

**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 5<sup>th</sup> Semester Laboratory*

Bachelor of Engineering  
in  
Computer Science and Engineering

*Submitted by:*

**SAKSHI B R**  
1BM22CS233

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 SAKSHI B R (1BM22CS233) during the 5<sup>th</sup> Semester Oct 24-Jan 2025.

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

## Problem Statement:

The current manual processes for managing hotel operations, including reservations, check-ins, check-outs, and billing, are inefficient and prone to human errors. These inefficiencies lead to issues such as inaccurate record-keeping, mismanagement of room availability, and delays in customer service. As a result, both operational effectiveness and customer satisfaction are compromised. There is a pressing need for a centralized and automated hotel management system that can streamline operations, reduce errors, enhance customer service, and ensure accurate and real-time data management.

## SRS-Software Requirements Specification:

| GoodLuck   |  | Page No. | Date |
|--|--|----------|------|
| [1] Hotel Management System:   |  |          |      |
| [2] Introduction:<br>Purpose: Development of Hotel Management System<br>Scope: This document includes functional and non-functional requirements, user interface requirement, budget and schedule.<br>Overview: This document helps development of hotel management system and managing all the functional and non-functional requirement. |  |          |      |
| [3] General Description:<br>This system enables the guests/ customers book the rooms as per their requirements.  |  |          |      |
| [4] Functional Requirement:<br>- Login / Signup page Register / Login<br>- Hotel Booking / Room Booking<br>- Viewing Ratings<br>- Customer Reviews<br>- Booking for service.<br>- Dining or Ordering Food ordering .<br>- Payment  |  |          |      |
| [5] Interface Requirement<br>This system is to be designed in such a way that user can easily use in their convenient system.  |  |          |      |

- 5] Performance Requirement  
 Scope Speed : Enables user to System's downtime should be less.  
 Multiple user can use this system without any inconvenience
- 6] Design Constraints:  
 This system is to be designed such that it can be used according to user convenient.
- 7] Non Functional Requirements:  
 Easily accessible without any time or place constraints.
- 8] Preliminary Schedule & Budget  
 Estimated Budget - ₹5,50,000  
 Estimated time - 5 months

## Class Diagram:

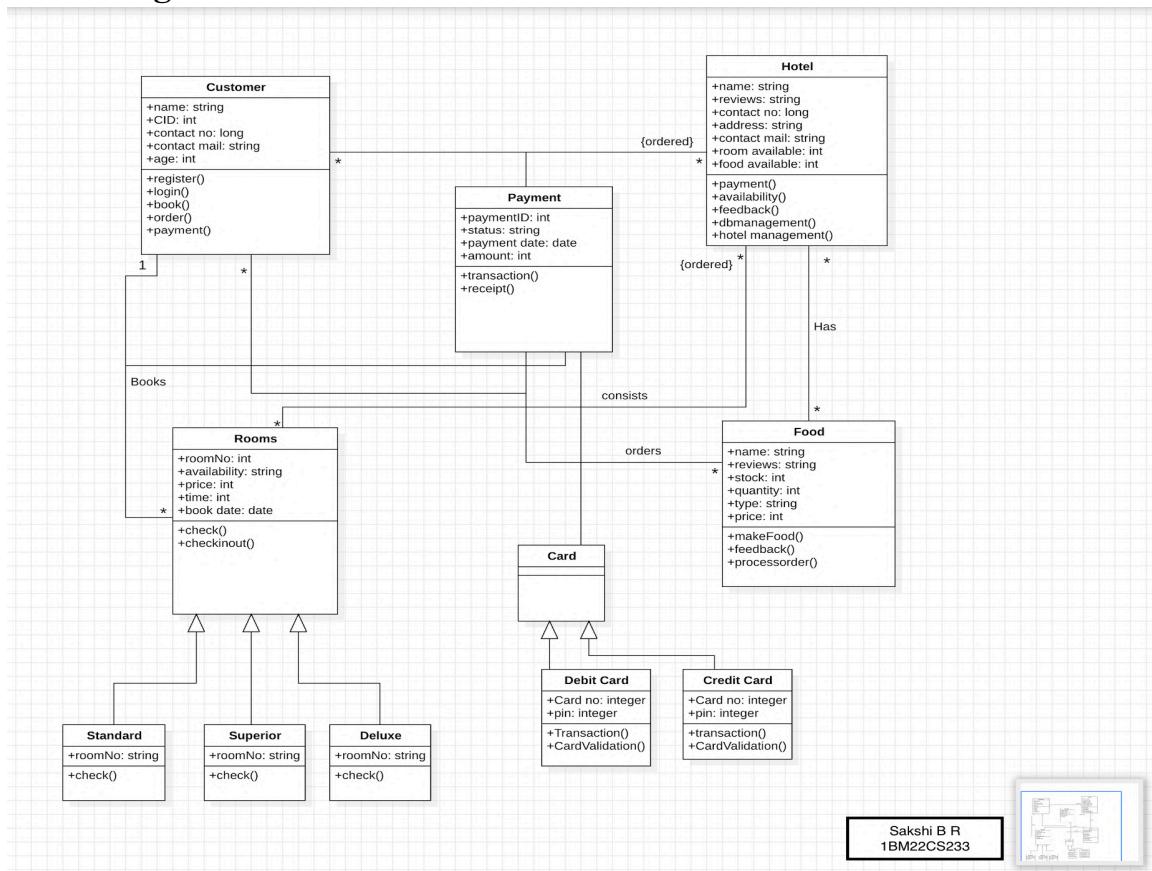


Fig 1.1

Sakshi B R  
 1BM22CS233

The Hotel Management System class diagram outlines the core entities and their interactions, ensuring efficient organization of hotel operations. The system includes entities such as Staff, with specialized roles like Manager and Receptionist, Hotel, Room, Guest, Reservation, and Payment. A hotel comprises multiple rooms that guests can reserve, and these reservations are linked to corresponding payments. Staff members handle various tasks such as cleaning and guest services, all managed by the Manager, while the Receptionist oversees guest interactions like check-ins and bookings. By effectively organizing reservations, payments, room management, and staff responsibilities, the system streamlines key operations within the hotel.

## State Diagram:

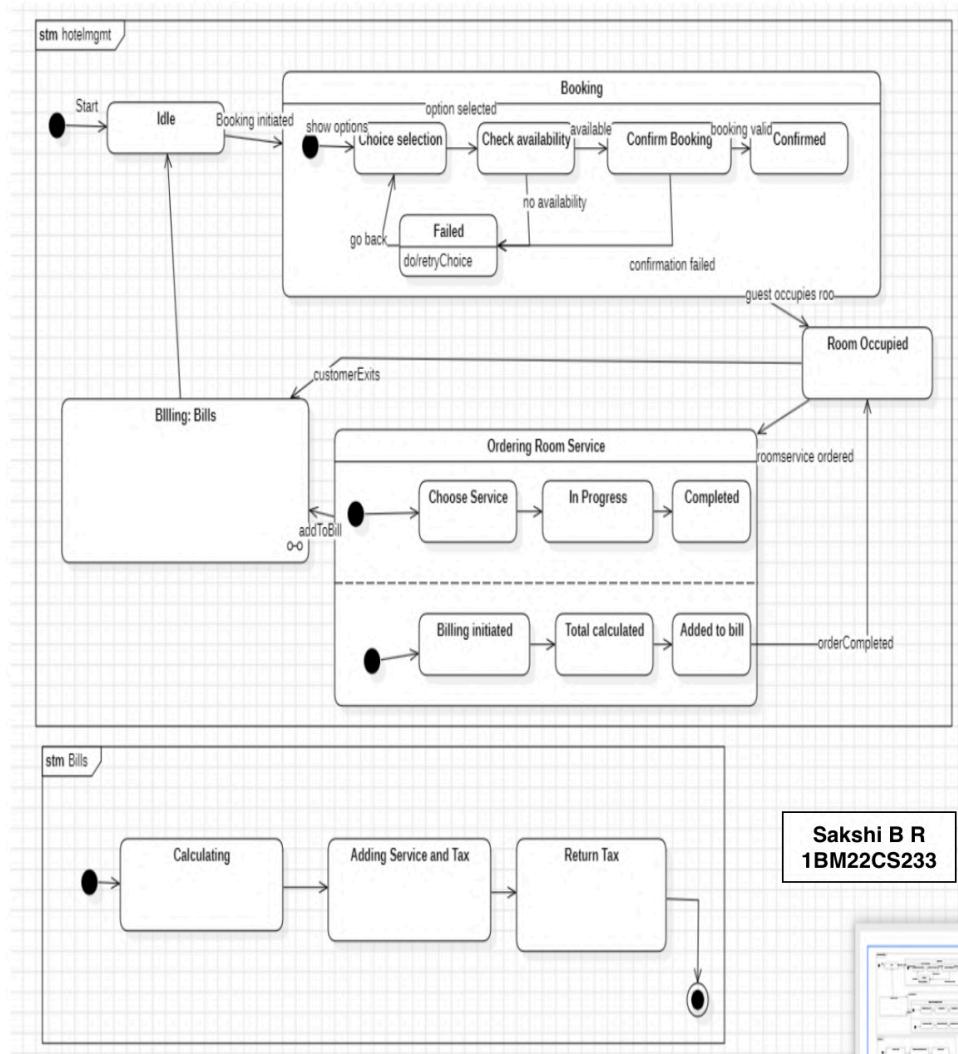


Fig 1.2

The state diagram for the Hotel Management System outlines the workflow, starting from an Idle state. When a booking request is received, the system transitions to the Booking state, where room availability is verified, bookings are confirmed, and confirmation details are sent to the guest. Guests then transition to the Check-In state, where rooms are assigned. During their stay (Occupied state), additional services like Room Service and Billing can occur. At Check-Out, the system generates bills and updates the room's status. Vacated rooms transition to a Cleaning state, where basic cleaning and inventory checks are conducted, after which the rooms return to the Idle state for reuse.

### Use Case Diagram:

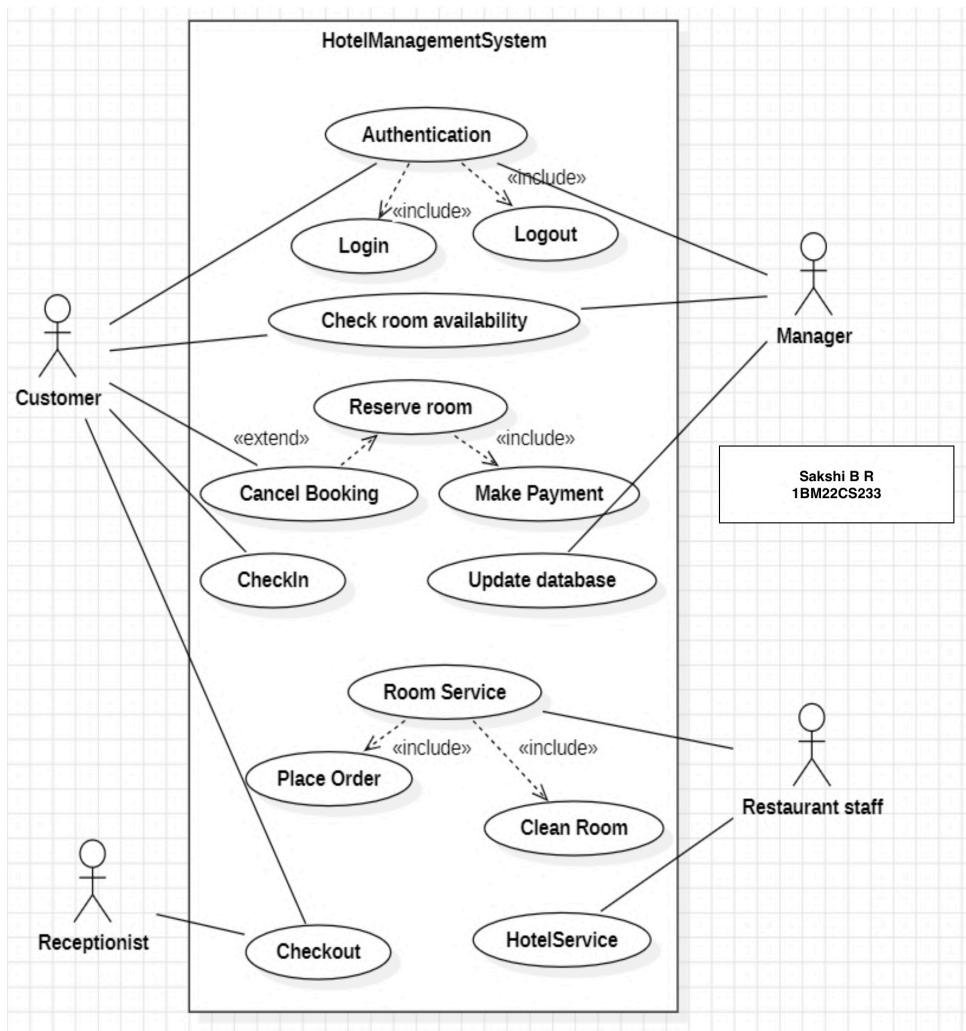


Fig 1.3

A use case diagram for a Hotel Management System highlights the interactions between actors such as Guests, Receptionists, Managers, and Staff with the system's key functionalities. Guests can perform actions like searching for rooms, making reservations, requesting services (such as room service), checking in, and checking out. Receptionists manage tasks like handling reservations, assigning rooms, processing check-ins and check-outs, and generating bills. Managers are responsible for supervising staff, monitoring bookings, and managing the system as a whole. Staff focus on operational tasks such as room cleaning, conducting inventory checks, and delivering requested services, providing a clear overview of the system's workflow and operations.

