VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED MODELING

Submitted by

Sarthak Gupta
(1BM22CS246)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
September-2024 to January-2025

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 **Sarthak Gupta (1BM22CS246)** during the 5th Semester Oct 2024 - Jan2025.

Signature of the Faculty Incharge:

NAME OF THE FACULTY: Prof Lakshmi Neelima

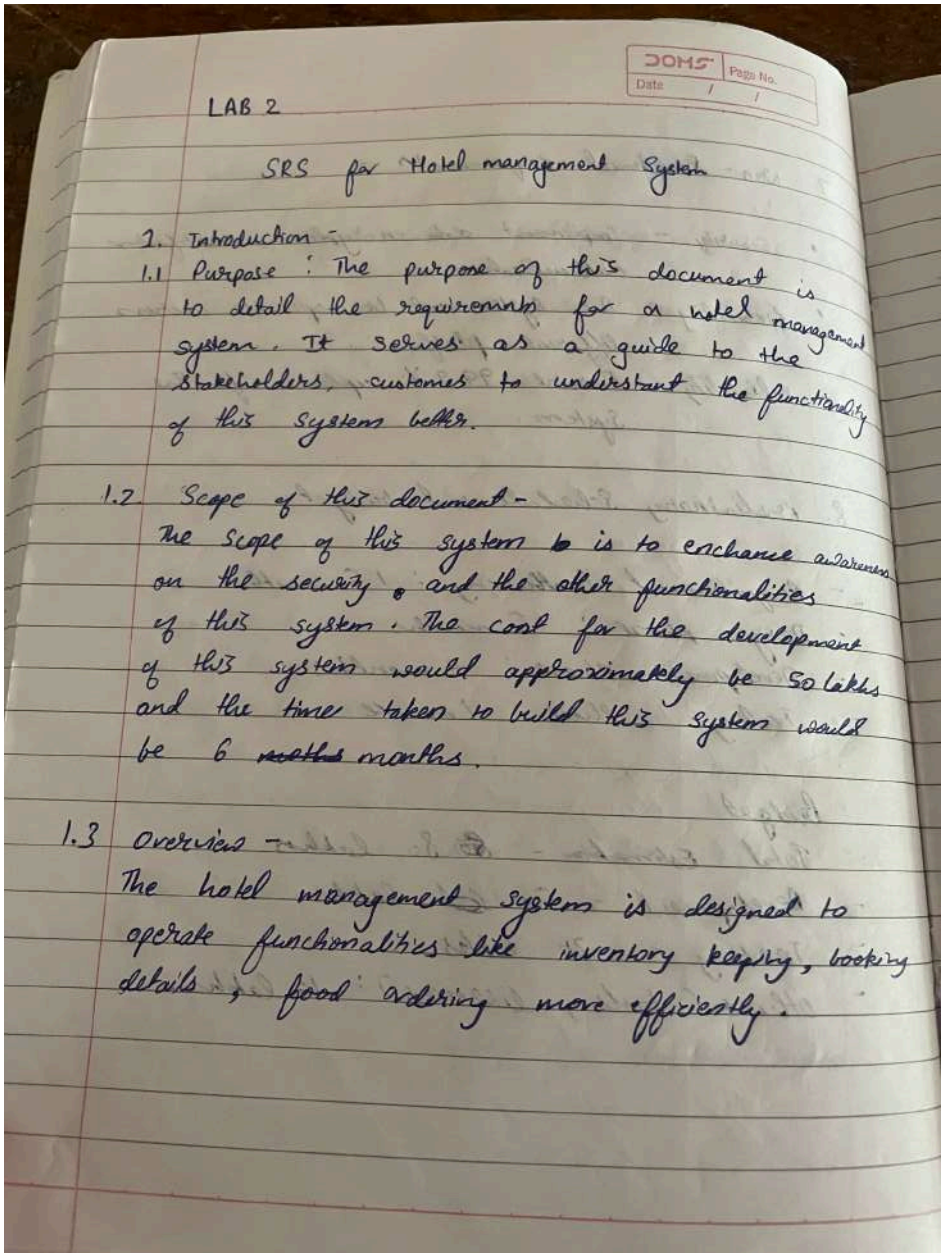
Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

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

a. SRS Document:



LAB 2

SRS for Hotel management System

1. Introduction -

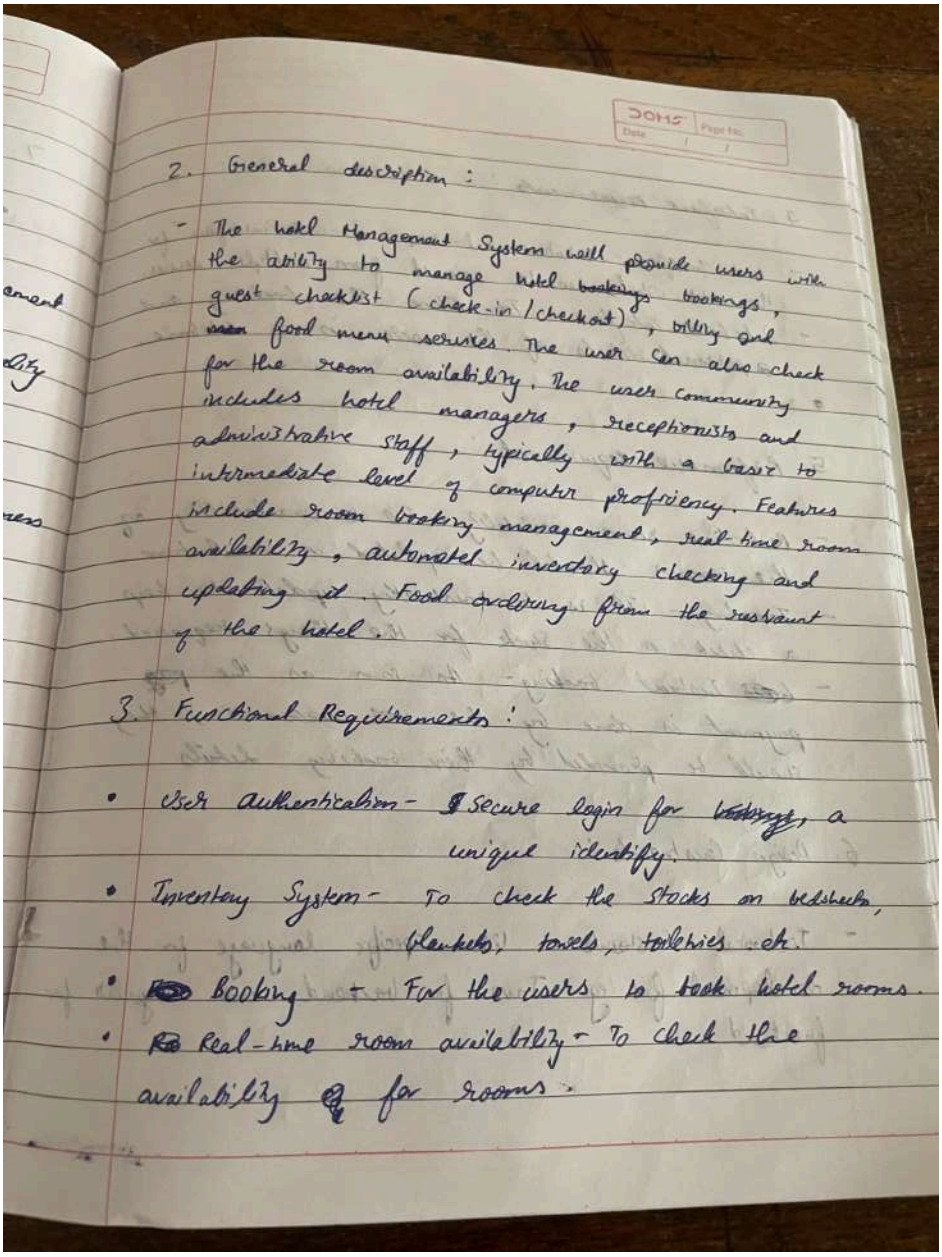
1.1 Purpose : The purpose of this document is to detail the requirements for a hotel management system. It serves as a guide to the stakeholders, customers to understand the functionality of this system better.

1.2 Scope of this document -

The scope of this system is to enchance awareness on the security, and the other functionalities of this system. The cost for the development of this system would approximately be 50 lakhs and the time taken to build this system would be 6 ~~months~~ months.

1.3 Overview -

The hotel management system is designed to operate functionalities like inventory keeping, booking details, food ordering more efficiently.

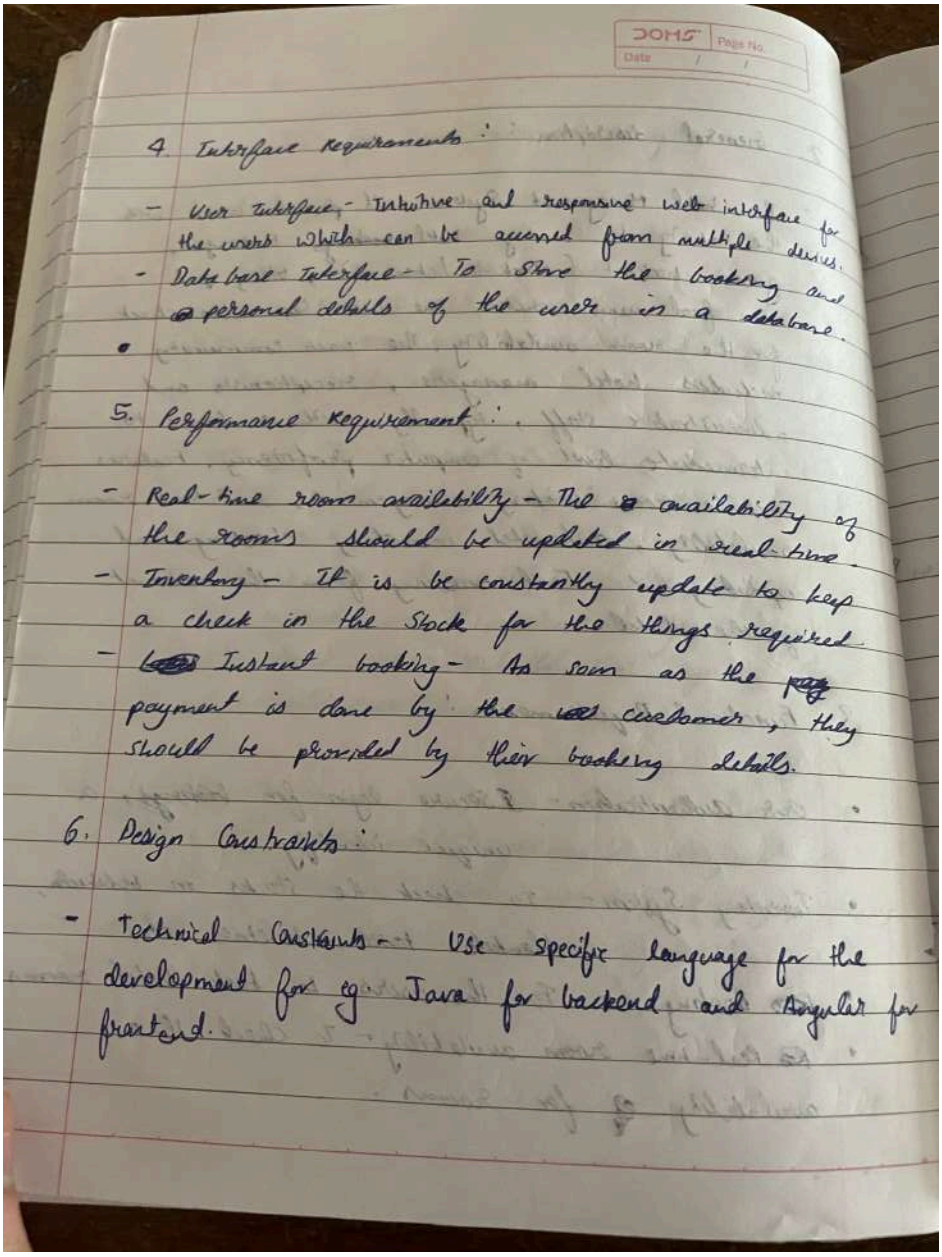


2. General description :

- The hotel Management System will provide users with the ability to manage hotel bookings, guest checklist (check-in / checkout), billing and food menu services. The user can also check for the room availability. The user community includes hotel managers, receptionists and administrative staff, typically with a basic to intermediate level of computer proficiency. Features include room booking management, real time room availability, automated inventory checking and updating it. Food ordering from the restraunt of the hotel.

3. Functional Requirements :

- User authentication - Secure login for ~~[illegible]~~, a unique identity.
- Inventory System - To check the stocks on bedsheets, blankets, towels, toiletries etc.
- Booking - For the users to book hotel rooms.
- Real-time room availability - To check the availability of for rooms.



4. Interface requirements:

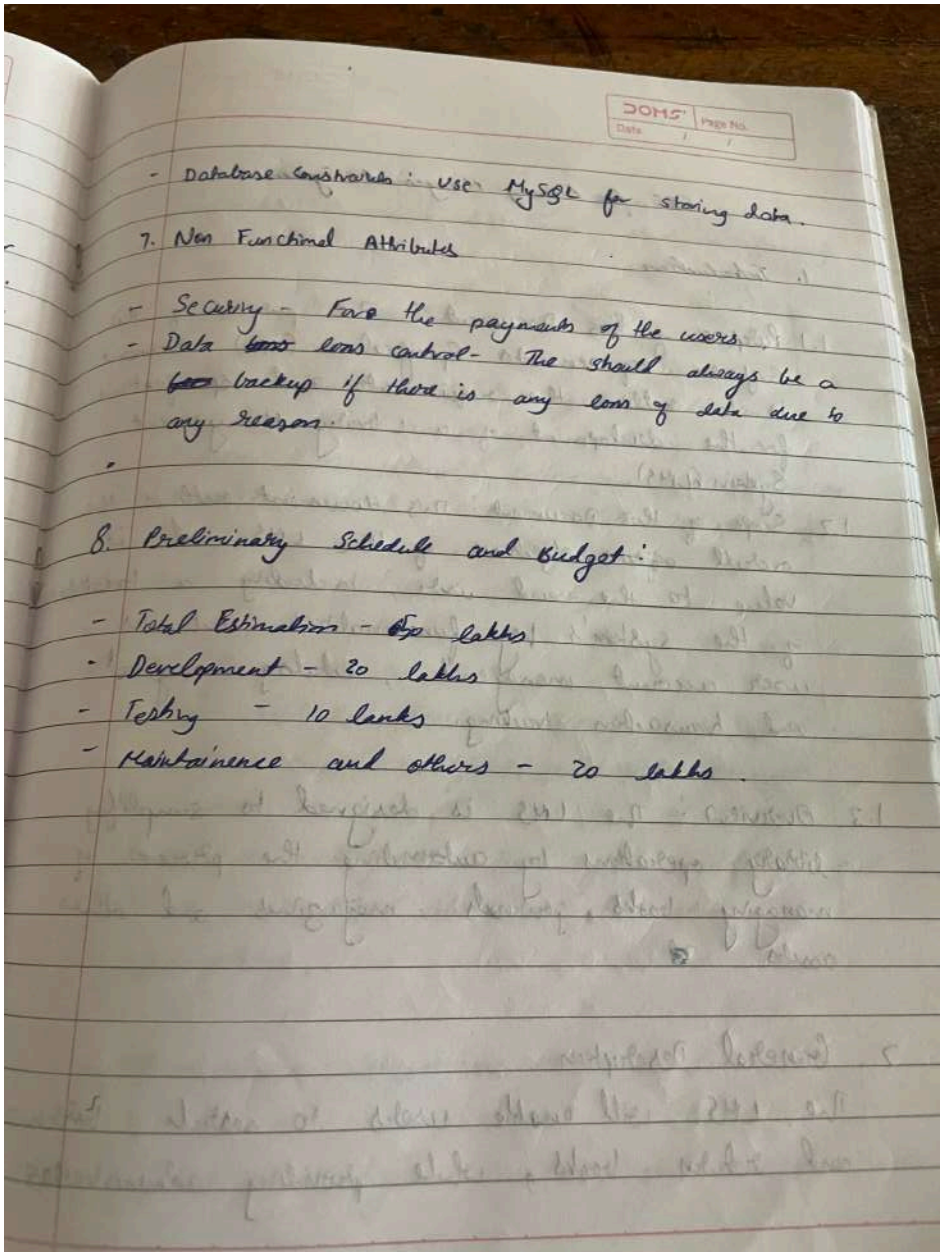
- User interface - Intuitive and responsive web interface for the users which can be accessed from multiple devices.
- Database Interface - To store the booking and personal details of the user in a database.

5. Performance requirement:

- Real-time room availability - The availability of the rooms should be updated in real time.
- Inventory - It is be constantly update to keep a check in the stock for the things required.
- Instant booking - As soon as the payment is done by the customer, they should be provided by their booking details.

6. Design constraints:

- Technical constraints - Use specific language for the development for eg. Java for backend and Angular for frontend.



- Database Constraints : Use MySQL for storing data.

7. Non Functional Attributes

- Security - Fare the payments of the users.
- Data loss control - The should always be a backup if there is any loss of data due to any reasons.

