

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



Lab Record

Object-Oriented Modeling

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Harsh Dev

1BM22CS107
Under the guidance

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
September-January 2025

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



CERTIFICATE

This is to certify that the Object-Oriented Modelling(23CS5PCOOM) laboratory has been carried out by Harsh Dev (1BM22CS107) during the 5th Semester Sep 24-Jan2025.

Signature of the Faculty Incharge:

Spoorthi DM

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

Table Of Contents

Sl.no.	Experiment Title	Page no.
1.	System Requirements Specifications	
2.	Class Modelling	
3.	State Modelling	
4.	Interaction Modelling: Use Case Models	
5.	Interaction Modelling: Sequence Models	
6.	Interaction Modelling: Activity Models	

1. SOFTWARE REQUIREMENTS SPECIFICATIONS (SRS)

Hotel Management System

Problem Statement:

Hotels often struggle with managing guest reservations, room availability, check-ins/outs, and billing manually. This leads to inefficiencies, overbookings, and potential customer dissatisfaction. The need exists for a Hotel Management System that automates these processes, ensuring accurate tracking of bookings, efficient room management, and seamless billing.

1. Introduction:

- 1.1 Purpose:**

The purpose of this document is to provide a detailed SRS for the Hotel Management System (HMS). This system will automate the management of hotel operations like room booking, check-ins/outs, and billing.

- 1.2 Scope:**

The HMS will reduce manual labor and improve efficiency. It will be accessible by hotel staff and management, ensuring faster customer service and accurate financial tracking. The development will require 3 months and an estimated budget of \$30,000.

- 1.3 Overview:**

HMS will provide a complete solution for hotel management, handling room inventory, guest details, billing, and reporting.

2. General Description:

The system will be used by hotel staff and management. It will have a user-friendly interface, allowing staff to check room availability, manage bookings, and process payments.

3. Functional Requirements:

- Room booking, check-in, and check-out system.
- Manage customer details and billing.
- Generate financial and occupancy reports.

4. Interface Requirements:

Graphical user interface (GUI) for hotel staff, with database integration for storing customer and room data.

5. Performance Requirements:

The system must handle up to 100 simultaneous users without performance degradation, processing room availability checks within 2 seconds.

6. Design Constraints:

The system must integrate with existing hotel security and camera systems for room monitoring.

7. Non-Functional Attributes:

- Security for customer data.
- Reliable and scalable architecture.

8. Preliminary Schedule and Budget:

Development will take 3 months, with an estimated cost of \$30,000.

Hotel Management System

Problem Statement

This system basically runs between customers and management. It improves business efficiency, room management, payment, food and delivery management. This online system overcomes the issue of manual system such as less security and speed. Also needs to automate and streamline and prevent double booking, delayed check ins, inconsistent guest service.

Software Requirement Specifications

(1) Introduction

1.1 Purpose of this document

Enhance the guest experience, optimal efficiency and real time room and service system, self driven decision system with minimized errors and centralized management.

1.2 Scope of this document

Streamline the daily operations of a hotel, operations like: staff management, guest review, guest cancellation and modifications, guest profile management, bedding and inuring and also providing security and trust and scalability. It can take upto \$100 and 1000 hours.

1.3 overview

2

HMV is a comprehensive software platform to automate hotel services. It can also tailor data for the users and services like:

- managing reservations
- guest check ins and check out
- guest relation management
- inventory and history and booking

(2) General description

Hotel management provides an interface for the users to book rooms on any day which is free and can be upgraded and also the user can view the room in VR mode and the user should pay on booking and only then it is confirmed. The hotel management can monitor staff allocation and shifts and can constantly improve on the user view and its more feasible by using this service.