## Sequence Diagram:

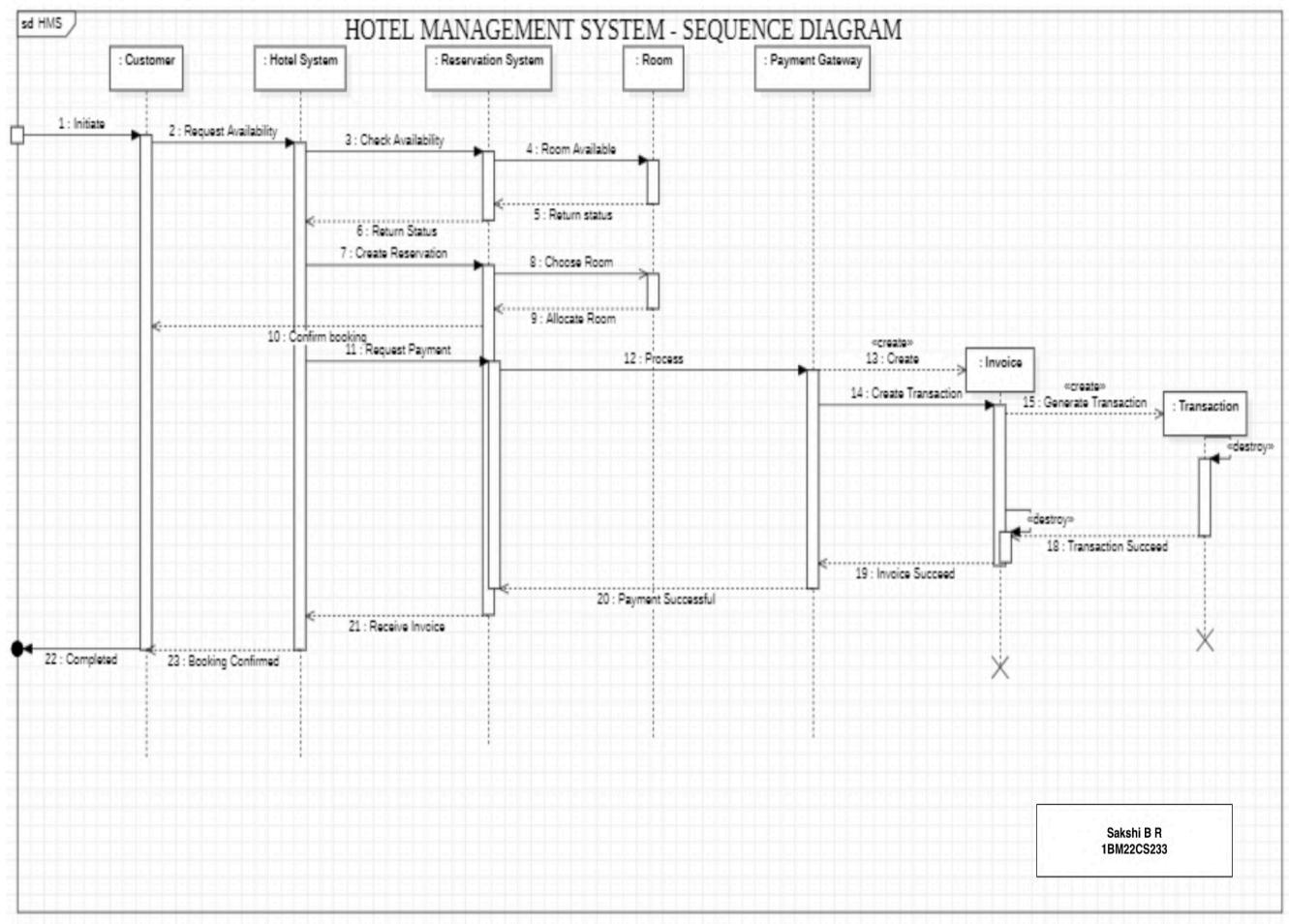


Fig 1.4

The sequence diagram for a Hotel Management System illustrates the flow of interactions between various entities like Guest, Receptionist, System and Room Service. It begins with the Guest requesting a room, followed by the Receptionist or system verifying room availability. Upon confirmation, the booking is processed, and payment details are handled. Once the guest checks in, interactions for additional services like room service or billing are shown. The diagram concludes with the Check-Out process, where the system generates a bill and updates room status. It emphasizes the chronological order of operations and communication between actors and system components.

## Activity Diagram:

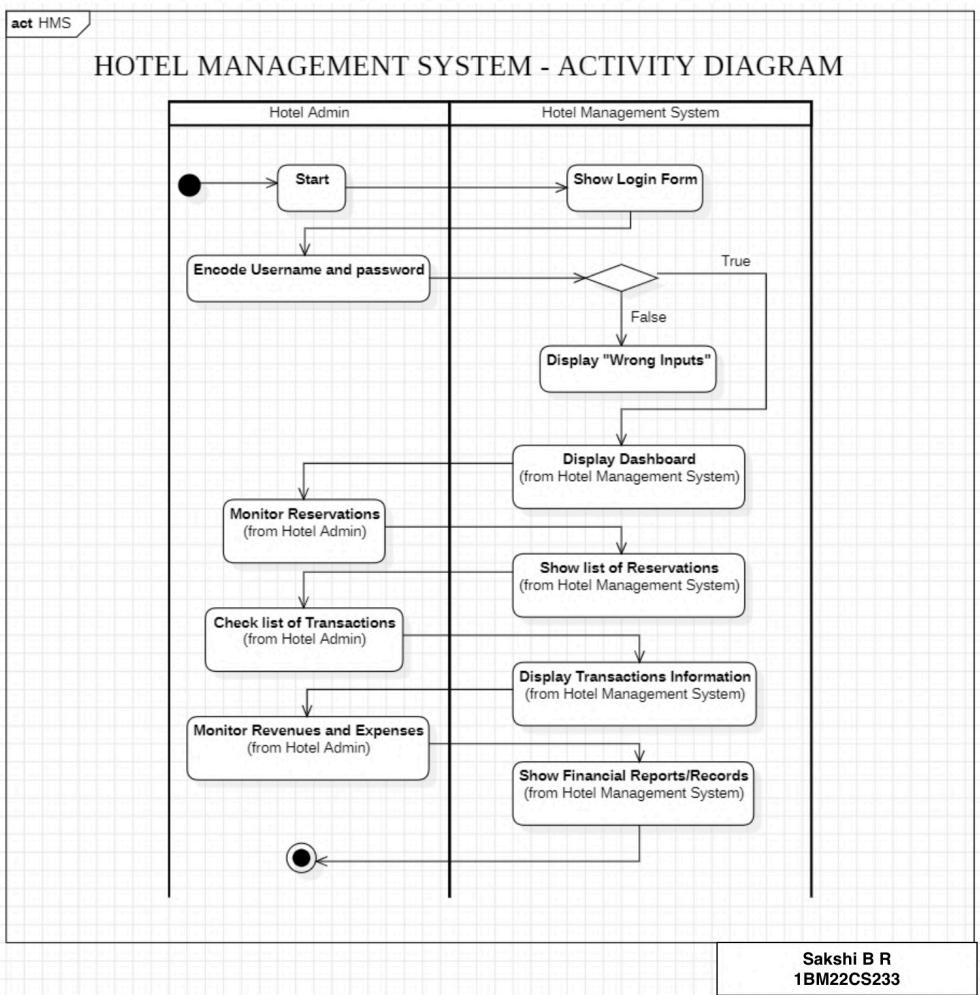


Fig 1.5

The sequence diagram for a Hotel Management System outlines the step-by-step interactions between key entities such as the Guest, Receptionist, System, and Room Service. It starts with the Guest requesting a room, followed by the Receptionist or system checking room availability. Once availability is confirmed, the booking is processed, and payment is handled. After the Guest checks in, the diagram captures interactions for additional services like room service or billing. It concludes with the Check-Out process, where the system generates the bill and updates the room's status. This diagram highlights the chronological flow of operations and communication among actors and system components.

## 2. Credit Card Processing System

### Problem Statement:

The existing manual processes for credit card transactions are inefficient, error-prone, and vulnerable to security breaches. Issues such as delayed payment processing, inaccurate transaction records, and the risk of fraud compromise both user experience and business operations. There is a need for a secure, automated credit card processing system that ensures accurate transaction handling, faster processing times, enhanced fraud detection, and compliance with payment security standards to improve reliability and user trust.

### SRS-Software Requirements Specification:

Good Luck Page No.  
Date 30.09.24

Software Requirement Specifications.

Credit Card System:

1] Introduction:-  
Development of Credit Card System which enables to read credit card.

Scope of the document : Working of credit card system that is it enables user to read their credit card for their payment transaction, user account management

2] Overview:  
This system help user manage their credit card and processing payments.

3] General Description:  
It enables user to make payments, track their transaction, manage their accounts.

3] Functionalities Requirements:  
Registration : User can securely register for new account with personal details.

Issuing of card : Credit card should be issued to user with unique number and CVV.

Transaction Processing and history :  
This system enables users to make payment for any of their purchases.  
The system also enables user to look into history of their transaction.

Security: Most important role is to provide security or safeguard user's personal information from frauds.

4] Interface Requirement:

This system is to be designed in such a way that user can easily use in their convenient system.

5] Performance Requirement:

The system should enable user to make payment within seconds.

Multiple users can use this system without any inconvenience.

6] Design Constraints:

This system should be designed such that it can be used in web browser or mobile which ever user are convenient to use.

7] Non Functional Requirements:

Easily Accessible: The system should be accessed easily by any user.

Portable: Can be accessed by user anywhere, <sup>anytime</sup> without any place constraints.

Portable: User can access the system on <sup>anytime</sup> ~~anyday~~ any convenient devices such as web browser or mobile phones - iOS / android.

8] Preliminary Schedule & Budget:

Estimated budget - ₹ 1,00,00,000

Estimated Time - 8 months

## Class Diagram:

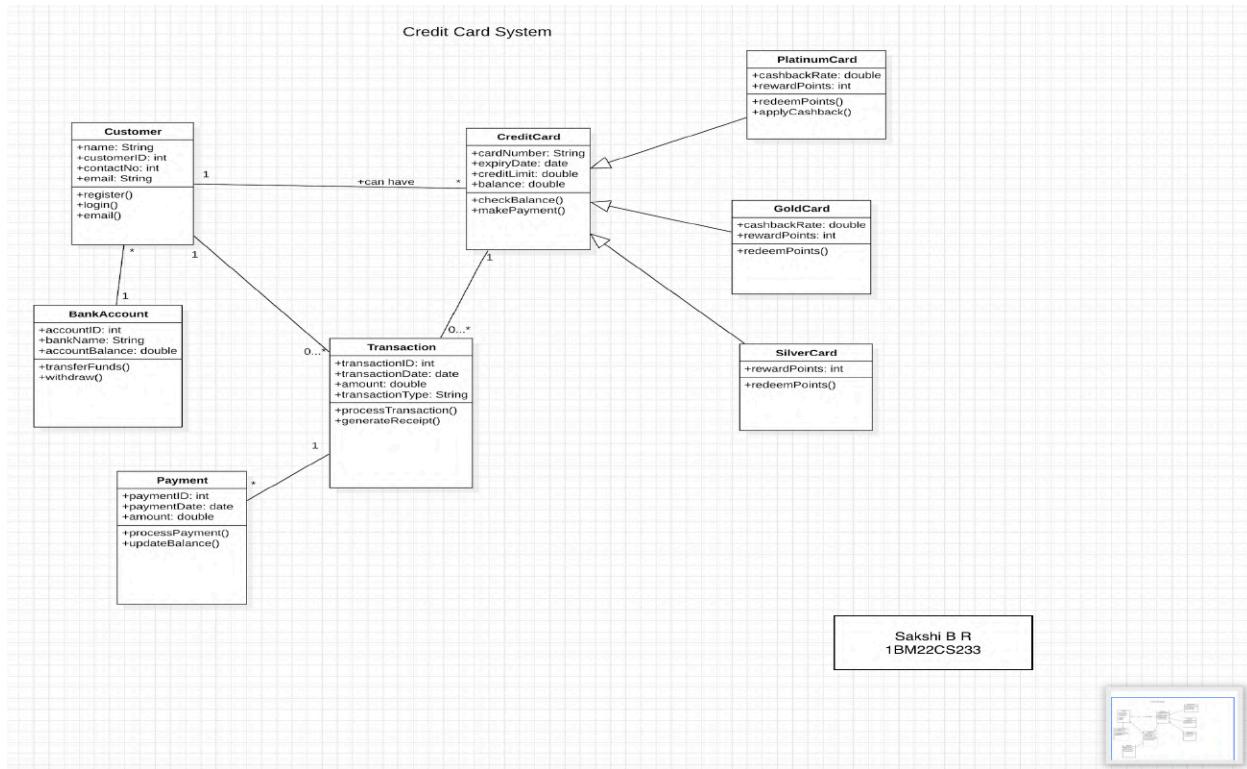


Fig 2.1

The Credit Card System class diagram illustrates the key entities and their relationships to manage credit card operations effectively. It includes entities such as Cardholder, Credit Card, Transaction, Merchant, Payment, Bank, and Credit Limit. A Cardholder owns one or more Credit Cards, which are issued by a Bank and come with a defined Credit Limit. When the Cardholder uses the card at a Merchant for purchases, a Transaction is generated and validated by the Bank to ensure it stays within the Credit Limit. Each Transaction is linked to the respective Credit Card, and Payments made by the Cardholder reduce the outstanding balance, restoring the available credit. This system efficiently handles cardholder details, transaction validation, and payment processing while coordinating interactions between banks, merchants, and cardholders.

