

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:

SAGAR I BANGARI

1BM22CS231

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 **SAGAR I BANGARI**(1BM22CS231) 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

Hotel Management System
SRS (Software requirement specification)

1) Introduction:-

1.1) Purpose of this document:- In a typical hotel ~~there is~~ there is a lot of work going on and we need a generalized system that collects all the information and manages them so that all the works are structured and efficient.

1.2) scope of this document:- scope of this document is to develop a software application that manages or controls the activity of hotel.

1.3) overview:- overall we have to develop a software application for the hotel manager to manage hotel efficiently and quickly with ease.

2) General description:-

This new software application provides hotel a easier life to hotel manager as it can be used for work allotment, payment of ~~or~~ to employees, advertisement for customers etc.

3) Functional requirements:-

- the system should be able to collect all information of employees/waiters and should be able ping a message to any of the waiter.
- It should be able to calculate daily customers cost, profit etc.
- It should be able to book or cancel a room.
- It should provide billing invoices, as soon as room is booked. The bill is generated.

4) Interface requirements:-

The application should have intuitive interface which is easy to navigate and access data quickly.

5) Performance requirements:-

The accuracy of calculations should be high, application should be able to run in heavy load, that is it should not be crashed.

6) Design constraints:-

At The application algorithm should follow standard procedure designed by requirements management.

7) Non-functional requirements:-

- Security should be provided to user data and only authorized entities can access data
- The interface should be simple
- The system should be reliable

8) Primary schedule and budget:-

The application development time depends on how big the hotel is to accommodate all information so for medium sized hotels it should take approximately 3-5 months to build software application and budget should be around ₹1,00,000 to ₹5,00,000