8. Preliminary Schedule and Budget :

- Total Estimation - 50 lakhs
- Development - 20 lakhs
- Testing - 10 lanks
- Maintainence and others - 20 lakhs.

b. Advanced Class Diagram:

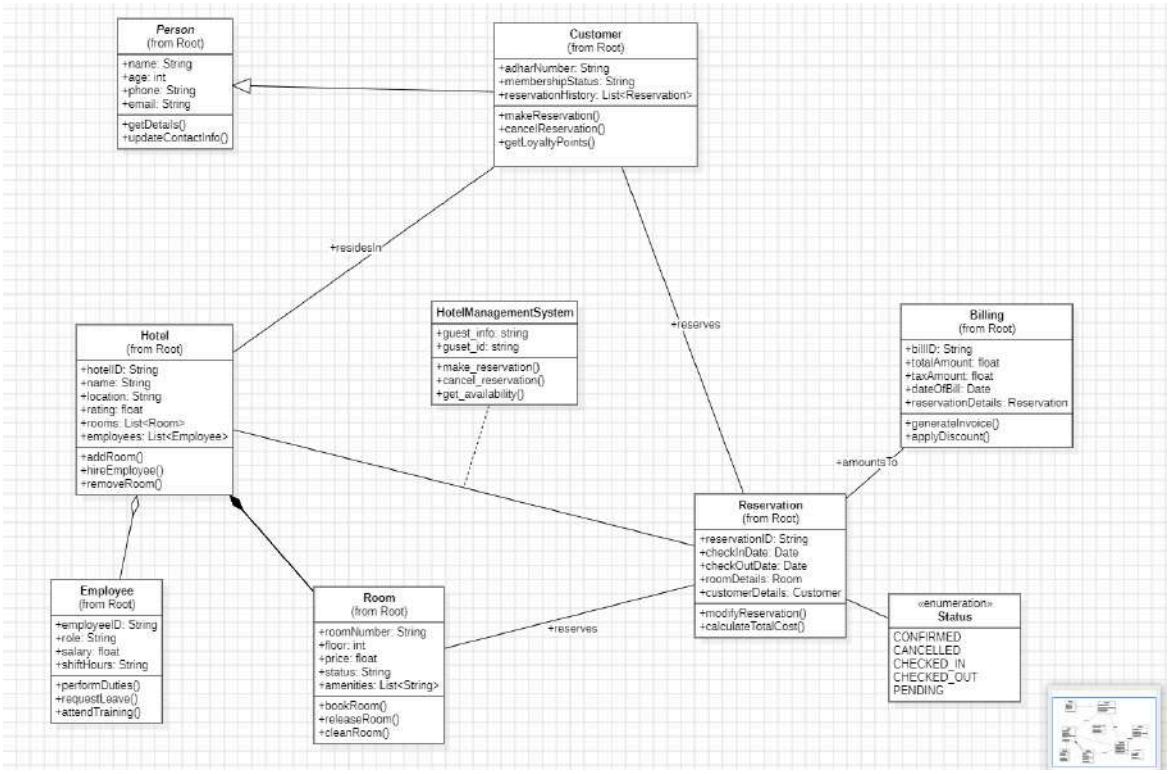


Fig 1.2 : Advanced Class Diagram for hotel management system

- **Person and Customer**: The Person class serves as a base for Customer, which includes details like membership and reservation history. Customers can make and cancel reservations.
- **Hotel**: The central entity managing Rooms and Employees. It includes methods to add or remove rooms and hire staff.
- **Room**: Represents individual rooms in the hotel, with details like room number, price, status, and amenities. Rooms can be booked, released, or cleaned.
- **Employee**: Represents hotel staff, with details like role, salary, and methods to perform duties or request leave.
- **Reservation and Billing**: The Reservation class handles booking details like check-in/check-out dates, room, and customer information. The Billing class generates invoices and applies discounts for reservations.
- **Hotel Management System**: Manages reservations, cancellations, and room availability, acting as the core system.

c. Advanced State Diagram:

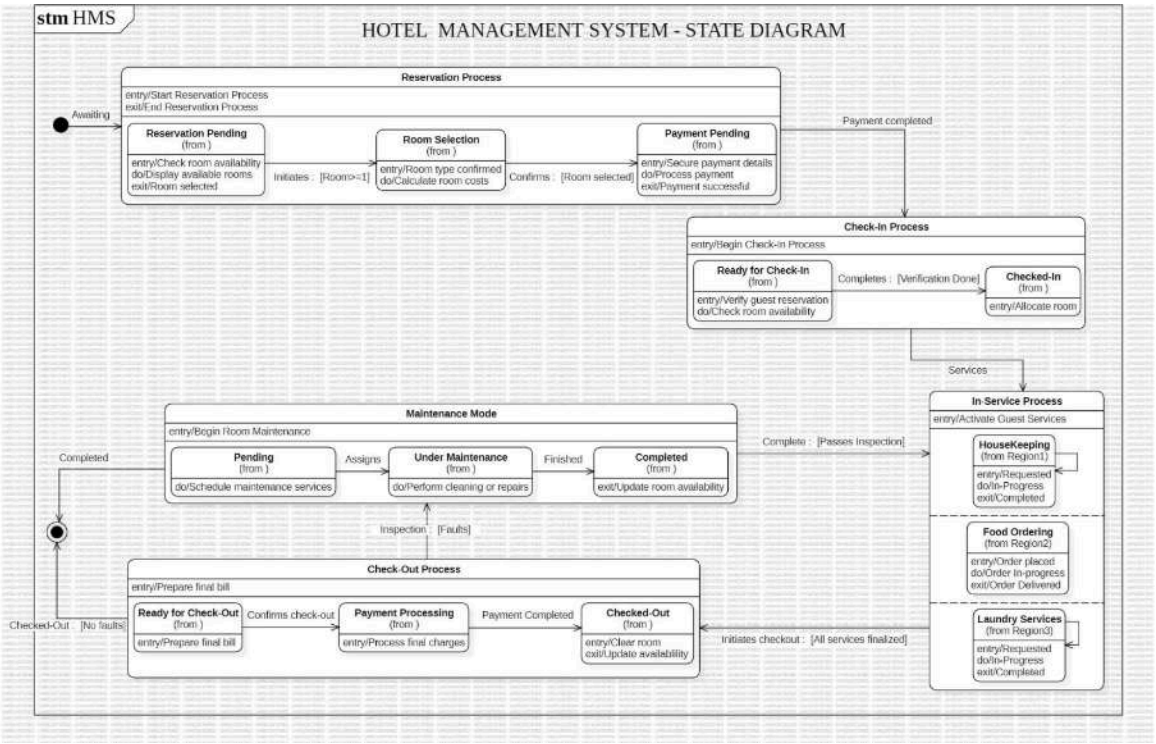


Fig 1.3

Key States:

- Reservation Process: This encompasses the initial reservation request, room selection, and payment processing.
- Check-in Process: This involves verifying the guest's reservation, assigning a room, and completing the check-in formalities.
- In-service Process: This state represents the ongoing services provided to guests, including housekeeping, food and beverage ordering, and laundry services.
- Maintenance Mode: This state handles room maintenance activities such as cleaning, repairs, and updating room availability.
- Check-out Process: This covers the final billing, payment processing, room clearance, and completion of the check-out procedure.

The diagram showcases the transitions between these states, triggered by events like "entry/Start Reservation Process," "exit/End Reservation Process," "entry/Check room availability," and others. These transitions depict the flow of the reservation process through the system.

d. Use Case Diagram:

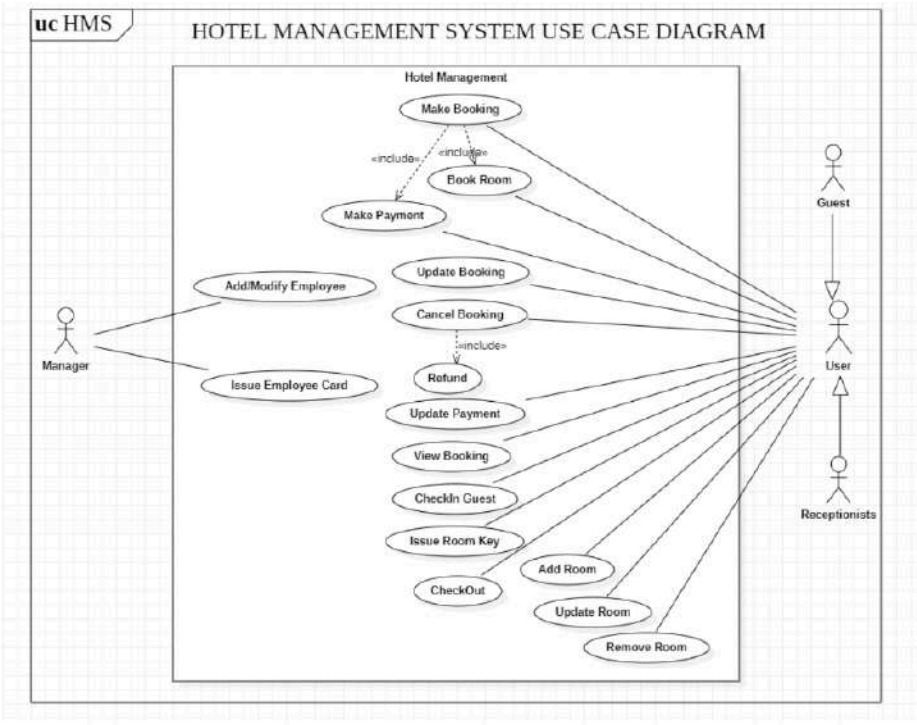


Fig 1.4

e. Sequence Diagram

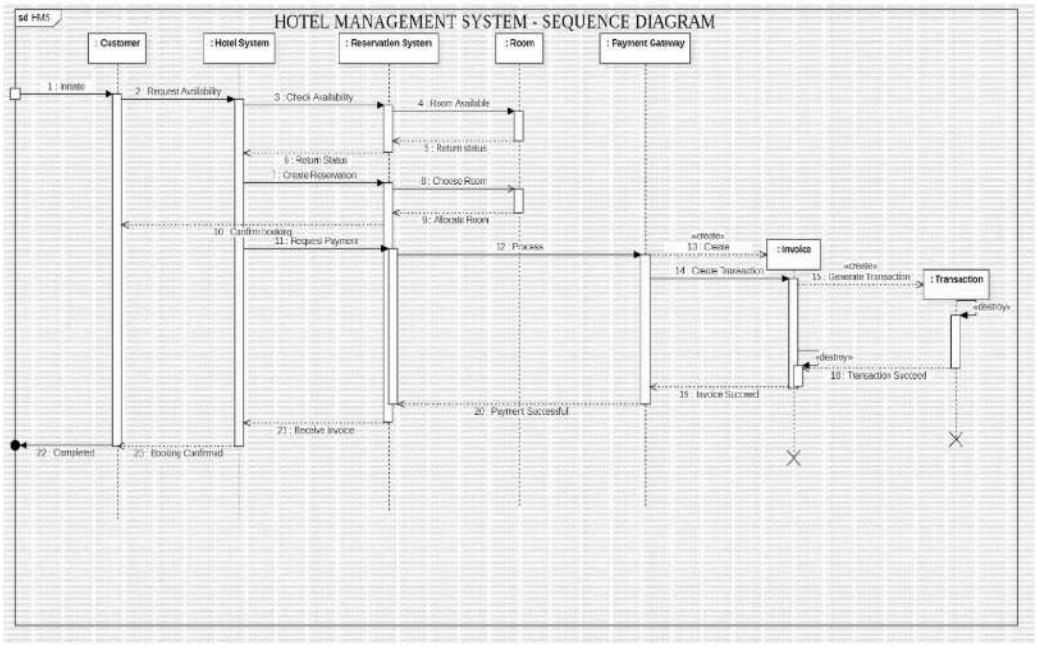

sd HMS
HOTEL MANAGEMENT SYSTEM - SEQUENCE DIAGRAM
: Customer
: Hotel System
: Reservation System
: Room
: Payment Gateway
: Invoice
: Transaction
1 : Initiate
2 : Request Availability
3 : Check Availability
4 : Room Available
5 : Return status
6 : Return Status
7 : Create Reservation
8 : Choose Room
9 : Allocate Room
10 : Confirm booking
11 : Request Payment
12 : Process
«create»
13 : Create
14 : Create Transaction
«create»
15 : Generate Transaction
«destroy»
16 : Transaction Succeed
19 : Invoice Succeed
20 : Payment Successful
21 : Receive Invoice
22 : Completed
23 : Booking Confirmed

Fig 1.5

Activity Diagram:

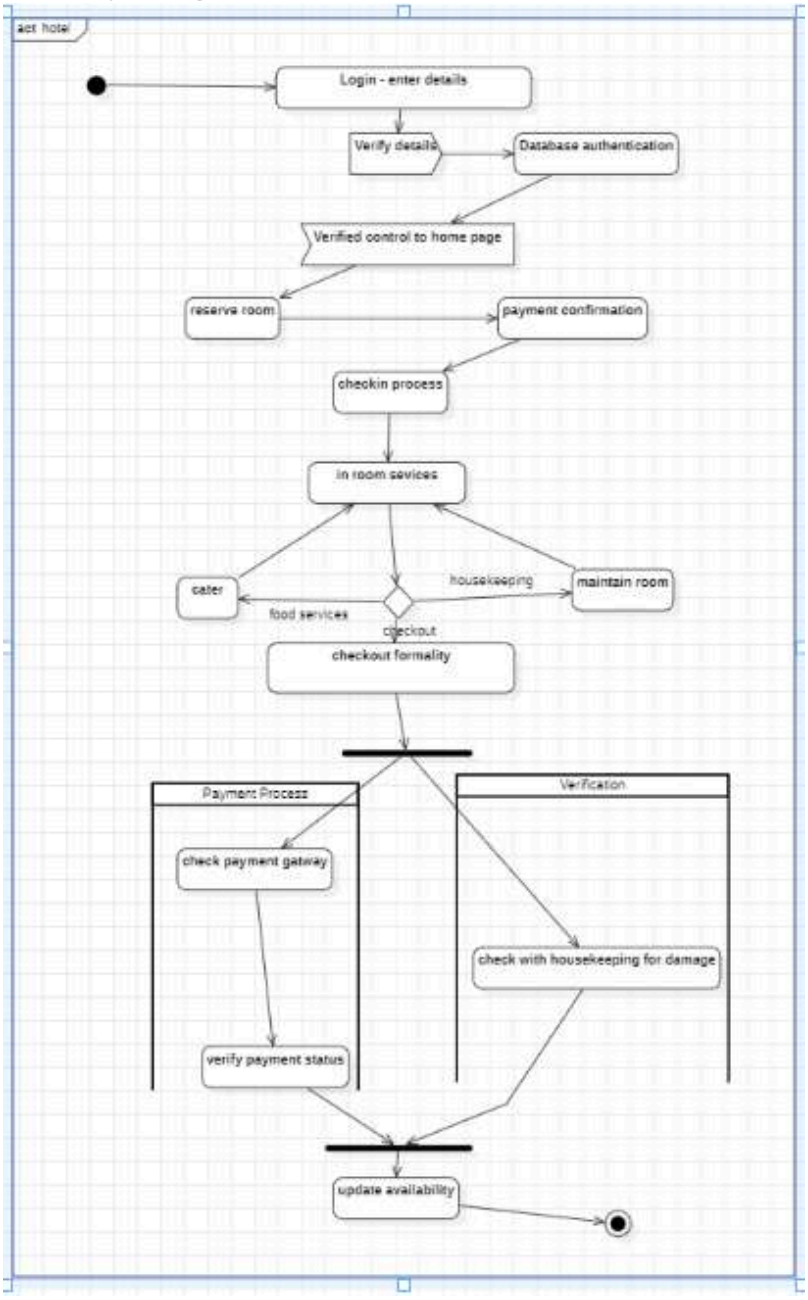
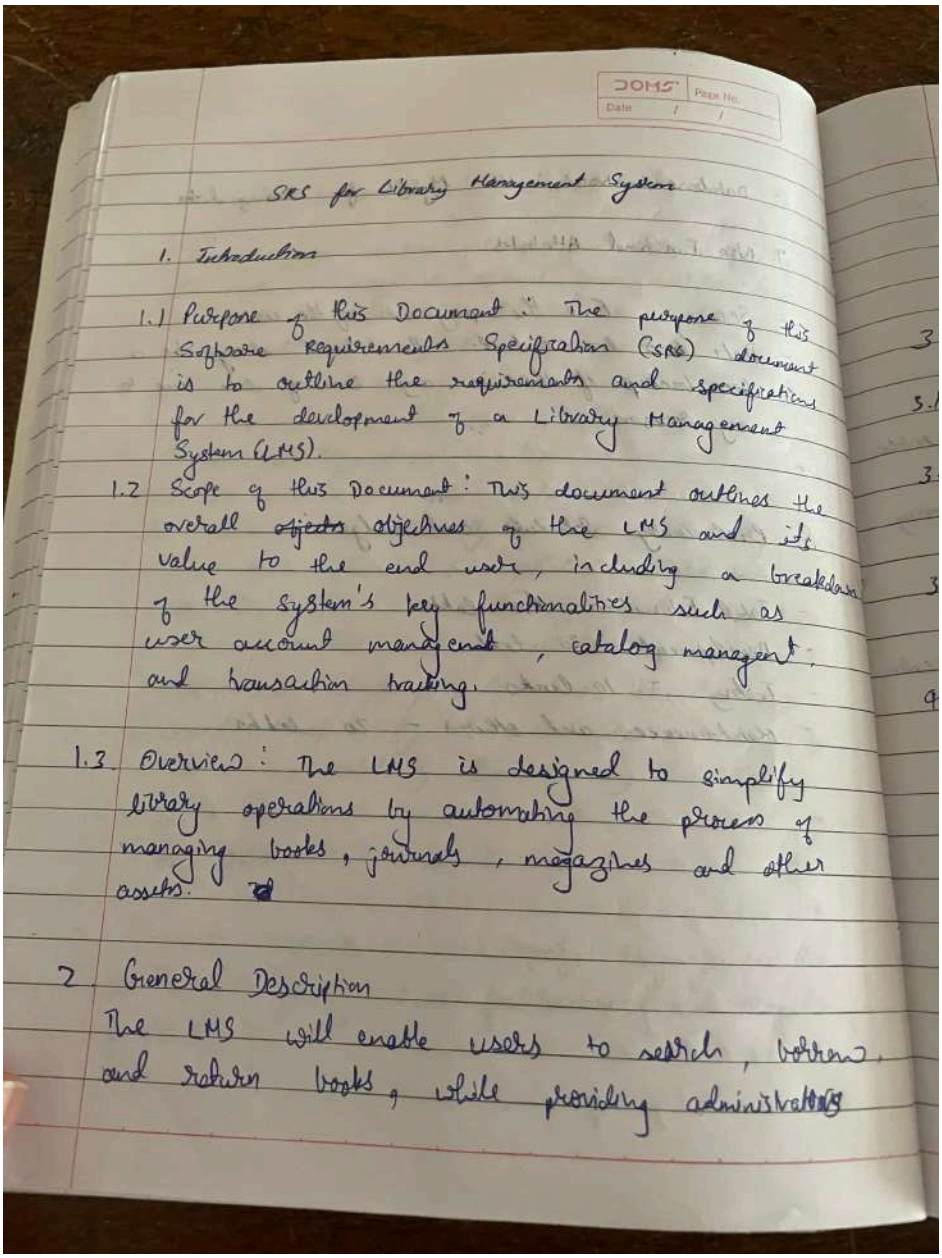


Fig 1.6

2. Library Management System

a. SRS Document:



SRS for Library Management System

1. Introduction

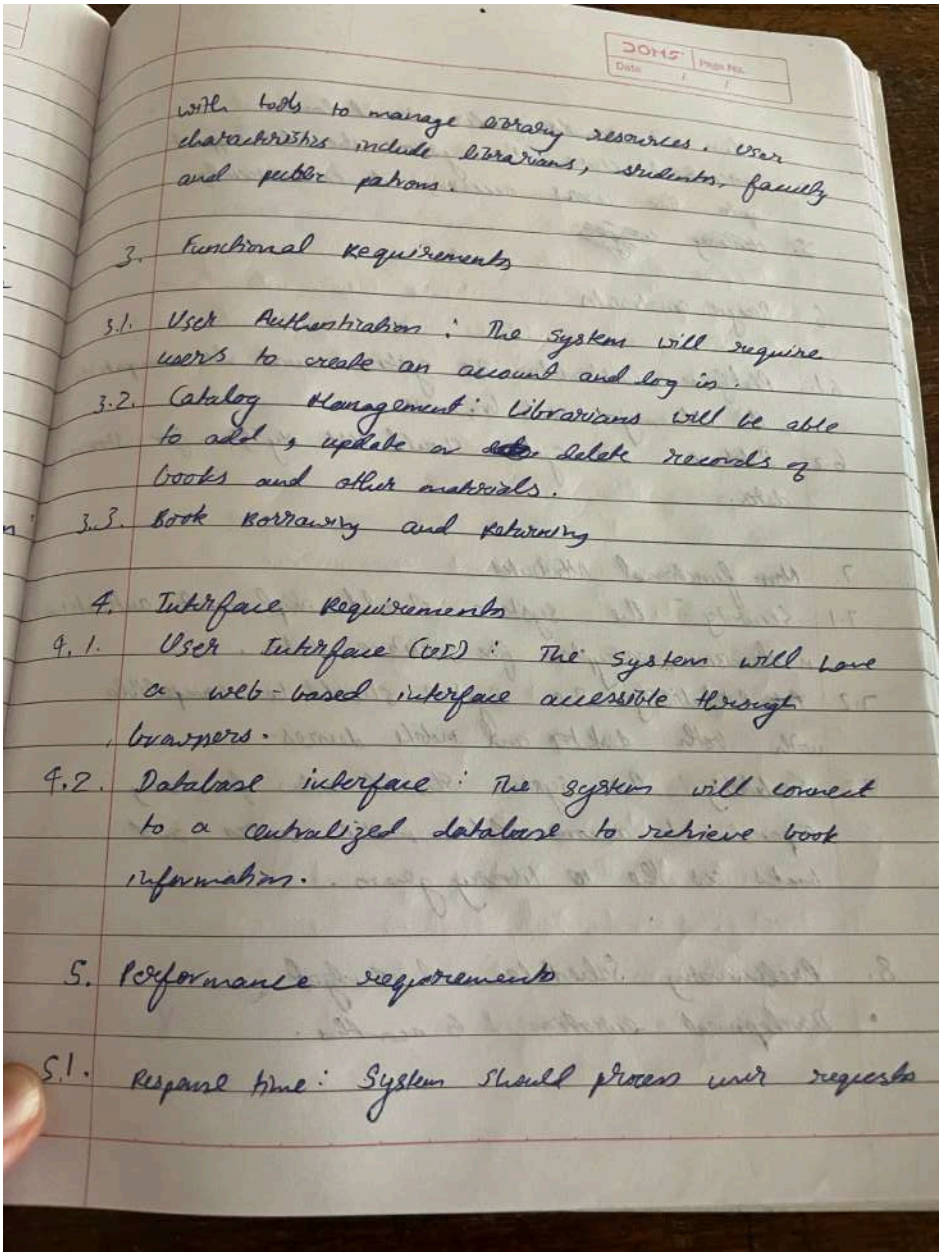
1.1 Purpose of this Document: The purpose of this Software Requirements Specification (SRS) document is to outline the requirements and specifications for the development of a Library Management System (LMS).