## State Diagram:

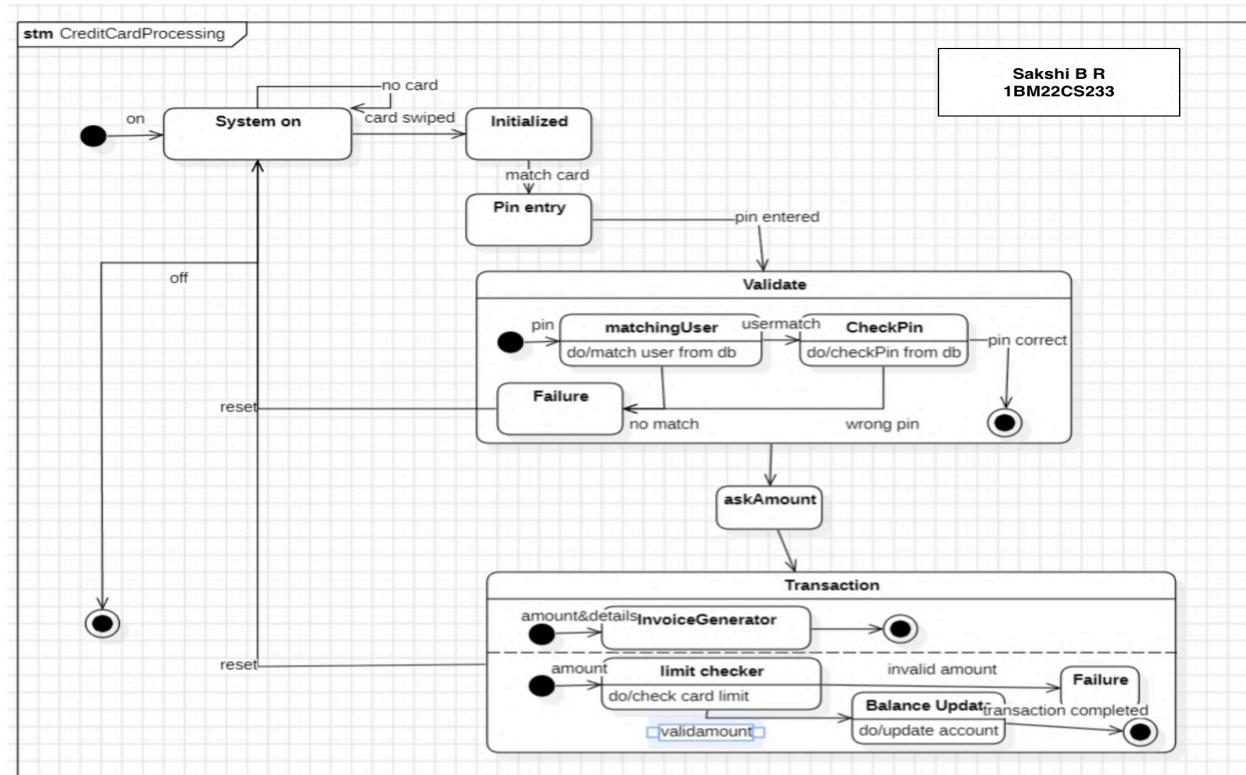


Fig 2.2

The state diagram for the Credit Card Processing System represents its key workflow. The system starts in an Idle state, transitioning to the Transaction Processing state when a transaction is initiated. The transaction is then verified for authenticity, credit limit availability, and fraud detection. If approved, the system updates the card balance and transitions to the Payment state, where the cardholder can make payments to settle outstanding balances. Payments restore the available credit, and the system returns to the Idle state, ready for new transactions. If the transaction is declined, the system generates a notification and resets to Idle.

## Use Case Diagram:

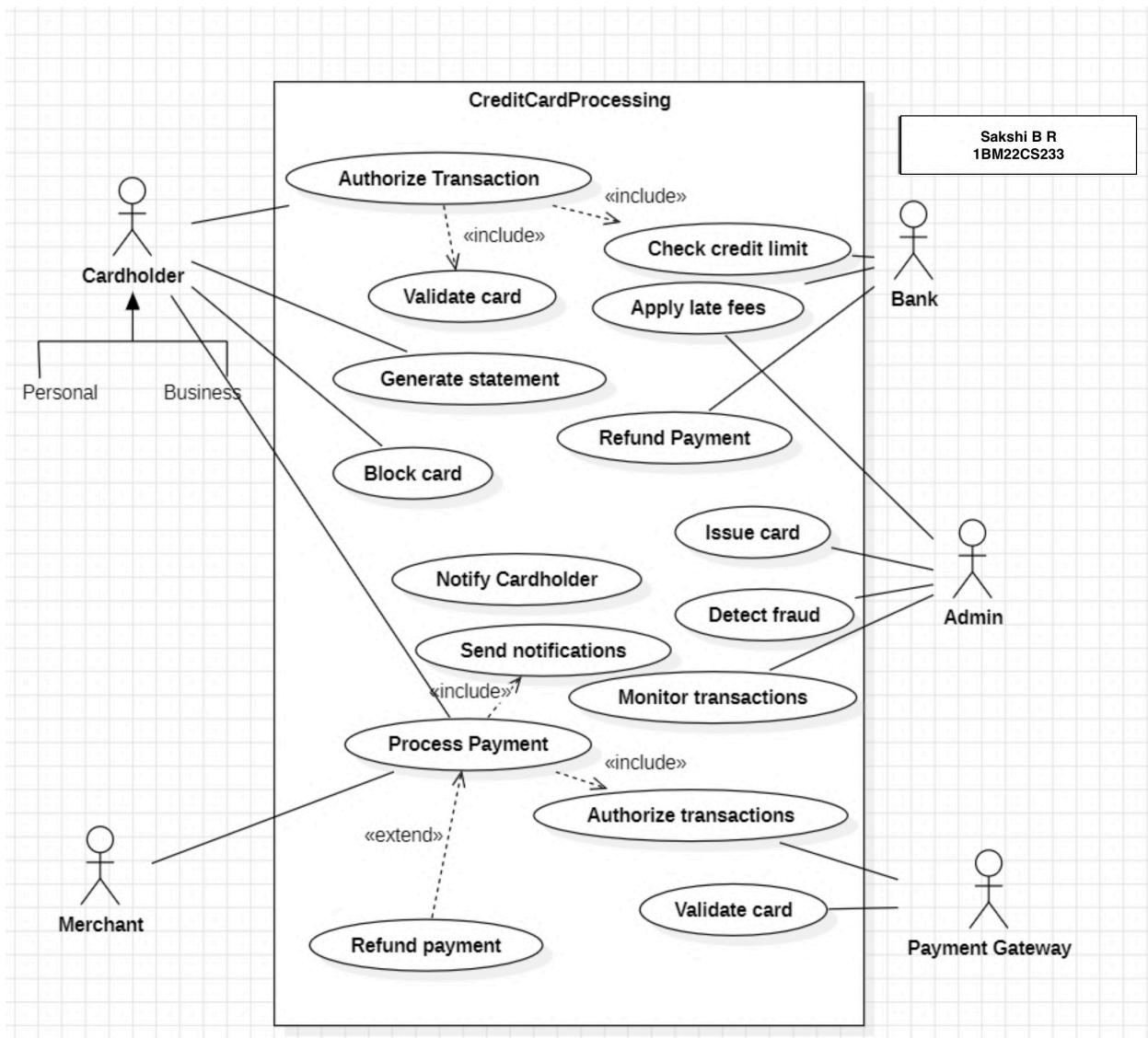
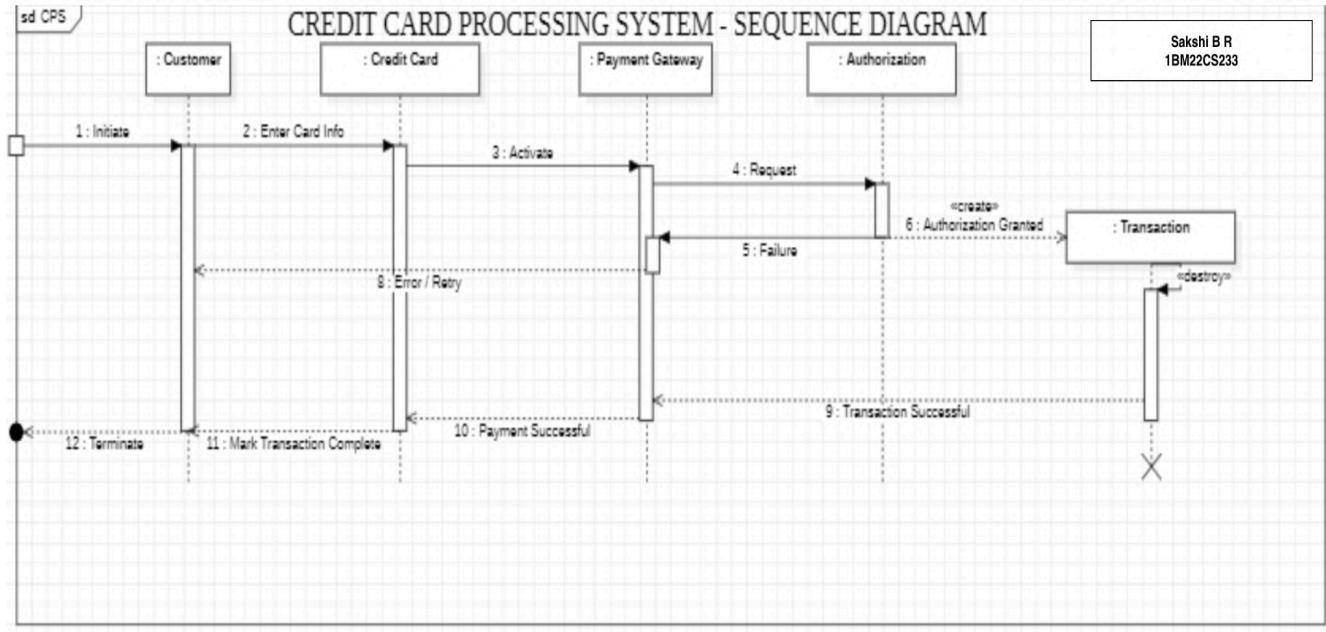


Fig 2.3

The Use Case Diagram illustrates the various functionalities of a credit card processing system. It outlines the interactions between different actors, including Cardholder, Merchant, Bank, Admin, and Payment Gateway. Use cases like Authorize Transaction, Process Payment, Generate Statement, Refund Payment, and Notify Cardholder are defined. Additionally, relationships like "include" and "extend" are used to represent optional or alternative behaviours, such as checking credit limits, applying late fees, and detecting fraud. This diagram provides a high-level overview of the system's capabilities and the interactions involved in credit card management.

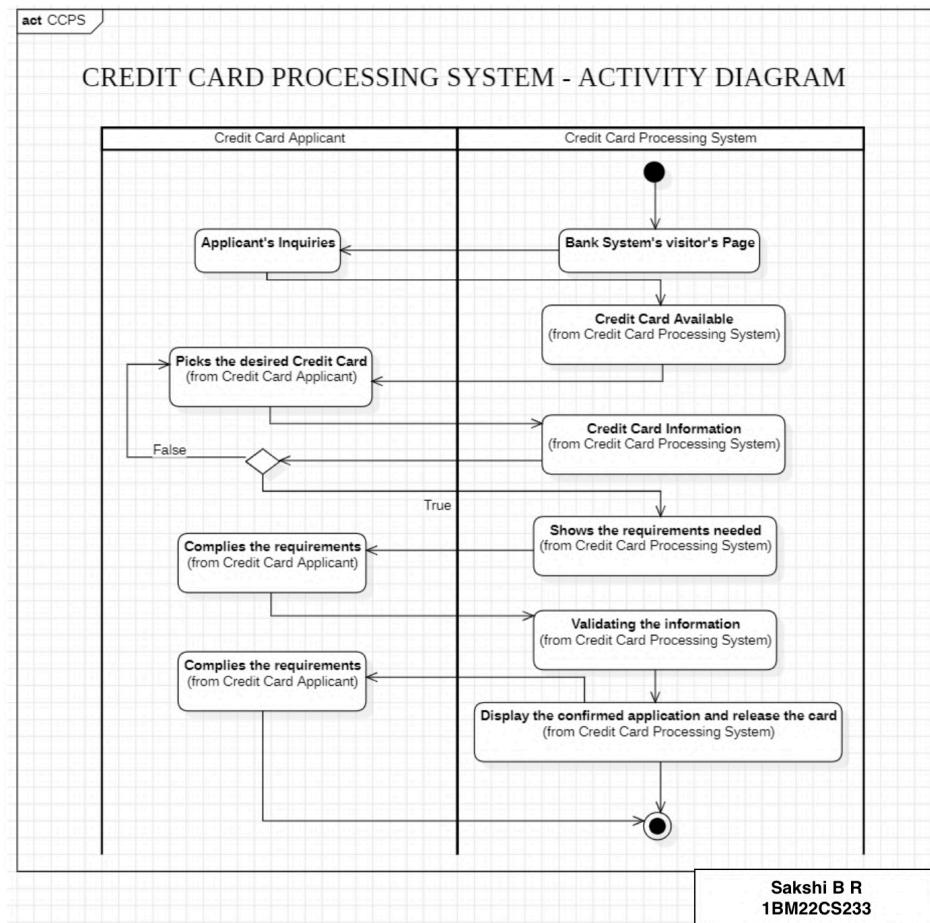
## Sequence Diagram:



**Fig 2.4**

The sequence diagram models the credit card transaction process. It starts with the customer initiating a purchase. The merchant then processes the payment and sends a request to the Payment Gateway for authorization. The Gateway forwards this to the Bank for approval. If authorized, the Bank informs the Gateway, and the transaction is confirmed. Both the customer and merchant are notified. However, if the Bank declines the authorization, the transaction is rejected, and notifications are sent accordingly. This diagram provides a step-by-step visualization of the interactions between different entities involved in a credit card transaction.