Class Diagram

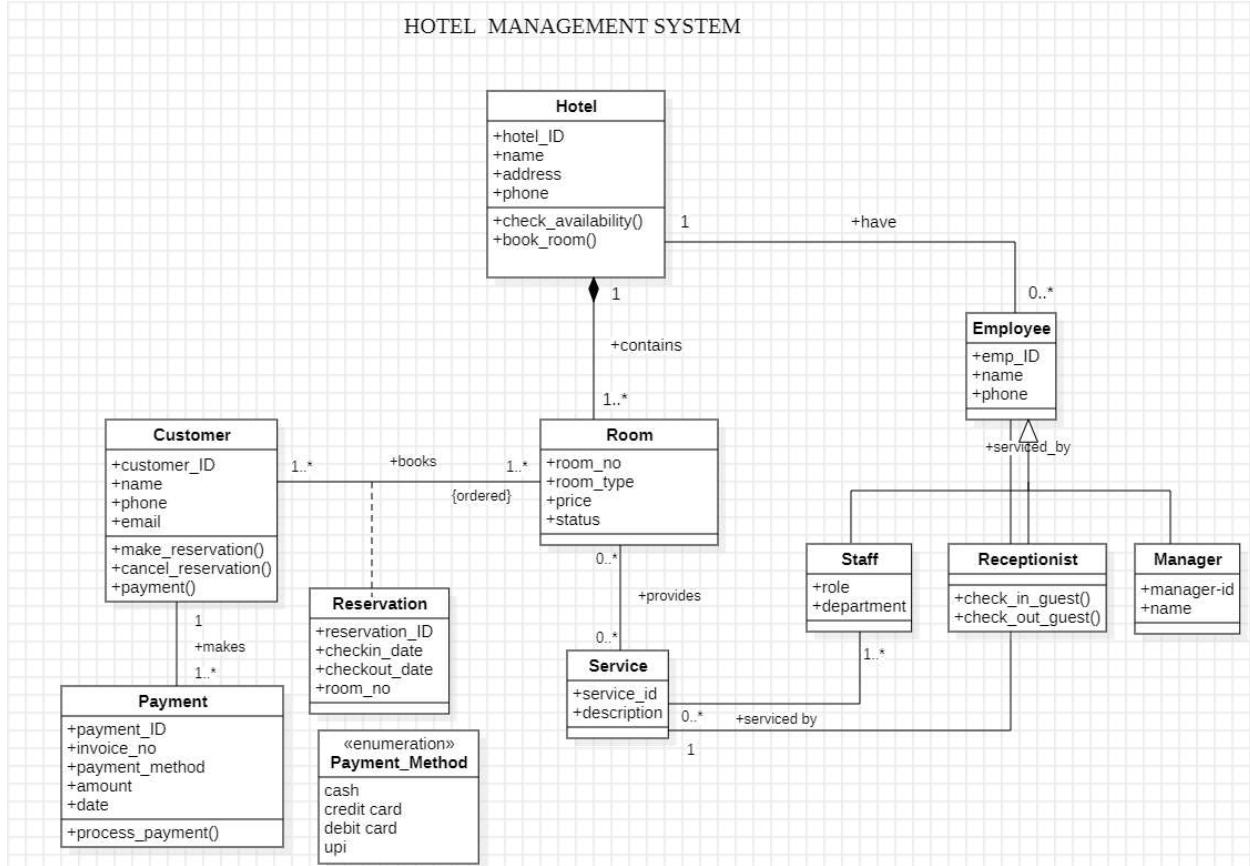


Fig1.1 Hotel Management System - Class Diagram

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

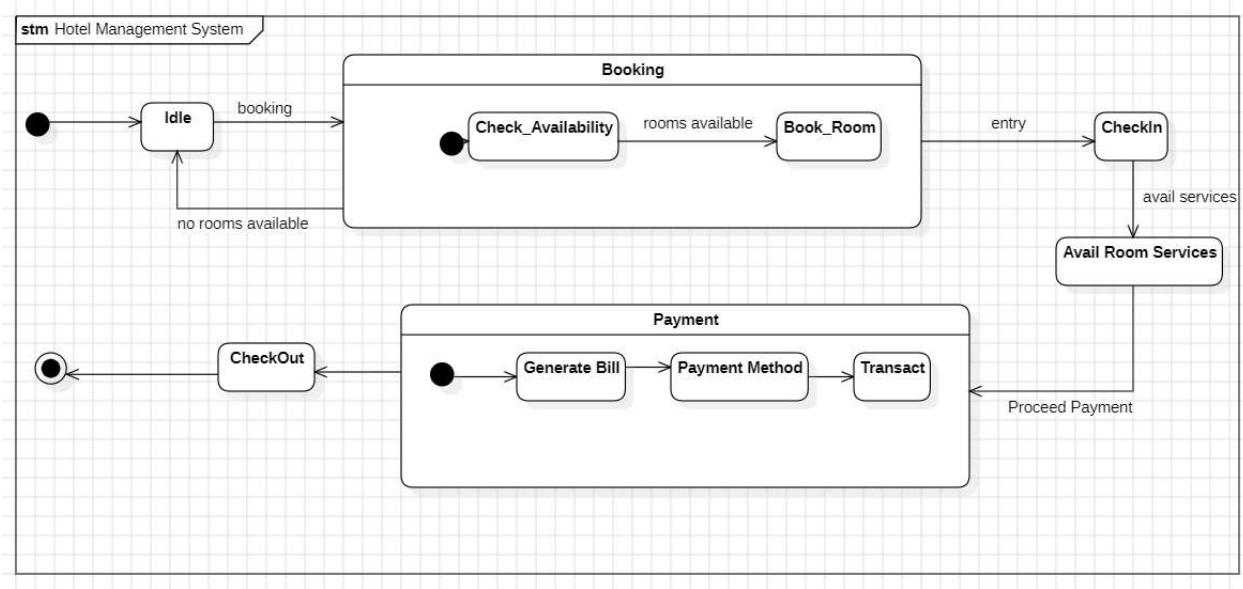


Fig1.2 Hotel Management System - State Diagram

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

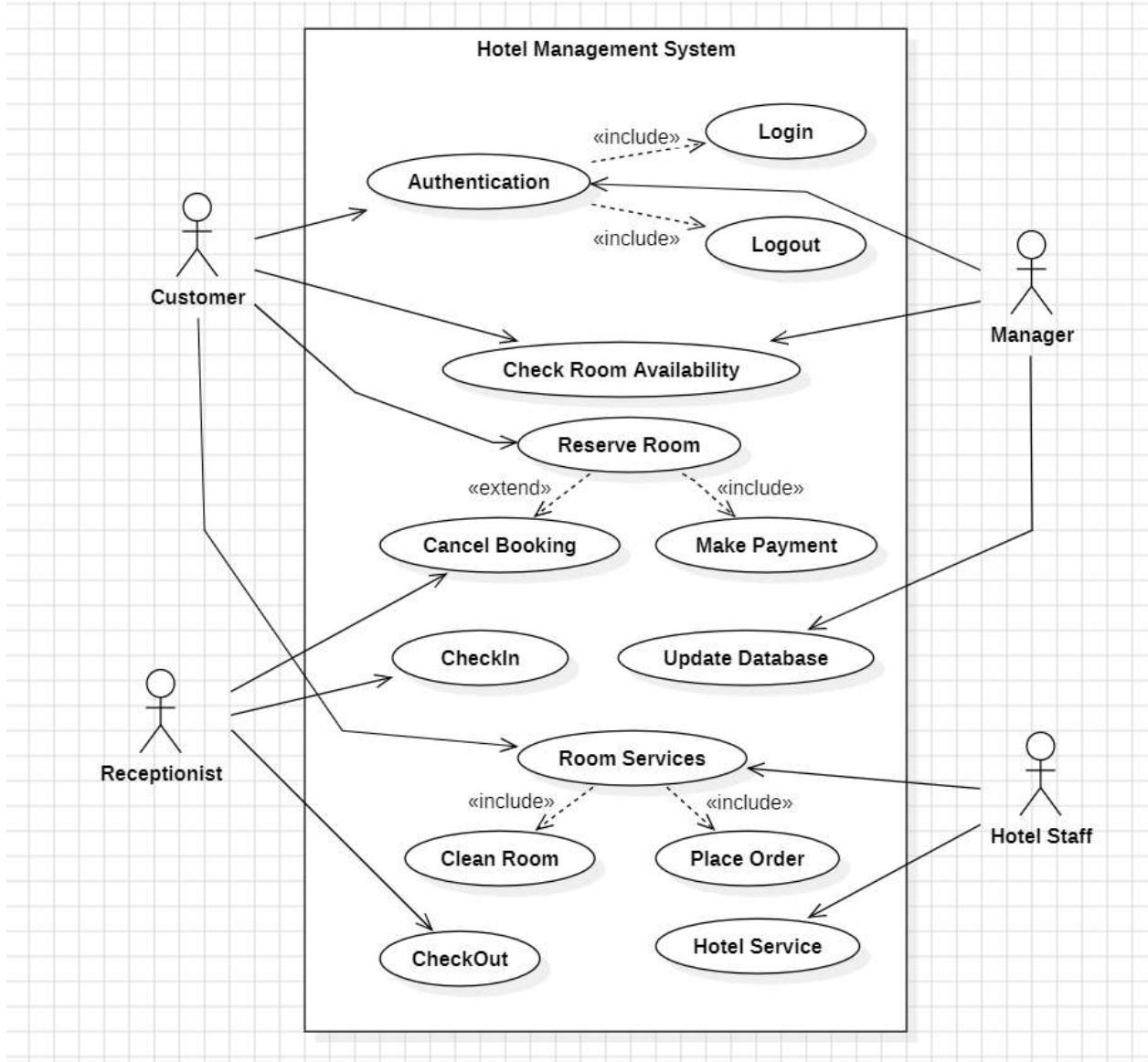


Fig1.3 Hotel Management System - Use Case Diagram

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

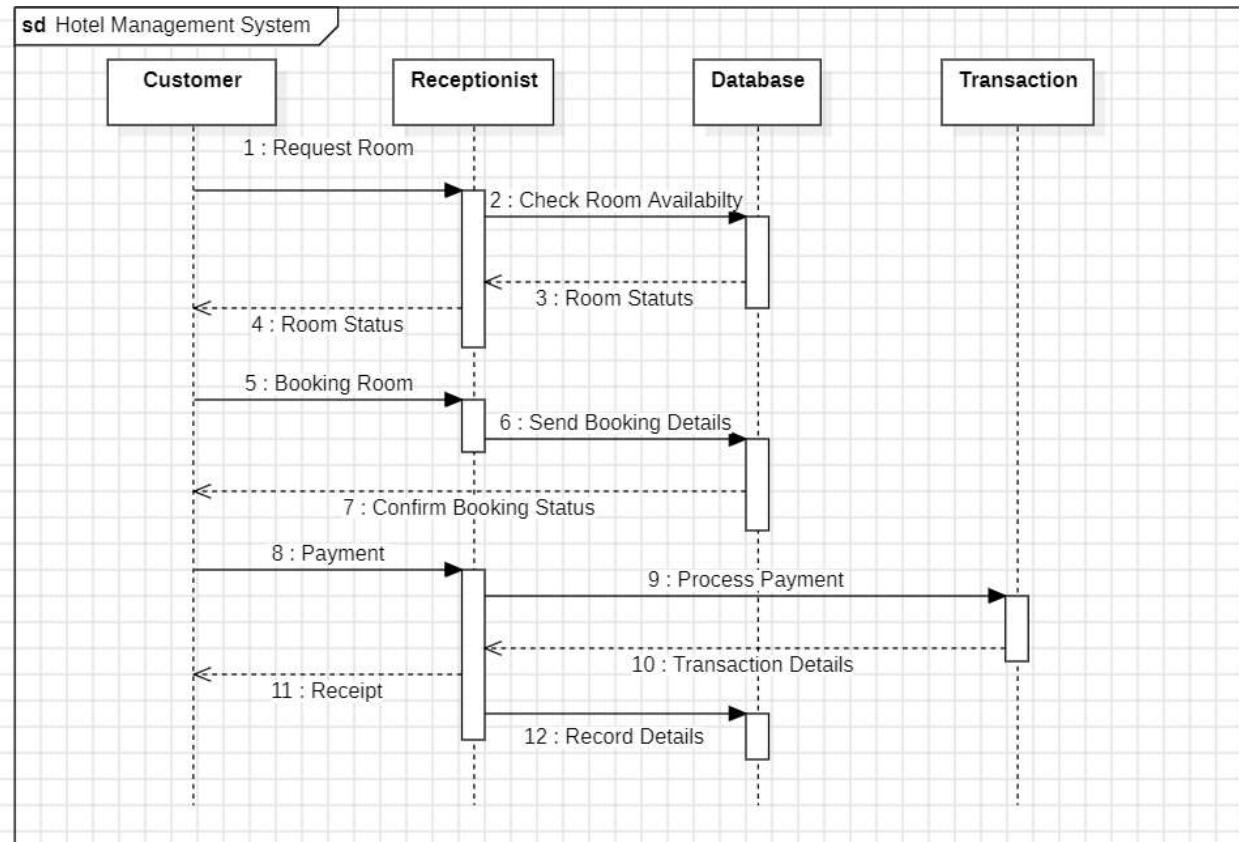


Fig1.4 Hotel Management System - Sequence Diagram

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

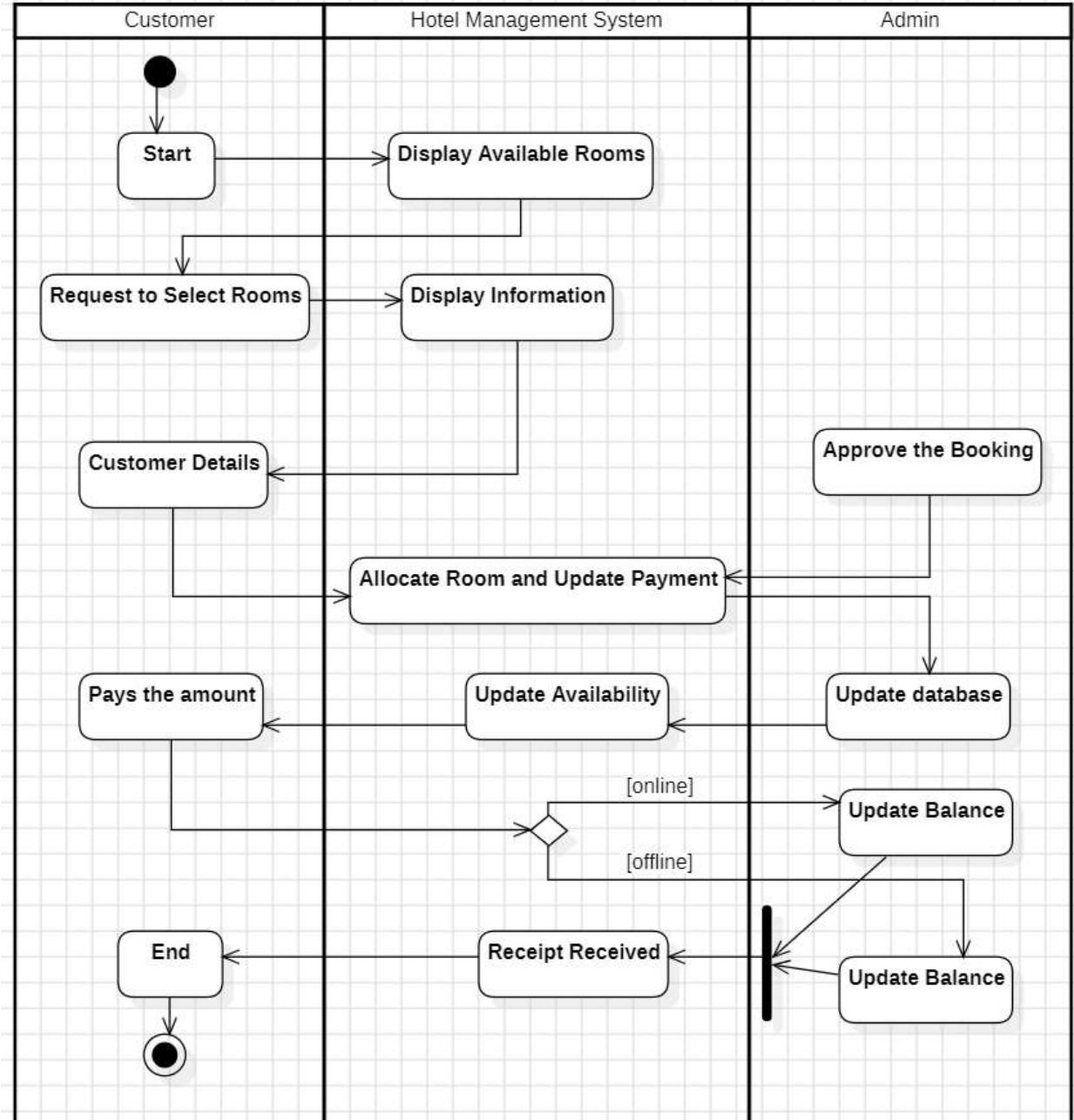


Fig1.5 Hotel Management System - Activity Diagram

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

Credit card system

SRS document

1) Introduction:-

1.1: Purpose of this document:- Aim of this document is to get familiar with software requirements specification of credit card system and its services.

1.2: Scope of this document :- The main objective of this document is it will provide detailed information about credit card system and its services like easy transaction, service charges with respect to time period. The development cost of the card is economical to the users.

1.3: Overview:- Credit card system is a financial tool which allows user to borrow the funds up to predetermined limit to purchase or to pay bills. It also provides various offers and rewards to users.

2) General description:- The objective of this credit card system is to make secure and easy transactions. The user characteristics are user must be above 18 years old to use credit card and user must have a bank account and pan card. Some of the features of credit card system is it provides credit amount when needed and we can pay it later to the system. Some of the benefits are that it provides offers, discounts.

3) Functional requirements:-

- * Notifications:- All the transactions are notified to the user by mail-id or mobile number.
- * User services:- login/logout - user can login and logout in bank account.
- * Real-time management:- When user does any transaction the user profile and his account history will be updated instantly.
- * customer support and help centre:- Users can clear their doubts and any queries by customer support.

4) Interface requirements:-

- User interface / online portal :- Through which a user can login or logout of his profile.
- Able to perform all necessary banking transactions in the portal.
- Able to manage his profile.

5) Design constraints:-

→ Data storage - Use of secure data storage solutions that ensure data redundancy and back up of data.

→ Regulatory compliance - Must adhere to local and international financial regulations.

6) Performance requirements:-

- Response time → The system must be able to authorize transactions within a maximum of 2 seconds from time of submission.
- System availability → System should be available continuously.

7) Non-functional requirements:-

- * Portability \Rightarrow A credit card can be used anywhere in the country or globally.
- * security \Rightarrow OTP based transaction authorization provides secure transaction processing (CVV). User authorization through profile password.
- * Reliability \Rightarrow A credit card can be used any number of times within 5 years from the date of issue of card.
- * Performance \Rightarrow The system should process transaction within 2 seconds even under peak loads to ensure a seamless user experience.

8) Primary schedule and budget:-

The budget of building this system depends on number of users. and we have to physically create card and provide it to the user. This may take up to 5 - 10 days.

