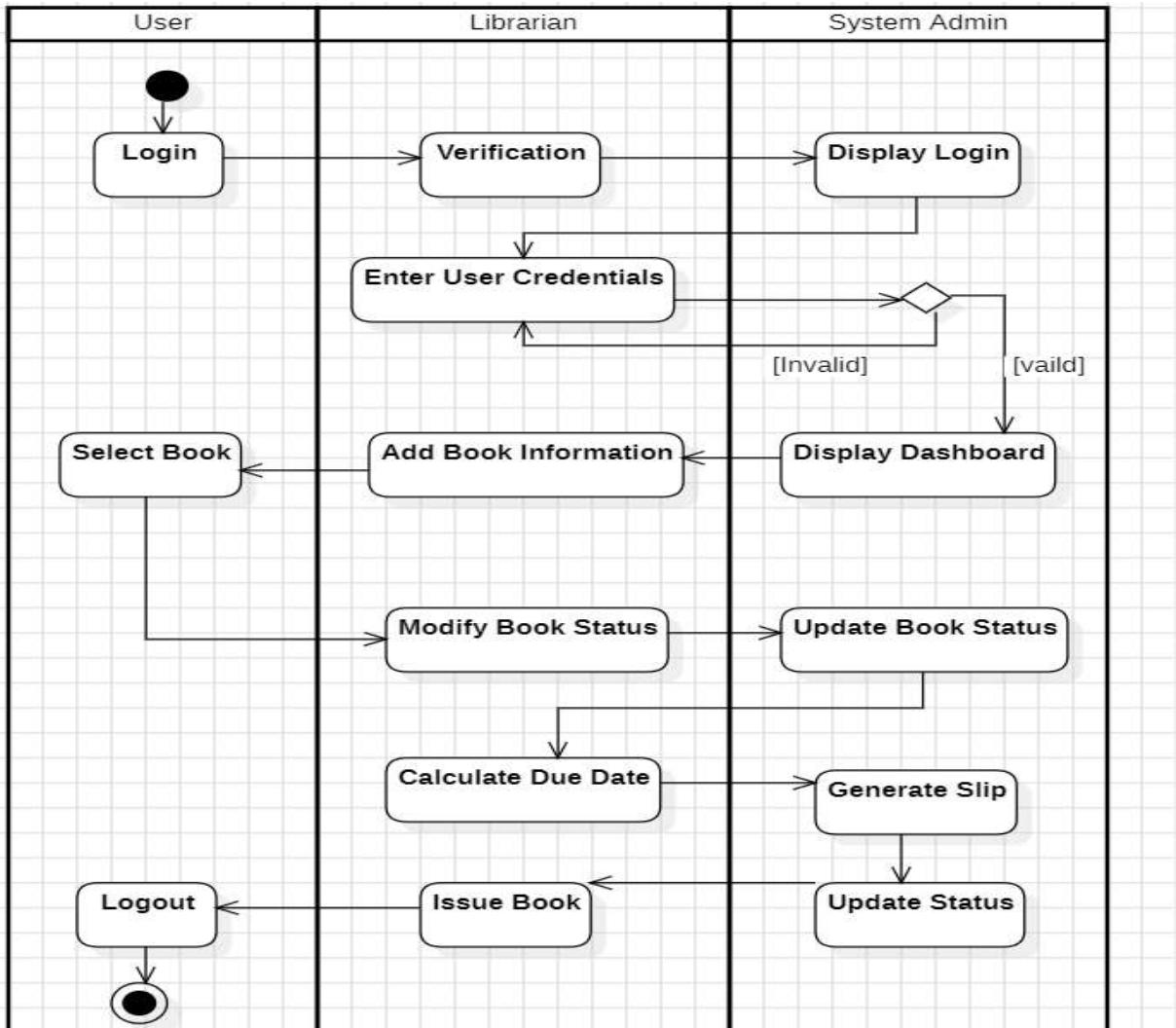


Fig 3.5 Library Management System - Activity Diagram

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. Finally, the system administrator updates the overall book status, and the user can log out. The diagram illustrates the interconnected roles and actions of users, librarians, and system administrators in the library's book borrowing process.

4. Stock Maintenance System

Software Requirement Specification

4) In	• In
ee	- It
- Tr	- Th
5. Des	- do
6. Perf	- Me
- Sta	- fin
- M	-
- E	-
7. No	- Sec
- See	- use
8. Per	- no
- Reli	- date
9. Pre	- sche
- soft	- Dep
- Busi	- Bud

w) Stock Maintenance System:

1. Introduction :

1.1 Purpose: The stock maintenance system ensures to capture the price & movements & also providing the real-time insights of the market.

1.2 Scope: The stock maintenance system will allow users to manage their stock portfolio, handling the buy & sell commands. It handles the inventory management, & records the movements of stocks.

1.3 Overview: The stock maintenance system records all the transactions of the warehouse. This document ensures all the requirements are specified & follow a standard procedure.

2. General description :

The stock management system will be integrated with sales system & existing procurement. It ensures the updation & management warehouse inventory, allowing the supplier to track real time stock-levels. It tracks the stores & will notify if the stock are low.

3. Functional Requirements :

- Allowing the user to add, update or remove products from the stock.
- Track inventory level, including incoming & outgoing items.
- Maintain product details, including price, descriptions & supplier information.
- Automatic notification system to notify when the stock levels are low.
- Generating the reports of stock levels, product management record transactions.