## Activity Diagram:



**Fig 2.5**

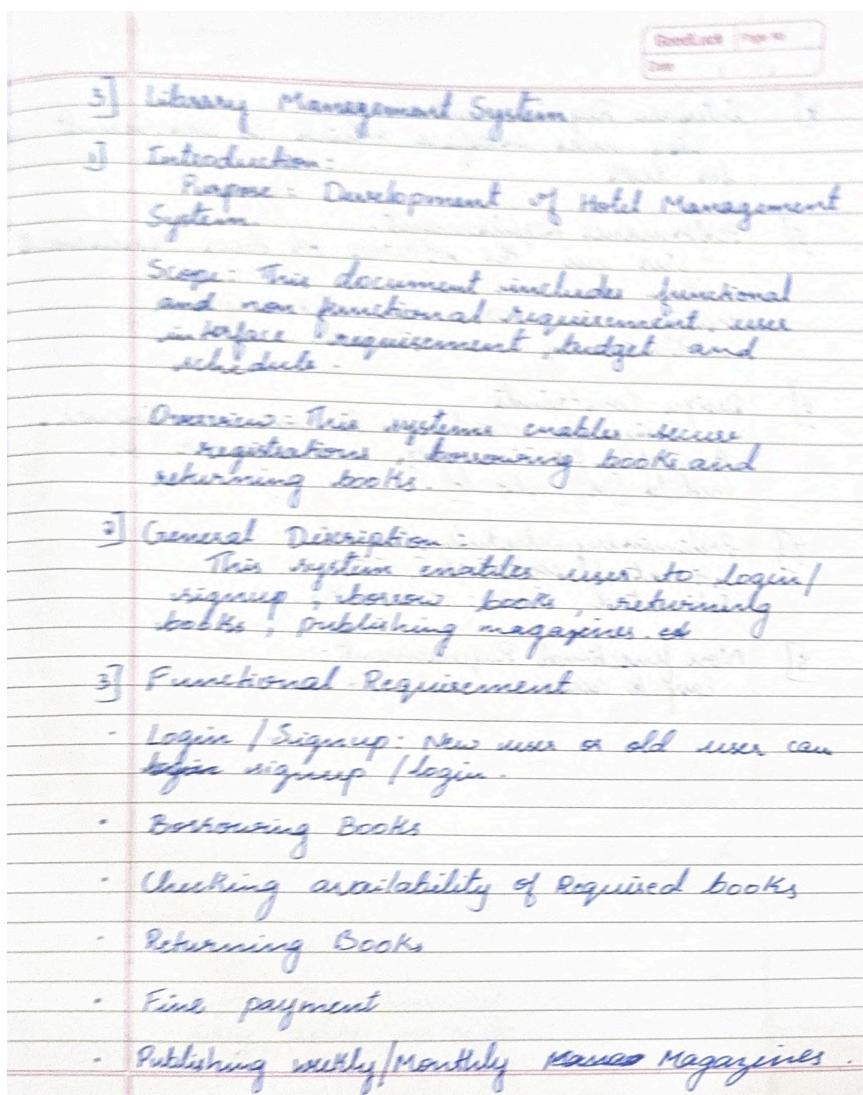
The Activity Diagram illustrates the process of a credit card transaction. It begins with the Customer initiating the transaction by entering payment details. The Merchant then requests authorization from the Payment Gateway. The Gateway sends the request to the Issuing Bank, which validates the card. If the card is valid, the Bank performs fraud detection and balance verification. Upon successful validation, the Bank approves the transaction, and the Gateway processes the payment. The Merchant then prints a receipt and sends a confirmation to the Customer. Finally, a signal is sent to the Notification System to complete the transaction.

### 3. Library Management System

#### Problem Statement:

The current manual system for managing library operations, including book cataloging, borrowing, returning, and tracking inventory, is inefficient and prone to errors. Challenges such as misplaced records, overdue books, and difficulty in locating resources lead to wasted time and reduced user satisfaction. There is a need for an automated library management system to streamline operations, maintain accurate records, improve resource accessibility, and enhance the overall user experience for both library staff and patrons.

#### SRS-Software Requirements Specification:



- Good Luck | Page No. \_\_\_\_\_  
Date \_\_\_\_\_
- 4] Interface Requirement: Easy user interface which is convenient for user.
- 5] Performance Requirement: User use the software for their requirement within less time multiple user can use the system without any inconvenience.
- 6] Design Constraints: Designed according to user convenience. Available for in any web browser or mobile (ios/android).
- 7] Preliminary Schedule / Budget  
Estimated cost: £60,65,595  
Estimated time: 10 months.
- 8] Non functional Requirement: Easy to access.

## Class Diagram:

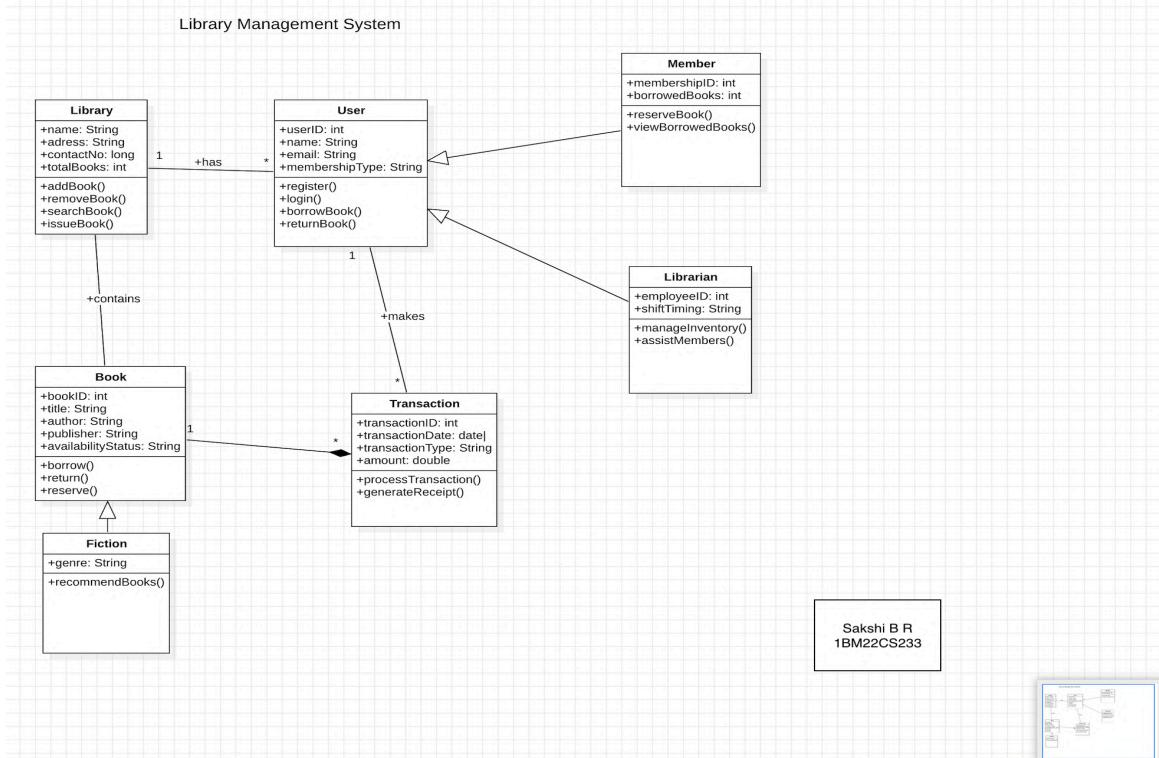
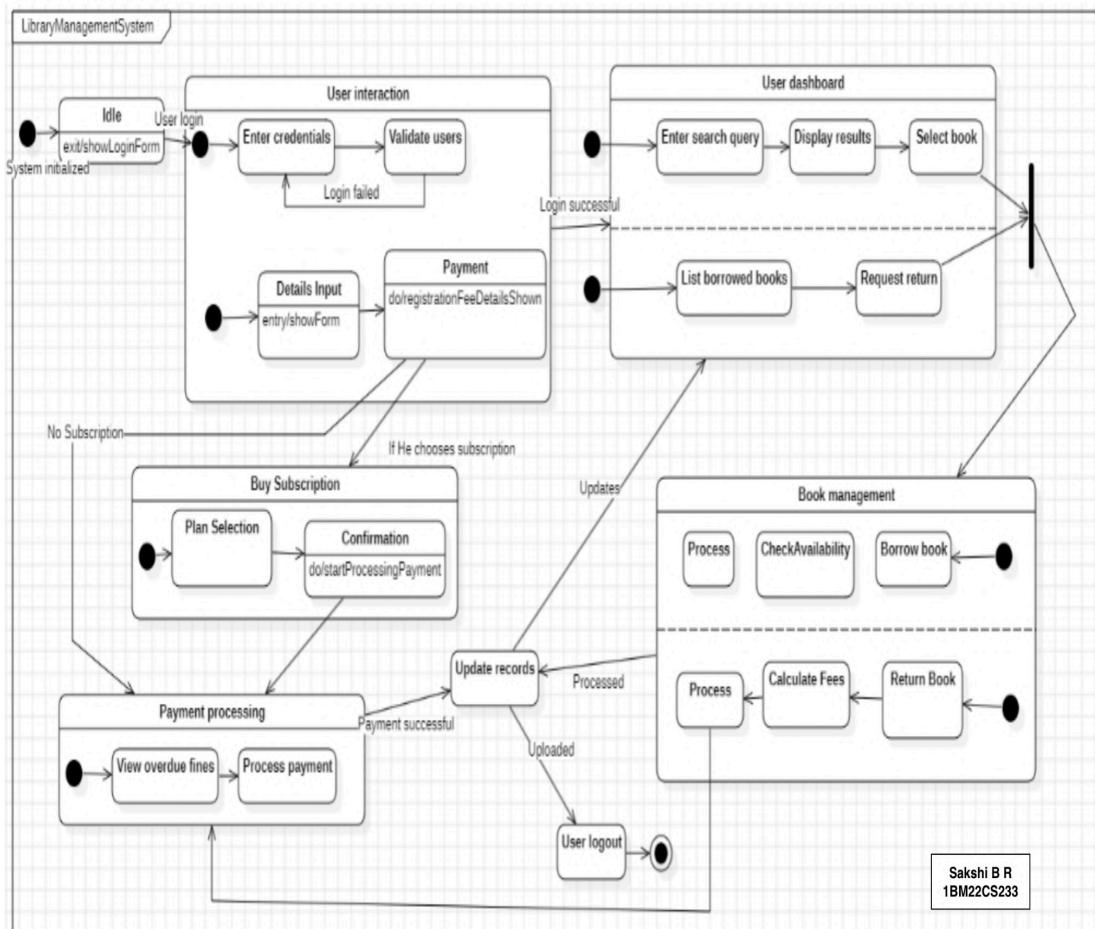


Fig 3.1

The Library Management System class diagram represents the key entities and their relationships, facilitating efficient library operations. The system includes entities such as Library, Book, Member, Librarian, Loan, and Fine. A Library manages a collection of Books, which can be borrowed by registered Members. Each Book has attributes like title, author, and ISBN, while Members have unique IDs and contact details. Loans track the borrowing process, linking Books to Members with information such as issue and return dates. If a Book is returned late, a Fine is calculated and linked to the respective Loan. The Librarian oversees the management of Books, Members, and Loans, ensuring smooth operations. This system effectively organizes borrowing, returns, and penalties, making library management seamless and efficient.