divide

Class Diagram

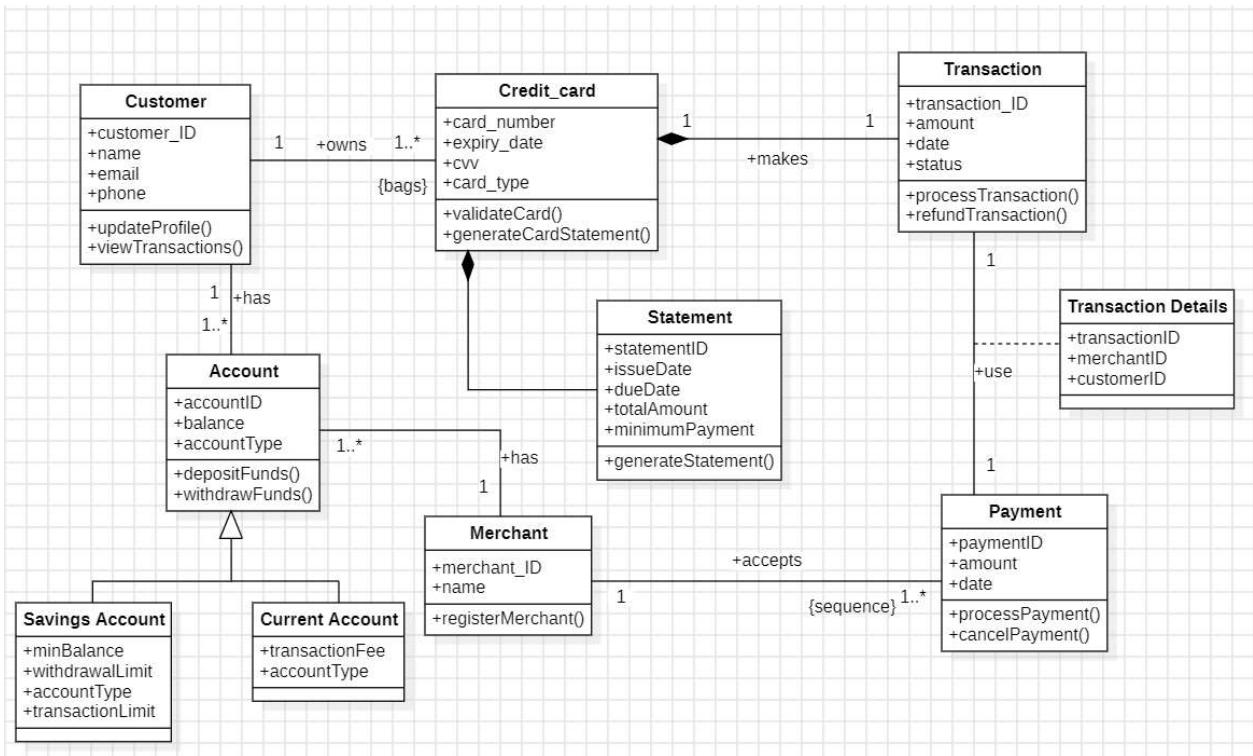


Fig 2.1 Credit Card Processing System - Class Diagram

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

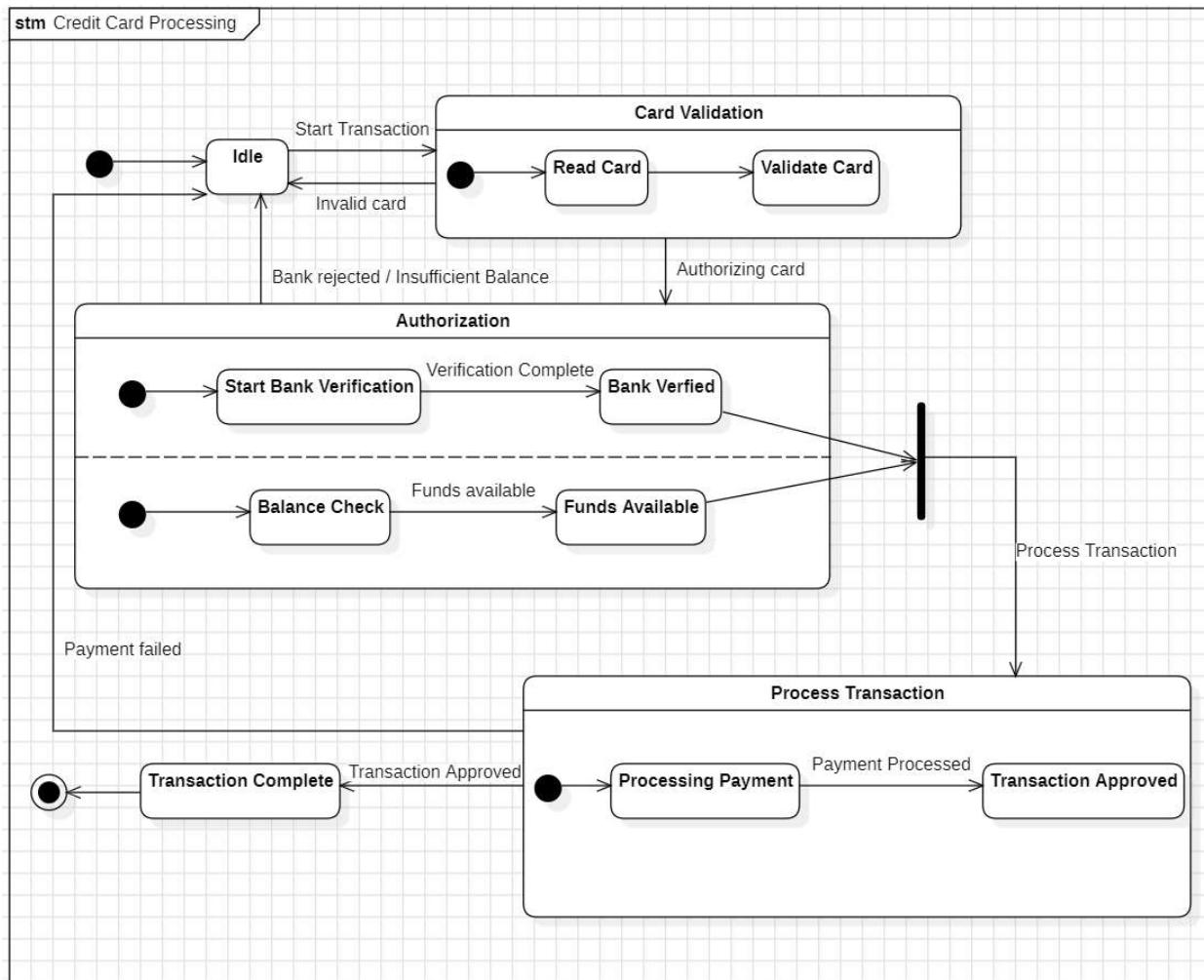


Fig 2.1 Credit Card Processing System - State Diagram

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

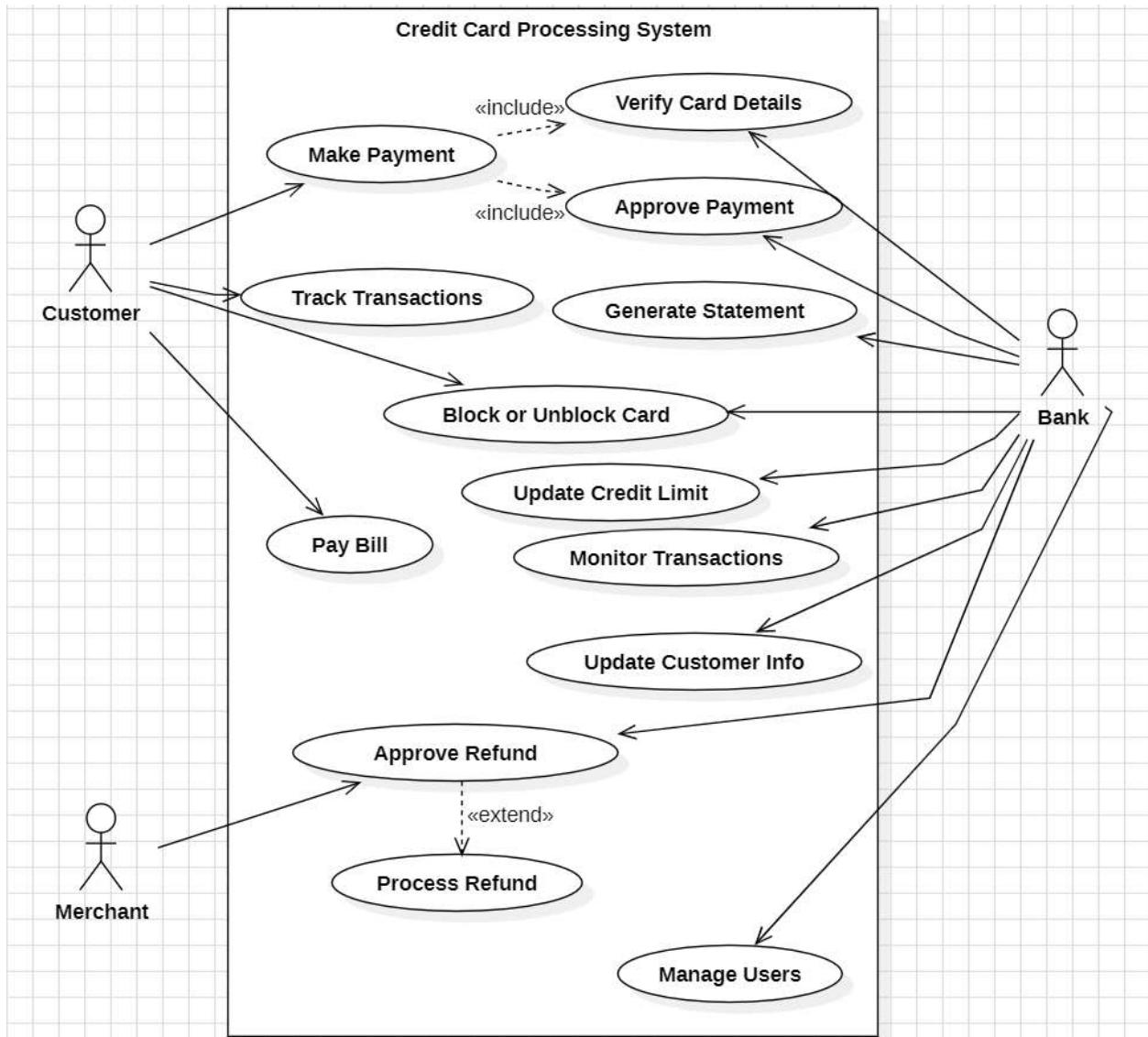


Fig 2.1 Credit Card Processing System - Use Case Diagram

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

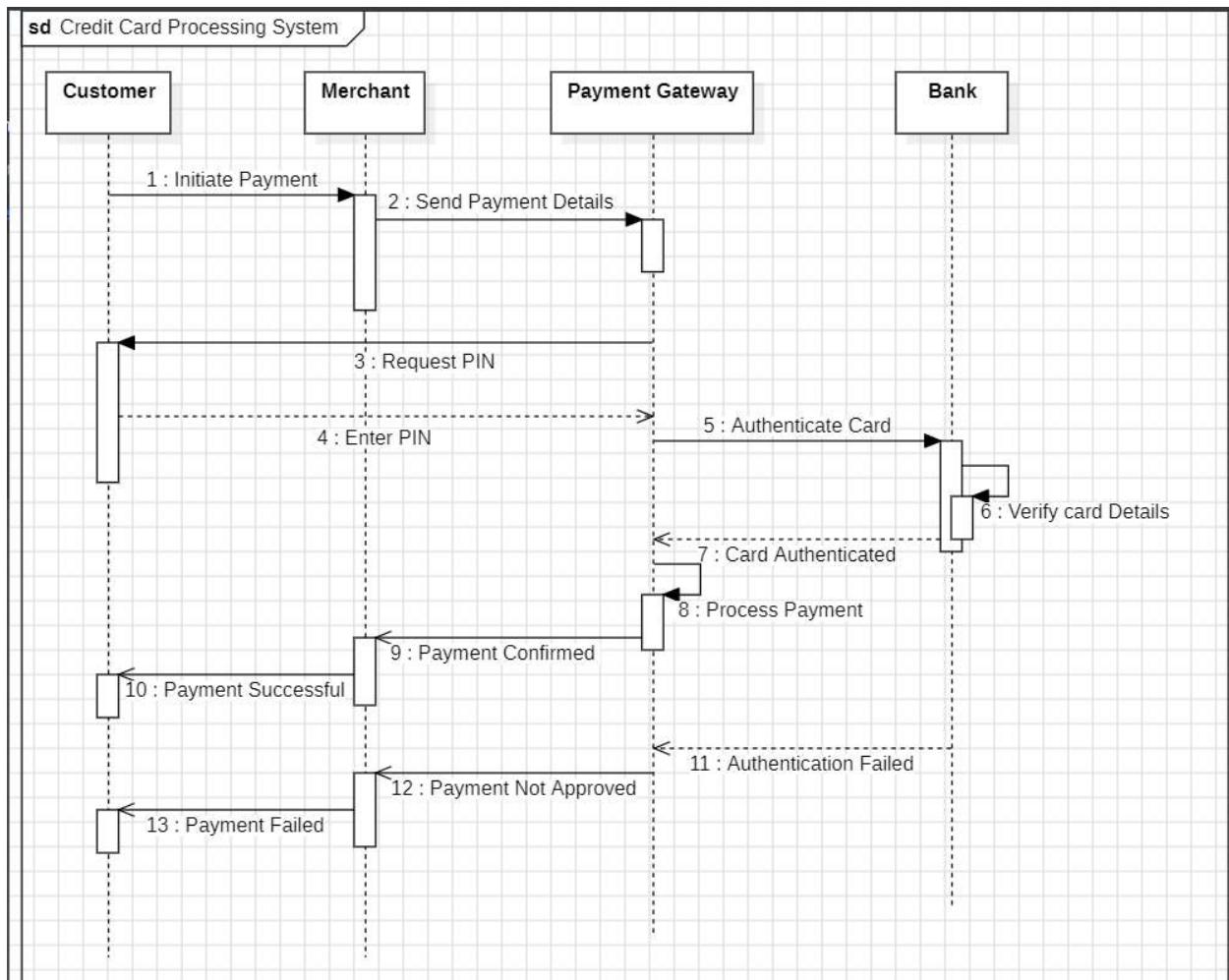


Fig 2.1 Credit Card Processing System - Sequence Diagram

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

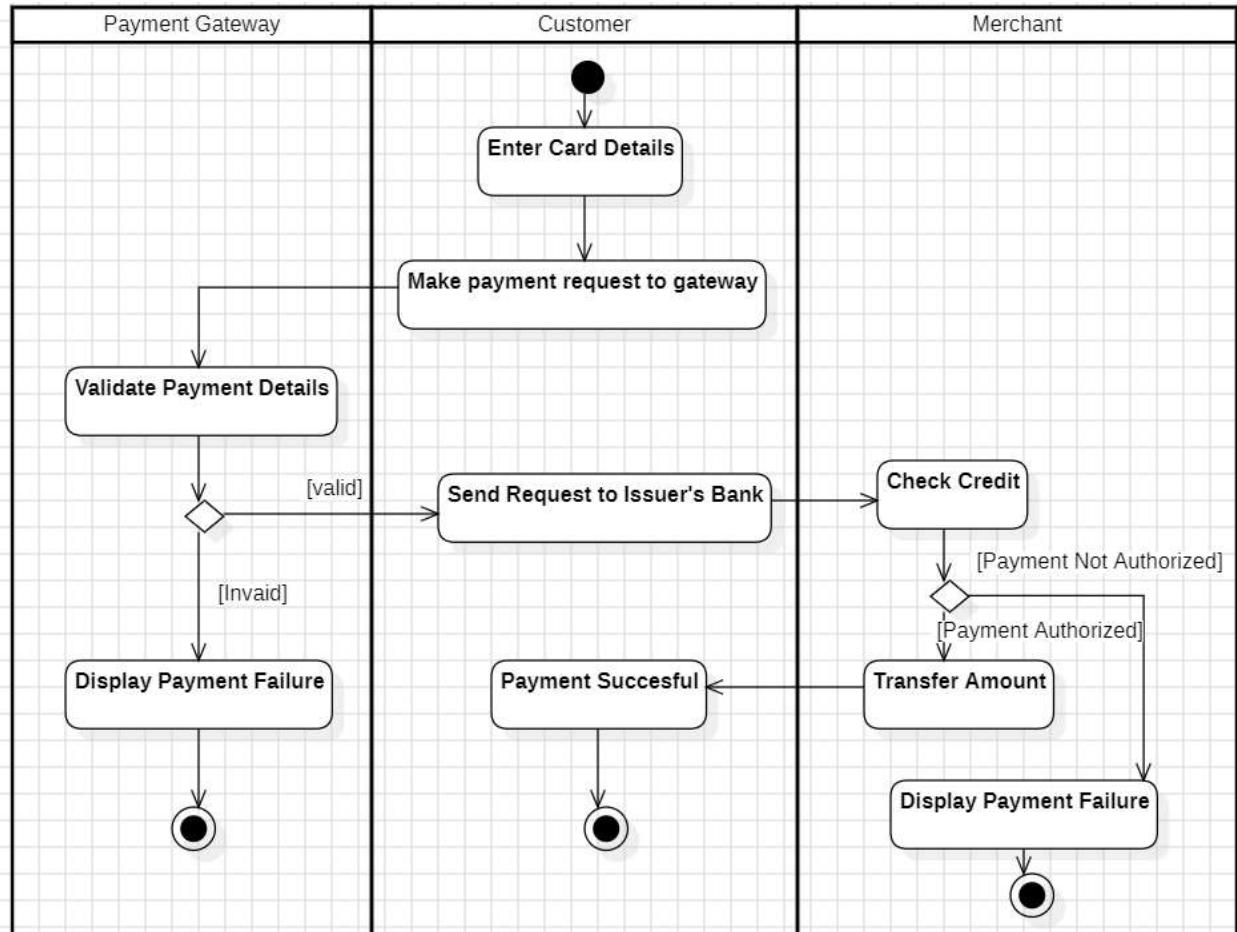


Fig 2.5 Credit Card Processing System - Activity Diagram

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

Library Management System

SRT document:

1) Introduction:

- 1.1. Purpose of this document: The purpose of this document is to define the requirements for a library management system that helps to automate and manage library operations like book borrowing, cataloging, etc.
- 1.2. Scope of this document: This system is designed to manage a library's book, member records and borrowing operations. It will provide functionalities for library staff members and administrators to efficiently manage and track resources.
- 1.3. Overview of the document: The document contains detailed descriptions of the functionalities, system requirements, user interactions, and non-functional requirements of the library management system.

2) General description:

- The library management system is designed to replace traditional and manual methods of managing libraries with a digitized system.
- That automates and optimizes the processes of cataloging books.
- Managing member records and tracking borrowing / returning transactions.

3) Functional requirements:-