4) Interface requirements:

- Interface dashboard for warehouse managers to view stock levels & movements, managing product data & purchase.
- It should maintain the transaction history.
- Interface must be able to sync data between the system.

5. Design Constraints:

- The system must adhere to industry standards for data storage & security.
- Must integrate with existing procurement, sales & financial systems used by the business.

6. Performance Requirements:

- Stock movements should be updated within short period.
- Must able to handle large product catalog & manage multiple inventory/warehouse.
- Ensure all time support to support continuous warehouse operations.

7. Non-functional requirements:

- Security: Implementing data encryption for sensitive information such as pricing & supplier details.
- Usability: Ensuring an easy to navigate interface for non-technical users.
- Reliability: The system must operate offline and sync data once the internet connection is re-established.

8. Preliminary schedule & Budget:

Schedule: Requirement specifications & analysis - 4 weeks,
Software development - 8 weeks, testing - 2 weeks,
Deployment - 2 weeks.

Budget: 30 lac hrs.

Class Diagram

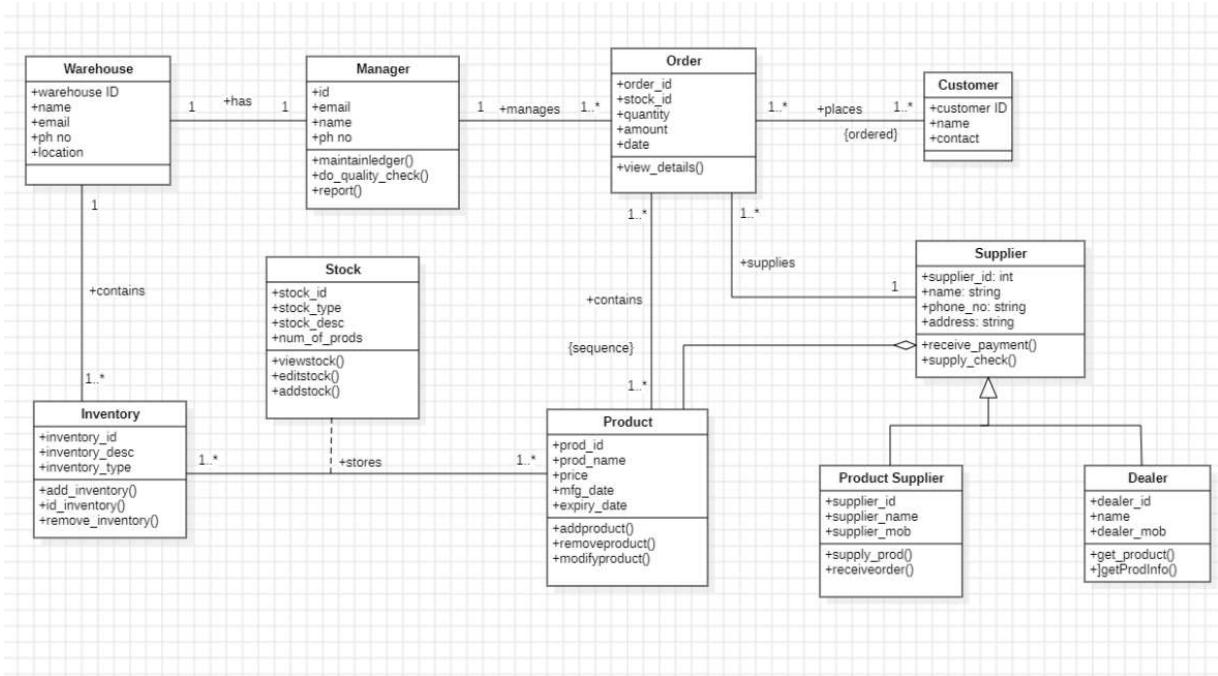


Fig 4.1 Stock Maintenance System - Class Diagram