### State Diagram:



**Fig 3.2**

The state diagram for the Library Management System showcases the process flow. The system starts in an Idle state, transitioning to the Book Search state when a user searches for a book. If the book is available, the system moves to the Borrowing state, where the loan is processed, and the book is assigned to the member. During the Loan state, the book remains borrowed until returned. Upon return, the system transitions to the Returning state, where late returns trigger Fine Calculation if applicable. The book undergoes inventory checks and updates before the system returns to the Idle state, making it available for borrowing again.

### Use Case Diagram:

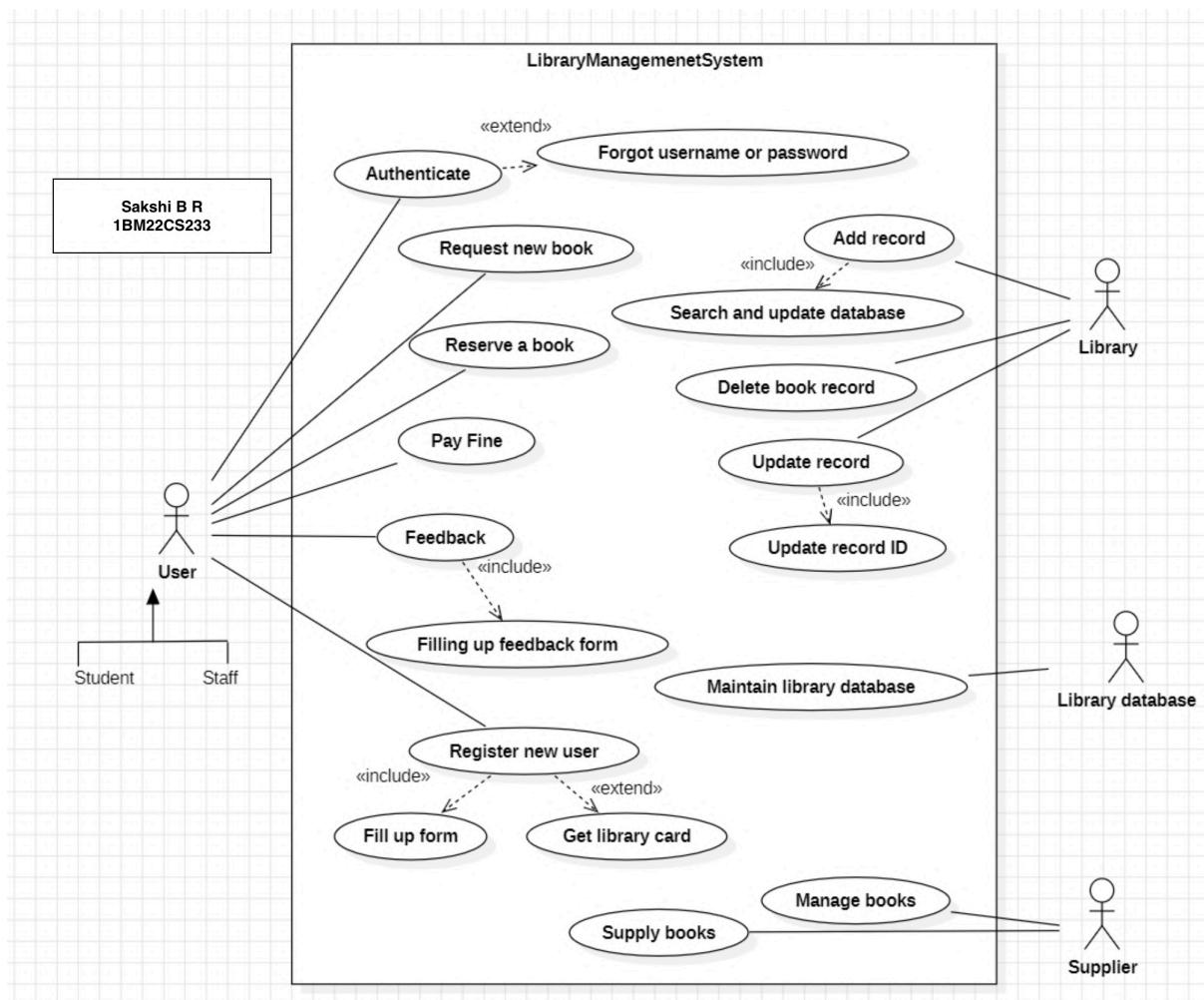


Fig 3.3

The Use Case Diagram illustrates the functionalities of a Library Management System. It shows various use cases like Authenticate, request new book, reserve a book, Pay Fine, Feedback, register new user, and Manage books. The diagram also includes relationships like “include” and “extend” to represent optional or alternative behaviours. For example, “Forgot username or password” extends the “Authenticate” use case. The system interacts with different actors such as User (including Student and Staff), Library, Library database, and Supplier. This diagram provides a high-level overview of the system’s capabilities and the interactions between different entities in the library management context.

### Sequence Diagram:

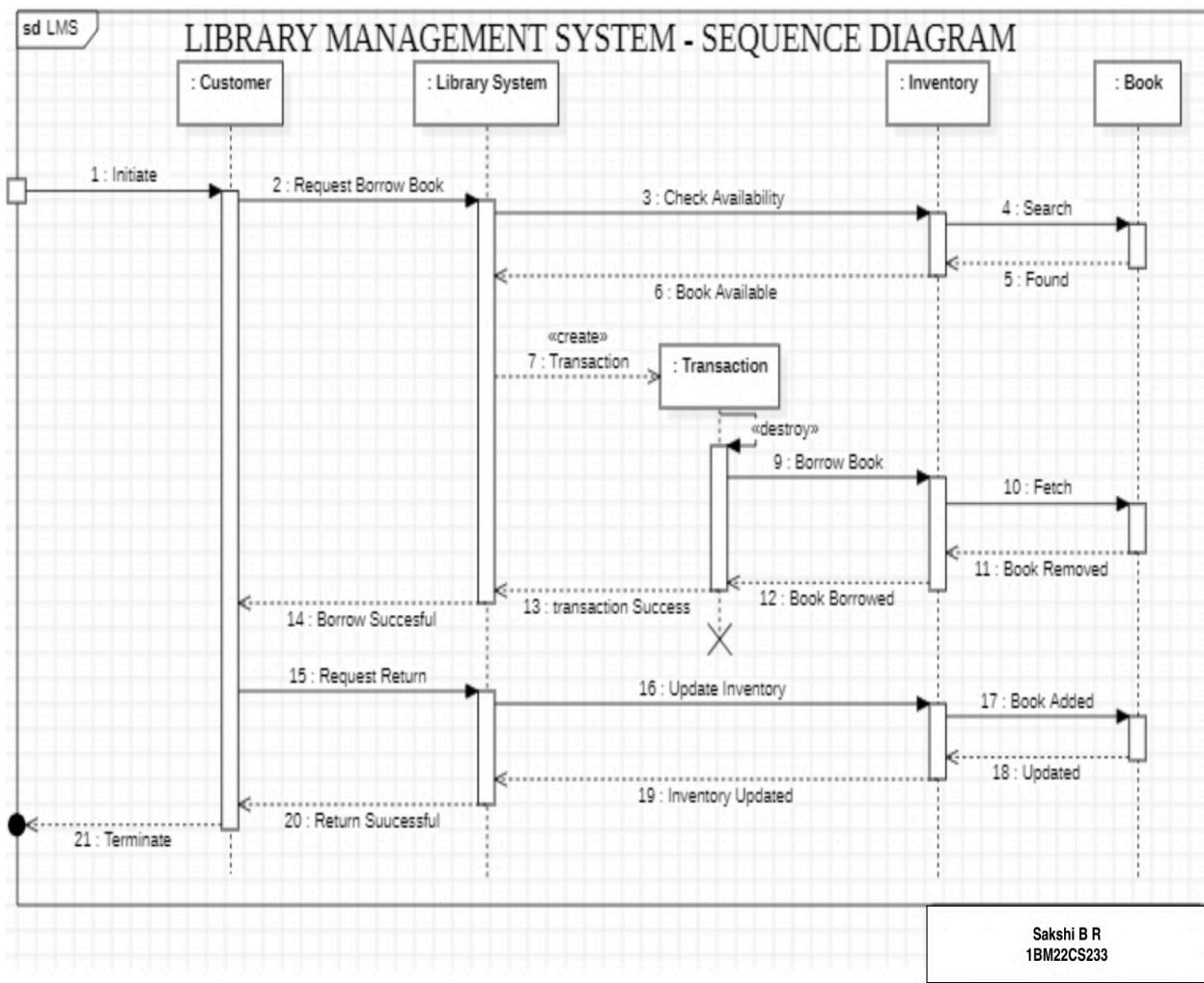
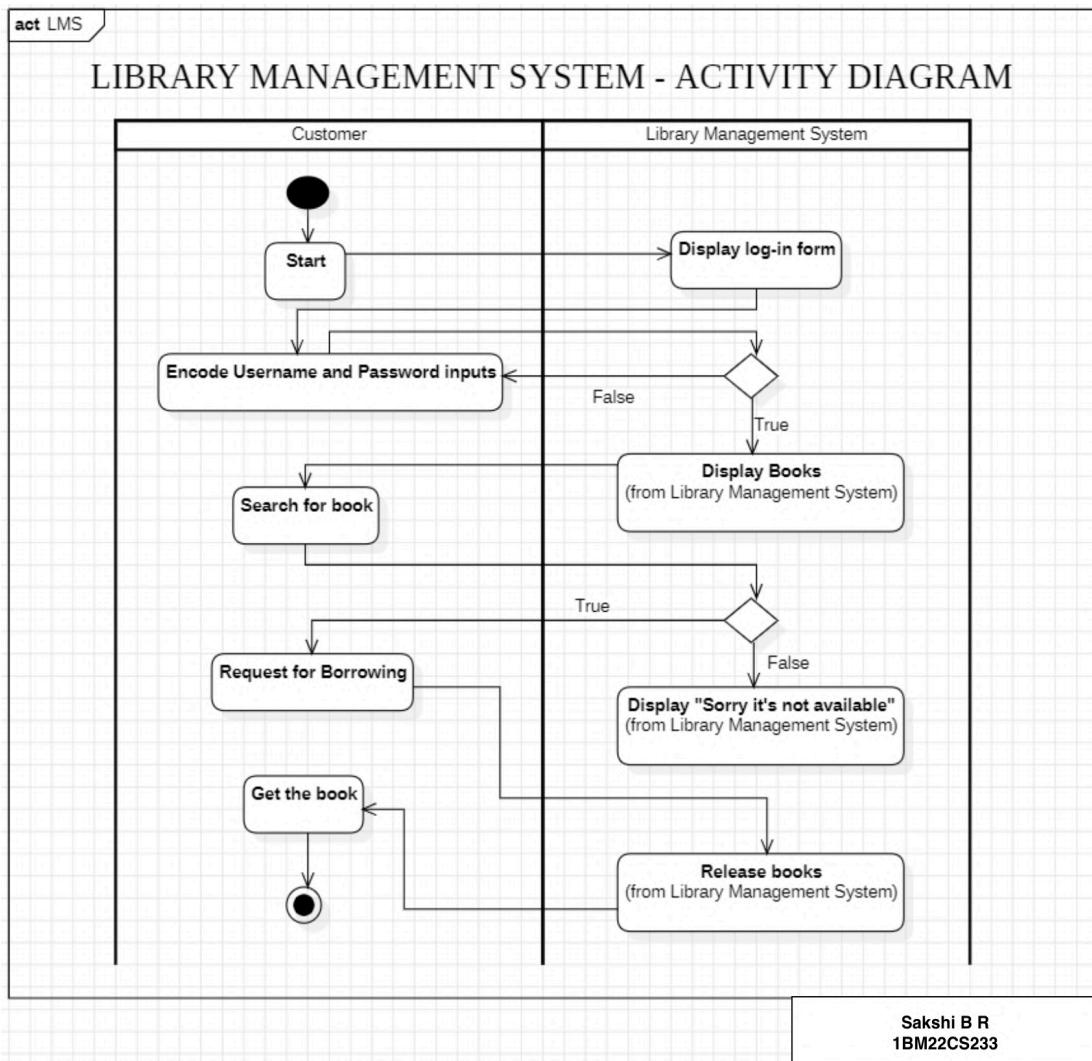


Fig 3.4

The sequence diagram illustrates the interaction between a User, the Library Management System (LMS), Database, and Library Inventory during a book search and borrowing process. The User initiates by searching for a book. The LMS queries the Database for book details, which then checks the Library Inventory for availability. Details are sent back to the User, who can then request to borrow the book. The LMS updates the Inventory accordingly and confirms the borrowing. Finally, the User can request to return the book, triggering another inventory update. This diagram provides a step-by-step visualization of the interactions involved in a book borrowing transaction within the Library Management System.

### Activity Diagram:



**Fig 3.5**

The Activity Diagram illustrates the workflow of a Library Management System. It starts with the User searching for a book. The Librarian checks the Book Database for availability. If available, the book is issued to the User, and the stock is deducted. A borrow time period is initiated. When the User returns the book, the Librarian checks the return date. If overdue, a fine is imposed. The stock is updated, and a return confirmation is sent. This diagram visually represents the key steps and decisions involved in book borrowing and returning within the library system.