- Member registration and authentication →
The system should allow users to register members by ~~approving~~ providing personal details
- The system should allow administrators to add, update or delete book records
- Members should be able to borrow the catalog and request to borrow books at the max 5
- The system should provide advanced search options allowing users to find books by book title, author, date of publication.

4) Interface requirements:-

- The system should be able to display all the books in ~~as~~ respective categories
- It should be able to show the borrowed books with their respective dates in the member profile.
- It should provide multiple screens for administrators to manage the system.

5) Design constraints:-

- The system should efficiently run on windows and macOS
- The system should ~~and~~ must support mobile devices with multiple screens.
- The system should enforce role-based access control to restrict access to certain features.

6) Performance requirements :-

- The system should handle multiple concurrent users.
- Response time for searching books and processing books transactions should be under 3 seconds.
- The system should decrease number of faults or errors happening.

7) Non-functional requirements :-

- * security → All user passwords must be stored securely.
- Role-based access should be implemented to insure only authorized personnel can access or modify sensitive data.
- * Usability → The interface should be user friendly and intuitive for all type of users.
- * availability → The system should be available all the time to support night readers and morning readers.

8) Preliminary schedule and budget:-

- A library management system can be build in college environment or in public.
- It should not take more than 2 months to build the system.
- The implementation cost can come around ₹ 50,000 to ₹ 60,000.