1.2 Scope of this Document: This document outlines the overall ~~objects~~ objectives of the LMS and its value to the end users, including a breakdown of the system's key functionalities such as user account management, catalog management, and transaction tracking.

1.3 Overview: The LMS is designed to simplify library operations by automating the process of managing books, journals, magazines and other assets.

2 General Description

The LMS will enable users to search, borrow, and return books, while providing administrators



with tools to manage library resources. User characteristics include librarians, students, faculty and public patrons.

3. Functional Requirements

3.1. User Authentication : The system will require users to create an account and log in.

3.2. Catalog Management: Librarians will be able to add, update or delete records of books and other materials.

3.3. Book Borrowing and Returning

4. Interface Requirements

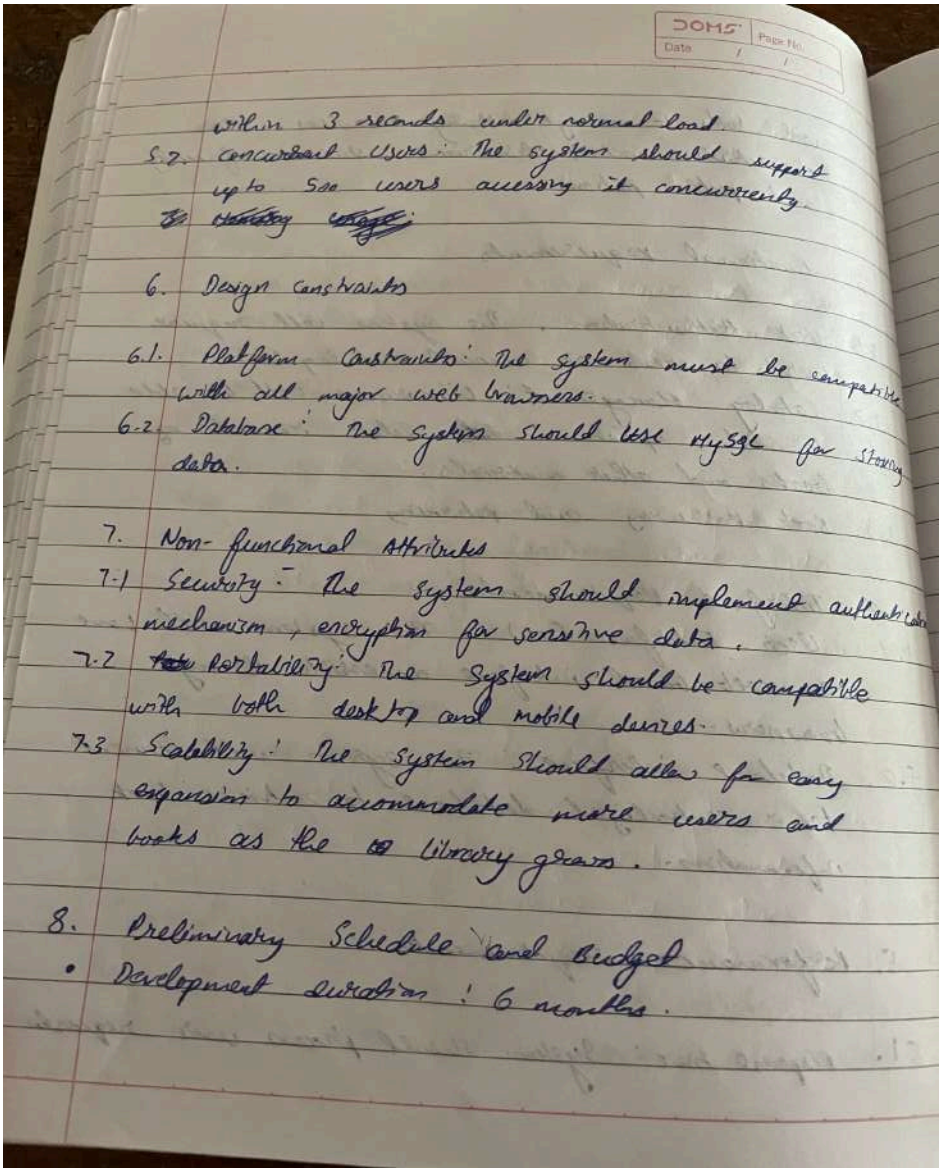
4.1. User Interface (UI) : The system will have a web-based interface accessible through browsers.

4.2. Database interface : The system will connect to a centralized database to retrieve book information.

5. Performance requirements

5.1. Response time: System should process user requests

b.



within 3 seconds under normal load.

5.2. Concurrent Users: The system should support up to 500 users accessing it concurrently

6. Design Constraints

6.1. Platform Constraints: The system must be compatible with all major web browsers.

6.2 Database: The System should use MySQL for storing data.

7. Non-functional Attributes

7.1 Security: The System should implement authentication mechanism, encryption for sensitive data.

7.2 Portability: The System should be compatible with both desktop and mobile devices.

7.3 Scalability: The system should allow for easy expansion to accommodate more users and books as the library grows.

8. Preliminary Schedule and Budget

- Development duration: 6 months.

b.Advanced Class Diagram:

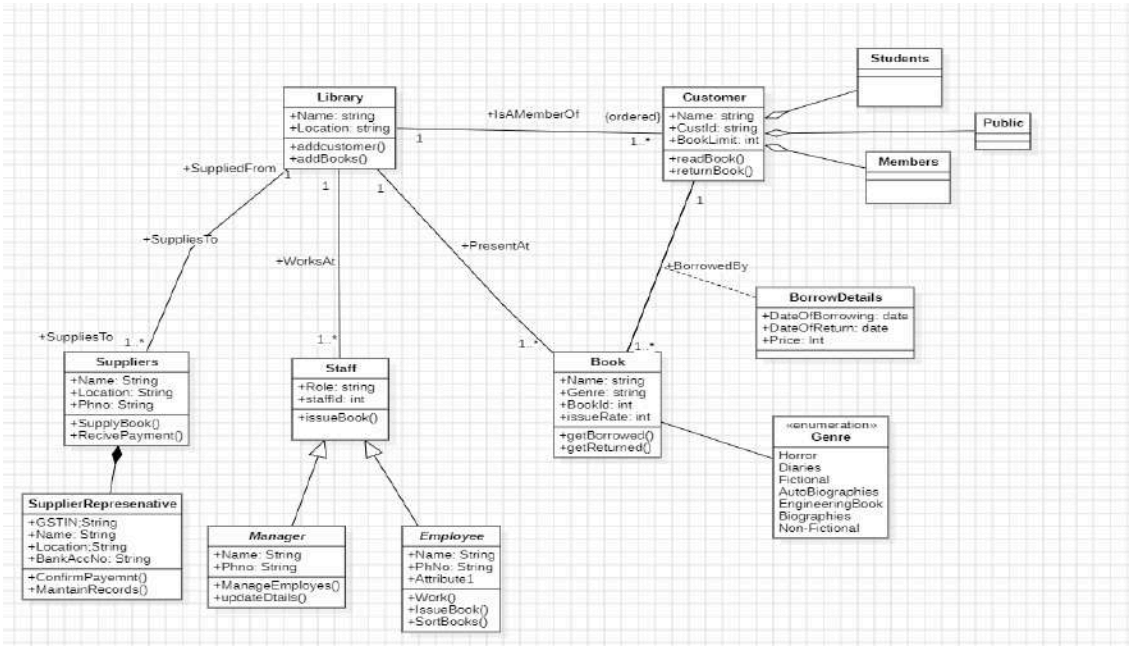


Fig 2.2:

c.Advanced State Diagram:

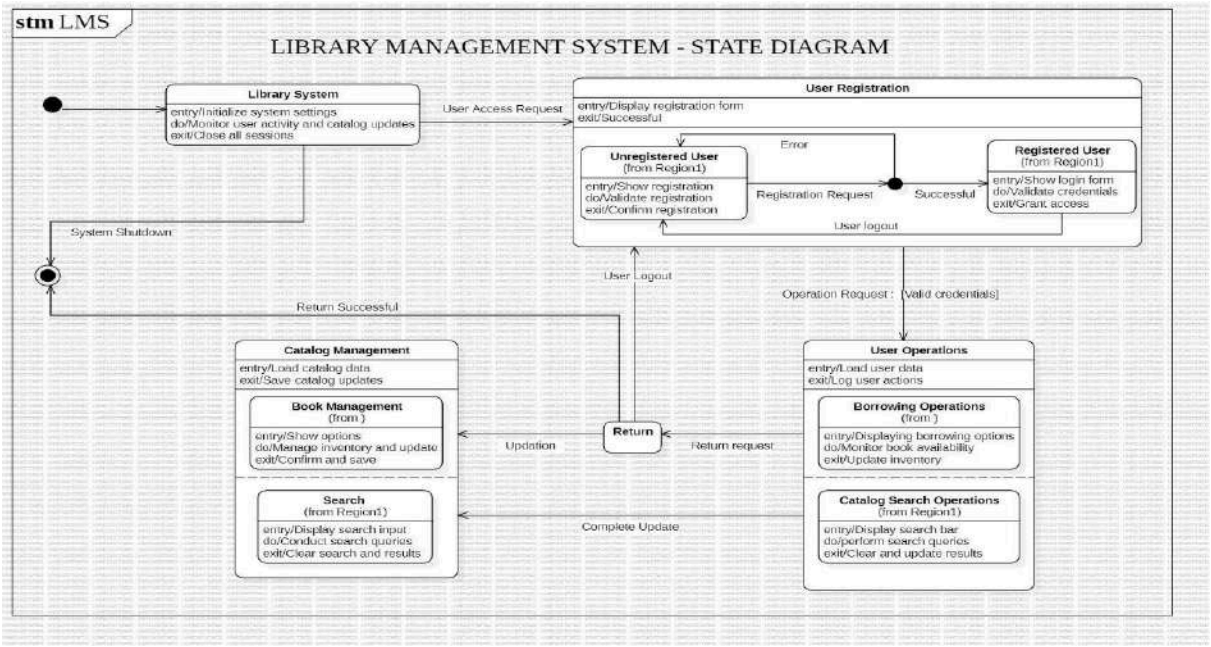


Fig 2.3:

d.Use Case Diagram:

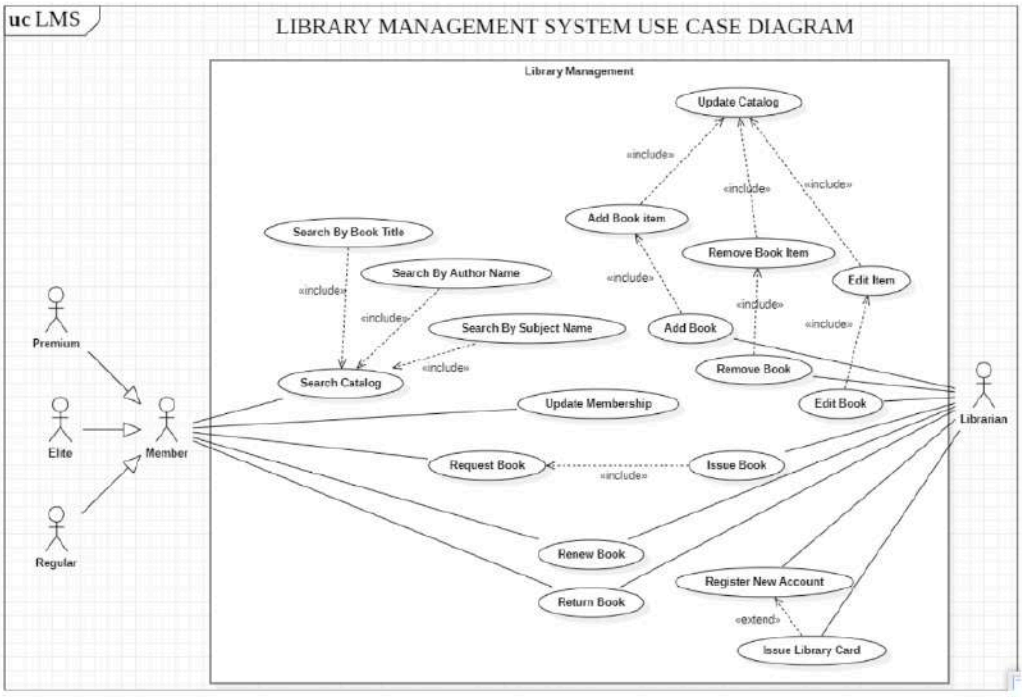


Fig 2.4:

The Library Management System Use Case Diagram illustrates the interactions between library members and staff within the system.

Actors are the entities involved, primarily "Member" representing library users and "Librarian" representing staff.

Use Cases describe the system's functionalities. Members can Search Catalog by title, author, or subject. Librarians can Add Book, Remove Book, and Edit Book information. "Update Catalog" encompasses these book management tasks. Librarians also manage member accounts by Updating Membership and can Issue Book and Renew Book for members. Members can Request Book if it's currently unavailable. For new members, librarians Register New Account and Issue Library Card.

Relationships between use cases are shown. "Include" signifies that one use case incorporates the functionality of another, like "Search By Book Title" being included in "Search Catalog." "Extend" indicates that one use case extends another under specific conditions, such as "Issue Library Card" extending "Register New Account."

e.Sequence Diagram:

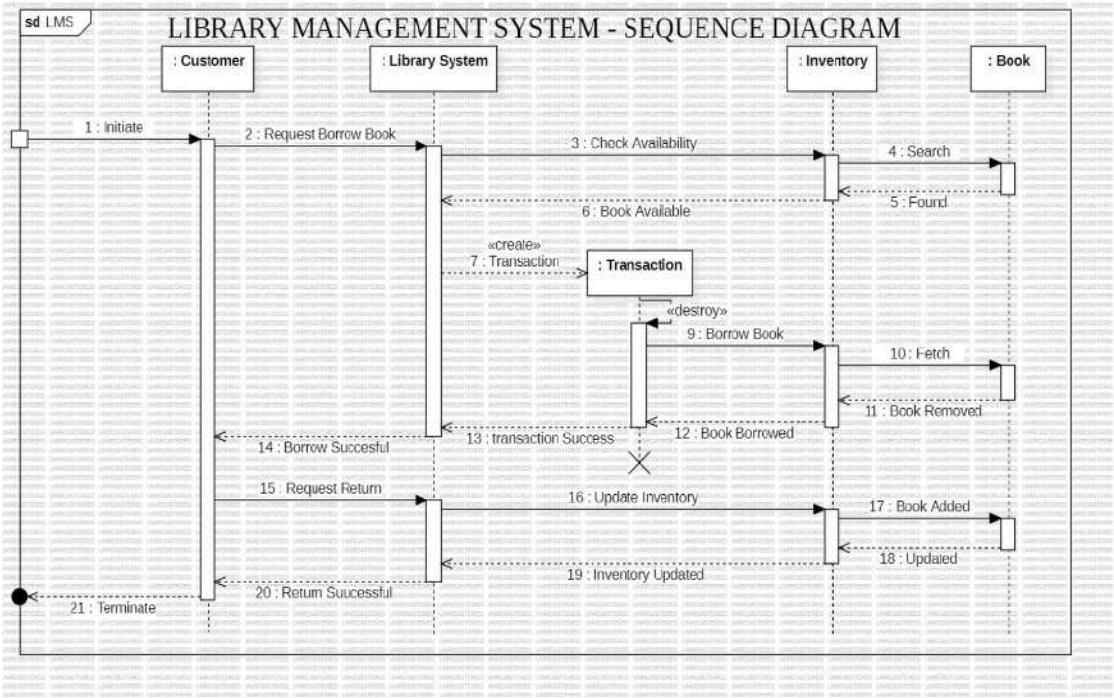


Fig 2.5:

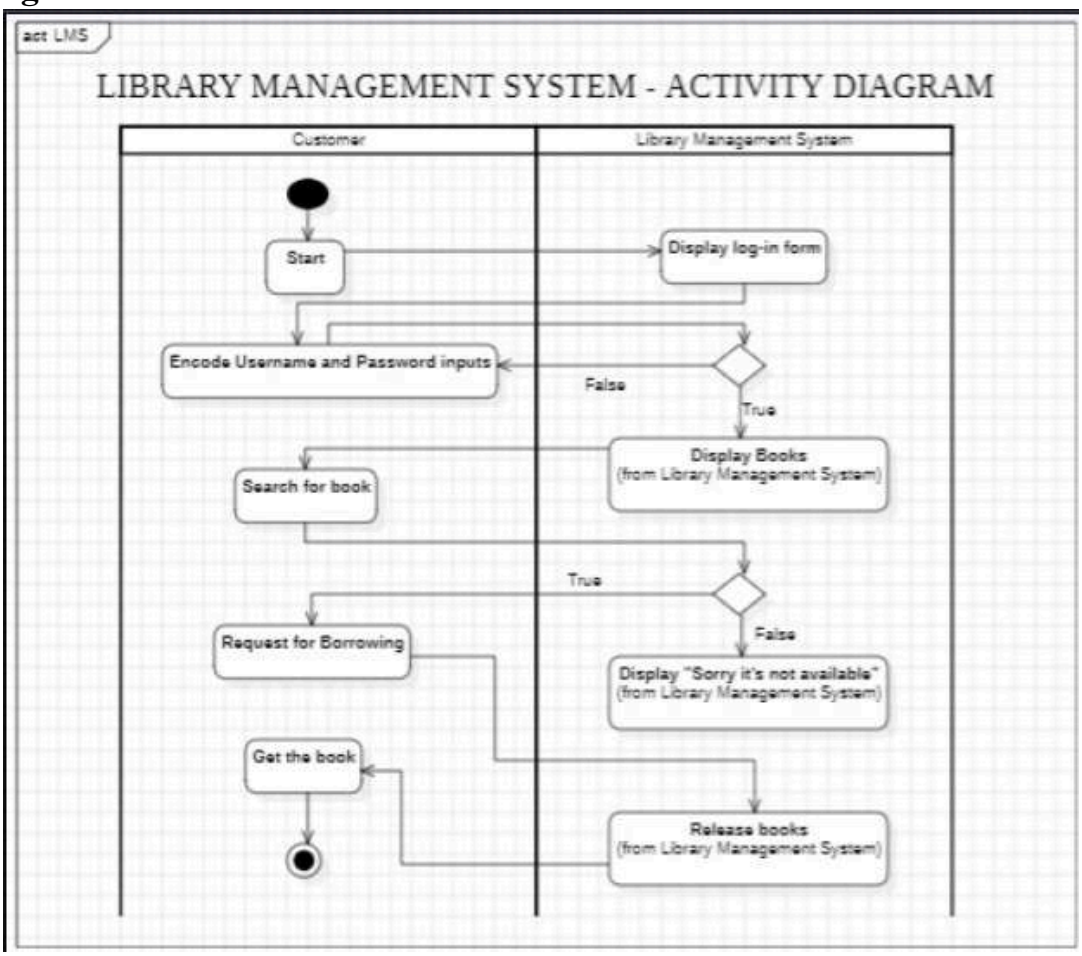
The sequence diagram depicts the interaction between a customer and the library system when borrowing and returning a book.

The customer initiates the process by requesting to borrow a specific book. The library system checks the inventory for the book's availability. If found, the system informs the customer and creates a transaction to manage the borrowing process. The customer then borrows the book, and the inventory records the book as borrowed and updates its availability status. The system confirms the successful borrowing to the customer.