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

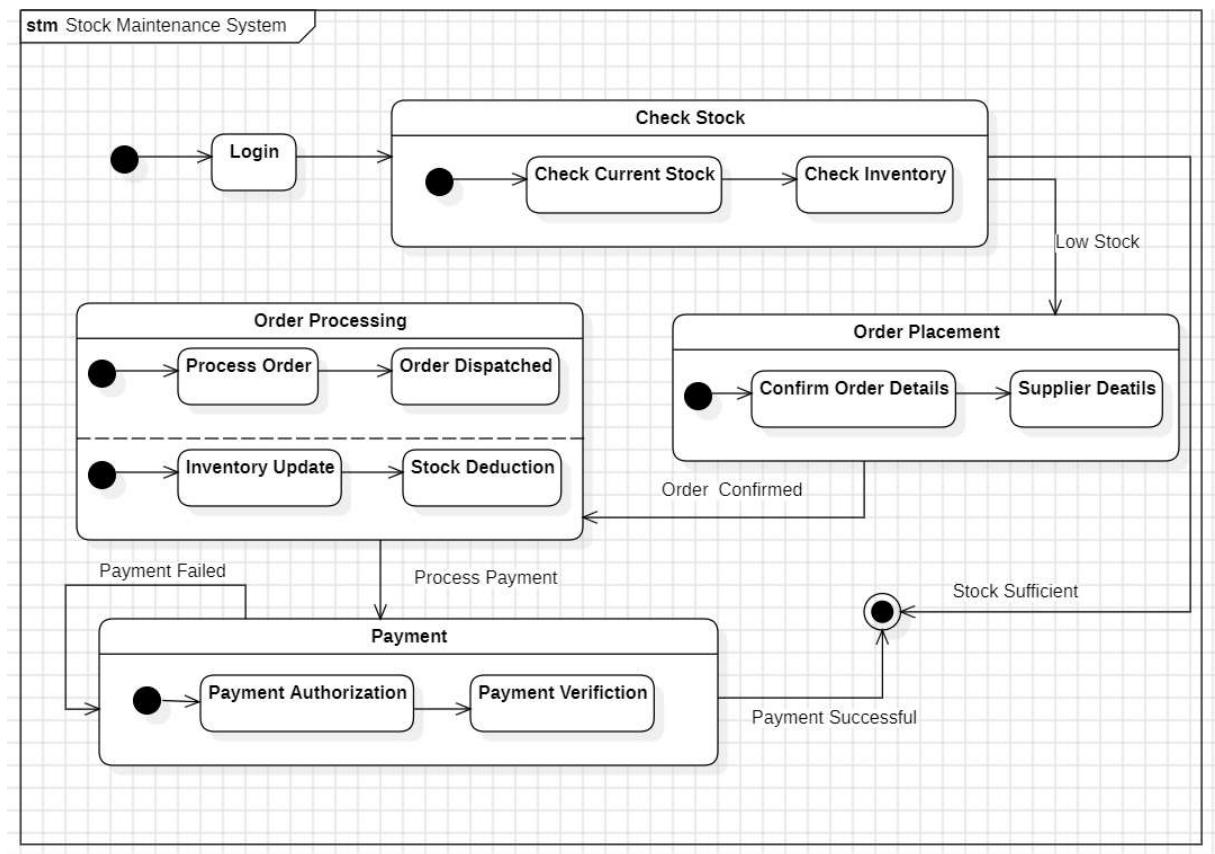


Fig 4.2 Stock Maintenance System - State Diagram

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

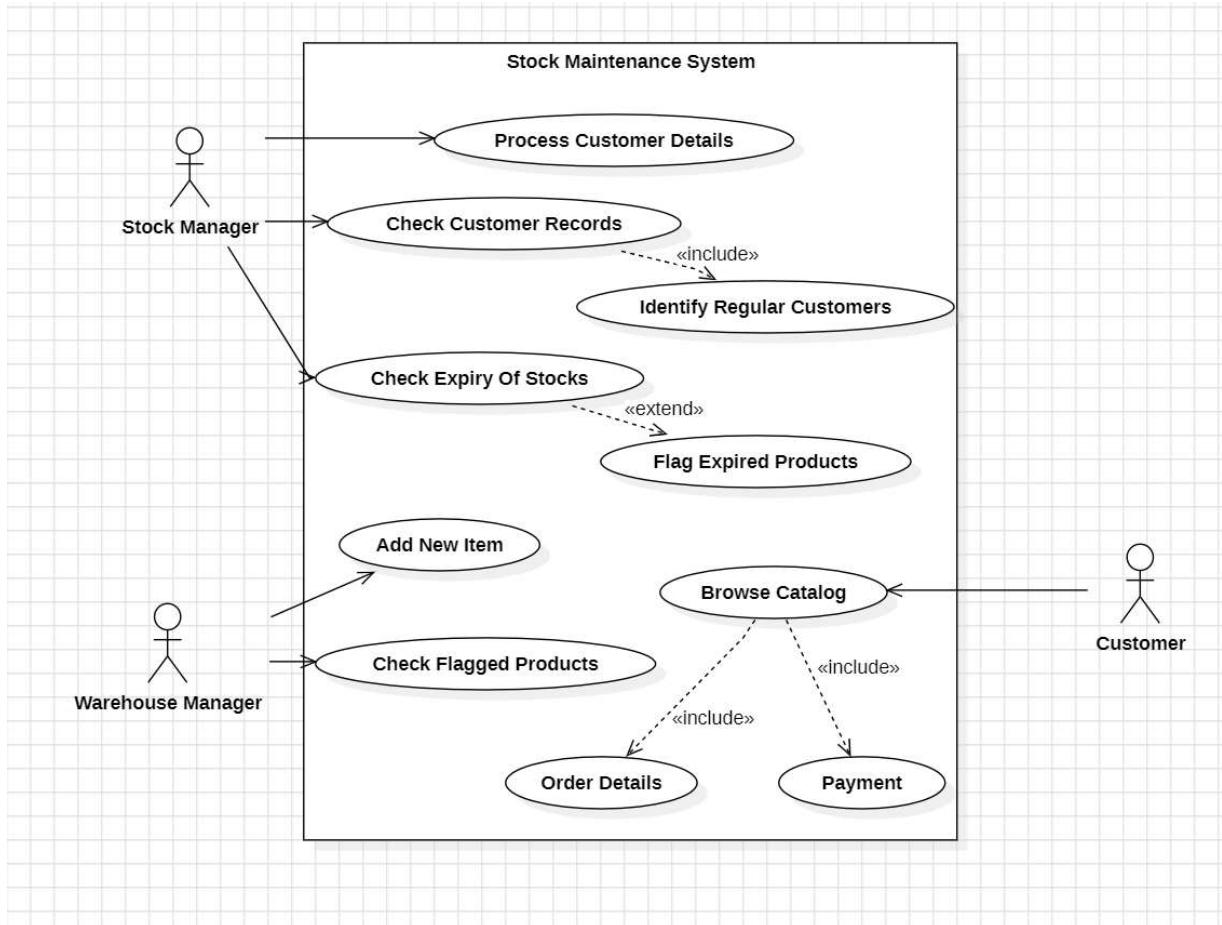


Fig 4.3 Stock Maintenance System - Use Case Diagram

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

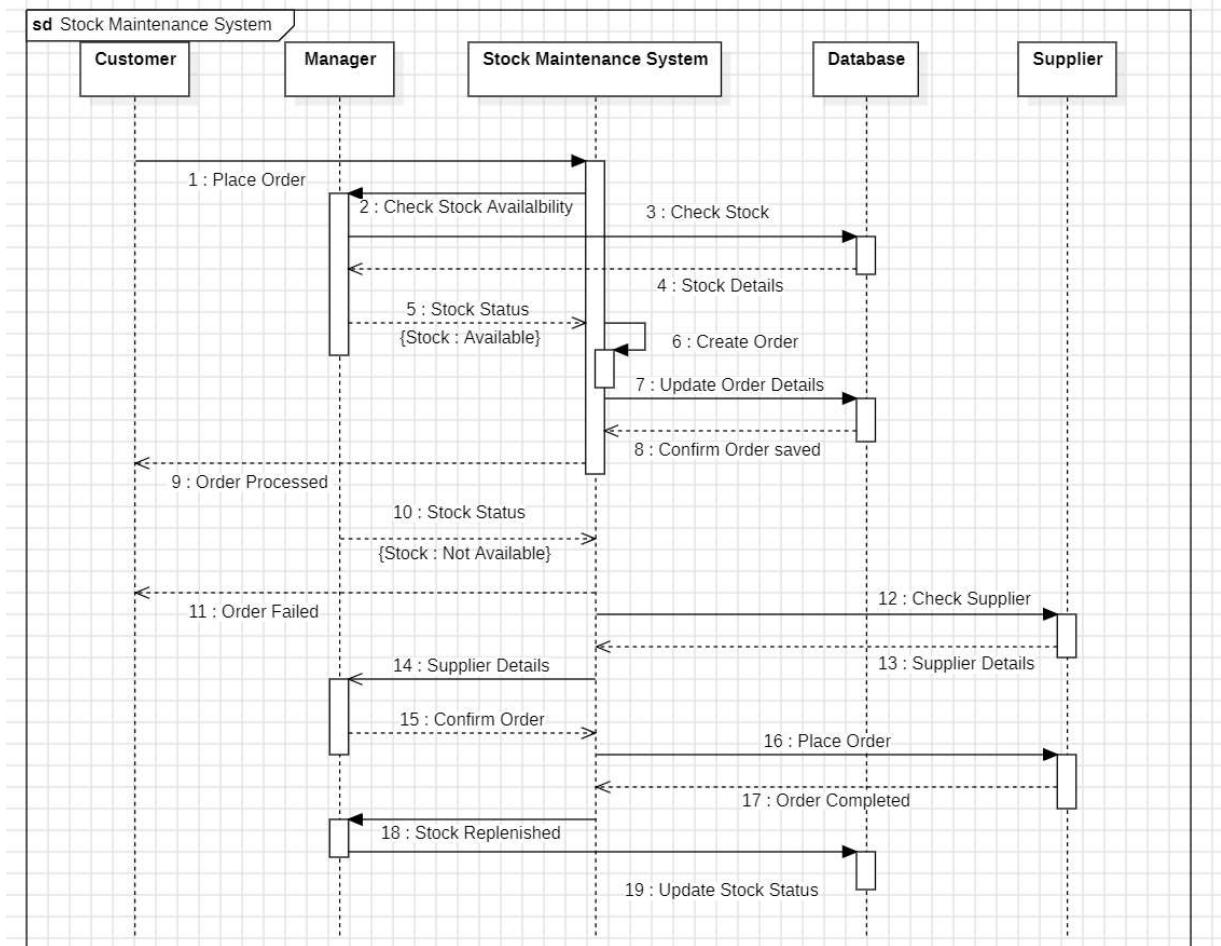


Fig 4.4 Stock Maintenance System - Sequence Diagram

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

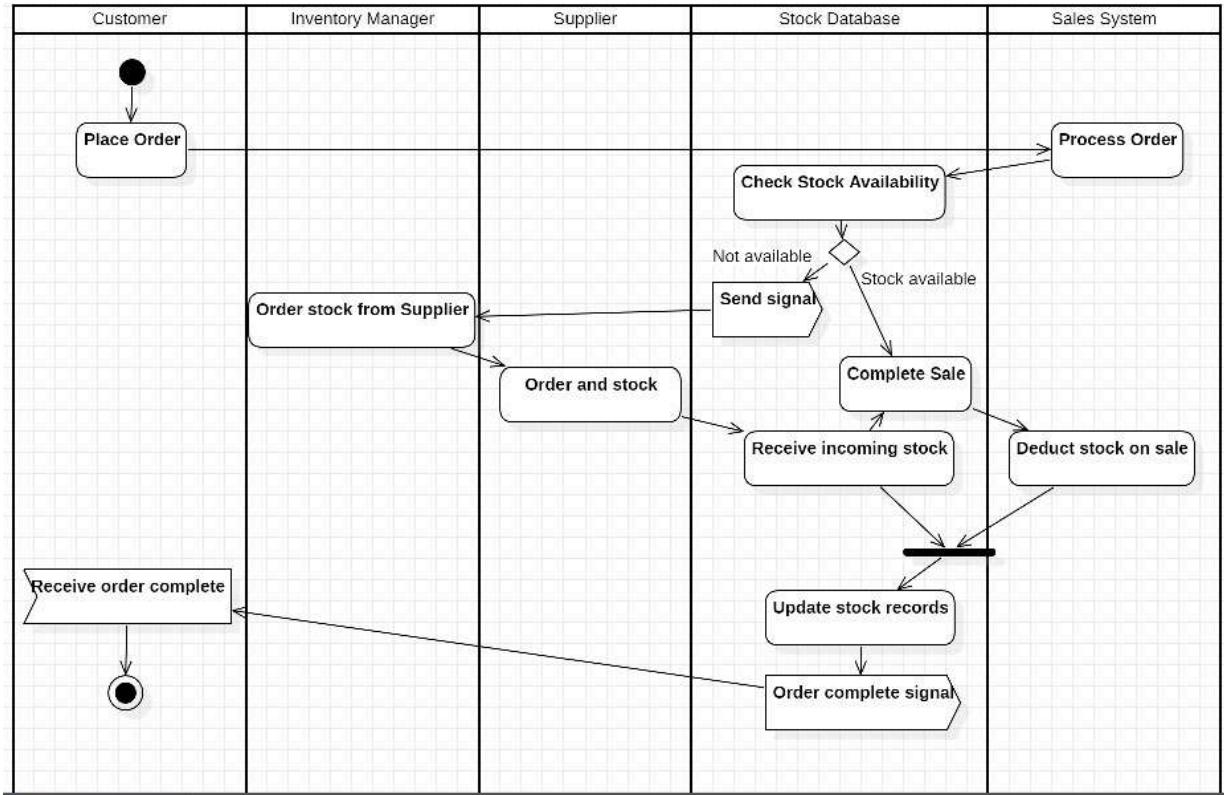


Fig 4.5 Stock Maintenance System - Activity Diagram

The activity diagram illustrates the stock maintenance system of a business. It outlines the flow of events starting with a customer placing an order. The inventory manager then checks stock availability. If sufficient stock exists, the order is processed, and stock levels are updated. However, if stock is insufficient, a request is sent to the supplier to replenish inventory. Upon receiving the order from the supplier, the stock database is updated, and the order is finally completed. The diagram employs swimlanes to clearly delineate the responsibilities of different roles involved in the system, such as the customer, inventory manager, supplier, and the database.