Class Diagram

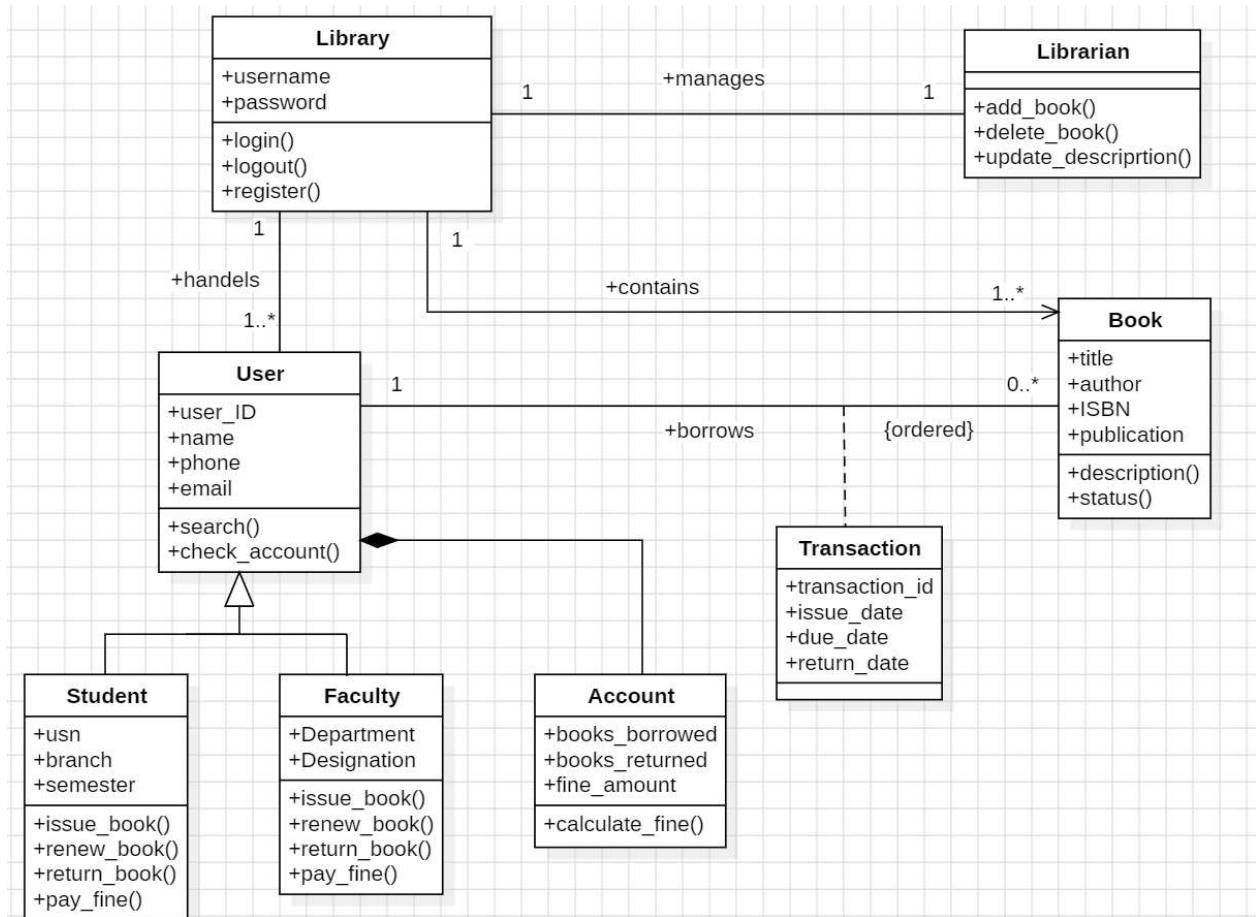


Fig 3.1 Library Management System -Class Diagram

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

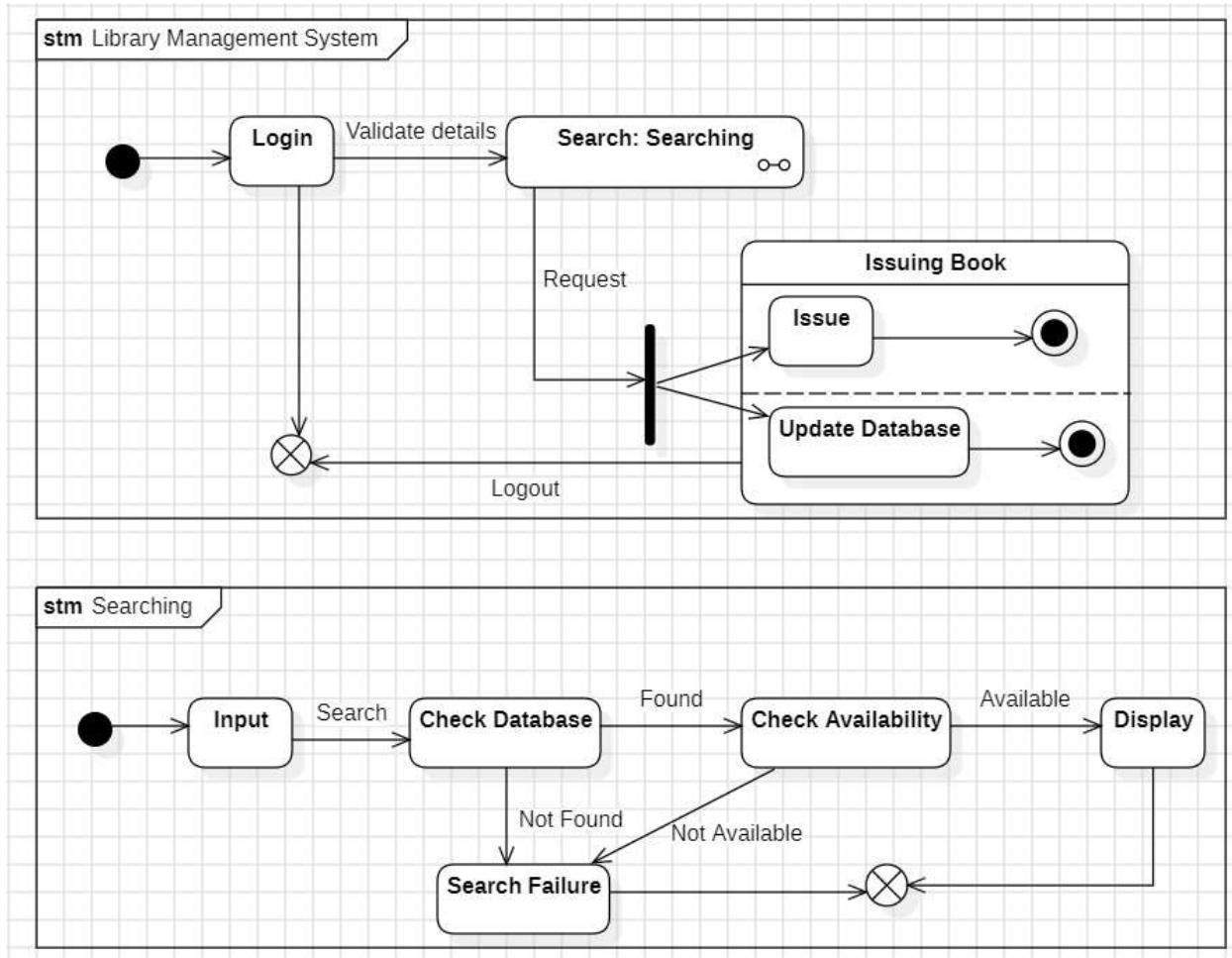


Fig 3.2 Library Management System - State Diagram

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

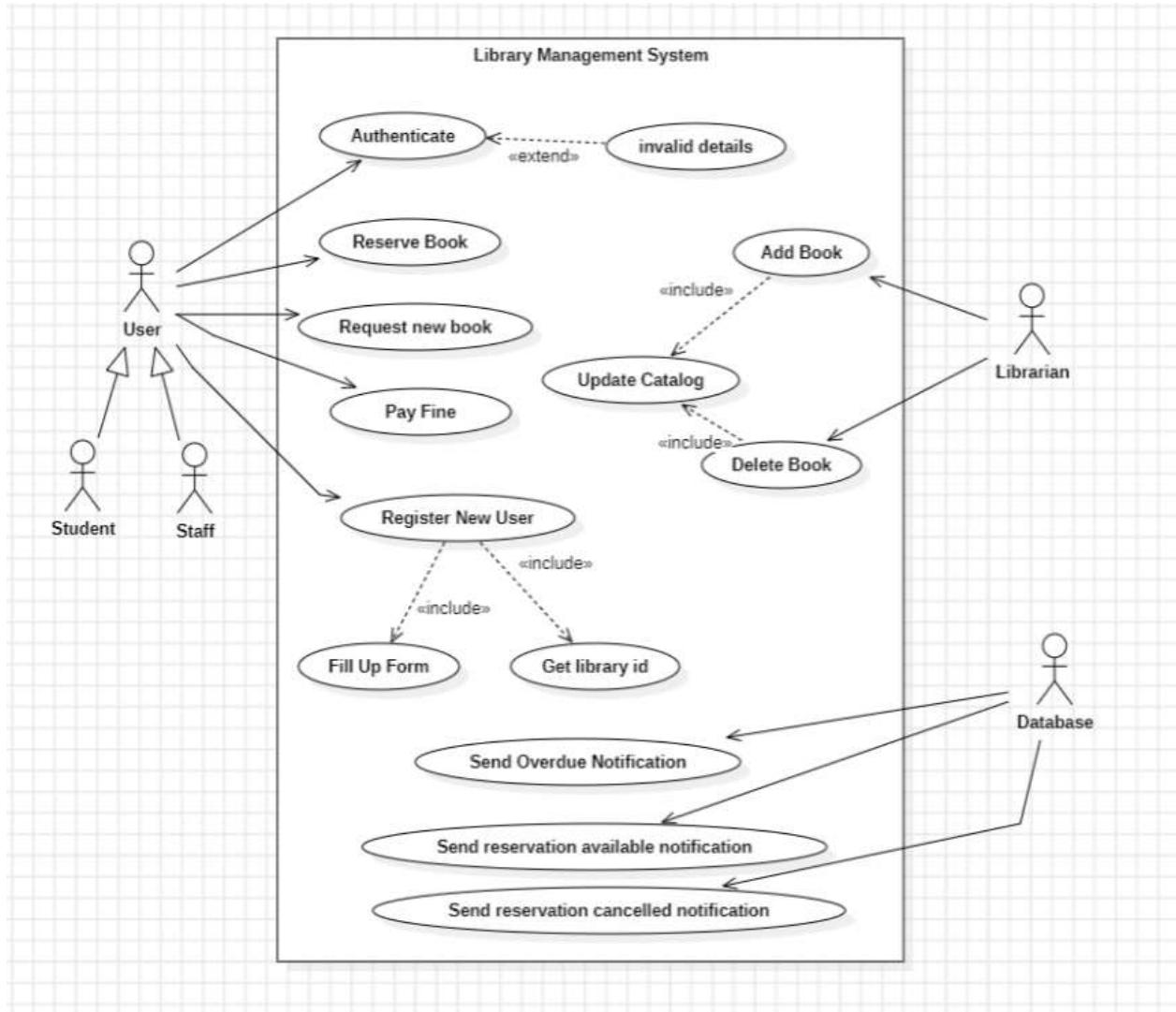


Fig 3.3 Library Management System - Use Case Diagram

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

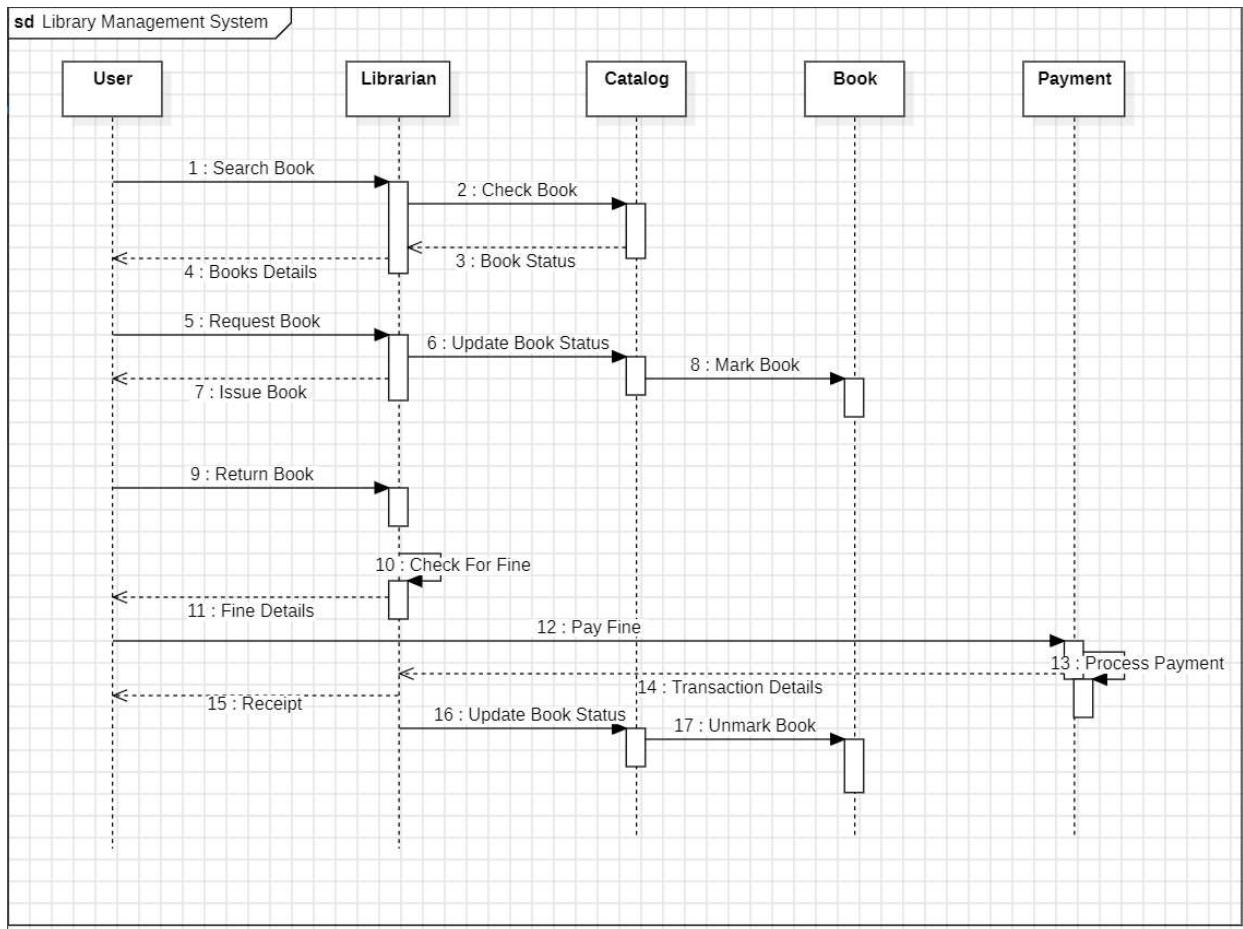


Fig 3.4 Library Management System - Sequence Diagram

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

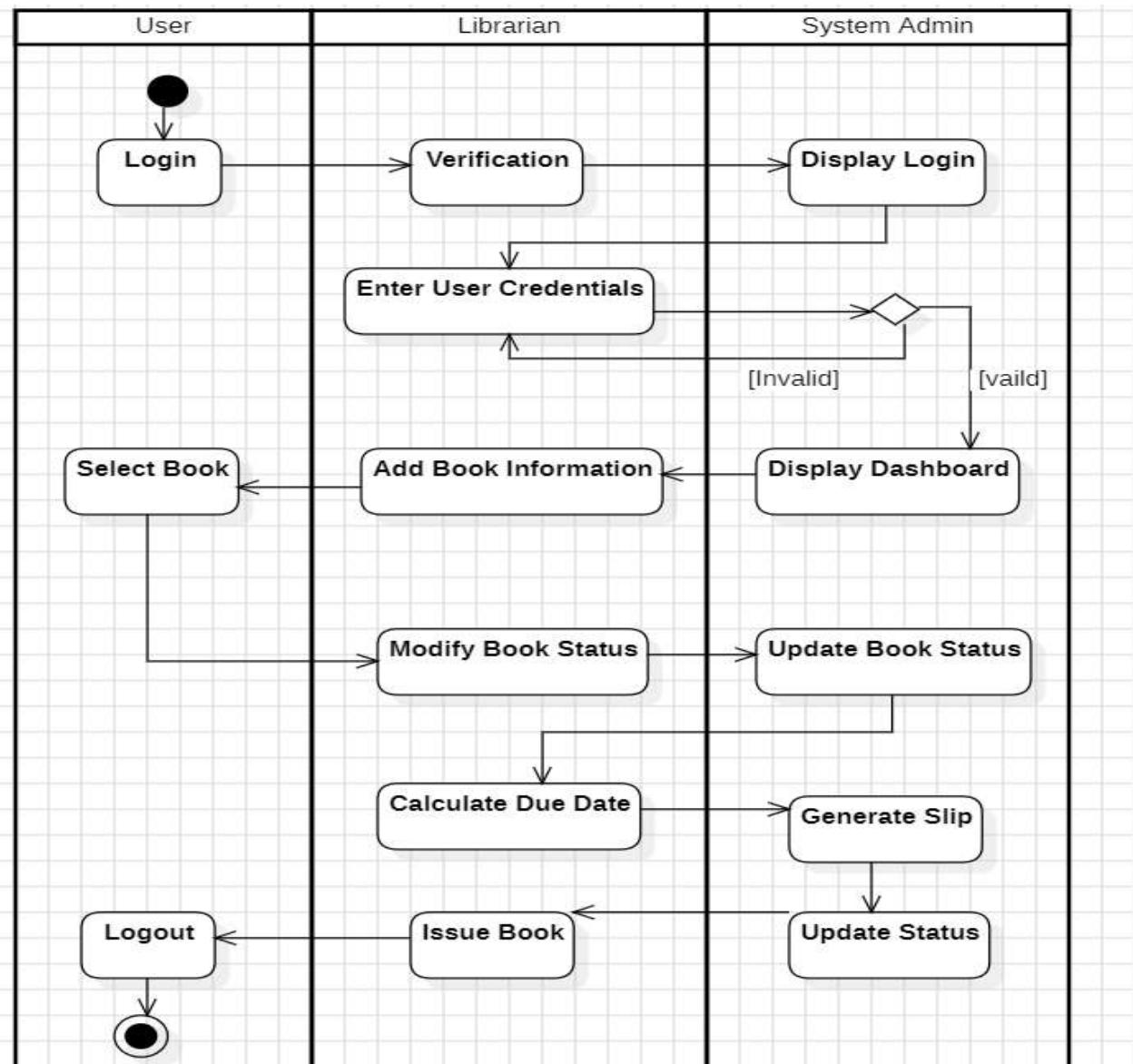


Fig 3.5 Library Management System - Activity Diagram

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

Stock Maintenance System

SRS document:-

1) Introduction:

1.1) Purpose of this document:- The purpose of this document is to provide detailed specification for a stock maintenance system that helps businesses track inventory, manage product stock levels.

1.2) Scope of this document - The system is designed to manage stock for businesses of all sizes, providing functionalities for stock tracking, order processing and report generation.

1.3) Overview:- This document provides requirements of stock maintenance system's functionality, user interactions, system architecture. The system will help businesses manage their inventory efficiently and reduce manual processes.

2) General description:-

→ Stock maintenance system is a stand alone system application designed to automate stock management tasks such as tracking inventory levels, managing product orders.

→ The system will be web based enabling access from multiple devices including desktops, laptops and mobile devices.

3) Functional requirements :-

- The system will allow users to create, view and update product information including SKU, product name & category and stock levels.
- Users can log stock movements into the system when receiving new inventory or when products are sold to other locations.
- The system will provide real-time stock tracking allowing users to monitor inventory levels, stock aging and usage trends.

4) Interface requirements :

- Interface should be very informative with lot of graphs showing the recent trends in respective stocks.
- Interface must have user profile where user can see, and maintain his stocks.

5) Design constraints :-

- The system should handle growing number of products, suppliers and stock movements of business expand.
- Role-based access control will be implemented to ensure that only authorized users can access certain data or perform sensitive operations.
- The system should be able to process stock updates, generate reports and handle multiple users.

2) Performance requirements:-

- The system should support up to 5000 concurrent users without any noticeable delay in loading.
- All stock related transactions should be processed within 2-3 seconds.
- It should be able to show live stock value any time.

3) Non-functional requirements:-

- All user data including stock information and supplier information should be stored securely.
- The system must be easy to use with an intuitive user interface designed to minimize errors.
- The system should maintain real-time management with regular back-ups of database to prevent data loss.