When the customer returns the book, the system updates the inventory to reflect the return and notifies the customer that the return was successful.

This sequence diagram provides a clear visualization of the steps involved in the book borrowing and returning process within the library system. It highlights the interactions between the customer, the library system, the inventory, and the book objects involved in each stage.

f.Activity Diagram:



- **Actors**:

  - **Customer**: Represents the user interacting with the system.
  - **Library Management System**: Represents the system itself.

- **Process Flow**:

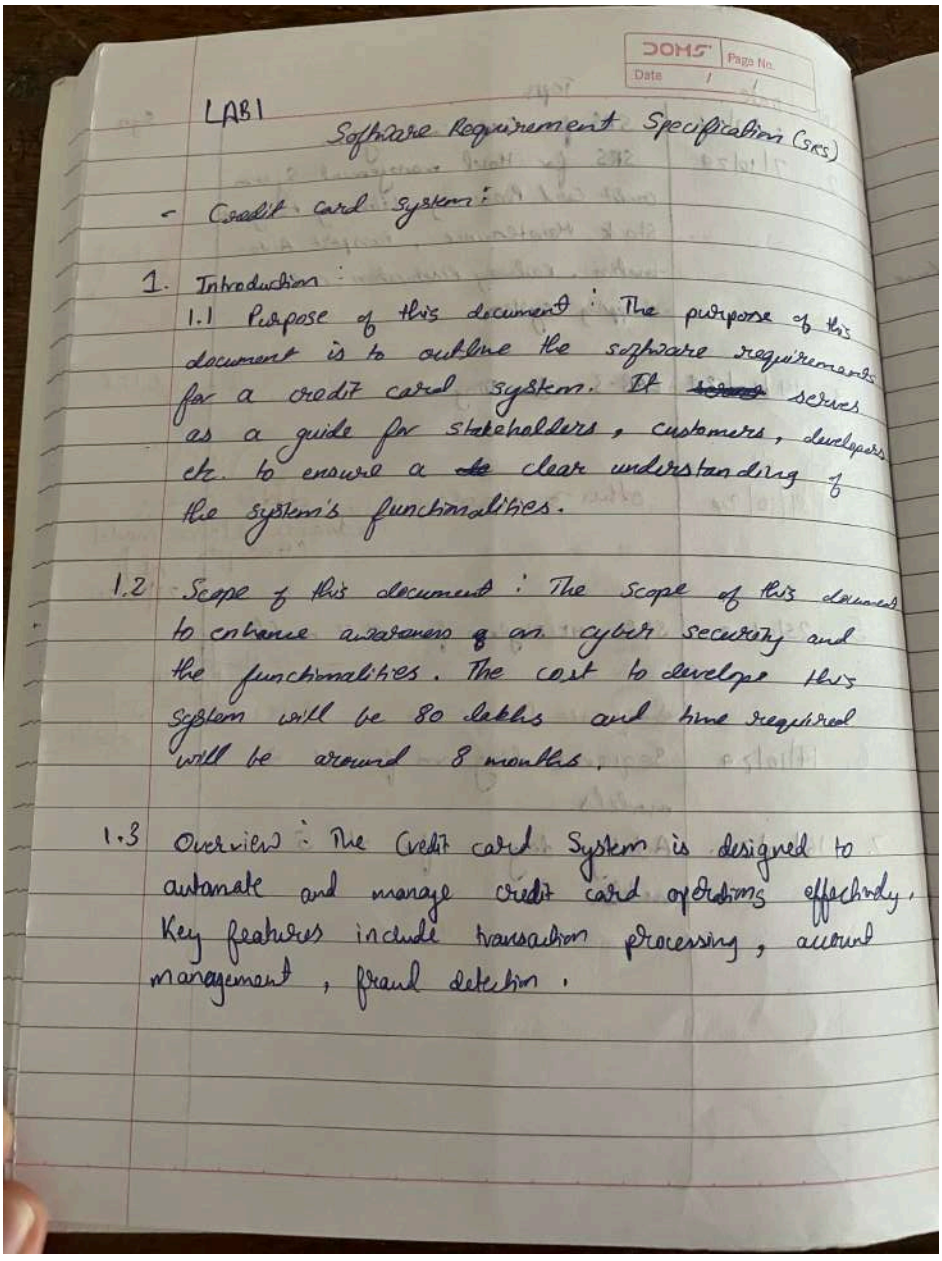
  - The customer starts by logging in, encoding their username and password.
  - The system validates the credentials. If invalid, the process stops; if valid,proceeds.
  - The customer searches for a book. The system checks the book's availability:
    - If the book is unavailable, the system displays a "not available" message.
    - If available, the customer requests to borrow the book, and the system releases it to them.

- **Decisions**:

- There are decision points for login validation and book availability, indicating alternate flows

3. Credit Card Management System

a. SRS Document:



LAB1

Software Requirement Specification (SRS)

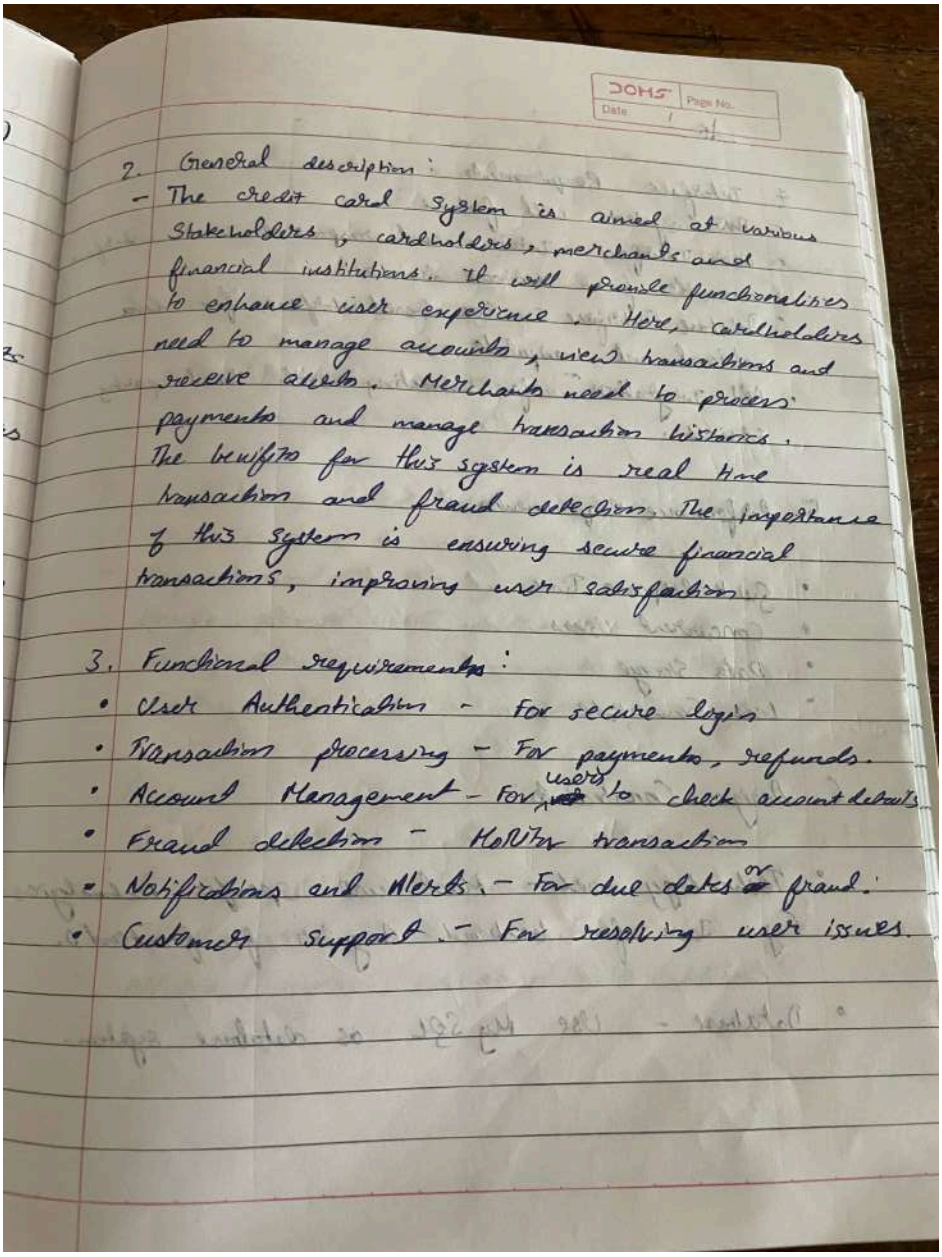
- Credit card system:

1. Introduction:

1.1 Purpose of this document: The purpose of this document is to outline the software requirements for a credit card system. It serves as a guide for stakeholders, customers, developers etc. to ensure a clear understanding of the system's functionalities.

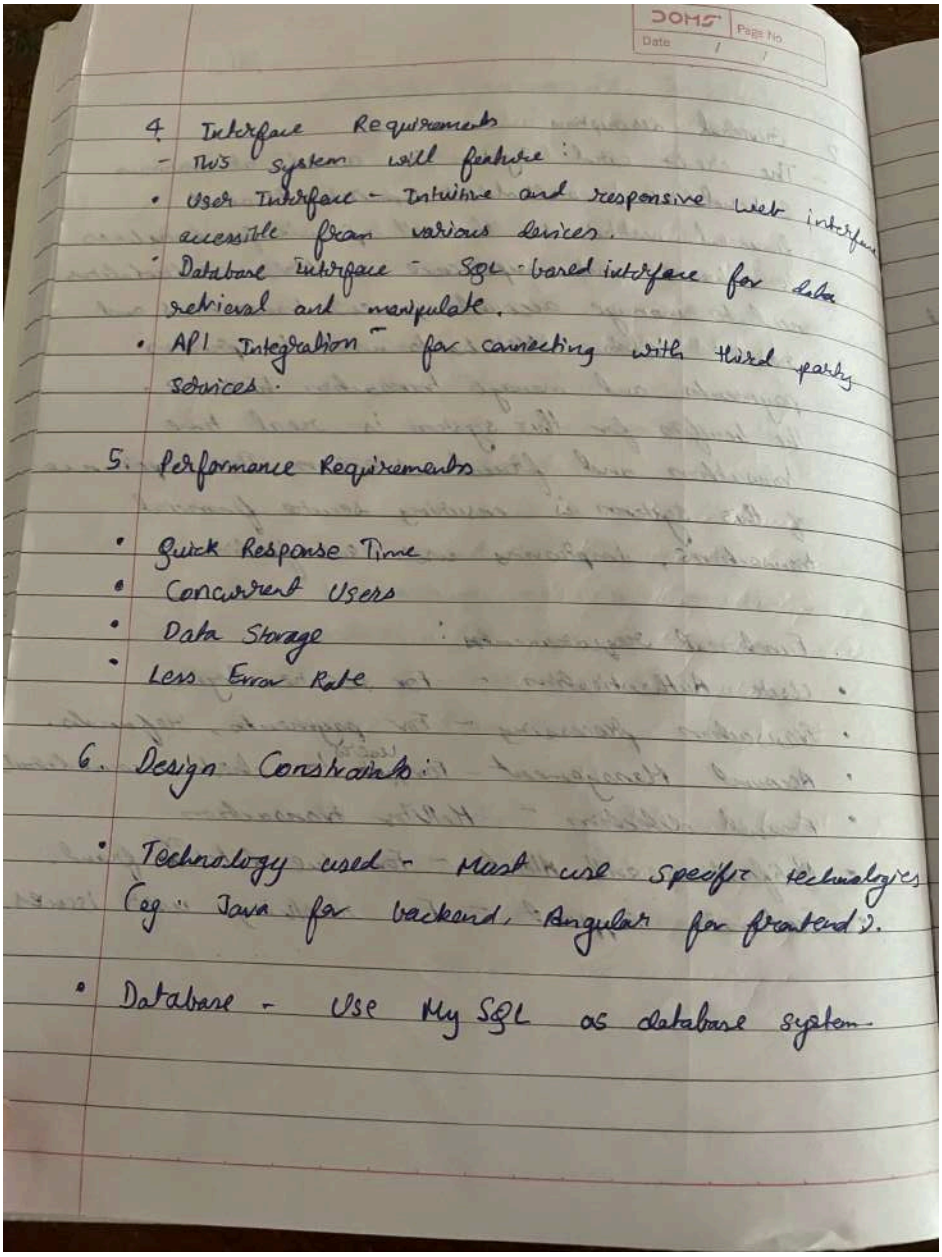
1.2 Scope of this document: The scope of this document to enhance awareness on cyber security and the functionalities. The cost to develop this system will be 80 lakhs and time required will be around 8 months.

1.3 Overview: The Credit card System is designed to automate and manage credit card operations effectively. Key features include transaction processing, account management, fraud detection.



2. General description:
- The credit card system is aimed at various stakeholders, cardholders, merchants and financial institutions. It will provide functionalities to enhance user experience. Here, cardholders need to manage accounts, view transactions and receive alerts. Merchants need to process payments and manage transaction histories. The benifits for this system is real time transaction and fraud detection. The importance of this system is ensuring secure financial transactions, improving user satisfaction.

3. Functional requirements:
- User Authentication - For secure login
- Transaction processing - For payments, refunds.
- Account Management - For users to check account details
- Fraud detection - Monitor transaction
- Notifications and Alerts - For due dates or fraud.
- Customer support - For resolving user issues.



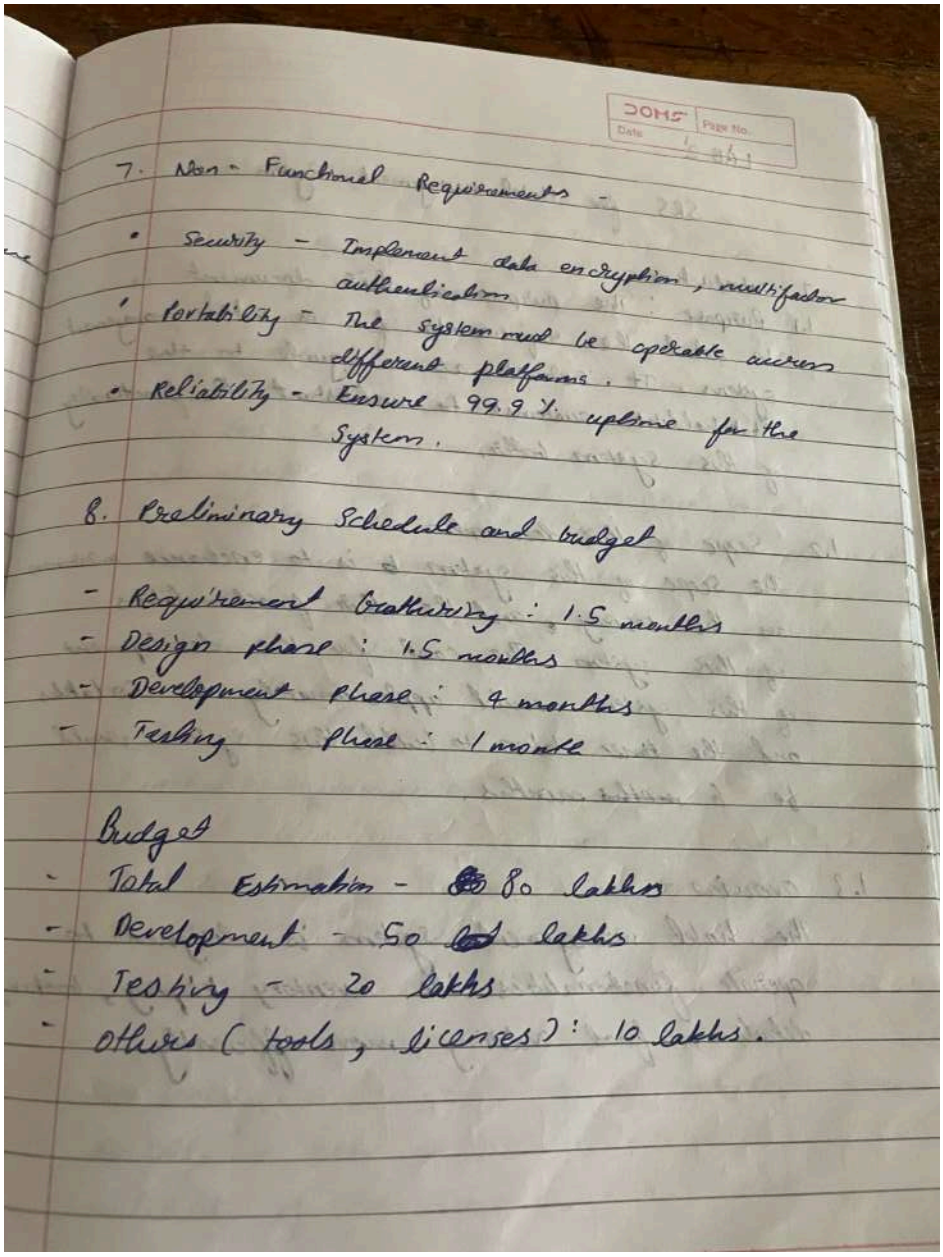
4. Interface Requirements
- This system will feature:
- User Interface - Intuitive and responsive web interface accessible from various devices.
- Database Interface - SQL-based interface for data retrieval and manipulate.
- API Integration - for connecting with third party services.

5. Performance Requirements

- Quick Response Time
- Concurrent Users
- Data Storage
- Less Error Rate

6. Design Constraints:

- Technology used - Must use specific technologies (eg: Java for backend, Angular for frontend).
- Database - Use My SQL as database system.



7. Non - Functional Requirements -

- Security - Implement data encryption, multifactor authentication
- Portability - The system must be operable across different platforms.
- Reliability - Ensure 99.9 % uptime for the system.

8. Preliminary Schedule and budget

- Requirement Gathering : 1.5 months
- Design phase : 1.5 months
- Development phase : 4 months
- Testing phase : 1 month

Budget

- Total Estimation - ₹ 80 lakhs
- Development - 50 lakhs
- Testing - 20 lakhs
- Others (tools, licenses) : 10 lakhs.

b. Advanced Class Diagram:

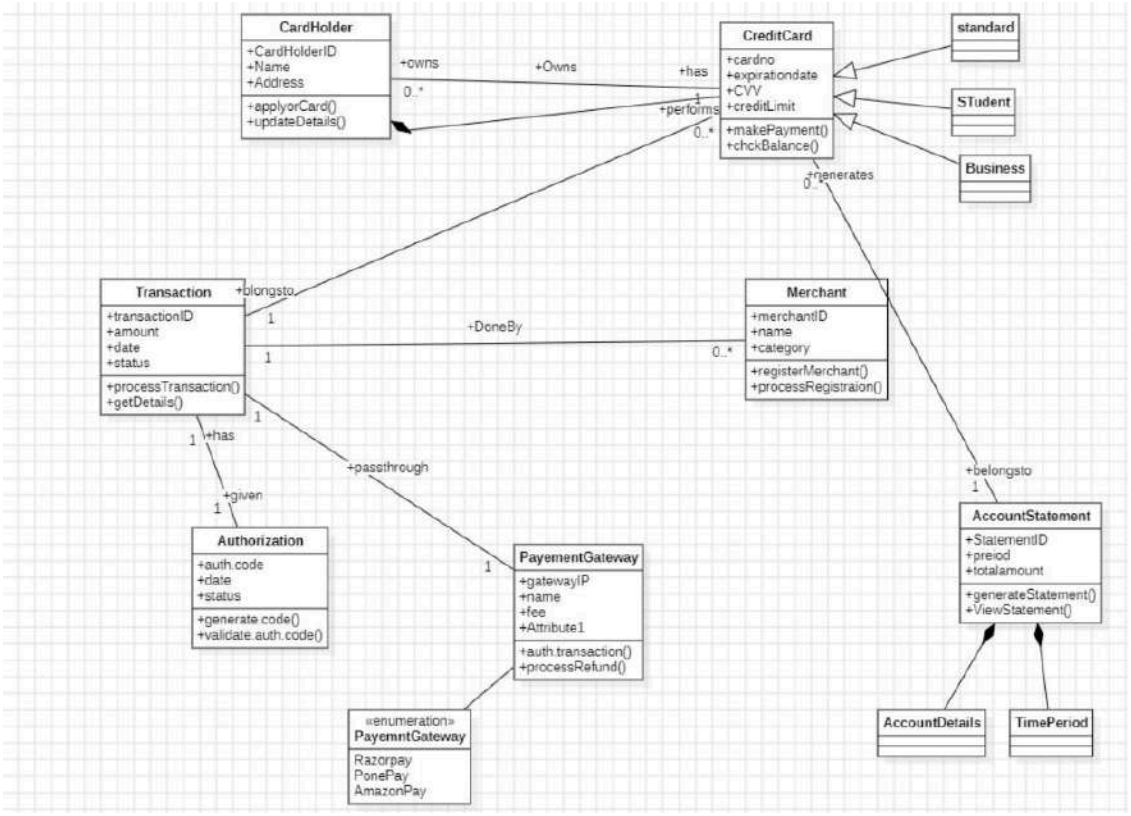


Fig 3.1:

The main entities include:

- Card Holder, who owns a Credit Card used for transactions. The credit card has details like card number, CVV, and credit limit, with operations for payments and balance checks.
- Transaction, which records details like amount, date, and status, is linked to both Authorization (for validation) and Payment Gateway (for processing or refunding payments).
- Merchant, which registers businesses accepting payments and processes transactions from customers.
- Account Statement, which tracks transaction history, generates statements, and is linked to periods and account details.