## 4. Stock Maintenance System

### Problem Statement:

Manual stock maintenance in businesses often leads to inaccuracies, overstocking, or understocking, impacting operational efficiency and customer satisfaction. An automated stock maintenance system is needed to track inventory levels, manage reorders, reduce wastage, and optimize supply chain processes.

### SRS-Software Requirements Specification:

| 7] Stock Maintenance System . |  |
|-------------------------------|--|
| 1.                            | Introduction<br>Purpose : The aim of this document is to define the purpose, scope and requirement of the Stock Maintenance System .   |
|                               | Scope : This document outlines the functionalities like stock monitoring, order management and reporting .   |
|                               | Overview : This document consists of functional requirement, non-functional requirement, user interface and budget .   |
| 2.                            | General Description : This system consists of features including real-time tracking of stock levels, automated alerts for low stock etc.   |
| 3.                            | Functional Requirements<br><ul style="list-style-type: none"><li>• User Authentication - User must log in with unique credentials ensuring secure access</li><li>• Stock Management</li><li>• Stock Alerts</li><li>• Reporting</li></ul> |

4. Interface Requirement  
Web-based dashboard for stock monitoring,  
management  
easy user interface

5. Performance Requirements  
The system must handle and support  
multiple users with less response time

6. Design Constraints  
Designed in such a way that user  
access through any web browser.

7. Non-Functional Requirements  
Security : Sensitive data is to be  
secured.  
The system should be accessible  
across multiple platforms.

8. Preliminary Schedule and Budget  
Estimated time is 6 months  
Estimated budget : ₹ 5,00,000.

Thank you

Study well

Picture

## Class Diagram:

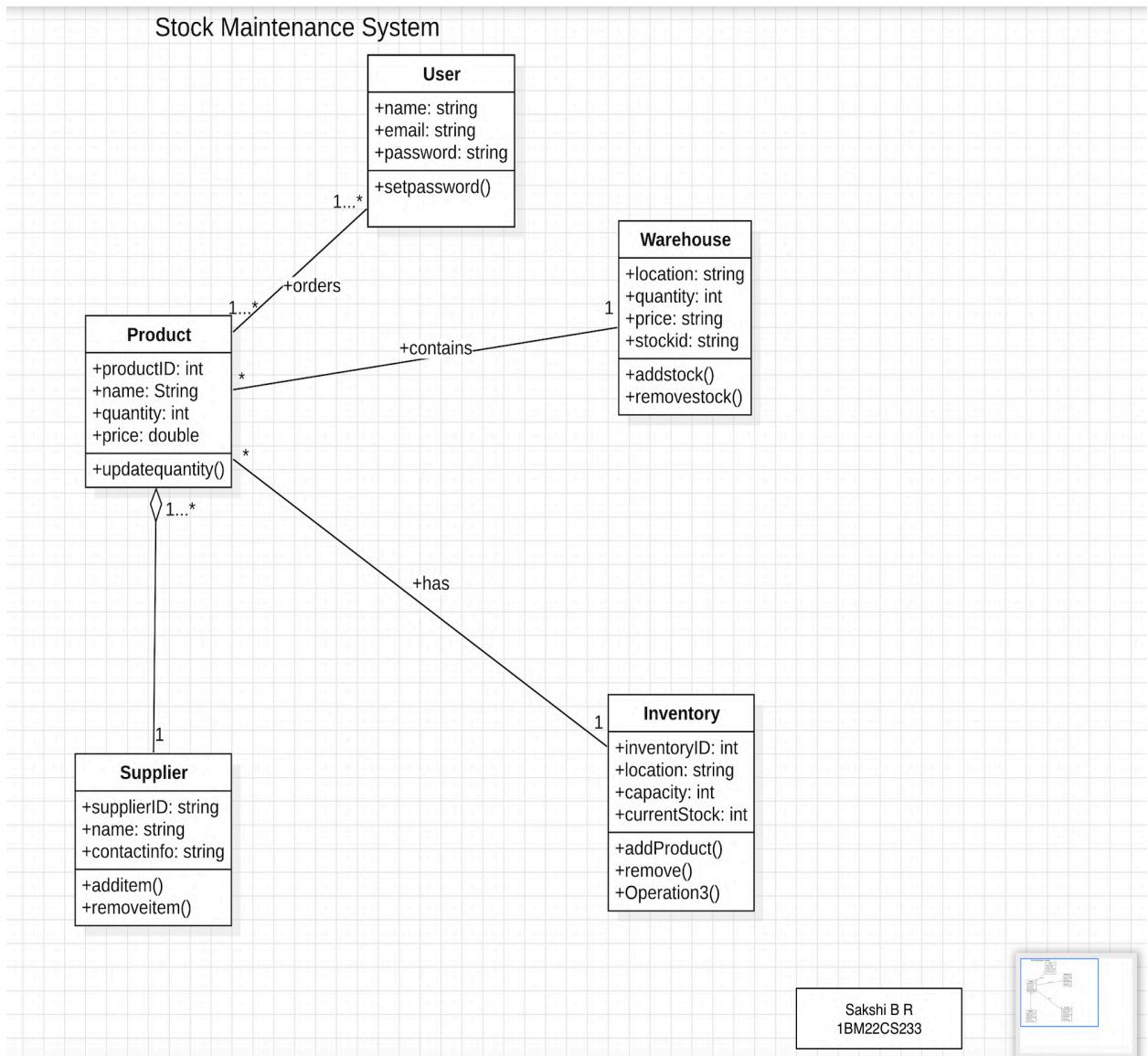


Fig 4.1

The Stock Maintenance System class diagram models the essential entities and their relationships to ensure efficient inventory management. The system includes entities such as Product, Supplier, Stock, Warehouse, Order, and Manager. Products, identified by attributes like name, ID, and category, are supplied by Suppliers and stored in Warehouses. Stock tracks the quantity of each Product available in the system, while Orders represent requests to replenish stock or fulfill customer demand. The Manager oversees the system, ensuring stock levels are maintained and orders are processed in a timely manner. By organizing the flow of products, supplier

Sakshi B R  
1BM22CS233

relationships, stock levels, and order management, the system ensures seamless inventory tracking and efficient stock maintenance.

## State Diagram:

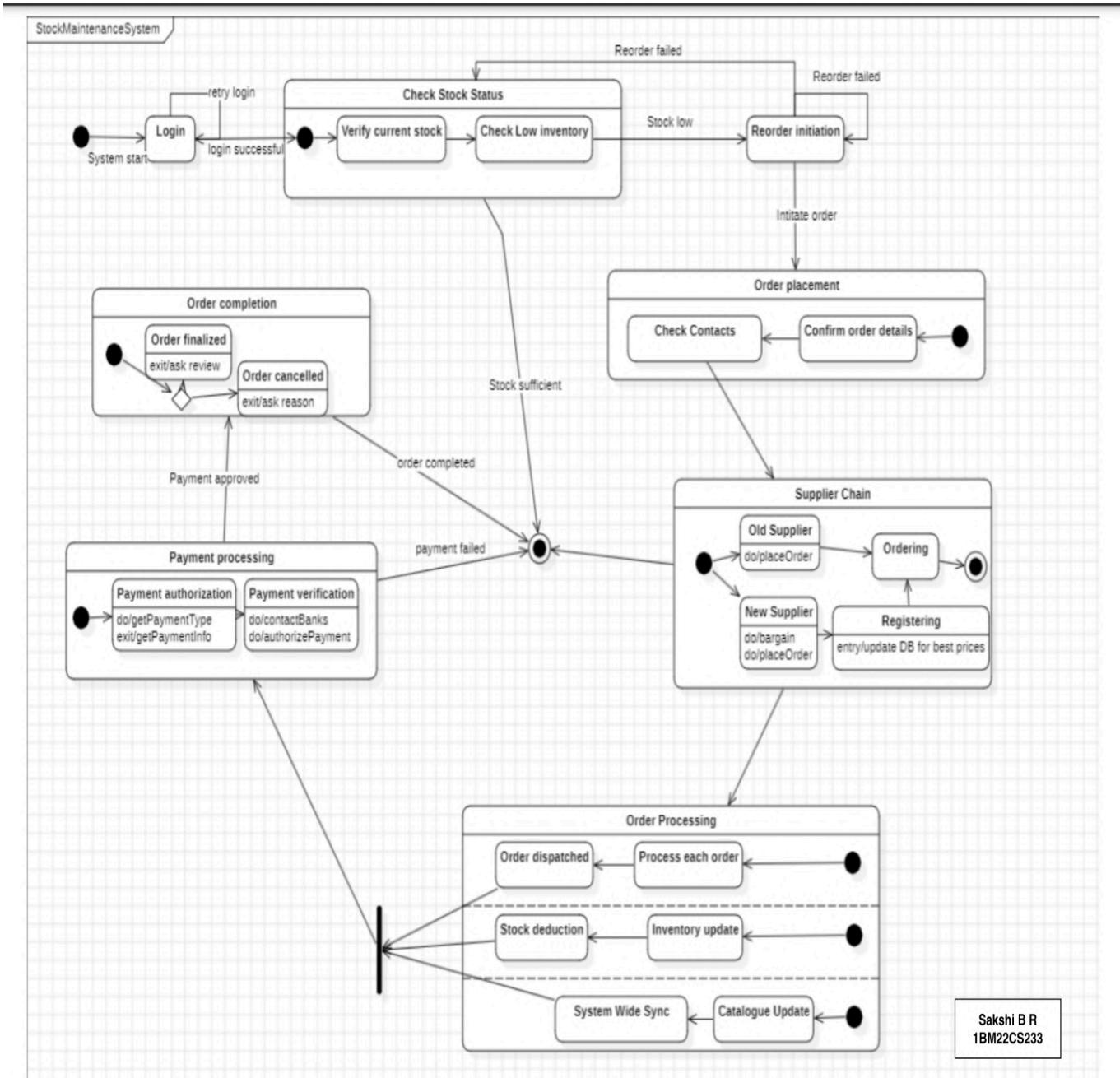
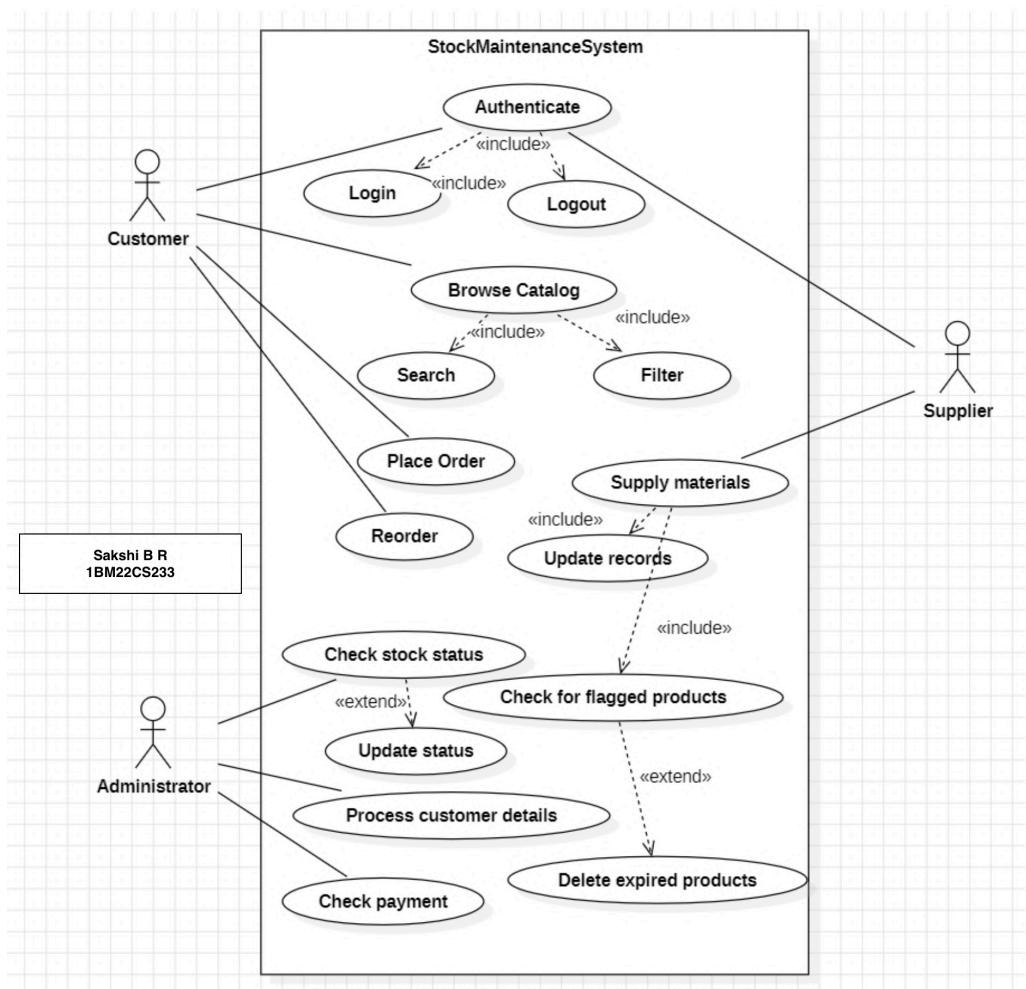


Fig 4.2

The state diagram for the Stock Maintenance System illustrates the inventory workflow. The system begins in an Idle state, transitioning to the Order Placement state when stock is requested or replenishment is required. After an order is placed, the system moves to the Inventory Update state, where incoming stock is verified and quantities are adjusted. Products are then assigned to their designated locations in the Warehouse state. When stock levels are sufficient, the system returns to the Idle state. For low stock alerts, the system transitions to a Reorder state to ensure timely replenishment, maintaining seamless inventory flow.