4) Preliminary schedule and budget

- The development of stock maintenance system should take up to 6 months.
- Total budget of system can come around 1-2 lakhs.

Class Diagram

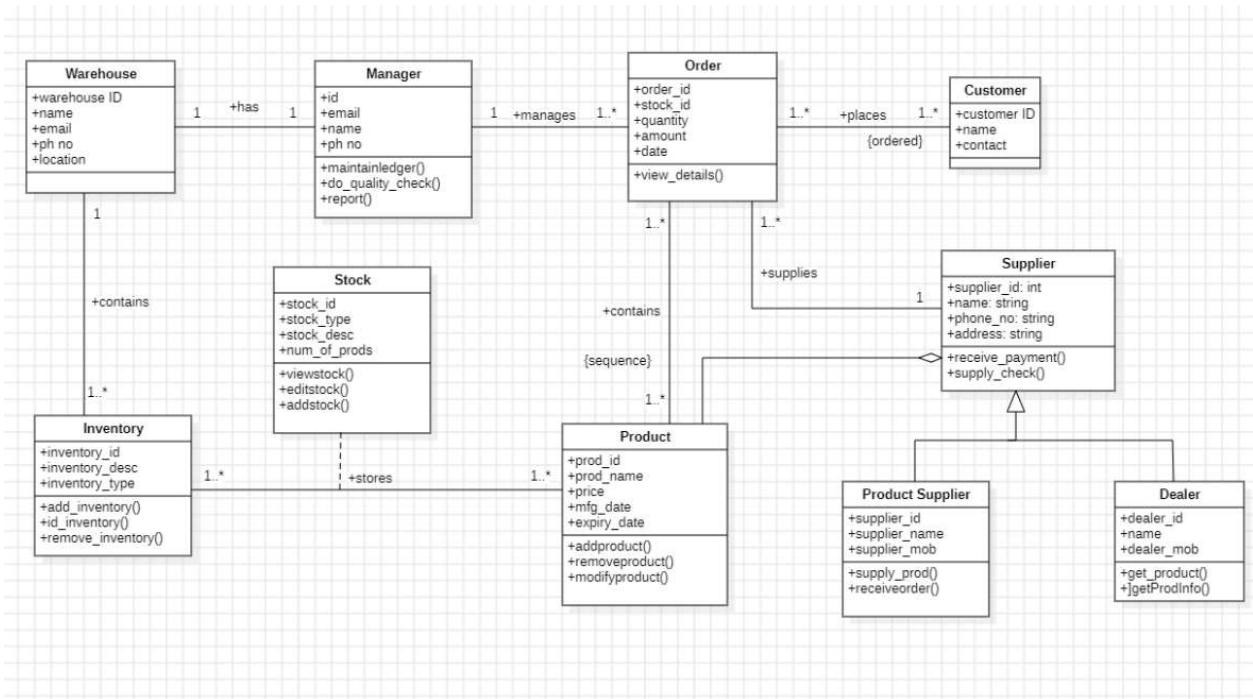


Fig 4.1 Stock Maintenance System - Class Diagram

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

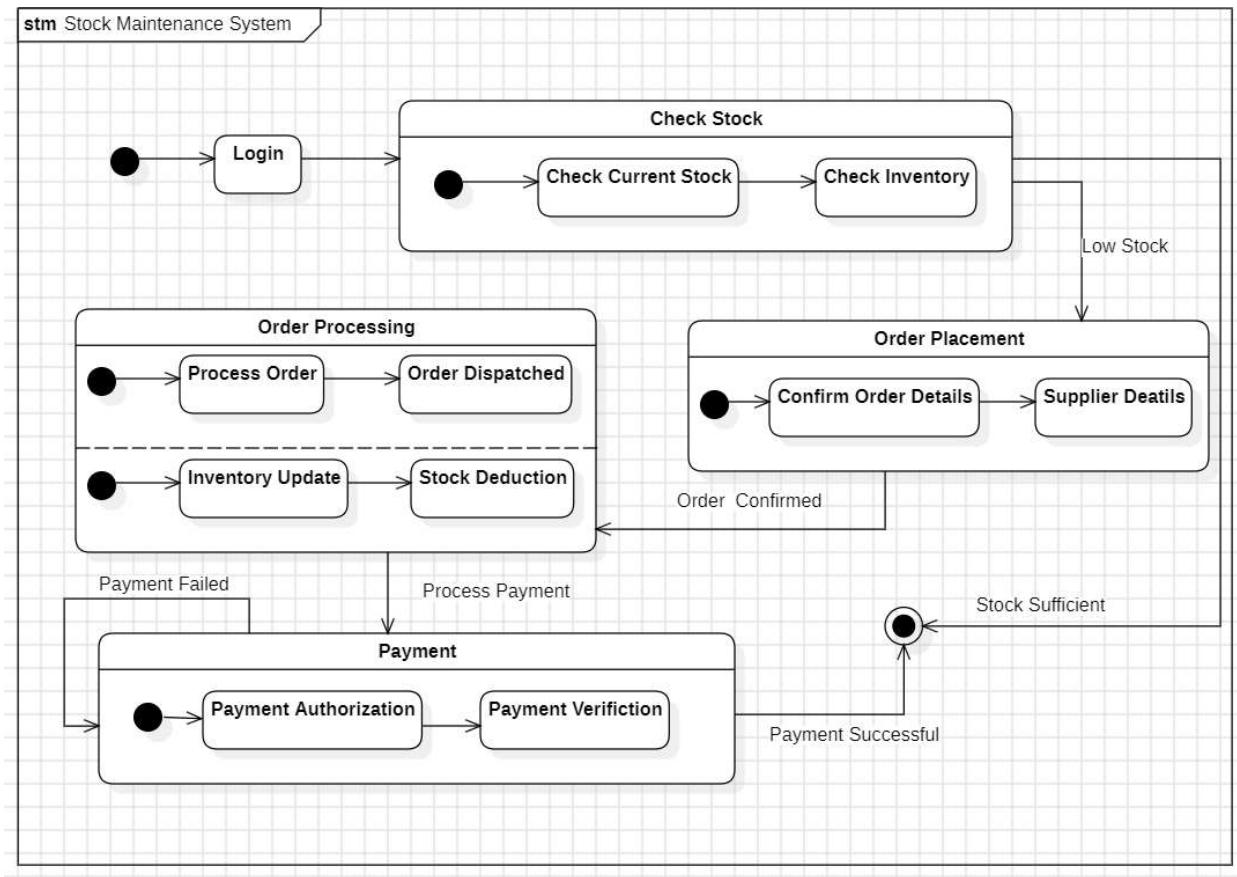


Fig 4.2 Stock Maintenance System - State Diagram

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

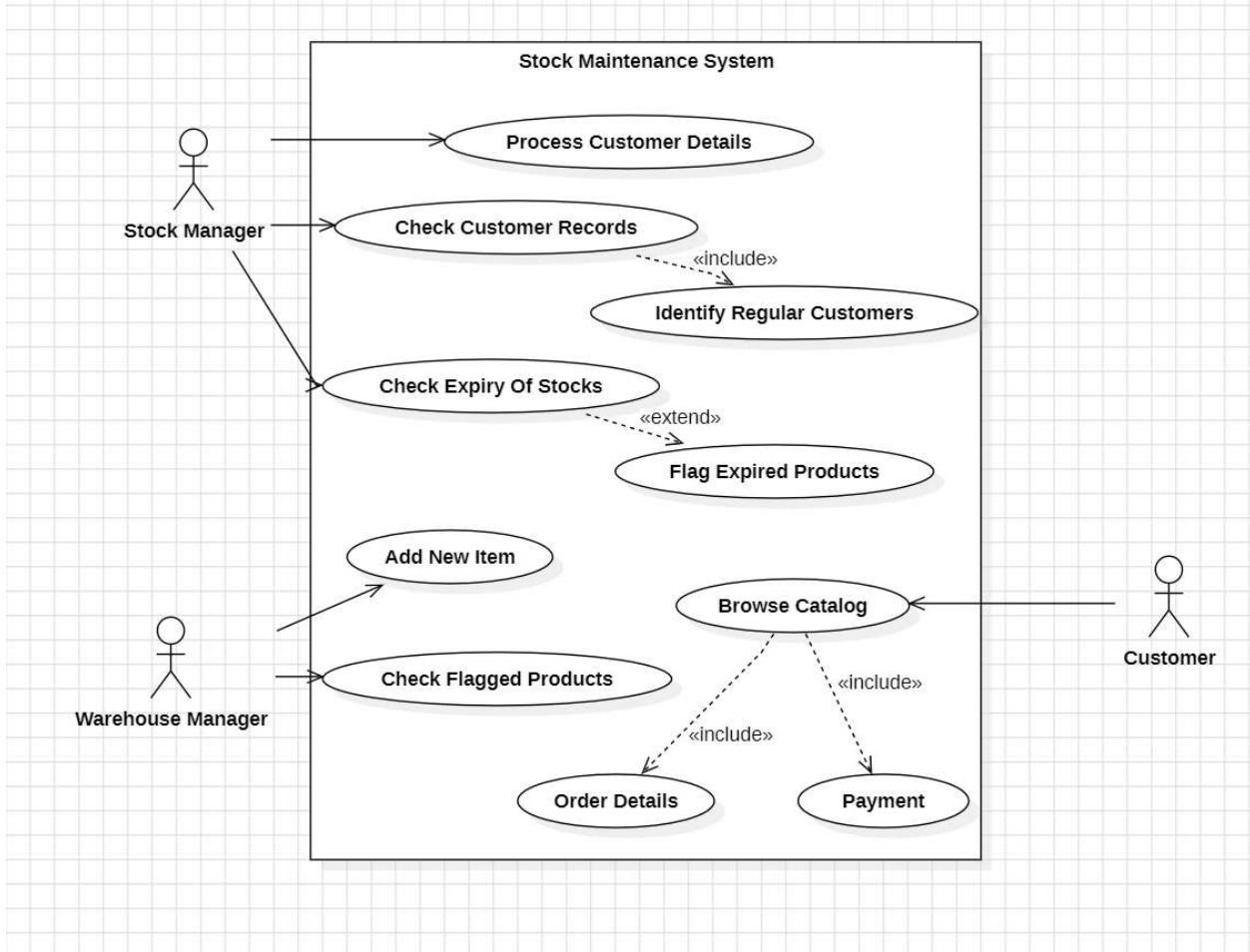


Fig 4.3 Stock Maintenance System - Use Case Diagram

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

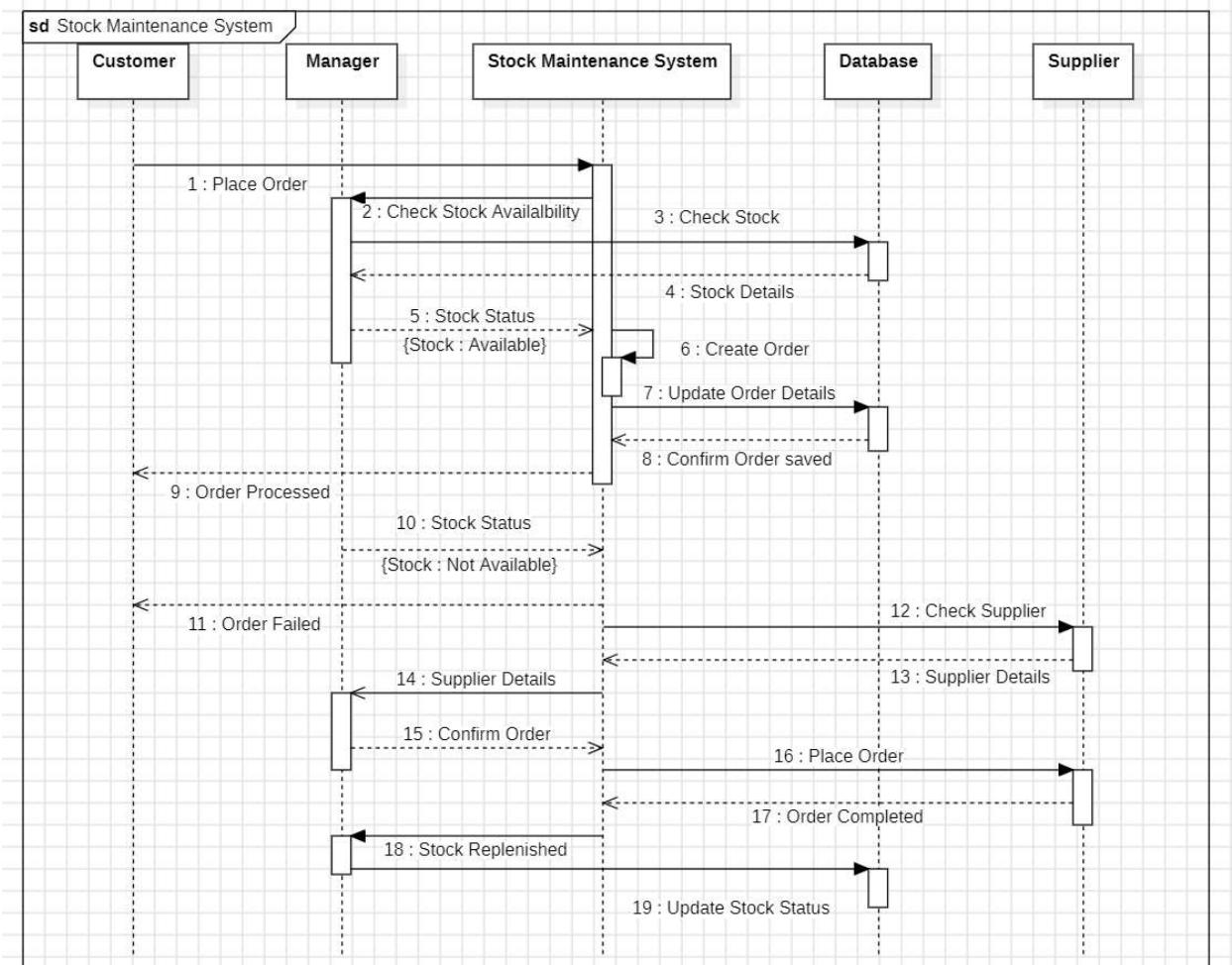


Fig 4.4 Stock Maintenance System - Sequence Diagram

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

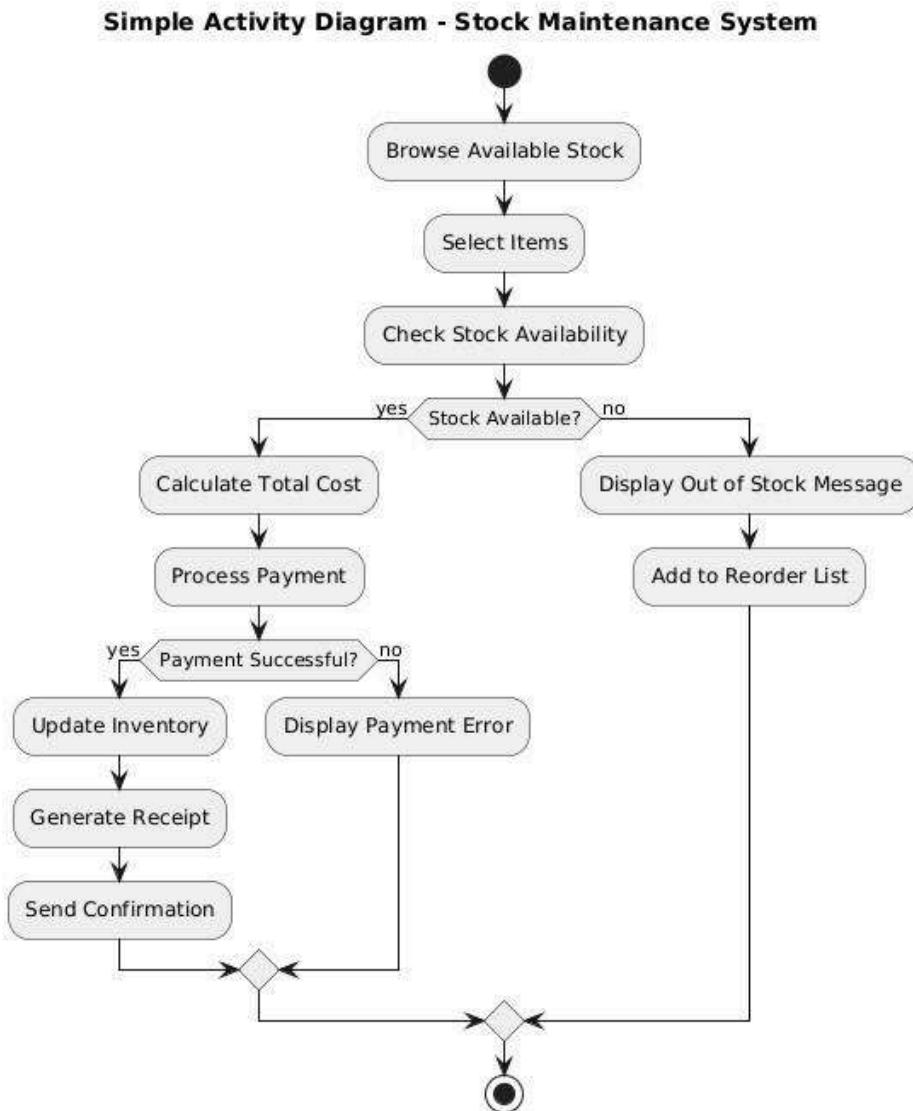


Fig 4.5.1

The stock maintenance process follows a linear flow, including key activities like stock browsing, availability checks, payment processing, inventory updates, and receipt generation. Decision points include stock availability and payment success/failure, ensuring simplicity and clarity for stakeholders while focusing on core functionality.

5. Passport Automation System

Software Requirement Specification

Passport Automation System

1) Introduction:-

1.1: Purpose:- This document defines a functional and non-functional requirements for a passport automation system (PAS). This system is designed to automate the application, processing and issuance of passports providing a streamlined solution.

1.2:- Scope:- The scope of this document covers the end to end process of passport application and issuance including online application submission, document verification, payment processing.

1.3: overview:- A web-based system allowing online passport applications, real time tracking and automation of passport processing.

2) General description:- The PAS is intended for use by passport applicants, processing officers and administrative personnel. The system will automate the passport applications. The system will ensure secure management of applicants data.

3) Functional requirements:-

- Application submission → The system must allow users to fill out and submit passport applications online.
- Status tracking → It should enable applicants to track the status of their applications real time.
- payment-processing → The system must support online payments for application fee through various payment gateways.

4) Interface requirements:-

- UI - The system must provide a web-based application portal for applicants.
- Database interface - It should be secure storage of applications and personal data.
- API - Integration with identity verification databases and payment systems.

5) Performance requirements:-

- It should handle maximum concurrent users. It should have real-time application tracking and status updates.

6) Design constraints:-

- The system should work on multiple web-based platforms. It should comply with data protection laws, encryption of data. It should support future growth in user base.

7) Non-functional Attributes:-

Security - Encryption and access control

usability - simple, user friendly interface

scalability - It should be capable of handling multiple users and their data

8) Preliminary schedule and budget:-

→ Requirement analysis (3 weeks), Design (3 weeks), Development (10 weeks), Testing (4 weeks) and deployment and training (2 weeks)

→ Development cost (£50000), Design (£20000), Testing (£10000), Deployment and training (£15000)