The diagram also shows inheritance, with specialized credit cards (e.g., Student or Business), and enumerates multiple payment gateways (e.g., RazorPay, PhonePay). It maps how users, cards, merchants, and gateways interact in the system to process payments seamlessly.

c. Advanced State Diagram:

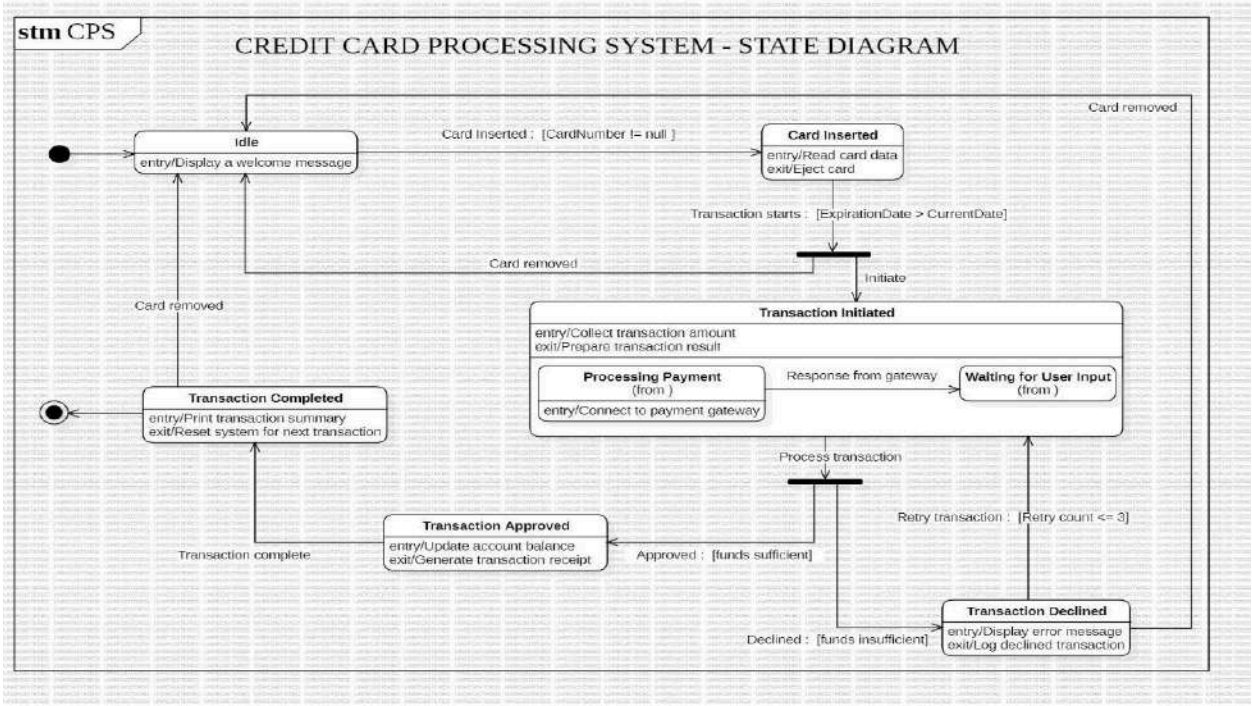


Fig 3.2:

d. Use Case Diagram:

Credit Processing System

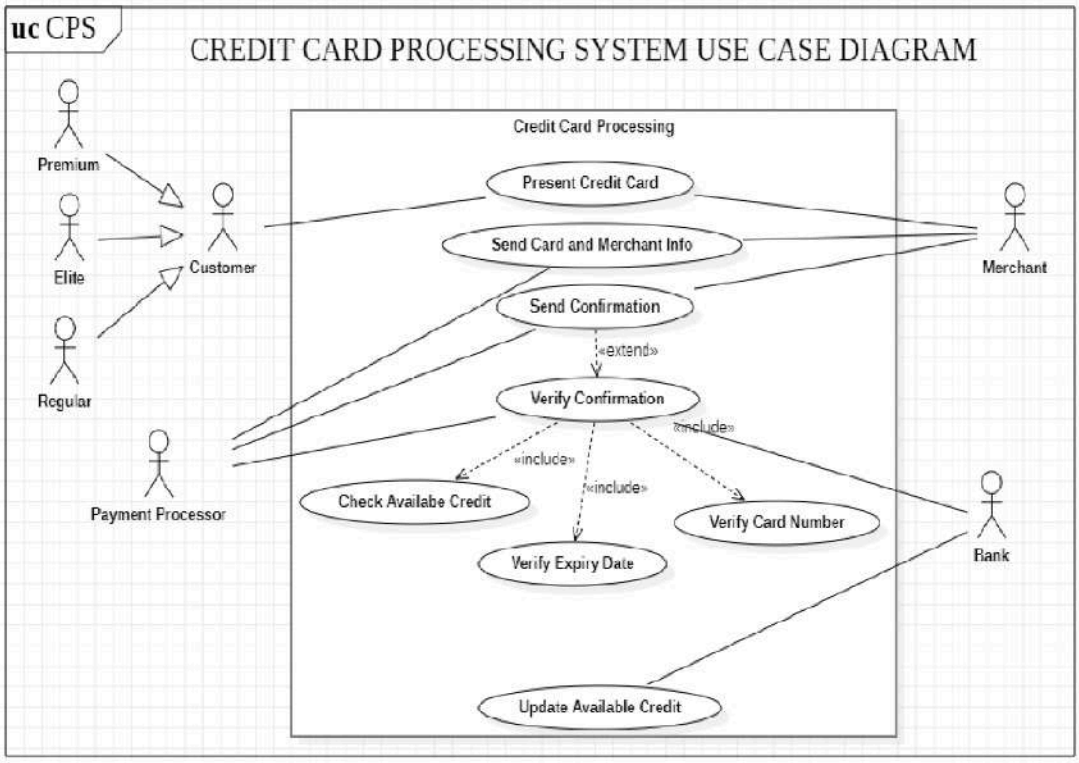


Fig 3.3:

The Use Case Diagram depicts the interactions within a Credit Card Processing System. Actors are the entities involved, primarily the Customer, the Merchant receiving payment, the Payment Processor handling the transaction, and the Bank that issued the card.

Use Cases describe the system's functionalities. The Customer presents their card. The Merchant sends card and merchant details to the Payment Processor. The Payment Processor confirms the transaction to the Merchant. The Merchant verifies this confirmation. The Payment Processor checks the card's available credit, verifies its number and expiry date, and updates the available credit post-transaction.

Relationships are shown. "Include" indicates that one use case incorporates the functionality of another, like "Verify Card Number" and "Verify Expiry Date" being included in "Check Available Credit." "Extend" signifies that one use case can extend another under specific conditions, such as "Verify Confirmation" extending "Send Confirmation" in case of a dispute.

This diagram provides a concise representation of the credit card processing system, outlining the key actors, use cases, and their interactions. It visually depicts the flow of information and actions within a typical credit card transaction.

e. Sequence Diagram:

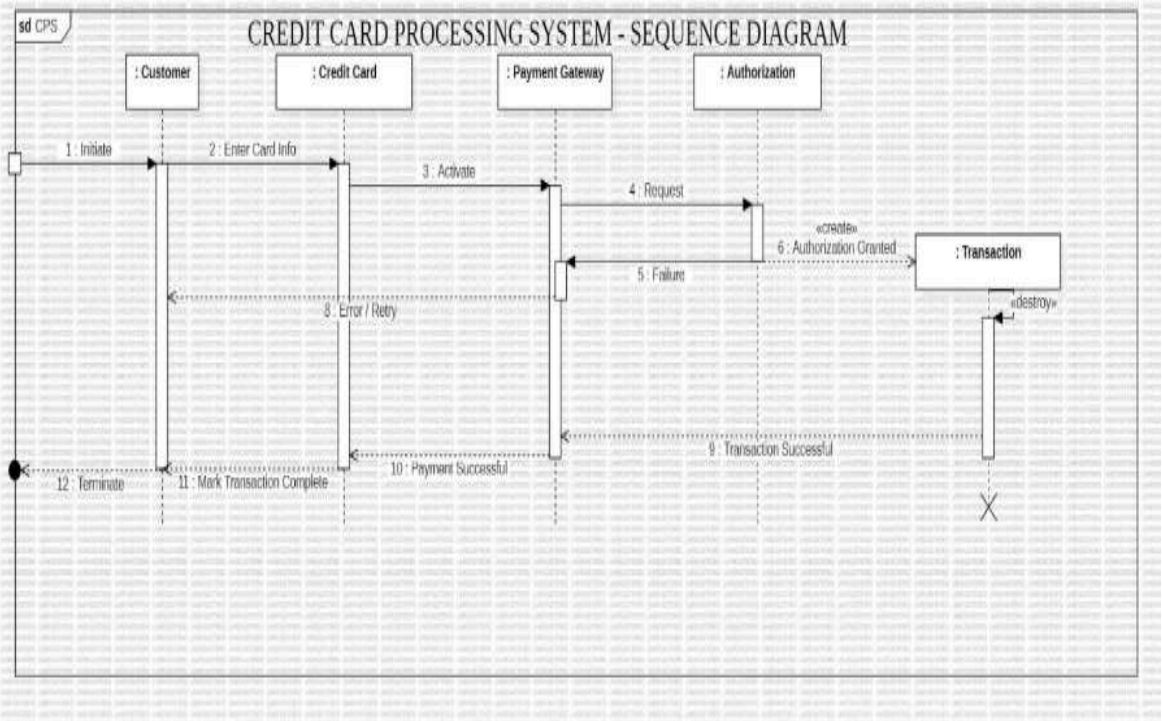


Fig 3.4:

f. Activity Diagram:

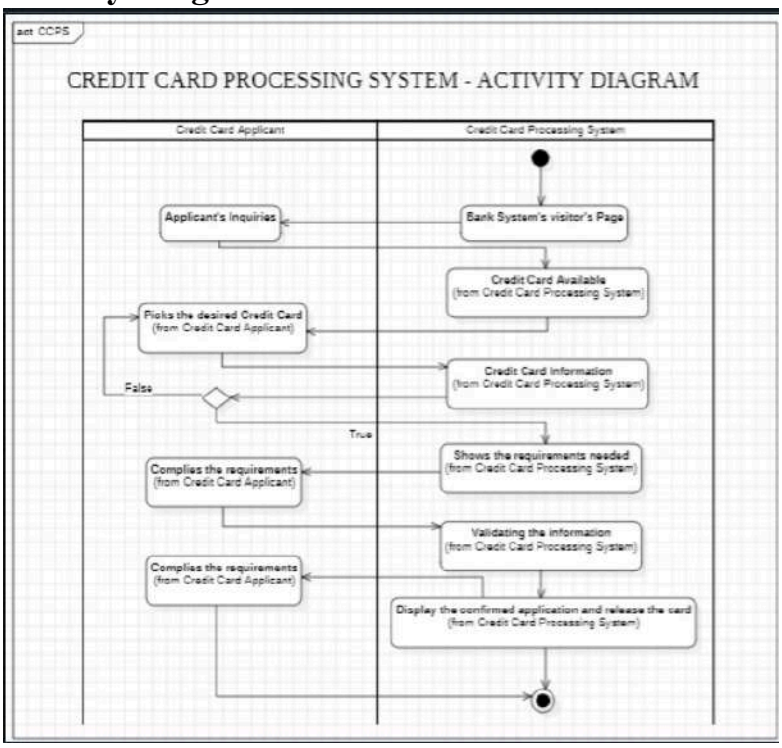


Fig 3.5:

Applicant's Journey:

- The process starts with the applicant making inquiries and selecting a desired credit card.
- If the chosen card is available, the applicant compiles and submits the required documents.

Processing System's Role:

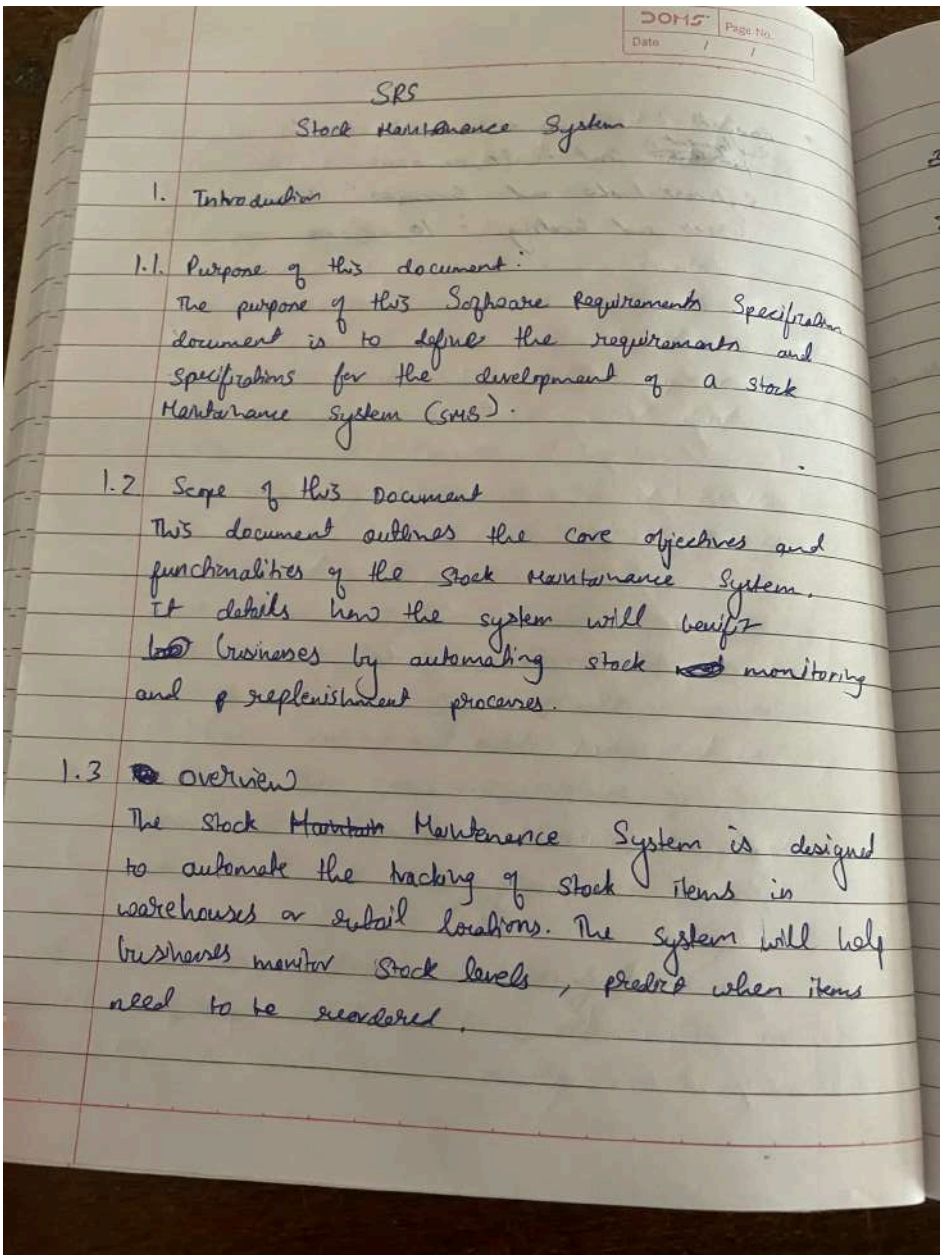
- The system displays the bank's visitor page and shows the available cards and their details.
- It provides the list of requirements needed and validates the submitted information.
- Upon successful validation, the system confirms the application and releases the card.

Decision Point:

- If the card is unavailable, the applicant must re-select or terminate the process.

4. Stock Maintenance system

a. SRS Document



SRS

Stock Maintenance System

1. Introduction

1.1. Purpose of this document:

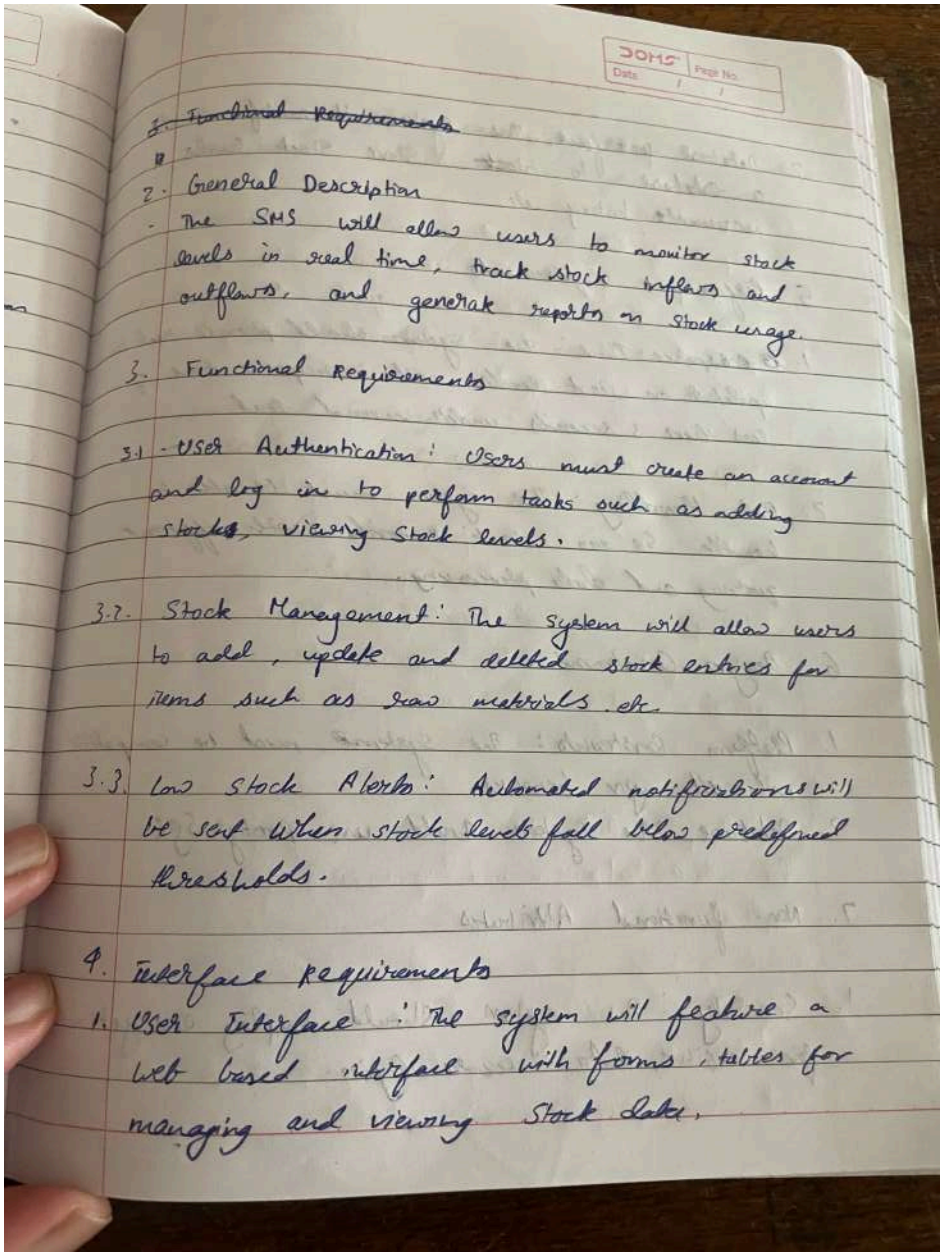
The purpose of this Software Requirements Specification document is to define the requirements and specifications for the development of a Stock Maintenance System (SMS).