5. Passport Automation System

Software Requirement Specification

5) Passport Automation system:

1. Introduction:

1.1 Purpose: The passport automation system streamlines the process of passport application, verification, issuance & renewal.

1.2 Scope: This system allows users to apply for new passports, renew existing ones, schedule appointments & track the status of their applications.

1.3 Overview: The passport automation system enables the user to use all the services. It will facilitate the verification & issuance of passport. This document gives the summary of the functional, non-functional, performance & other requirements.

2 General Description: The passport automation system will be a web-based application accessible to both applicants & government officials. It will automate various stages of the passport process, including application submission, verification, fee payment, and document issuance. Integration with external government systems will ensure real-time verification & tracking.

3. Functional Requirements:

- Applicants must be able to submit personal details, upload required documents & pay application fee online & also to track the application status.
- Automating the document verification process & integrate with police & immigration databases for background checks.
- The applicants must be notified when the passport generating process is completed.

4. Interface Requirements:

- Must provide user friendly forms for filling out applications, uploading documents, scheduling appointments & tracking status.
- The government officials must be able to manage & verify applications, schedule appointments & issue passports.

5. Design Constraints:

- The system must obey the data protection laws & must ensure secure storage & handling the personal data.
- The sensitive data like biometric information must be encrypted & only the authorized person can be accessed.

6. Performance Requirements:

- Scalability: It must be capable of handling large numbers of applications.
- Application submission must be processed as early as possible.
- Must ensure high uptime for uninterrupted access to the system.

7. Non-functional Requirements:

- Security: Multi-factor authentication for applicants & officials.
- Usability: Easy to use interfaces for both users with minimal training required for new users.

8. Preliminary schedule & budget

- Schedule: Requirement Specification & analysis: 4 weeks, System Design: 6 weeks, Development: 12 weeks, Testing: 3 weeks, Deployment: 2 weeks.
- Budget: 50 lakhs.