Class Diagram

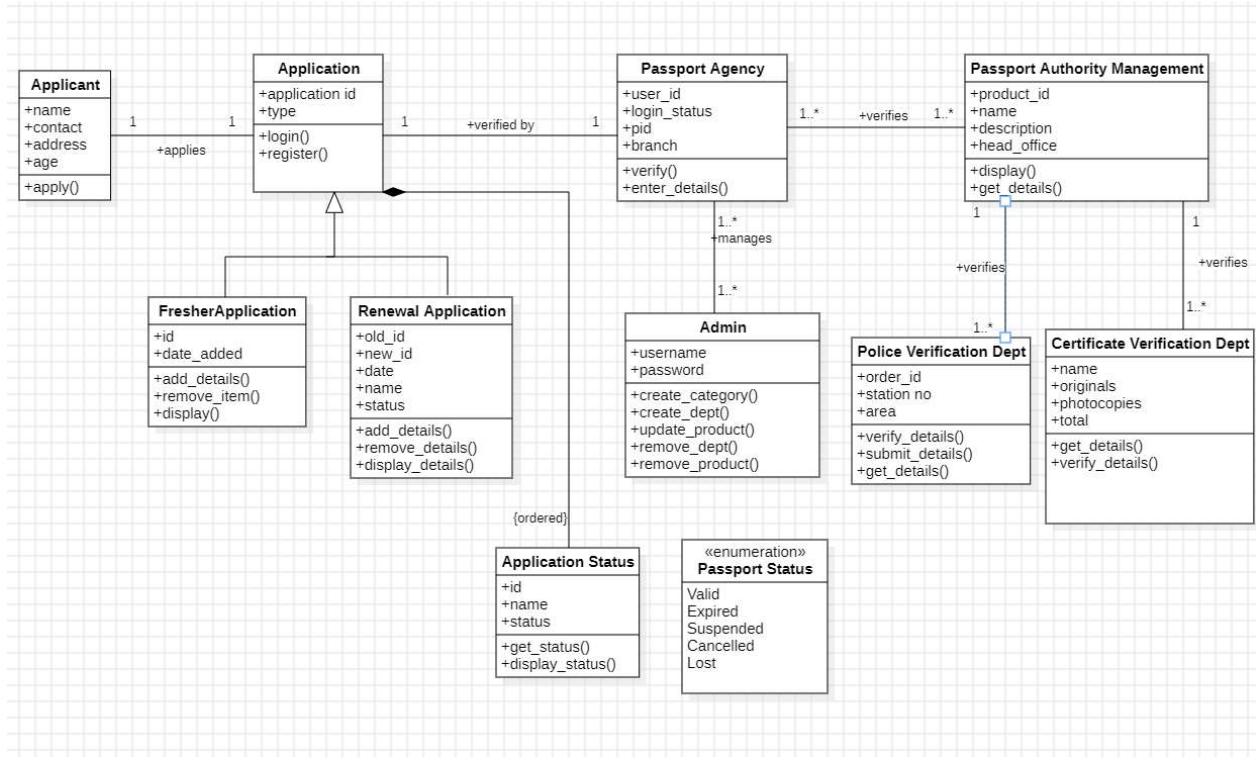


Fig 5.1 Passport Automation System - Class Diagram

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

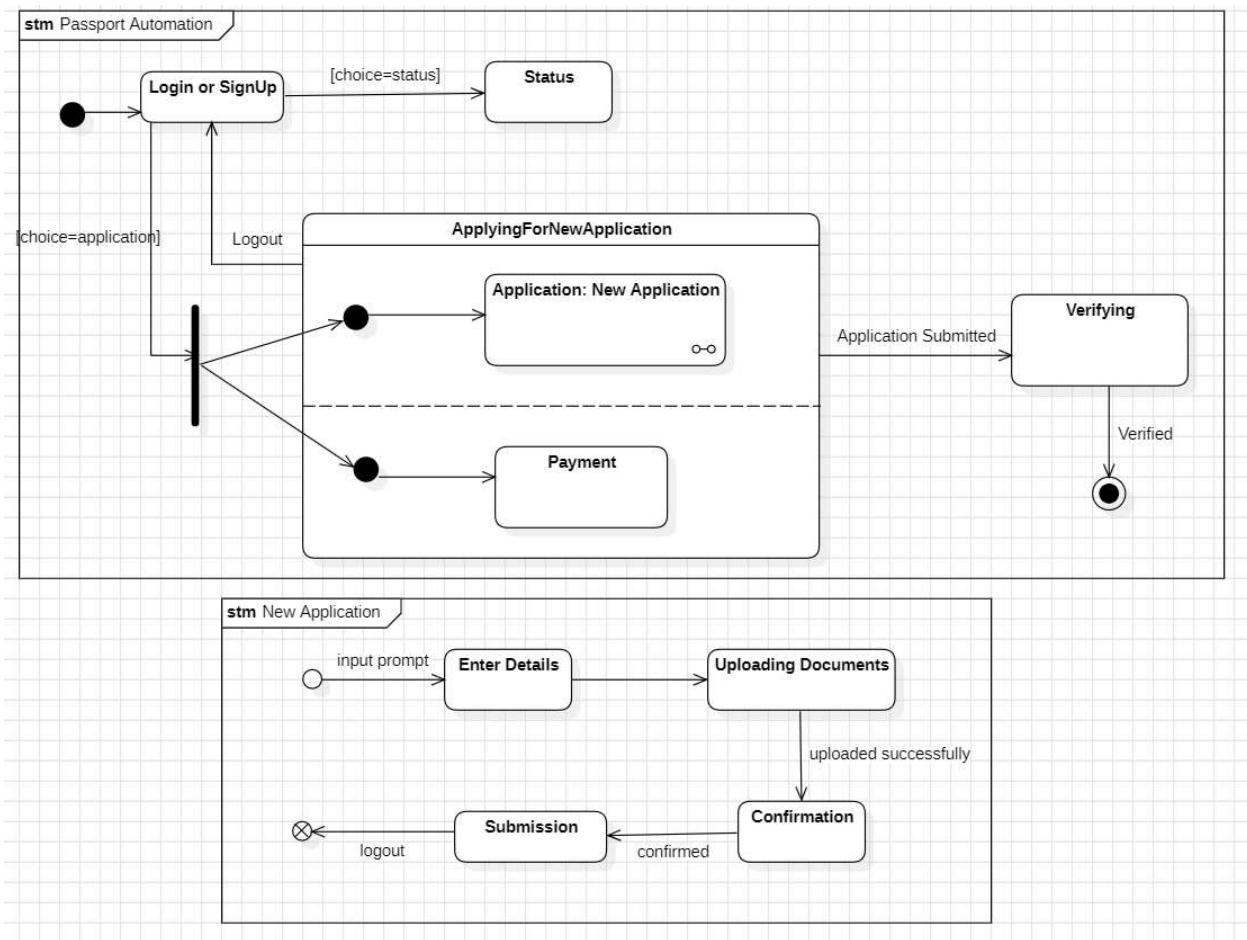


Fig 5.2 Passport Automation System - State Diagram

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

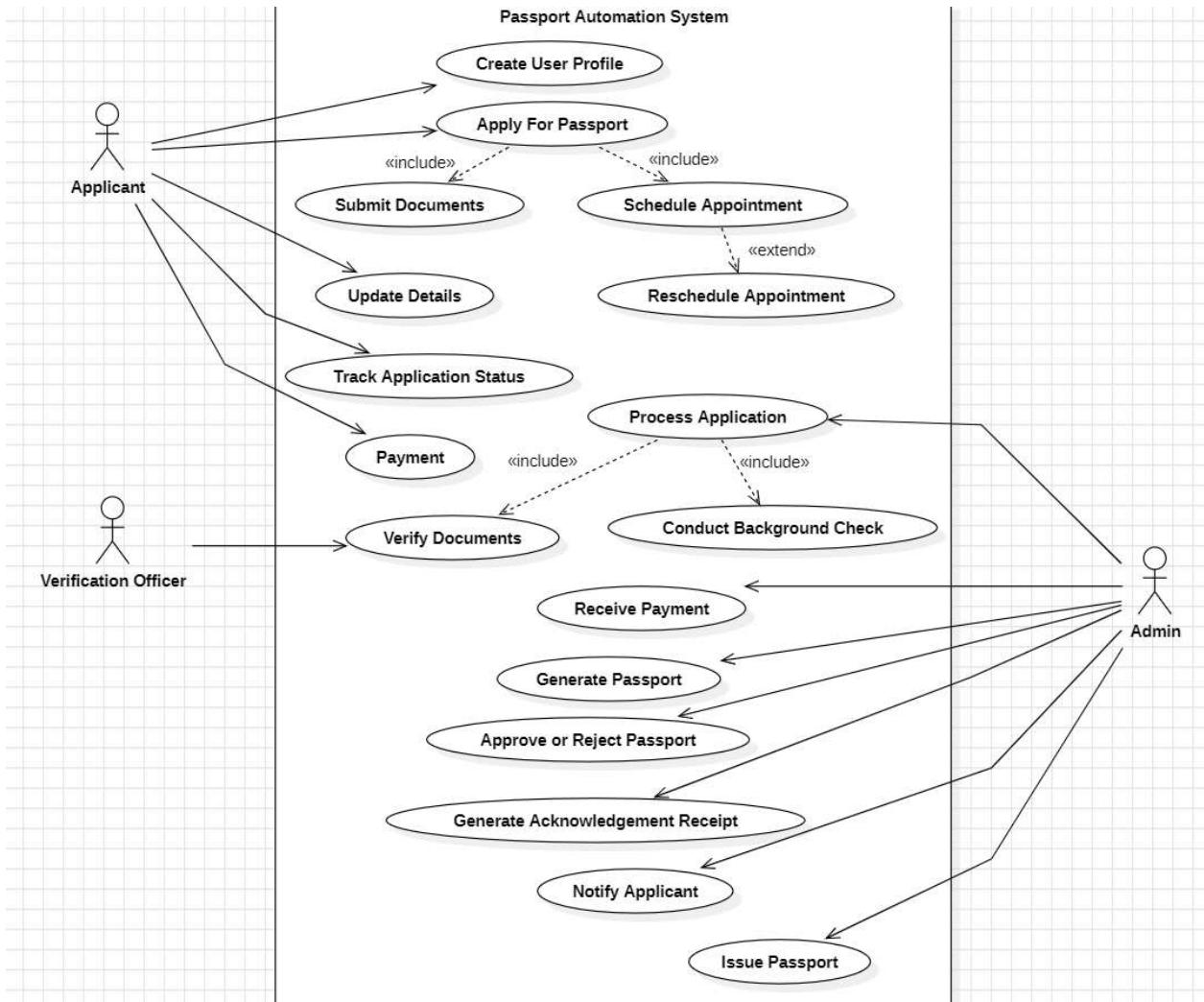


Fig 5.3 Passport Automation System - Use Case Diagram

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

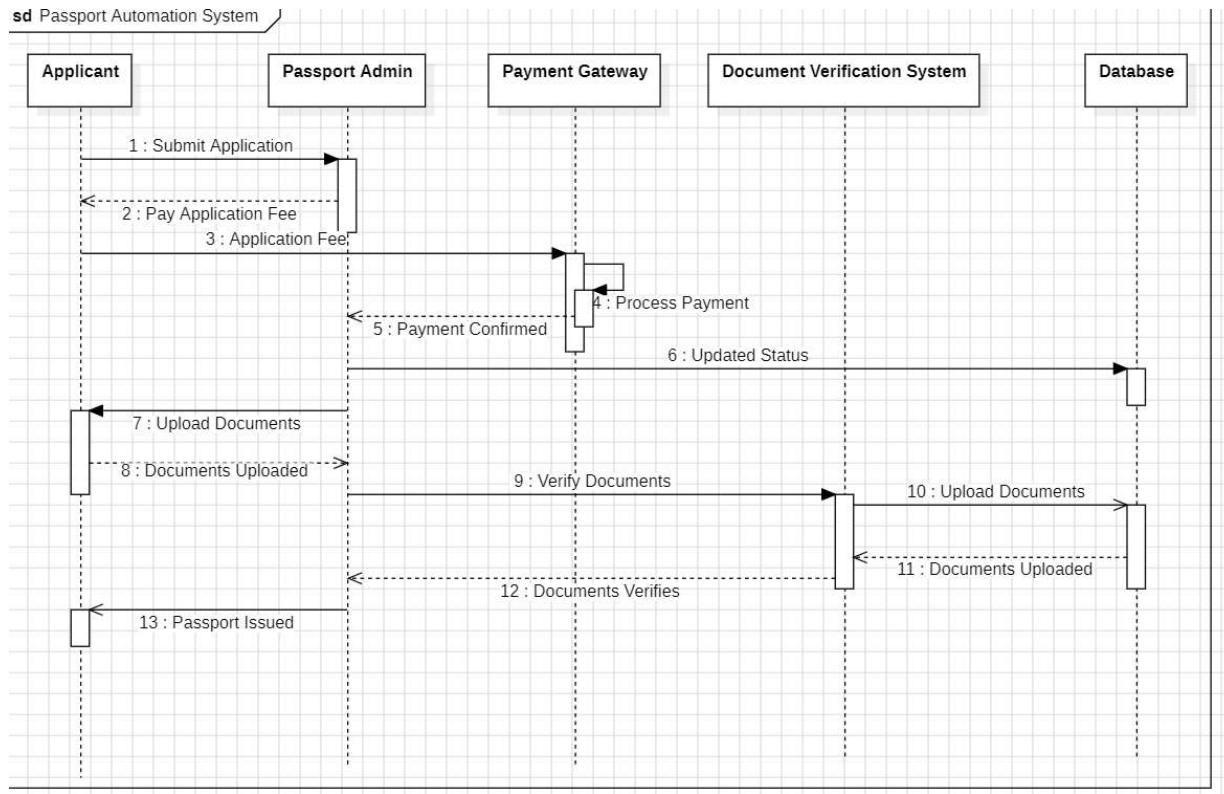


Fig 5.4 Passport Automation System - Sequence Diagram

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

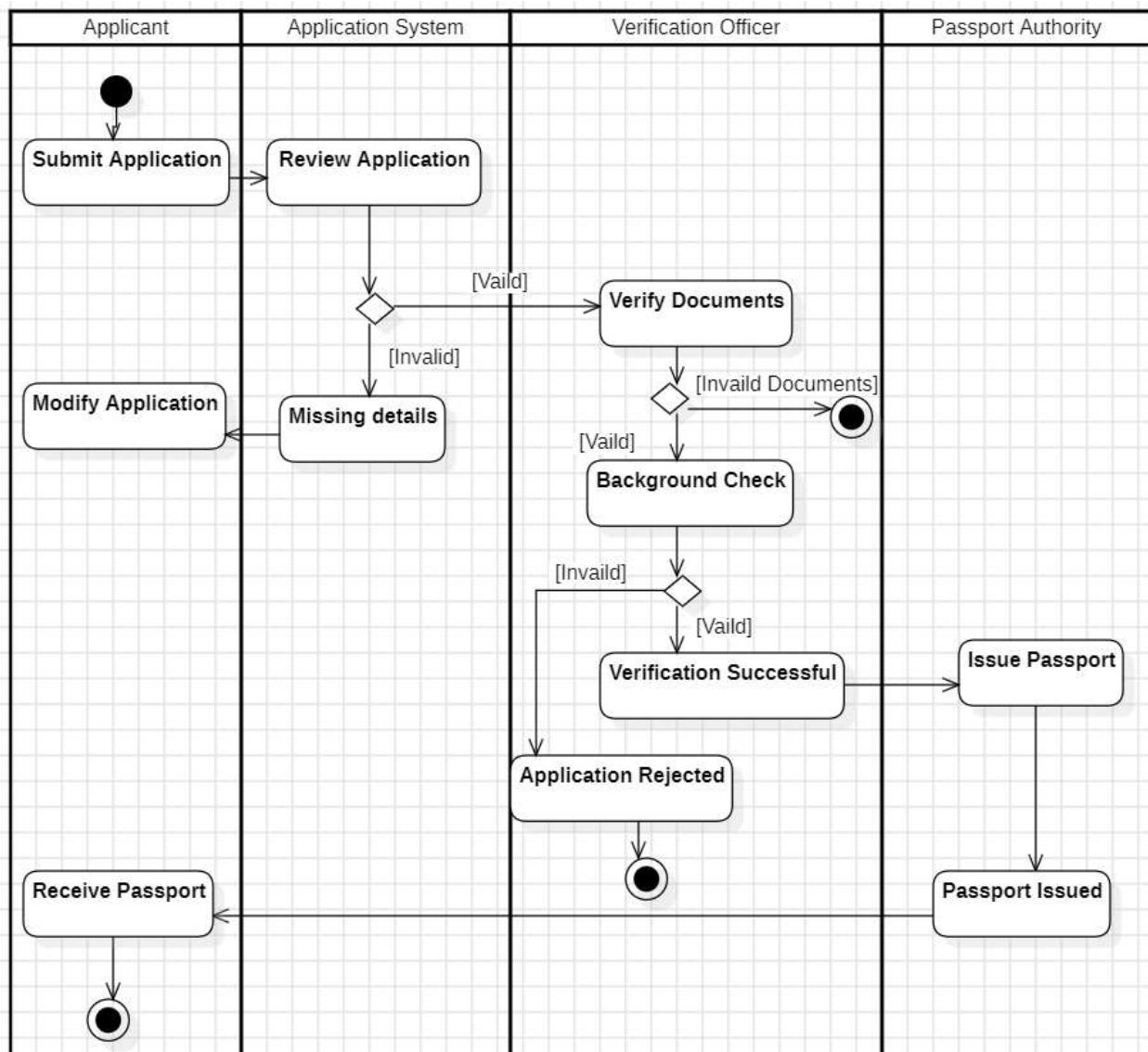


Fig 5.5 Passport Automation System - Activity Diagram

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.