1.2 Scope of this Document

This document outlines the core objectives and functionalities of the Stock Maintenance System. It details how the system will benefit businesses by automating stock monitoring and replenishment processes.

1.3 Overview

The Stock Maintenance System is designed to automate the tracking of stock items in warehouses or retail locations. The system will help businesses monitor stock levels, predict when items need to be reordered.



2. ~~Functional Requirements~~

2. General Description

- The SMS will allow users to monitor stock levels in real time, track stock inflows and outflows, and generate reports on stock usage.

3. Functional Requirements

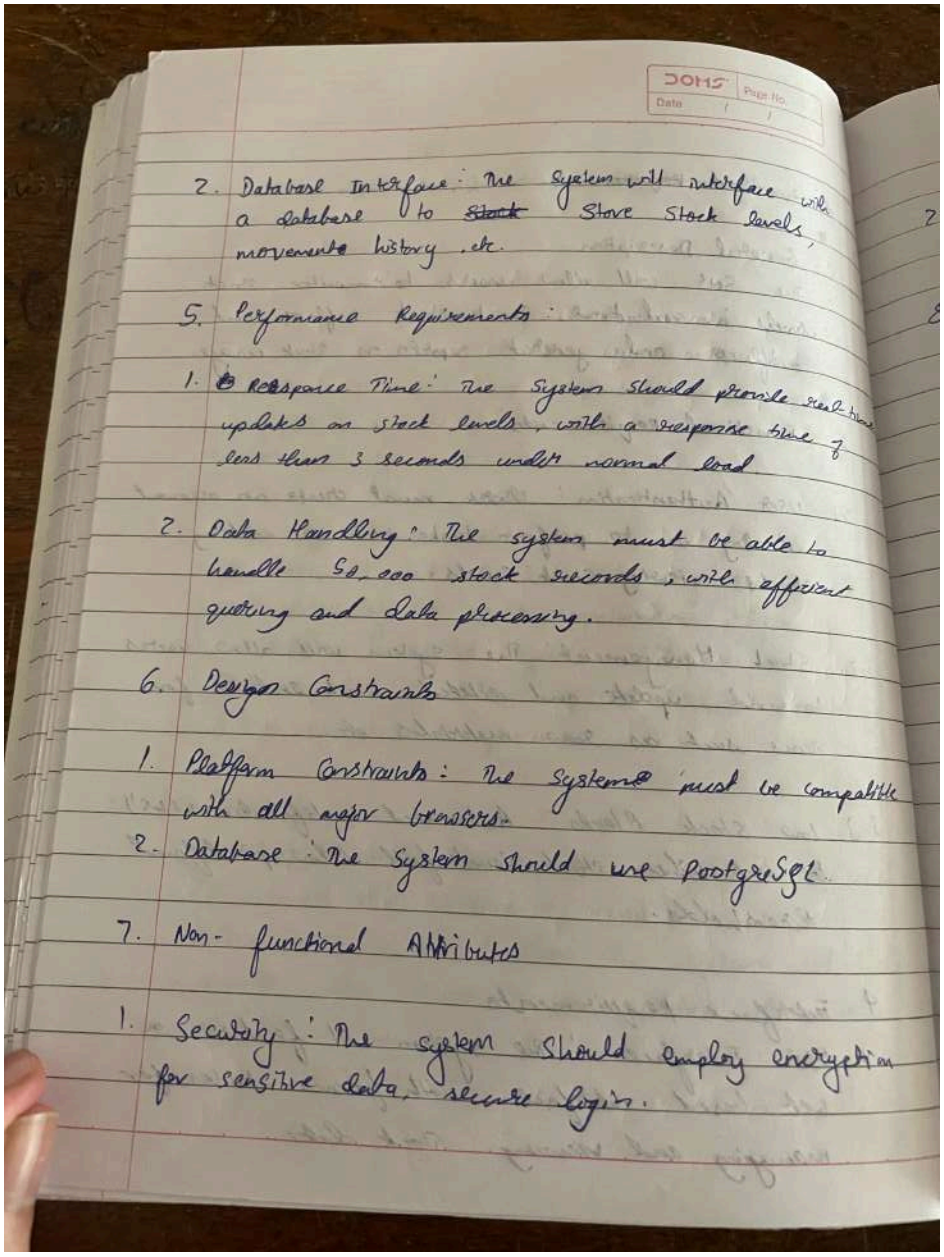
3.1 - User Authentication: Users must create an account and log in to perform tasks such as adding stocks, viewing stock levels.

3.2. Stock Management: The system will allow users to add, update and delete stock entries for items such as raw materials etc.

3.3. Low Stock Alerts: Automated notifications will be sent when stock levels fall below predefined thresholds.

4. Interface requirements

1. User Interface: The system will feature a web based interface with forms, tables for managing and viewing stock data.



2. Database Interface: The system will interface with a database to ~~Stock~~ Store stock levels, movements history .etc.

5. Performance Requirements:

1. Response Time: The System should provide real-time updates on stock levels, with a response time of less than 3 seconds under normal load

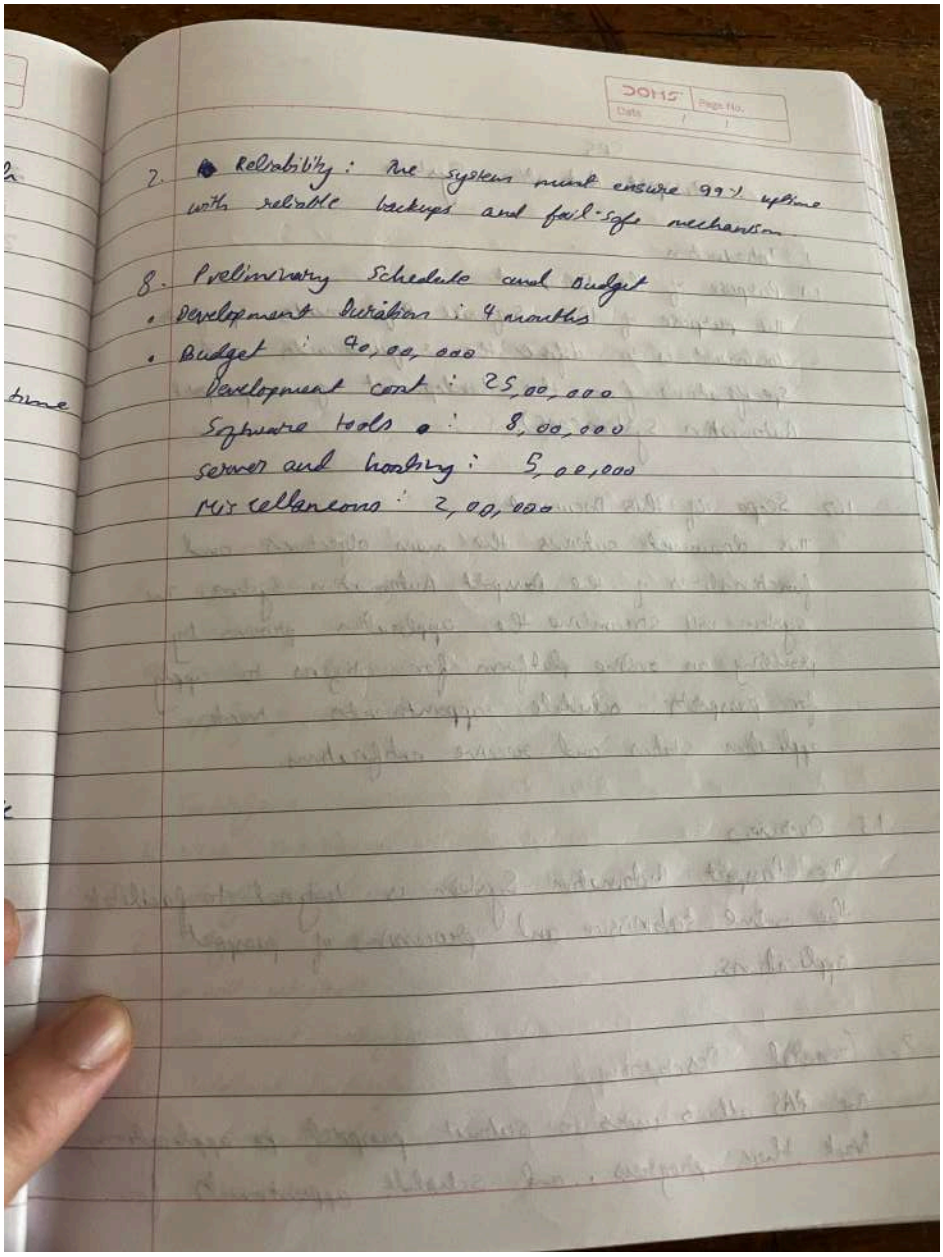
2. Data Handling: The system must be able to handle 50,000 stock records, with efficient querying and data processing.

6. Design Constraints

1. Platform Constraints: The System must be compatible with all major browsers.
2. Database: The System should use PostgreSQL.

7. Non-functional Attributes

1. Security: The system should employ encryption for sensitive data, secure login.



b.Advanced Class Diagram:

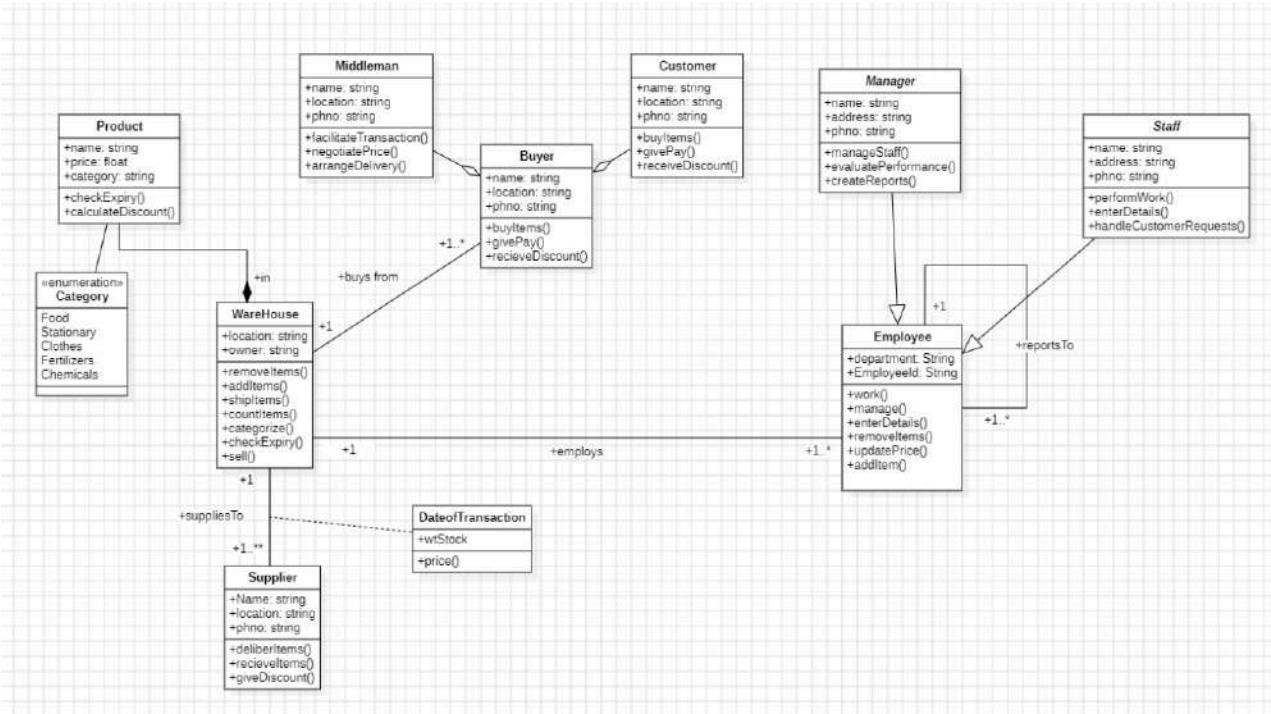


Fig 4.1:

Class diagram represents a system for managing a supply chain or retail business, showcasing the entities involved, their attributes, behaviors, and relationships. It includes classes like Product, Warehouse, Supplier, Middleman, Buyer, Customer, Employee, Manager, and Staff.

The Product class represents items categorized as Food, Stationary, Clothes, Fertilizers, or Chemicals, with methods to check expiry and calculate discounts. Products are stored and managed in the Warehouse, which handles inventory operations like adding, removing, and categorizing items. Suppliers provide products to the warehouse, while Buyers and Customers purchase them, facilitated by a Middleman who negotiates prices and arranges deliveries.

The Employee class represents workers managing warehouse or customer-related tasks, with roles divided into Staff (handling customer requests) and Managers (supervising staff and generating reports). The Date Of Transaction class records transaction details like stock and pricing.

- Warehouses being supplied by Suppliers and serving Buyers.
- Employees reporting to Managers and working within the warehouse or customer service.

This system models the complex interactions between entities in a supply chain, ensuring efficient inventory and transaction management.

b. Advanced State Diagram:

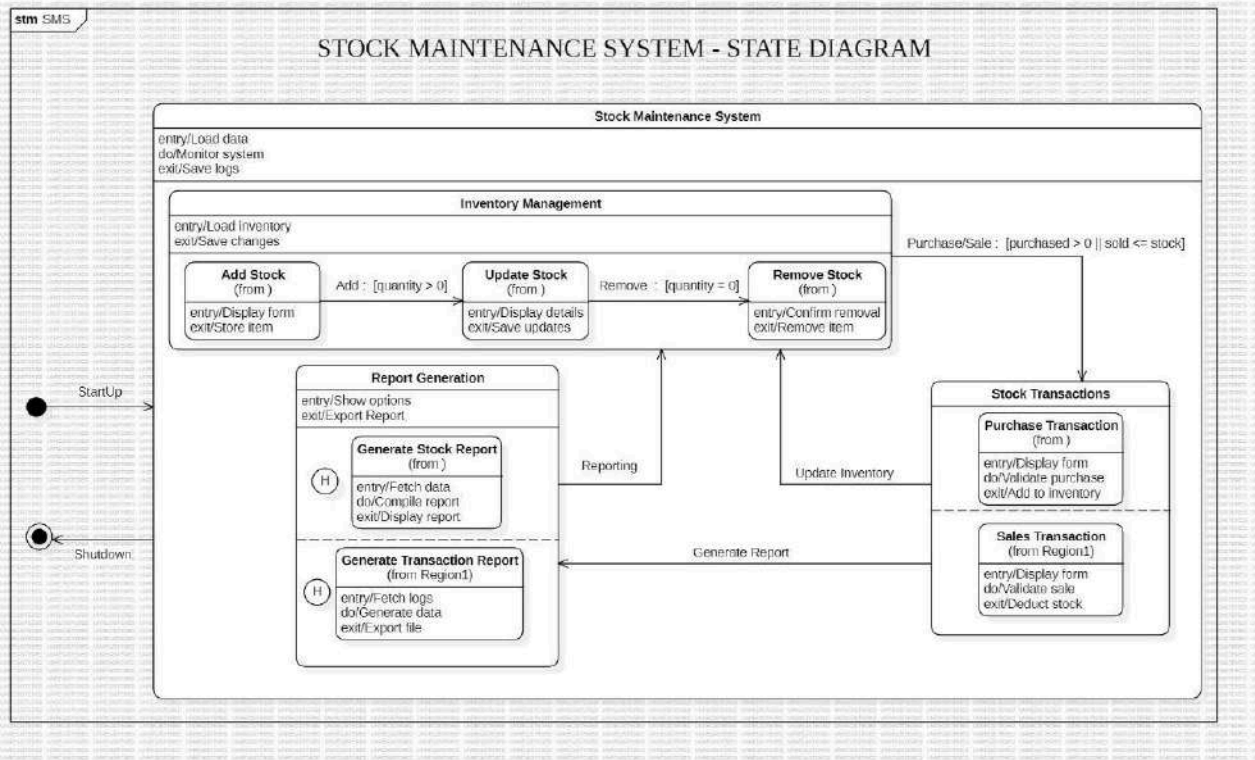


Fig 4.2:

The diagram illustrates the state machine for a Stock Maintenance System.

- **StartUp:** The initial state where the system loads data, monitors the system, and saves logs.
- **Shutdown:** The final state where the system exits and saves logs.
- **Inventory Management:** This state handles actions related to managing stock inventory, including adding, updating, and removing stock items.
- **Report Generation:** This state focuses on generating various reports, such as stock reports and transaction reports.
- **Stock Transactions:** This state manages stock transactions, including purchase transactions and sales transactions.