(3) Functional Requirements

- one person undergoes database checks for room availability
- validate reservation details
- Assign rooms, process payments and update rooms status upon checkout
- update profiles with stay history
- System tracks progress, assigns tasks and monitor
- keep audit logs of all system operation

(4) Interface Requirements

Performance: Handle multiple concurrent users and response time

Security: Must use SSL for secure communications

Usability requirements: The interface should be intuitive for both

(5) External Interface Requirements

Third Party Payment gateways, Point of Sale Systems for initial

(6) Design Constraints

The system must be developed using the specified technology stack.
(JavaScript, Node.js)

The system must be compatible with major web browsers

(7) Non-functional Attributes

Security: The system shall implement user authentication

Reliability: The system shall ensure uptime and recoverability from failures. Data migration and the system shall ensure data accuracy

(8) Preliminary Schedule and Budget

Project Duration: The estimated development timeline is 3 months, including planning and development.

Budget Estimate: The projected cost for development is approximately \$6000, which includes technology and manpower.

Functional objectives:

1. To generate target from next level of analysis (e.g. by skill

Methodology:

1. Data collection and target market

2. Data analysis and reporting (including discussions)

3. Data collection and target market analysis

Activities:

1. Data collection and target market analysis

2. Data analysis and reporting (including discussions)

3. Data collection and target market analysis

Credit Card Processing System

Problem Statement:

Businesses face challenges in securely processing credit card transactions while ensuring speed and accuracy. Without proper fraud detection mechanisms and secure data handling, companies risk financial losses and compromised customer information. A robust Credit Card Processing System is required to handle transactions securely, detect potential fraud, and provide a fast, reliable user experience.

1. Introduction:

- 1.1 Purpose:**

This document defines the SRS for a Credit Card Processing System (CCPS), designed to facilitate secure credit card transactions.

- 1.2 Scope:**

The CCPS will allow businesses to process credit card payments with integrated fraud detection and transaction history tracking. Development will take 4 months with a budget of \$40,000.

- 1.3 Overview:**

The system will offer secure processing of credit card transactions, reduce fraud risks, and provide reporting tools.

2. General Description:

The system will be used by merchants and customers to process payments. It will ensure safe and fast transactions.

3. Functional Requirements:

- Secure credit card processing (charge, refund, etc.).
- Transaction history and reporting.

4. Interface Requirements:

The system will interface with bank APIs for transaction validation and fraud detection.

5. Performance Requirements:

It must process each transaction in less than 3 seconds with a 99.99% uptime.

6. Design Constraints:

Must comply with PCI-DSS standards for secure payment processing.

7. Non-Functional Attributes:

- High security.
- Scalability to handle peak transaction loads.

8. Preliminary Schedule and Budget:

Development is estimated at 4 months with a budget of \$40,000

Credit Card Processing Document (S)

Problem Statement

The current credit card processing system faces high transaction failure rate, slow processing times, and security vulnerabilities, risking PCI DSS non-compliance and potential fraud. Lack real time reporting.

(1) Introduction

1.1 Purpose of this document

Outline the functional and non-functional requirements for accredited card processing system, aims to improve transaction success rates, reduces time, provides real time reporting.

1.2 Scope of this document

The credit card processing will cater to merchants, customers and financial institutions providing a secure, and suitable payments solutions. It will handle transaction processing, fraud detection, data encryption.

1.3 Overview

The credit card processing system designed to facilitate secure and efficient processing of credit card transaction for various businesses, also handles payment authorizations, refunds and reporting. It will ensure compliance with payment card industry data security standards.

(2) General Description

Transaction Authorization: Validate credit card information and authorize

Transaction capture: Process and record completed details

Refund: Also have a refund process

Fraud and reporting: Generate reports on transaction details including sales and refunds

User characteristics

Merchants: Business owners who will use the system to process credit card transactions

Customers: End user making purchases using credit cards

Administrator: User managing the system monitoring transactions and generating reports

(3) Functional Requirements

FR1: System shall allow merchants to input and process credit card information

FR2: Authorize transaction within 2 seconds

FR3: Provide real time transaction

FR4: Issue of refunds for completed

transactions

FR5: The system shall generate detailed transaction reports for merchants

FR6: The system shall generate detailed transaction reports for administrators

(4) Interface Requirements

User Interface: The ODS will feature a web-based dashboard for merchants to manage transactions.

Payment Gateway Integration: The system shall interface with external payment gateways to facilitate transaction processing.

API: The system shall provide APIs for integration with e-commerce platforms and third-party applications.