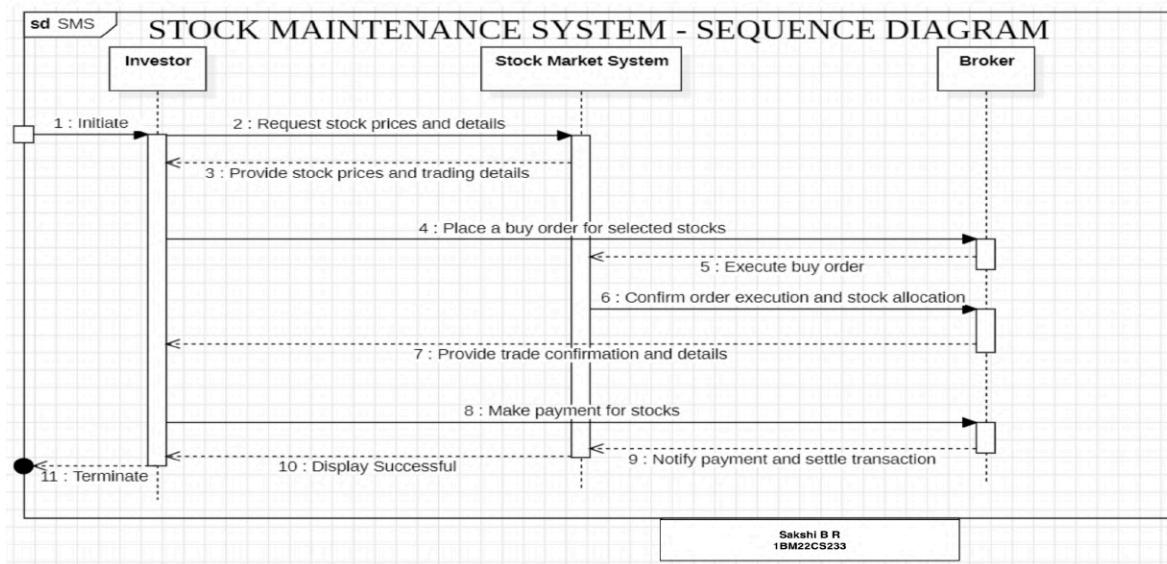
## Use-Case Diagram:



**Fig 4.3**

The Use Case Diagram illustrates the functionalities of a Stock Maintenance System. It shows various use cases like Authenticate, Browse Catalog, Place Order, Supply materials, Update records, and Check stock status. The diagram also includes relationships like "include" and "extend" to represent optional or alternative behaviours. For example, "Search" and "Filter" are included in "Browse Catalog". The system interacts with different actors such as Customer, Supplier, and Administrator. This diagram provides a high-level overview of the system's capabilities and the interactions between different entities in the stock management context.

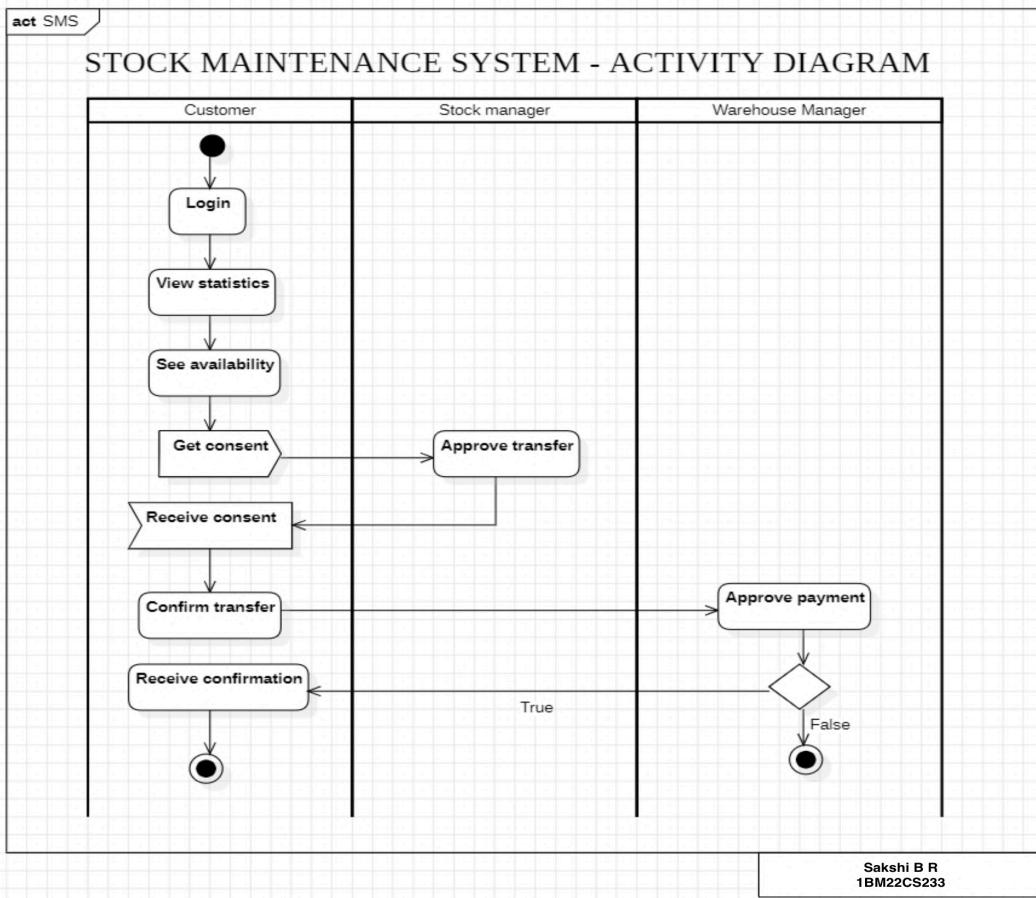
### Sequence Diagram:



**Fig 4.4**

The sequence diagram illustrates the interactions between different entities in a stock management system. It begins with the Customer browsing the Catalog. The Supplier then supplies materials, and the Stock System updates its status. The Customer can place orders, and the Stock System confirms the order. The Customer can also request notifications when an item becomes available. Finally, the Admin can generate and review reports on stock levels. This diagram provides a stepby-step visualization of the interactions involved in managing stock within the system.

## Activity Diagram:



**Fig 4.5**

The Activity Diagram illustrates the workflow of a Stock Maintenance System. It starts with the Customer placing an order. The Inventory Manager checks the Stock Database for availability. If the stock is not available, an order is placed with the Supplier. Once the Supplier sends the order and stock, the Inventory Manager receives it and deducts the stock on sale. The Sales System processes the order and completes the sale. Finally, the Inventory Manager updates the stock records and sends an order complete signal. This diagram visually represents the key steps and interactions involved in managing stock and fulfilling orders within the system.

## 5. Passport Automation System

### Problem Statement:

The manual process of issuing and renewing passports is slow, cumbersome, and prone to errors, causing delays for applicants and inefficiencies in data management. A passport automation system is required to digitize and streamline the process, ensure data accuracy, enhance security, and provide real-time status tracking for users.

### SRS-Software Requirements Specification:

| 6] Passport Automation System: |  |
|--------------------------------|--|
| 1.                             | Introduction<br>Purpose : This document outlines the development of passport Automation System.  |
|                                | Scope : This document consists of functional requirement, non functional requirement, user interface, budget   |
|                                | Overview : This document outlines of everything from user registration to authentication.  |
| 2)                             | General Description :<br>This system ensures secure registration of user, user authenticates authentication, secure maintenance of user personal information.                              |
| 3]                             | Function Requirements : <ul style="list-style-type: none"><li>Registration of user</li><li>Application of passport</li><li>Verification of documents</li><li>User authentication</li></ul> |
| 4]                             | Interface Requirements<br>User interface which is convenient for user to use   |
| 5]                             | Performance Requirement<br>Multiple user can user without any  |

any inconvenience

6] Design constraints  
Designed such that it is available  
in any web browser

7] Non-functional:  
Easy accessibility

8] Preliminary Schedule

Estimated budget: ₹ 50,00,000

Estimated Time: 5 months

## Class Diagram:

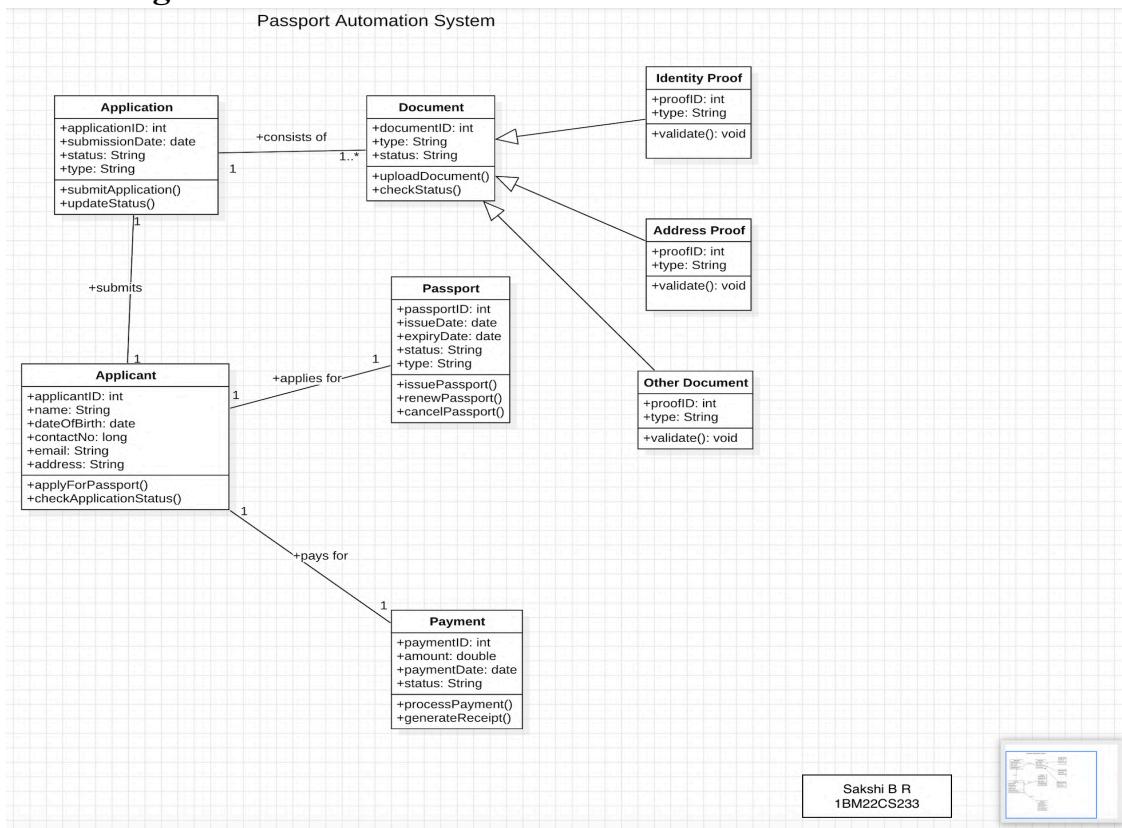
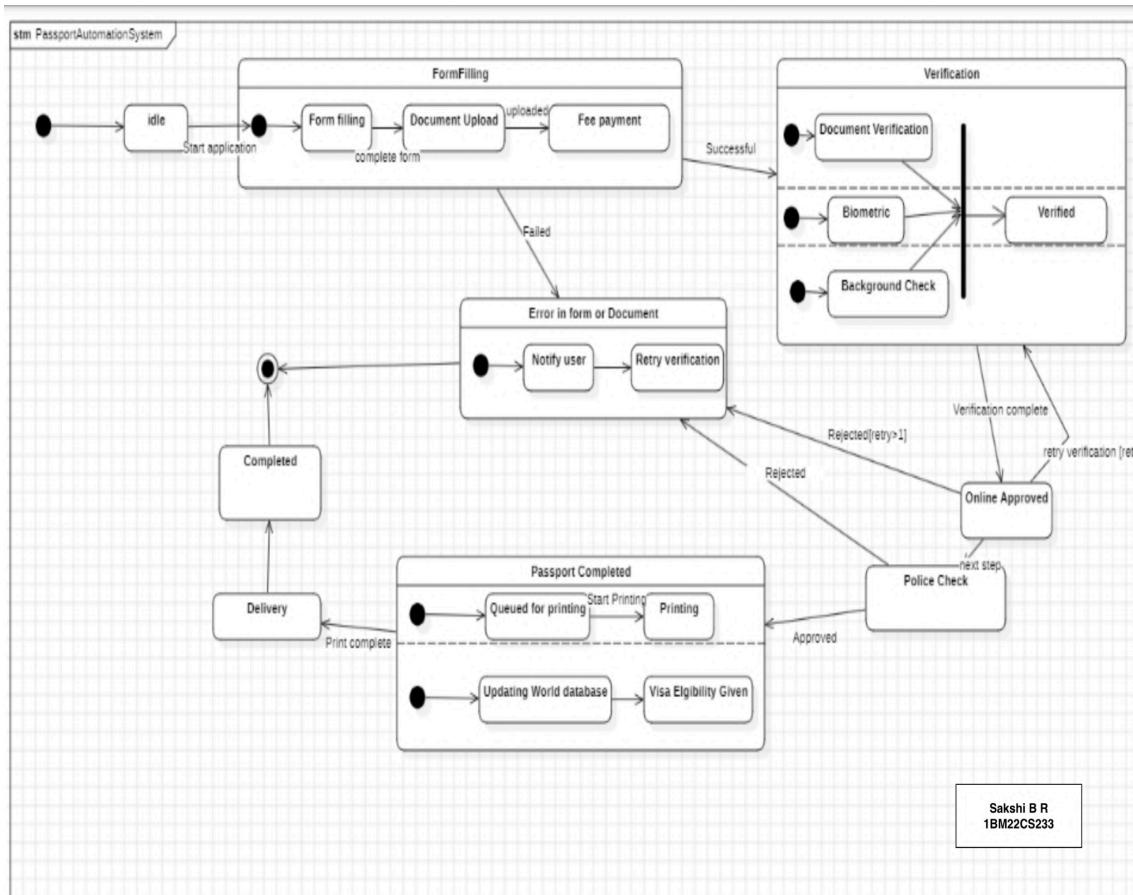


Fig 5.1

The Passport Automation System class diagram highlights the key entities and their relationships to streamline passport issuance and management. The system includes entities such as Applicant, Passport, Application, Verification, Officer, and Payment. An Applicant submits an Application containing personal details, which undergoes a Verification process to ensure the accuracy and validity of the provided information. Verification is managed by Officers, who approve or reject applications based on the results. Once approved, a Passport is generated and linked to the Applicant. The system also handles Payments associated with the application process. By organizing the workflow of application submission, verification, payment, and passport issuance, the system ensures a smooth and efficient process for managing passport-related services.