The diagram shows the transitions between these states, triggered by events like "entry/Load data," "exit/Save logs," "Add Stock," "Remove Stock," "Generate Stock Report," and others. These transitions depict the flow of the system through different states based on user actions or system events.

c. Use Case Diagram:

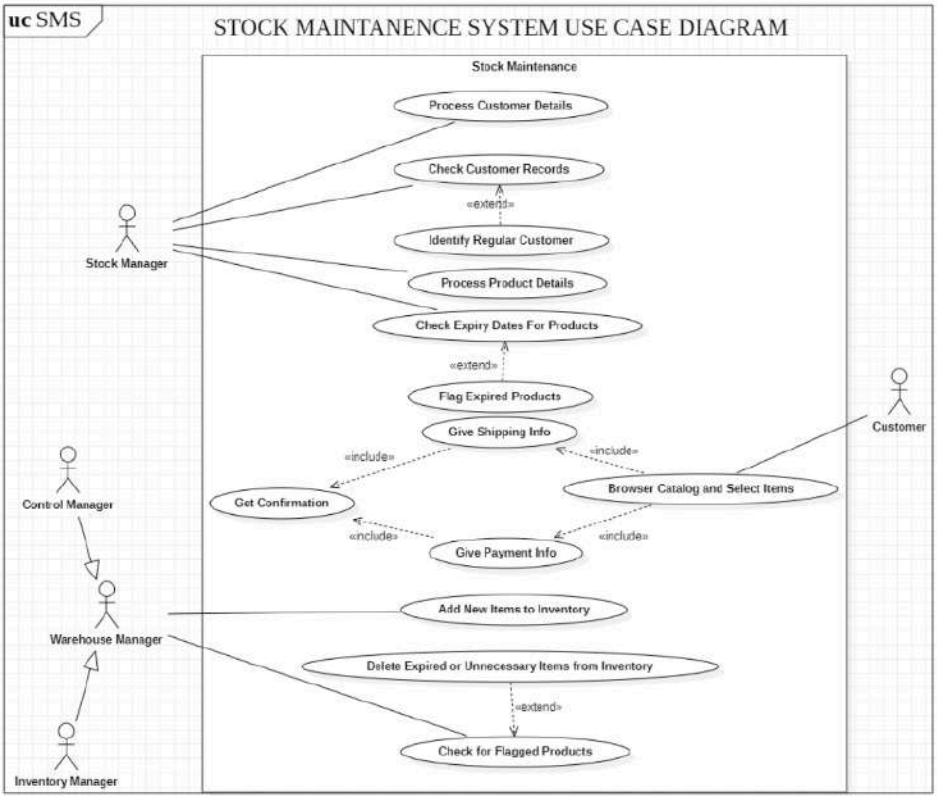


Fig 4.3:

Use case diagram represents a Stock Maintenance System, highlighting the interactions between key actors and system processes:

1. Actors:
    - Stock Manager: Handles customer and product details, checks expiry dates, and flags expired products.
    - Warehouse Manager: Adds new items to inventory.
    - Inventory Manager: Deletes expired or flagged items from inventory.
    - Customer: Browses the catalogue, selects items, and provides payment and shipping details.
2. Key Use Cases:
    - Stock Maintenance: Includes managing customer records, identifying regular customers, and processing product details.
    - Inventory Management: Adding new items and removing expired or unnecessary products.
3. Relationships:

- Include: Mandatory actions, such as payment info during order processing.
- Extend: Optional tasks, like flagging expired products during expiry checks.

d. Sequence Diagram:

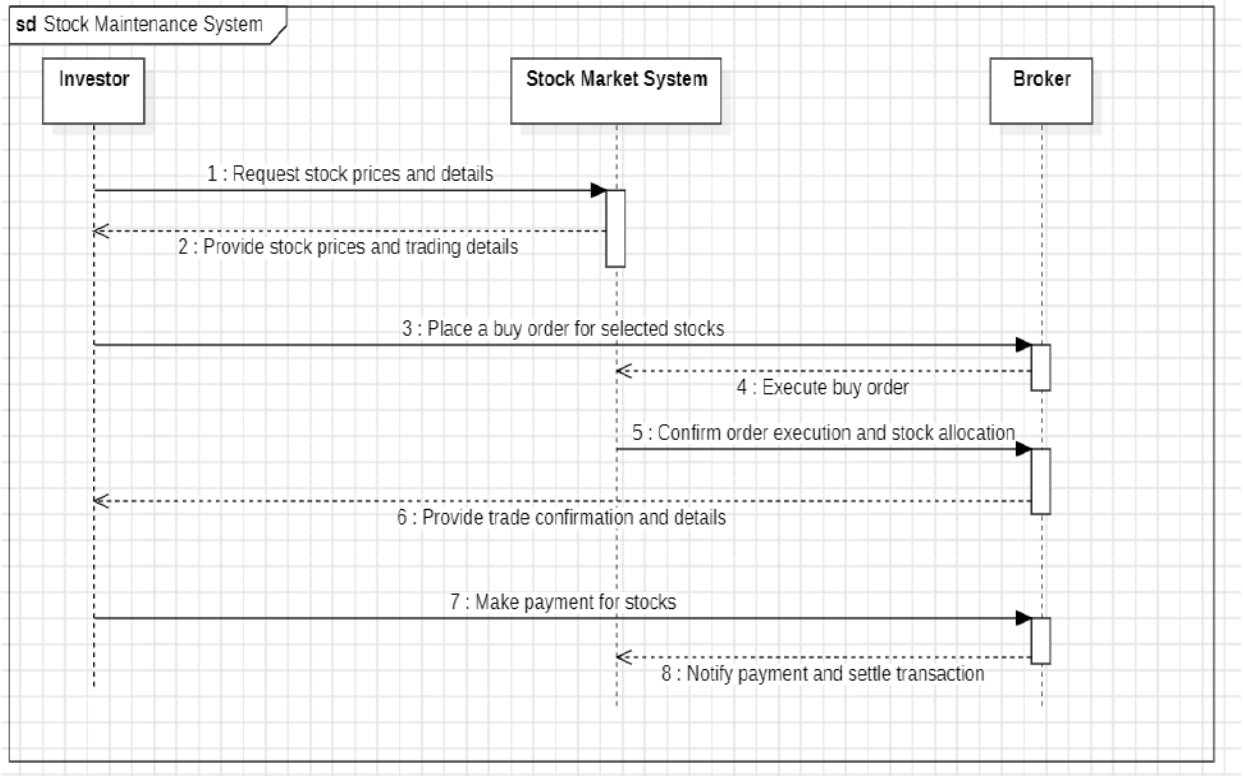


Fig 4.4:

e. Activity Diagram:

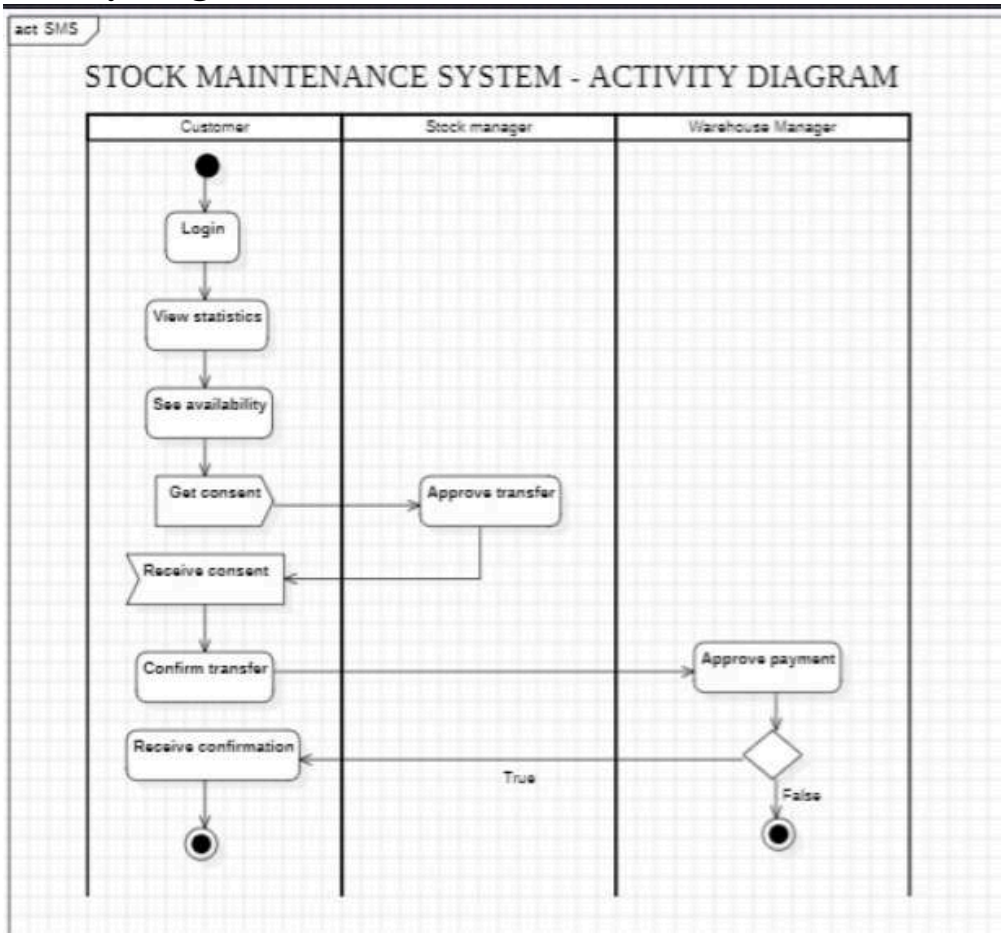


Fig 4.5:

The activity diagram outlines the workflow for transferring stock within a stock maintenance system.

The process begins with the Customer logging into the system. They then view relevant statistics and check the availability of the stock they wish to transfer. Next, the Customer requests consent from the Stock Manager for the transfer. The Stock Manager reviews the request and either approves or denies it. The Customer receives the decision from the Stock Manager.

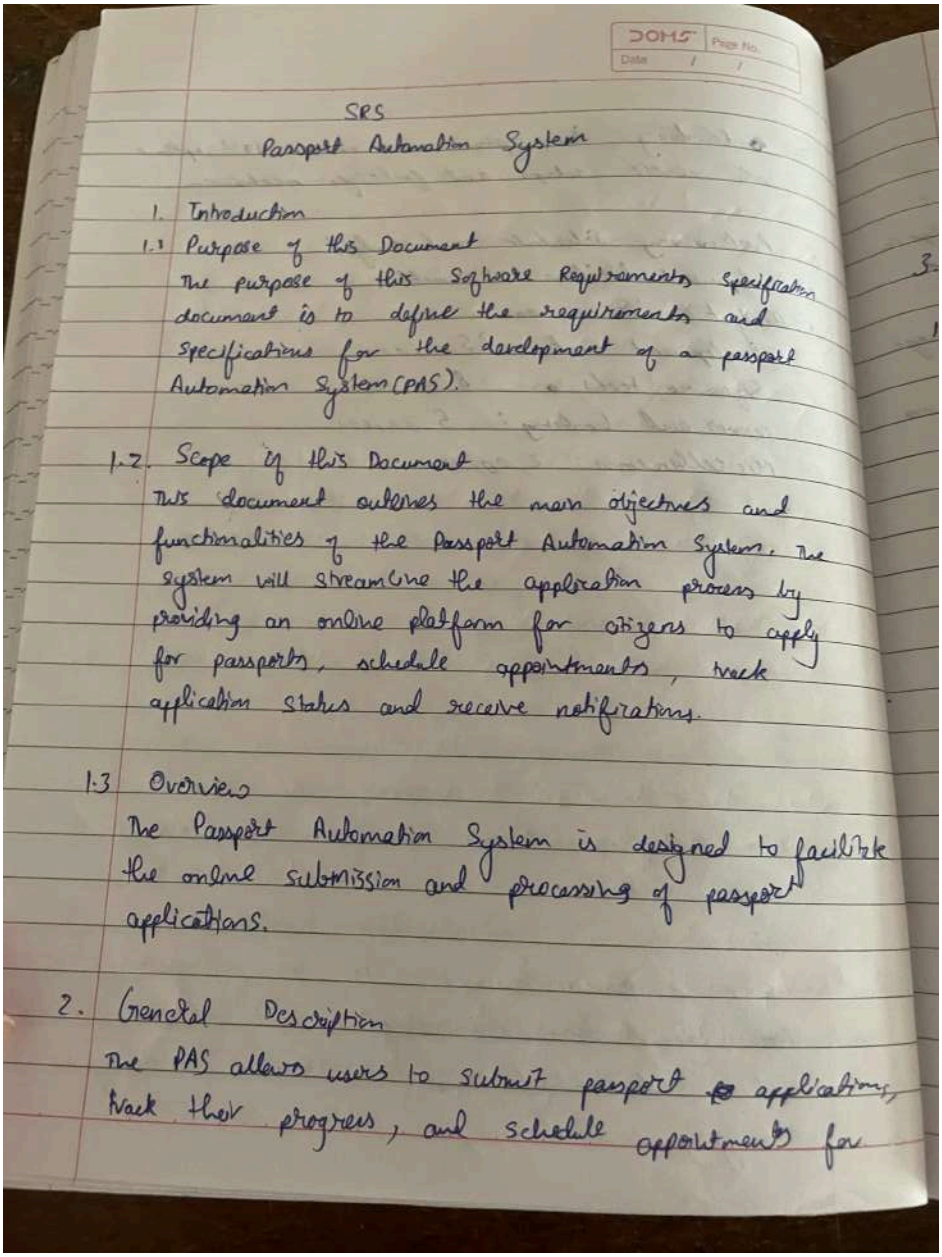
If the transfer is approved, the Customer confirms the transfer. Subsequently, the Warehouse Manager approves the payment for the transfer. Finally, the Customer receives confirmation that the payment has been approved.

If the Warehouse Manager does not approve the payment, the process terminates.

This activity diagram visually represents the sequential steps involved in the stock transfer process, highlighting the interactions between the key actors within the system.

5. Passport Automation

a. SRS Document:



SRS

Passport Automation System

1. Introduction

1.1 Purpose of this Document

The purpose of this Software Requirements Specification document is to define the requirements and specifications for the development of a passport Automation System (PAS).

1.2 Scope of this Document

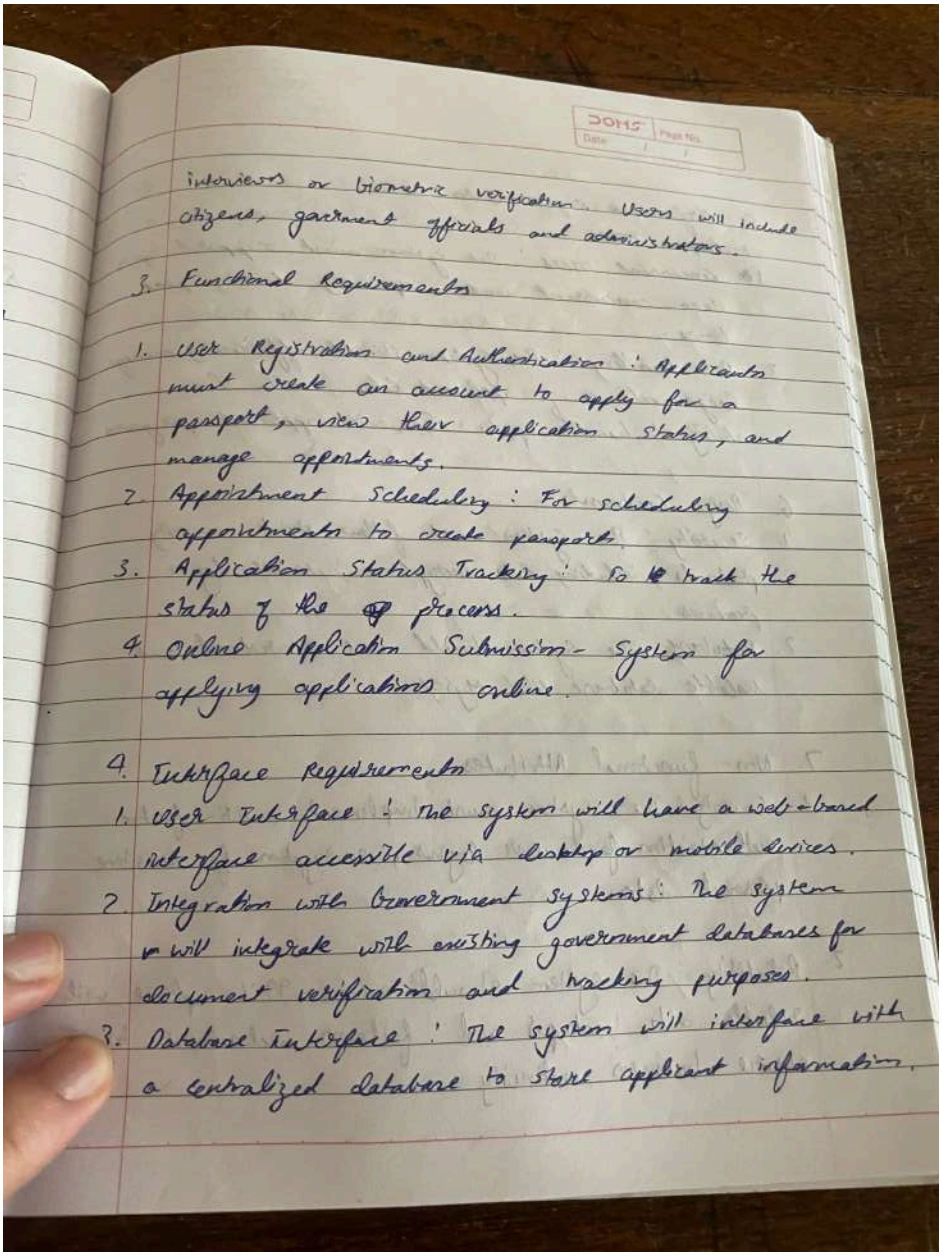
This document outlines the main objectives and functionalities of the Passport Automation System. The system will streamline the application process by providing an online platform for citizens to apply for passports, schedule appointments, track application status and receive notifications.

1.3 Overview

The Passport Automation System is designed to facilitate the online submission and processing of passport applications.

2. General Description

The PAS allows users to submit passport applications, track their progress, and schedule appointments for



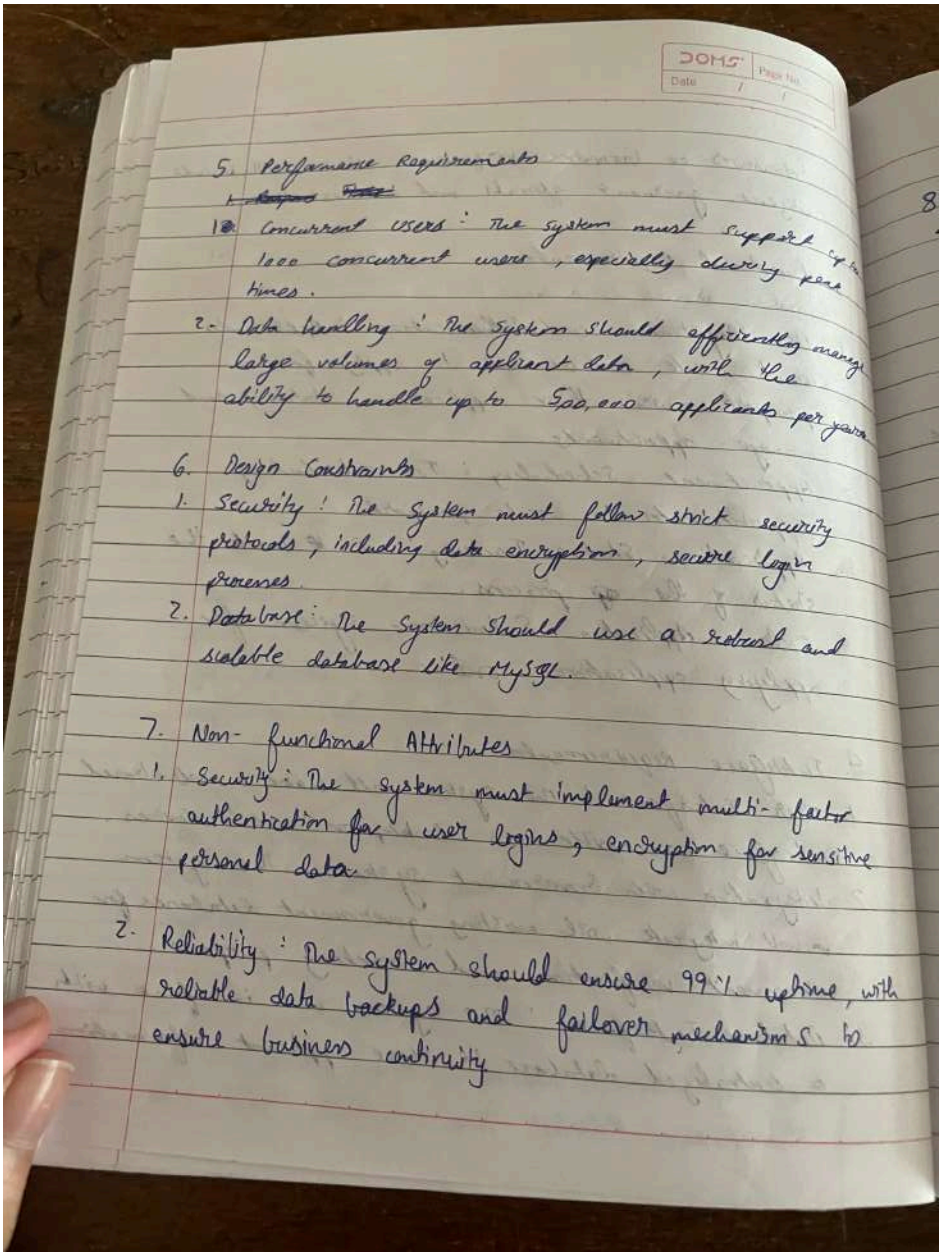
interviews or biometric verification. Users will include citizens, garment officials and administrators.

3. Functional Requirements

1. User Registration and Authentication: Applicants must create an account to apply for a passport, view their application status, and manage appointments.
2. Appointment Scheduling: For scheduling appointments to create passports.
3. Application Status Tracking: To track the status of the process.
4. Online Application Submission - System for applying applications online.