Class Diagram

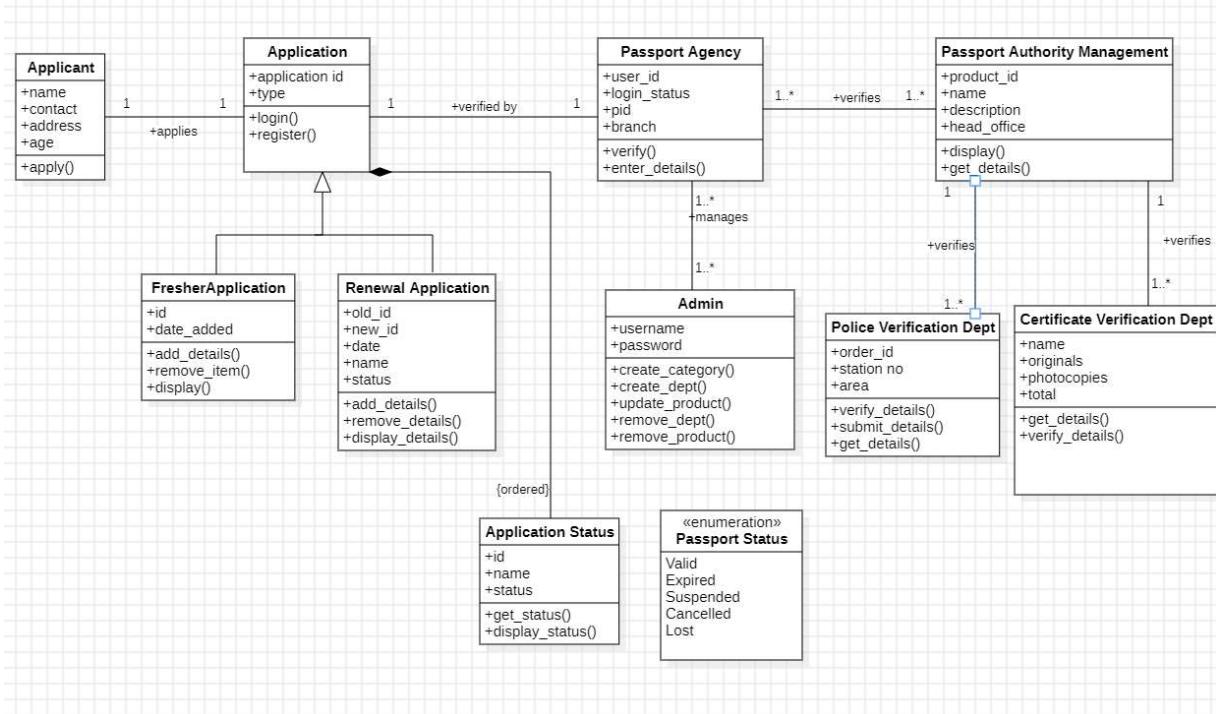


Fig 5.1 Passport Automation System - Class Diagram

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

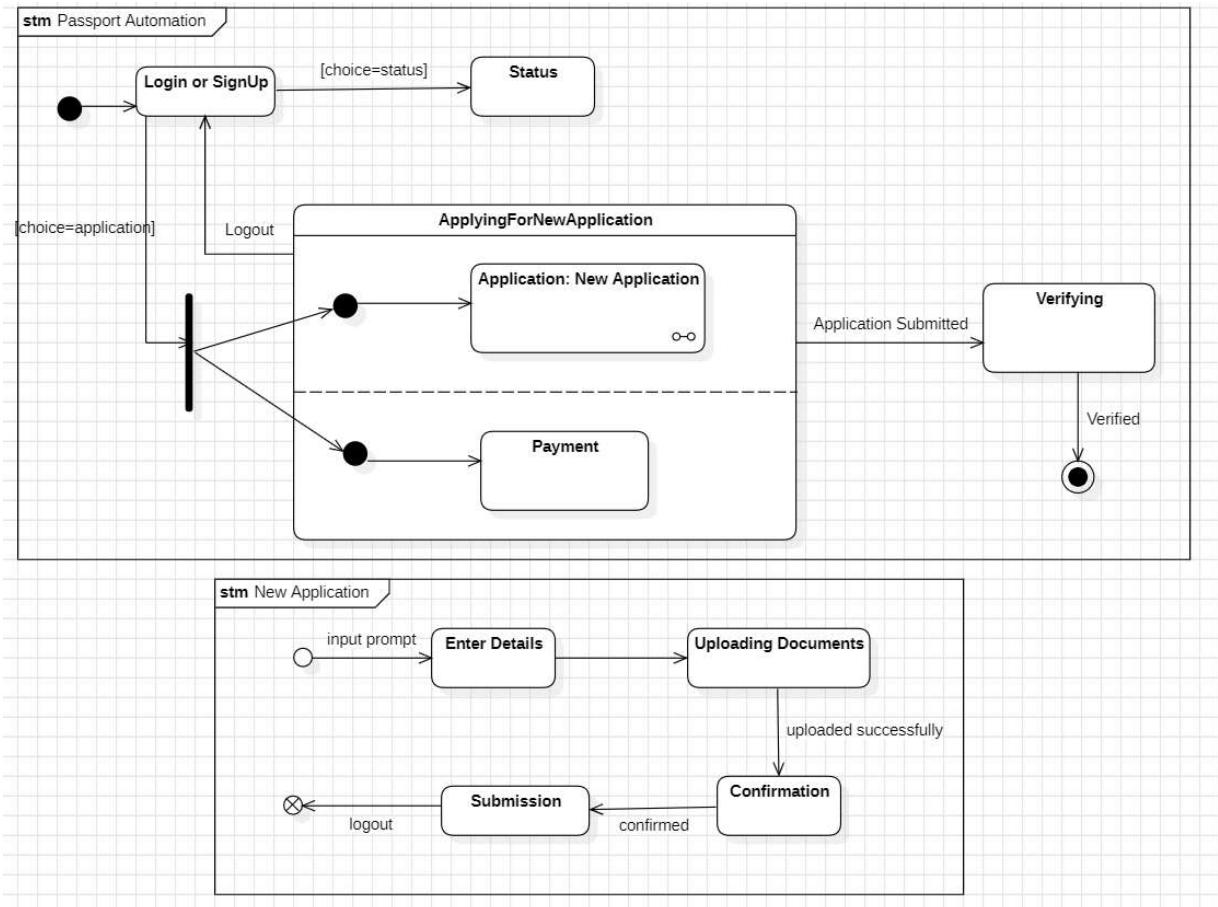


Fig 5.2 Passport Automation System - State Diagram

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 "Applying For New Application" 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.