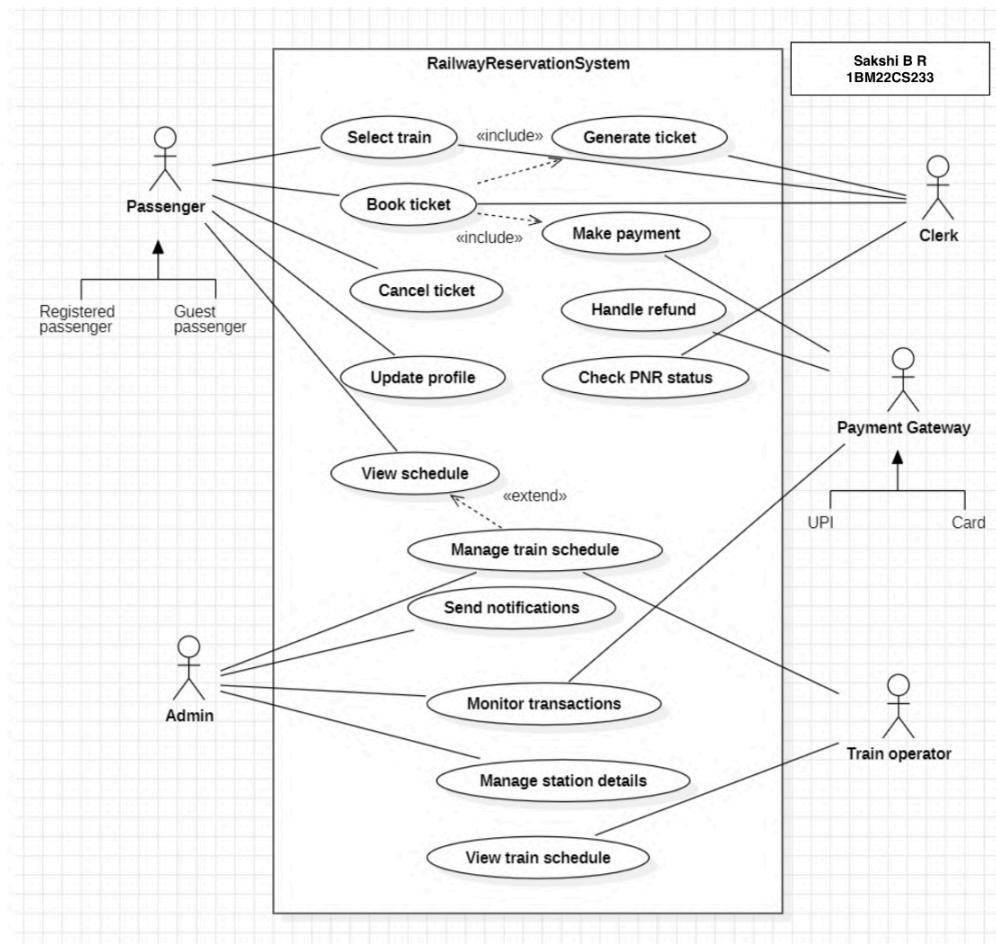
### State Diagram:



**Fig 5.2**

The state diagram for the Passport Automation System outlines the streamlined workflow. The system starts in an Idle state, transitioning to the Application Submission state when an applicant submits their details. The application moves to the Verification state, where documents and information are checked for accuracy. If verified, the system transitions to Passport Issuance, where the passport is generated and linked to the applicant. Payments are processed during the Payment state, and once complete, the system sends the passport to the applicant. Rejected applications return to Idle, while approved ones complete the cycle with issuance.

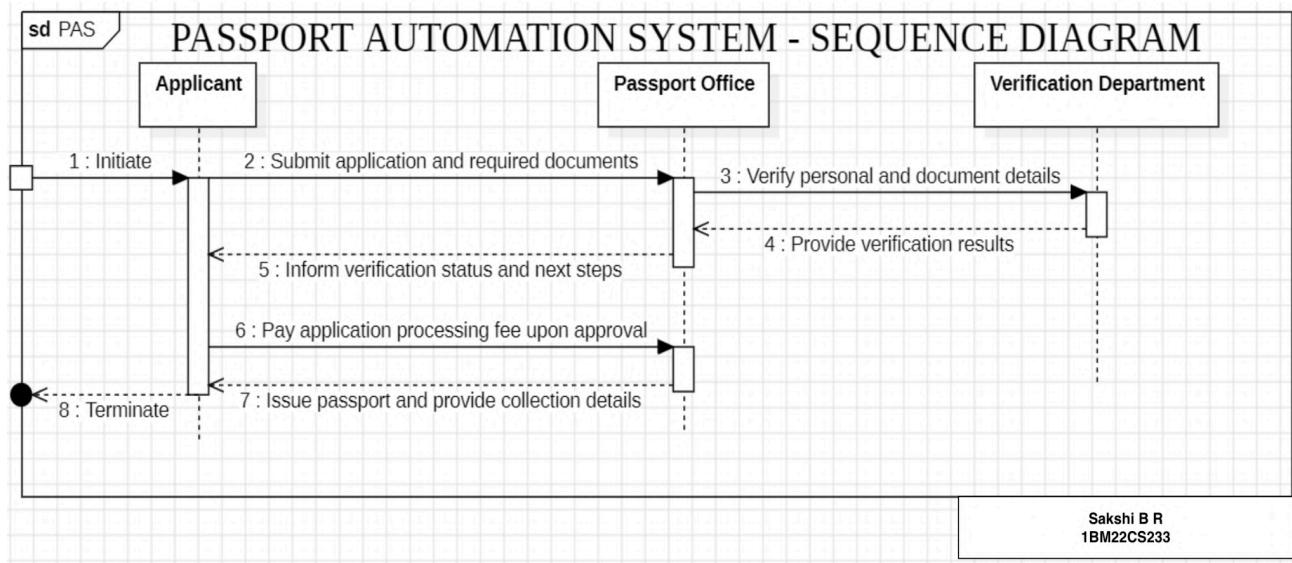
### Use-Case Diagram:



**Fig 5.3**

The Use Case Diagram illustrates the functionalities of a Passport Automation System. It shows various use cases like Apply for passport, Schedule appointment, Make payment, Verify documents, Approve application, and Deliver Passport. The diagram also includes relationships like "include" and "extend" to represent optional or alternative behaviors. For example, "Upload documents" is included in "Apply for passport" and "Forgot username or password" extends "Authenticate". The system interacts with different actors such as Applicant, Verification Officer, Police, Courier, Payment Gateway, and Admin. This diagram provides a high-level overview of the system's capabilities and the interactions between different entities in the passport application process.

### Sequence Diagram:

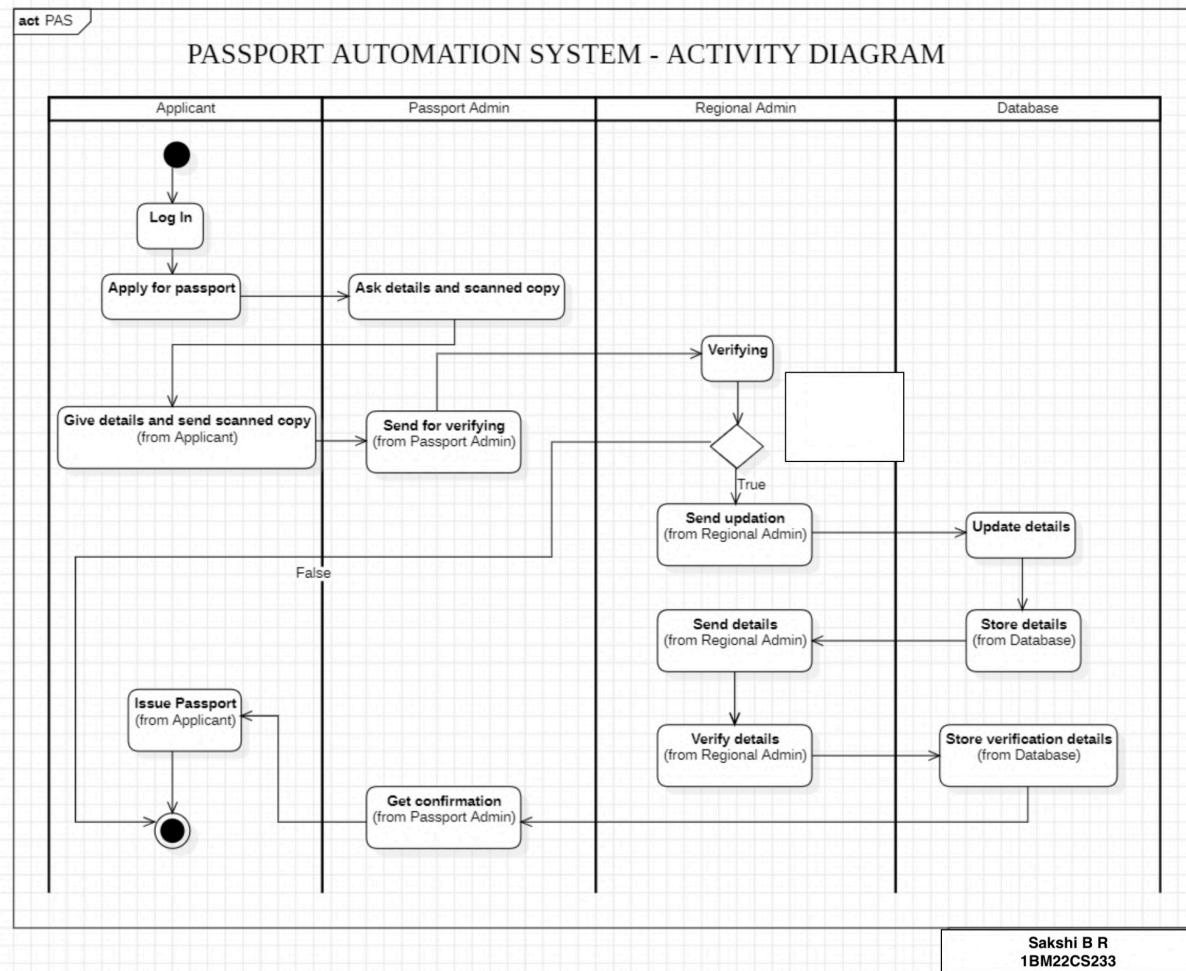


**Fig 5.4**

The sequence diagram illustrates the workflow of a passport application system. It starts with the Applicant applying for a passport and submitting documents to the System Database. The Verification Officer verifies the documents and schedules an appointment if successful. If verification fails, the Applicant is requested to submit additional documents. The Applicant attends the appointment, and the Verification Officer approves the application. The admin dispatches the passport, and the Courier Service delivers it to the Applicant. Finally, the System Database updates the application status, and the Applicant is notified. This diagram provides a

step-by-step visualization of the interactions between different entities involved in the passport application process.

## Activity Diagram:



**Fig 5.5**

The Activity Diagram depicts the workflow of the passport application process, beginning with the Applicant submitting their application. The system then reviews the application and verifies the submitted documents. If any documents are missing, the Applicant is notified to provide the required details. A background check is conducted by the Police Department, and upon a successful clearance, the application moves forward, and the Passport Office issues the passport.

If the background check fails or the application is rejected, the process is terminated. This diagram effectively visualizes the key actions and decision points in the passport issuance process.