4. Interface Requirements

1. User Interface: The system will have a web-based interface accessible via desktop or mobile devices.
2. Integration with Government Systems: The system will integrate with existing government databases for document verification and tracking purposes.
3. Database Interface: The system will interface with a centralized database to store applicant information.



5. Performance Requirements

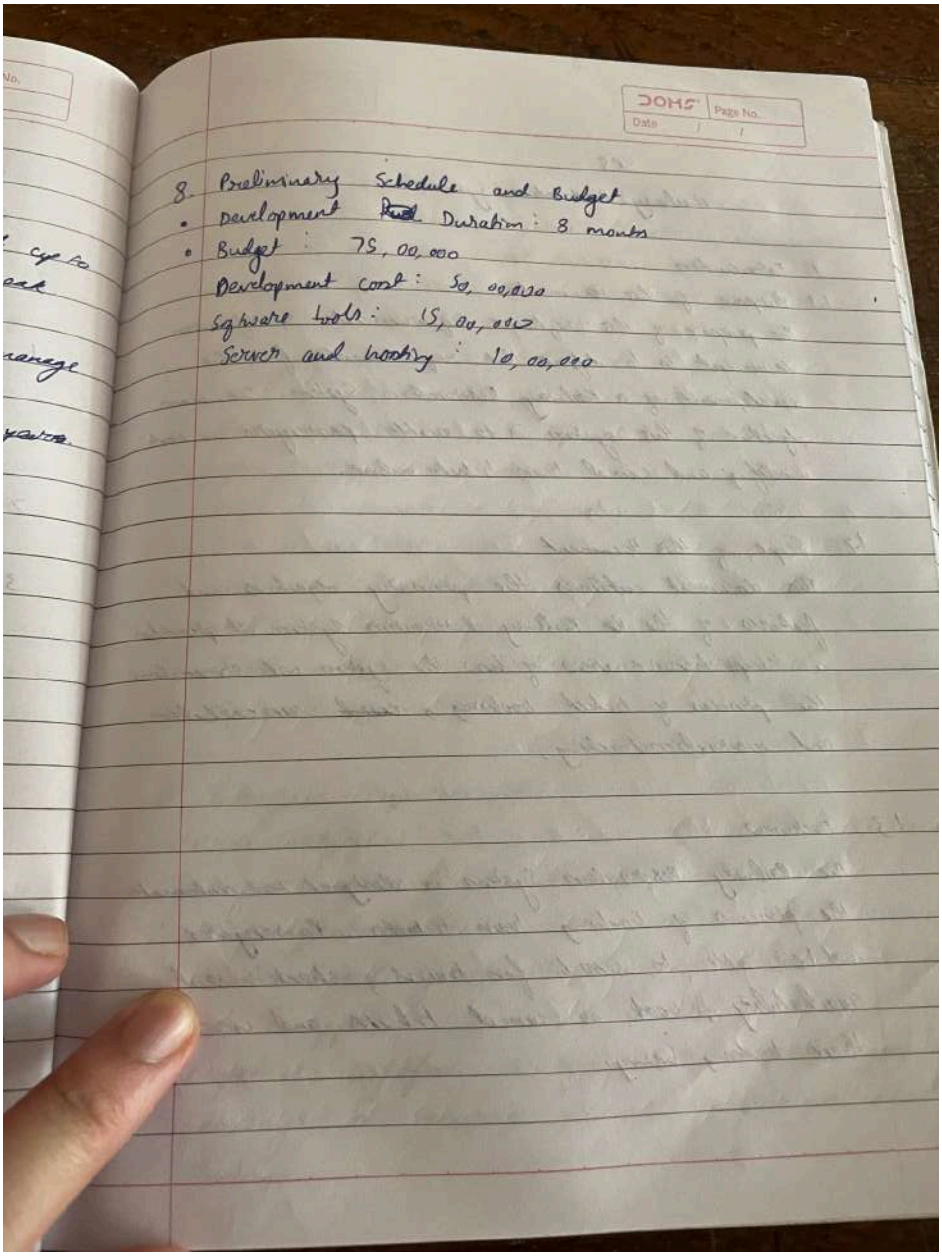
1. Concurrent users : The system must support up to 1000 concurrent users, especially during peak times.
2. Data handling : The system should efficiently manage large volumes of applicant data, with the ability to handle up to 500,000 applicants per year.

6. Design Constraints

1. Security : The system must follow strict security protocols, including data encryption, secure login processes.
2. Database : The system should use a robust and scalable database like MySQL.

7. Non- functional Attributes

1. Security : The system must implement multi- factor authentication for user logins, encryption for sensitive personal data.
2. Reliability : The system should ensure 99% uptime, with reliable data backups and failover mechanisms to ensure business continuity.



b. Advanced Class Diagram:

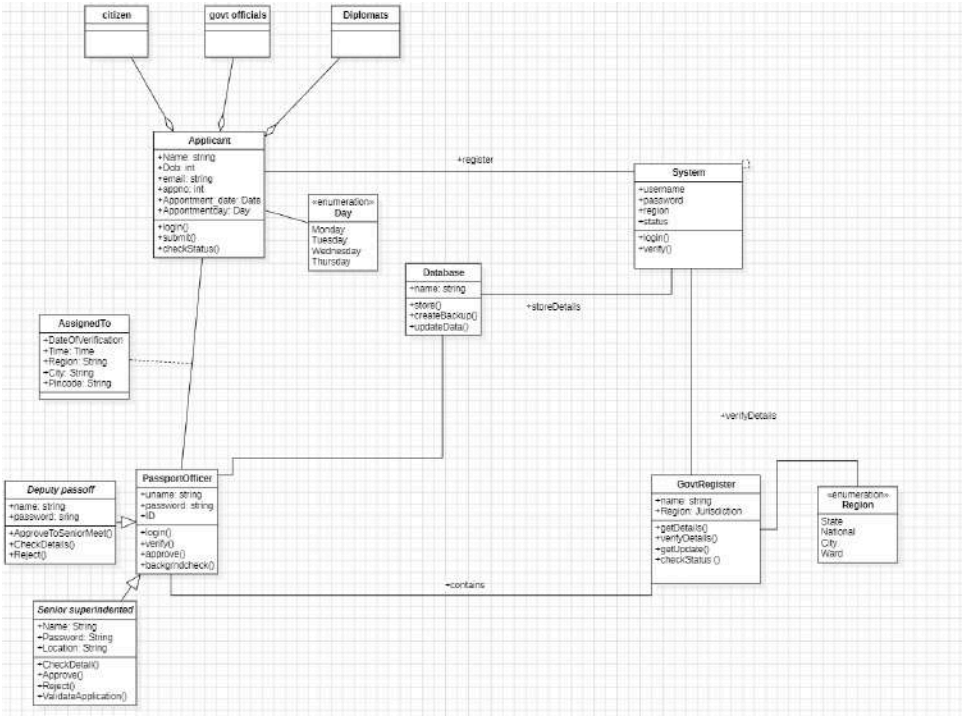


Fig 5.1

Passport management system that facilitates the process of applying for and managing passport applications. The key actors include applicants such as citizens, government officials, and diplomats, who interact with the system to submit their applications, book appointments, and track their status. Applicants provide essential details like their name, date of birth, and appointment information and can log in to the system to check updates on their application.

The system itself handles the core processes, such as user registration, login, and verification of details. All the information is stored and managed in a central database, which ensures that application data is securely stored, backed up, and updated as needed. Applications are assigned to specific Passport Officers based on the region, appointment schedule, and other factors. These officers are responsible for verifying documents, approving or rejecting applications, and updating the system with their decisions.

The process is overseen by Senior Superintendents, who validate and approve final decisions to ensure accuracy and compliance. Additionally, a Cover Register tracks applications based on their jurisdiction, categorizing them by state, city, or ward, and ensuring they are routed to the correct authorities.

This system organizes tasks efficiently by linking the roles of applicants, officials, and system components. It simplifies the workflow, ensures data security, and provides transparency for applicants to monitor their application progress.

c. Advanced State Diagram:

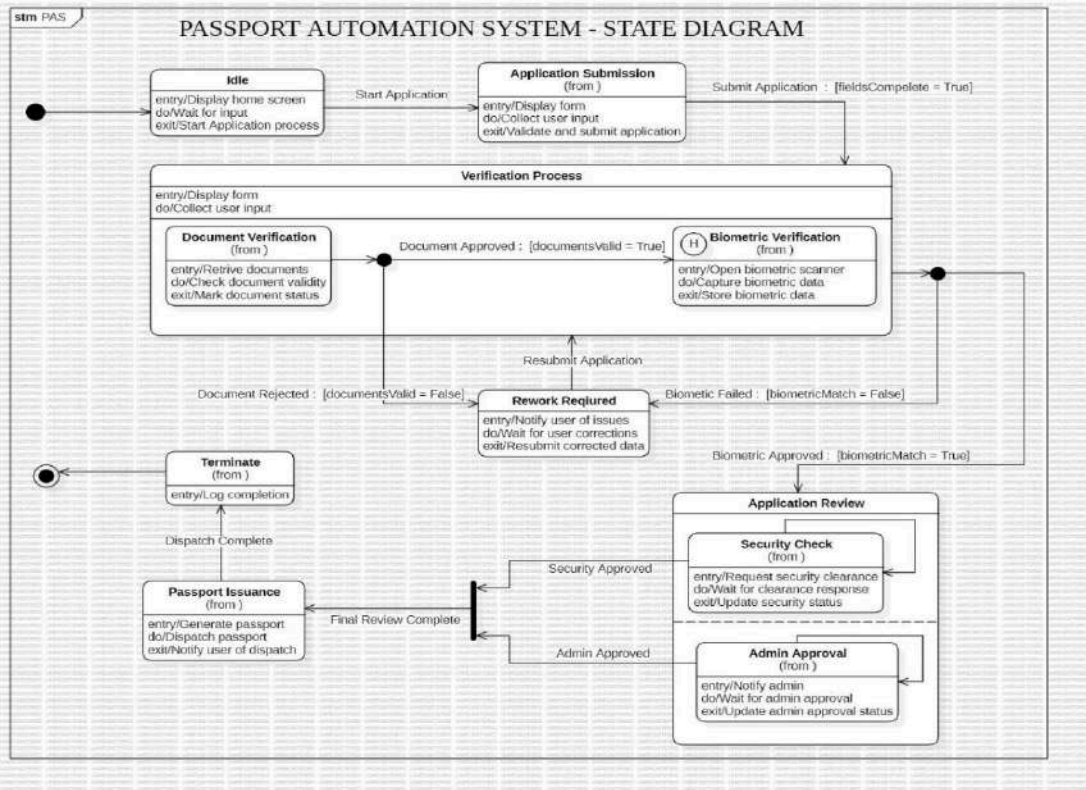


Fig 5.2

The diagram illustrates the state machine for a Passport Automation System. It shows the various stages a passport application goes through, from initial submission to final dispatch. The system includes states for application submission, document verification, biometric verification, security check, admin approval, and final review, with transitions between states based on user input and system decisions.

d. Use Case Diagram:

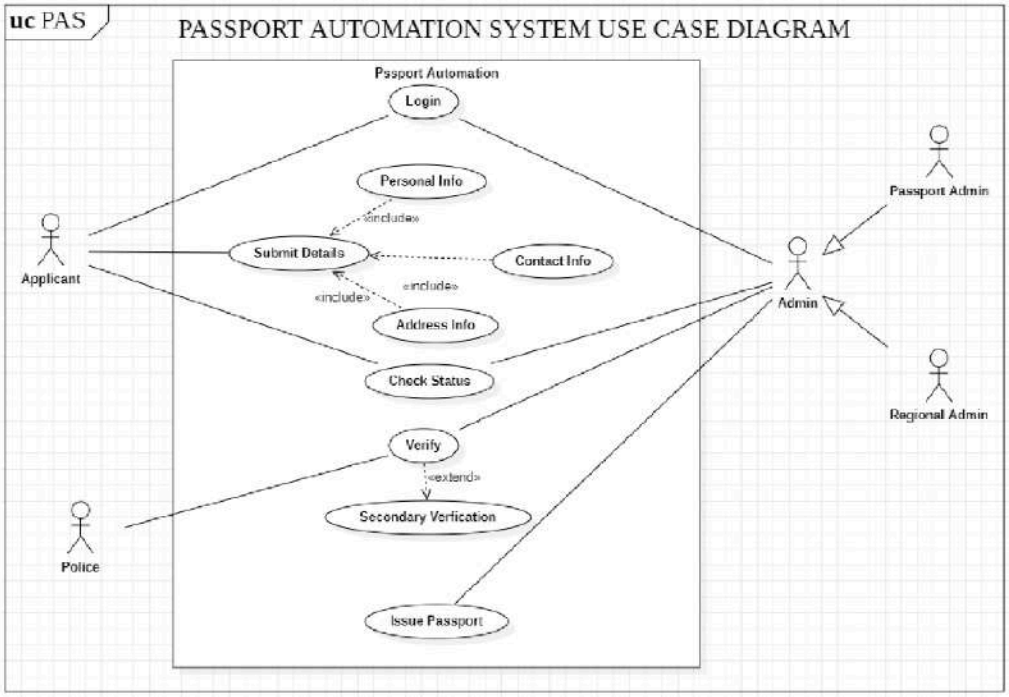

uc PAS
PASSPORT AUTOMATION SYSTEM USE CASE DIAGRAM
Passport Automation
Login
Personal Info
«include»
Submit Details
Contact Info
«include»
«include»
Address Info
Check Status
Verify
«extend»
Secondary Verfication
Issue Passport
Applicant
Police
Admin
Passport Admin
Regional Admin

Fig 5.3

e. Sequence Diagram:

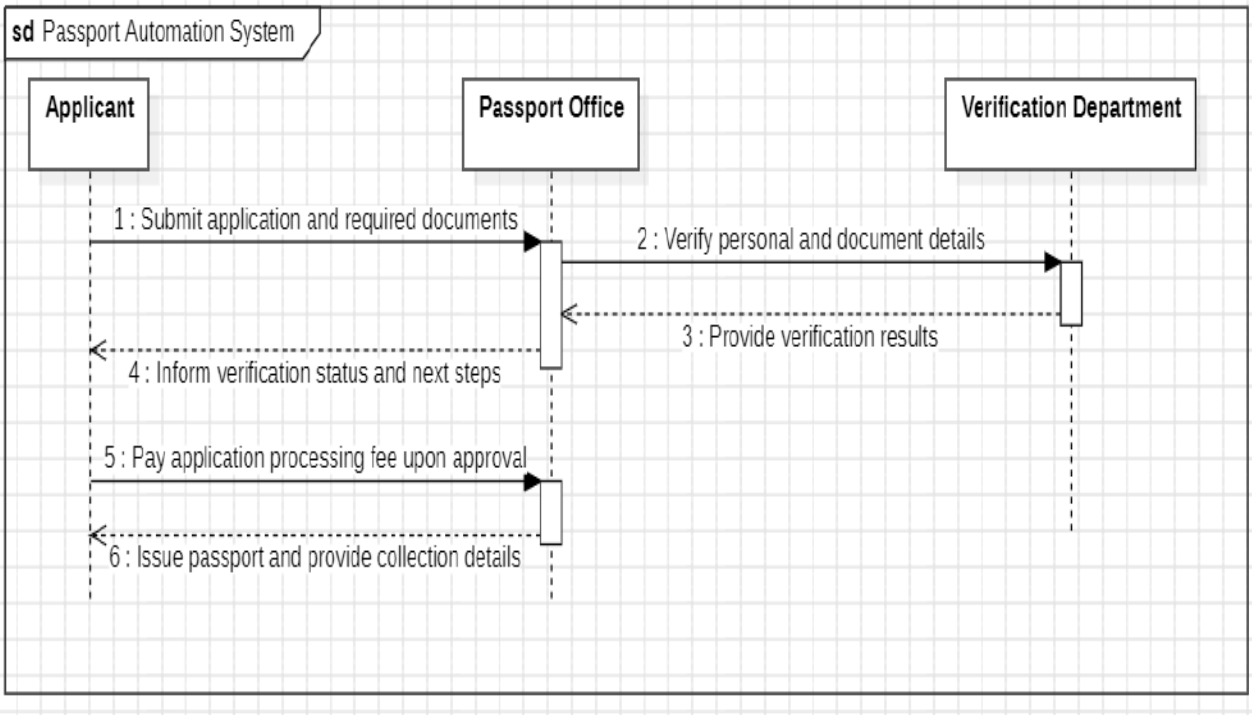


Fig 5.4

f. Activity Diagram:

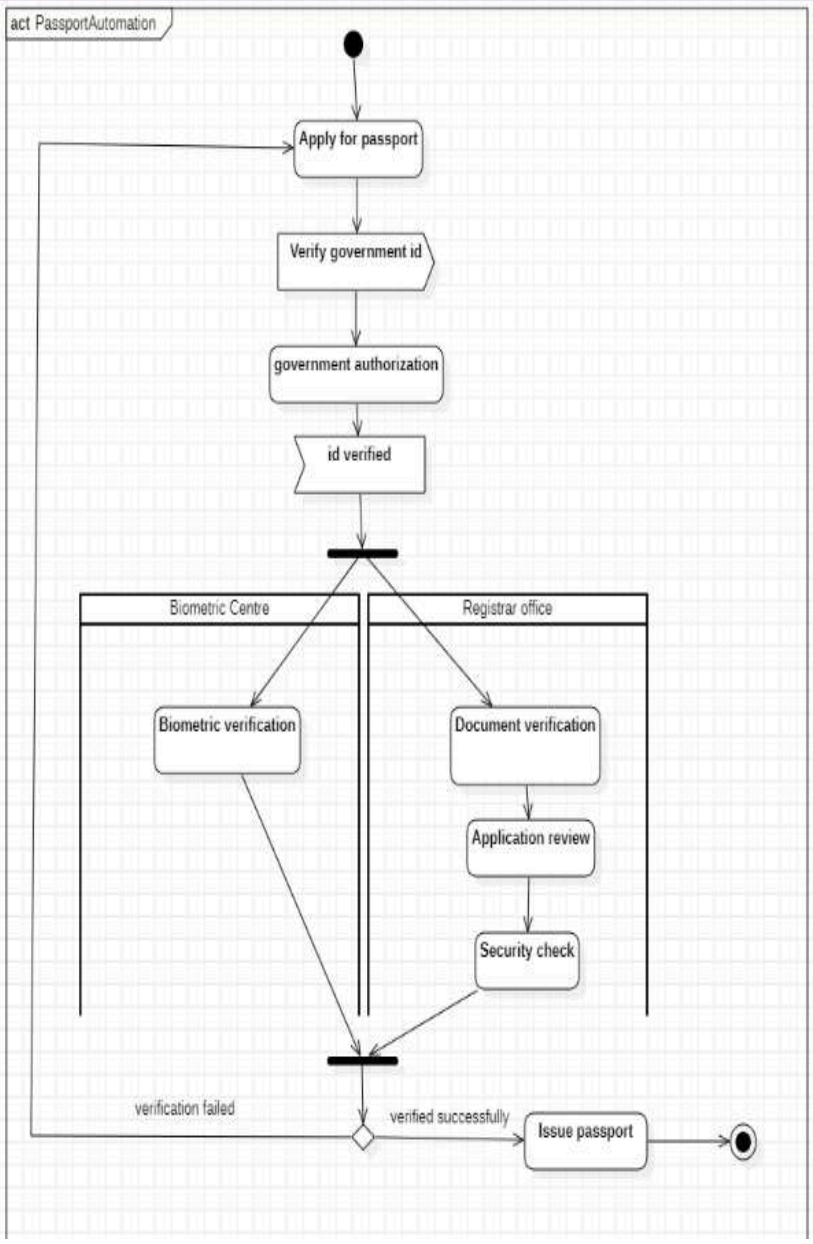


Fig 5.5

Passport Automation Activity that showcases the events that needs to take place to issue passport to a person. It includes biometric centre and registrar office where both activities must be completed indication by split and merge of control for passport to be verified successfully.
Verify Government Id signal is sent to the database and if true or legitimate the signal is received and Id is verified successfully.