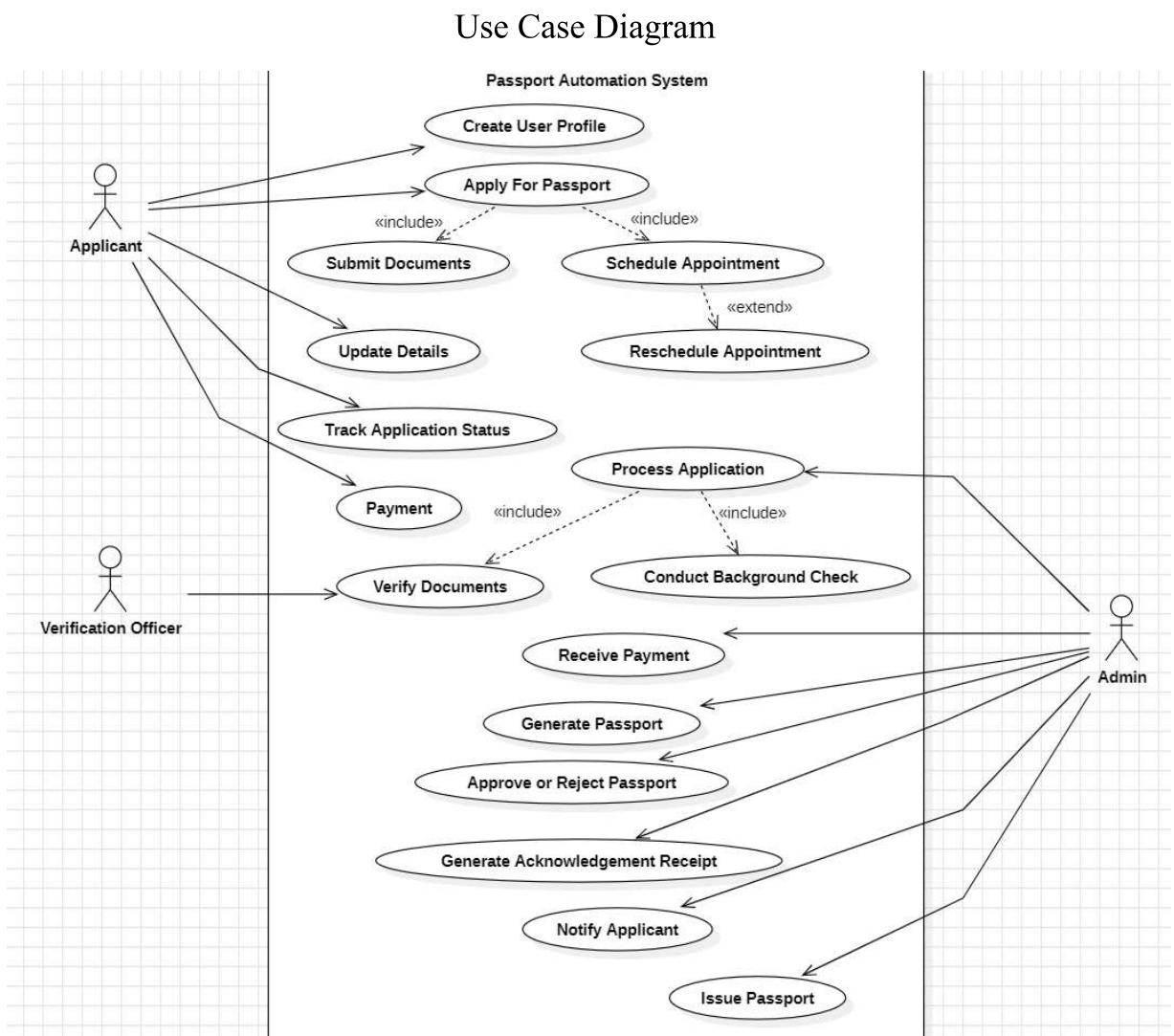


Fig 5.3 Passport Automation System - Use Case Diagram

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

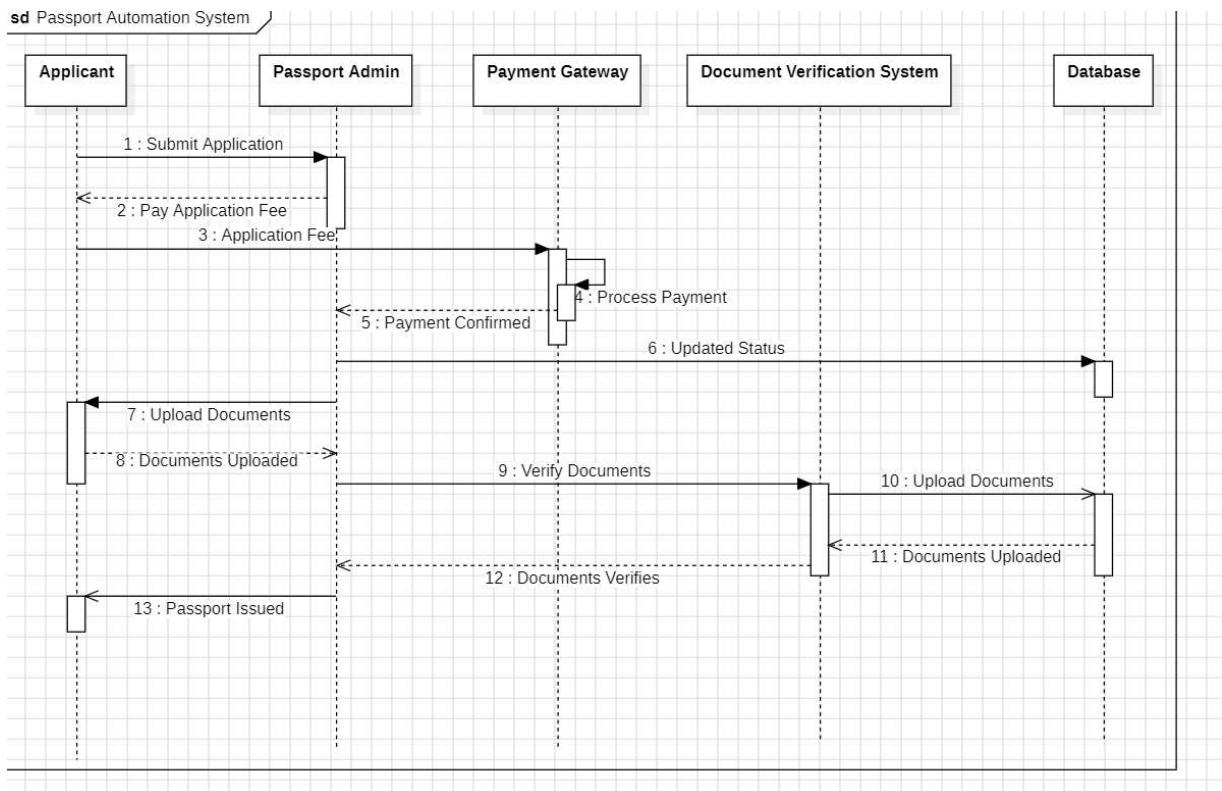


Fig 5.4 Passport Automation System - Sequence Diagram

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

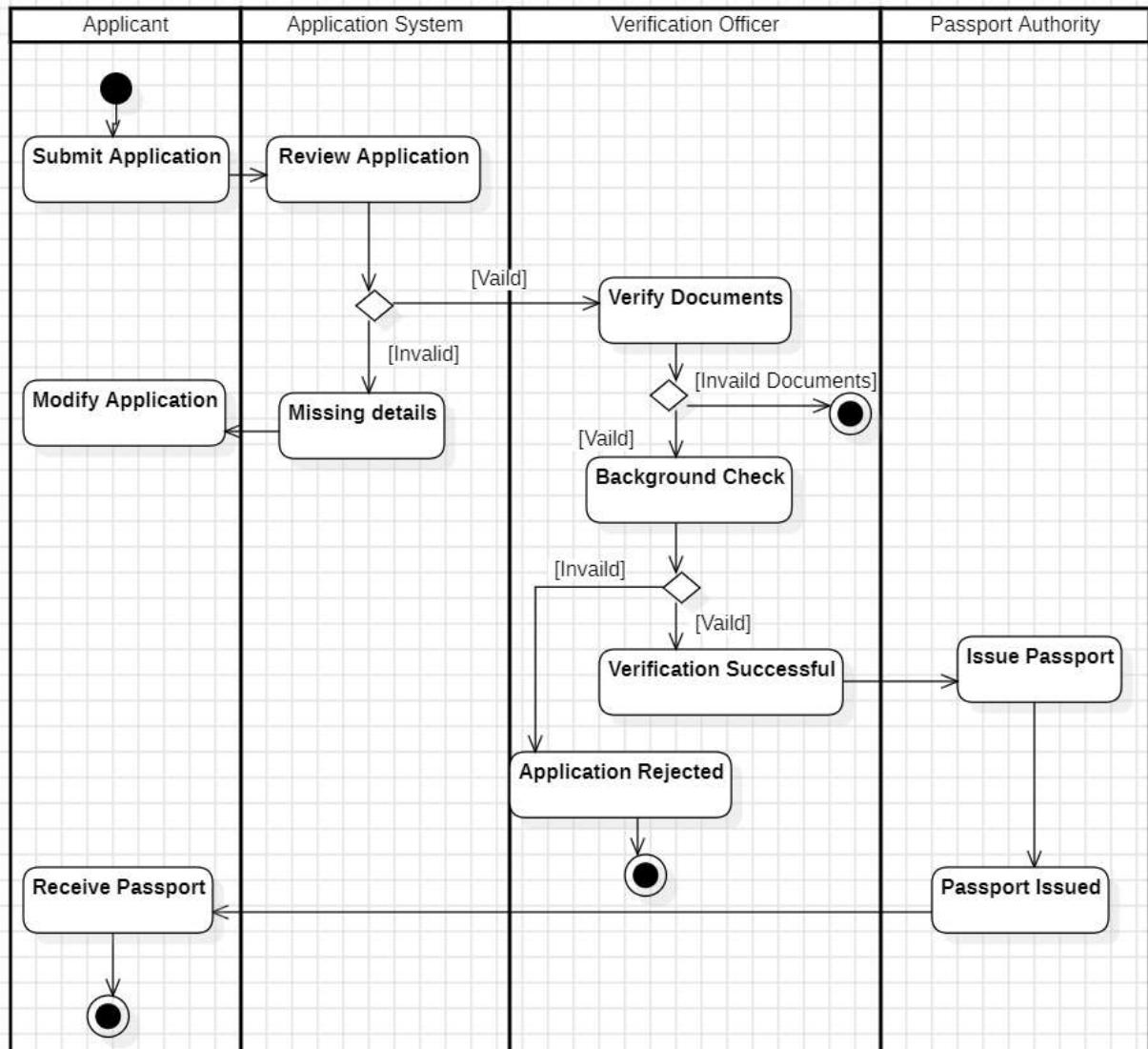


Fig 5.5 Passport Automation System - Activity Diagram

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.