(5) Performance Requirements

Transaction Processing time within 2 seconds under normal cases, and Max - 5 seconds in peak time.

Memory usage → The system should have enough memory space and utilize properly.

Error Rate → The system will maintain an error rate of less than 0.1%.

(6) Design Workarounds

The credit card details should not be stored in any database and should be masked.

It should support third-party gateways like ~~Paypal~~.

It should handle multiple transactions at a time and be efficient at some time.

(7) Non-functional Attributes

Security System will use AES-256 cryptosystem for storing sensitive data ensuring security, driven data transaction and storage. It should be reliable and maintain uptime of 99.99% and ensure uninterrupted payment.

It should be portable across different platforms including Mac OS / Win etc

(8) Preliminary Schedule and Budget

Initial 1-2 months gather measurements and initial design.

Form 3-5 months develop (will supersede)
Month 6 may be testing

API integration Month 6 May 2018
By end of 8 months cleanup and integrate
into system with code review

The estimated cost of project P.S. & 2000
including all works will be the cost of
borrowing all the expenses
of construction, plus 10% for the cost of
labor and materials + 5% for the cost of
travelling and the cost of maintenance
and cost of 10%

Library Management System

Problem Statement:

Manual tracking of library books, loans, and returns is time-consuming and prone to errors, leading to lost books and inefficient member services. The lack of an automated system results in delays and errors in managing book circulation and overdue fines. There is a need for a Library Management System to automate book cataloging, track borrowing and returning of books, and manage fines efficiently.

1. Introduction:

• 1.1 Purpose:

This document describes the SRS for the Library Management System (LMS), automating book lending and cataloging.

• 1.2 Scope:

The LMS will allow librarians to manage book inventory and track borrowing. The development will take 2 months with a budget of \$20,000.

• 1.3 Overview:

The LMS will facilitate book borrowing, returning, and inventory management for libraries.

2. General Description:

The system will be used by librarians and members for book cataloging and loan management.

3. Functional Requirements:

- Book cataloging and search functionality.
- Borrowing and returning management.

4. Interface Requirements:

GUI for librarians and members, with database integration for tracking books.

5. Performance Requirements:

The system must process book transactions within 1 second.

6. Design Constraints:

Integration with external book databases may be required.

7. Non-Functional Attributes:

- User-friendly interface.
- Secure and scalable architecture.

8. Preliminary Schedule and Budget:

Development will take 2 months with a budget of \$20,000.

Library management System

Problem Statement

Library is an outdated system with problems like misplaced books, loss of materials, user experiences difficulties in creating and reserving books due to an inefficient search functionality. Also collecting funds and managing membership is complicated. Furthermore, managing funds and optimizing collections based on user demands

(1) Introduction

1.1 Purpose of the document

The system will manage library operators such as book borrowing, returns, user registration and catalog management

1.2 Scope of this document

This provides detailed overview of system objectives, features and user interactions. LMS will automate management of library resources, including cataloging, borrowing, returning and placing books

1.3 Overview

The system will facilitate the operations of physical library by allowing libraries to track inventory, borrowers to borrowed books, returned books generate reports. It will support multiple user roles, administration, user

, members, efficient & platform to manage library activities digitally

(2) General description

User roles and permissions:

library staff: Librarians, assistants, administrators with varying access levels
members: General public with limited access to their accounts

The system will function as a centralized platform for managing library resources. It will enable users to search books etc. Librarians, administrators will have ability to add / update genuine books. System will also generate reports for library management. Easy tracking of books inventory and borrowing activities. It reduces manual errors and helps in efficient way

(3) Functional Requirements

- User registration, authentication, new users to register and existing users to log in using credentials
- Book catalog - administrator, can add update or genuine books
- Book Searching and Borrowing
- Book Return & fine calculation

• notification system

• Report generation (e.g. books issued, fines)

• User interface: system provides an easy

user based interface for both librarians and users

API: System will have API's for integration. System will send reminders for users to return books on time

(4) Performance Requirements

Load test: conduct load test to ensure it can handle peak usage

Disaster Recovery: Implement robust disaster recovery plans to minimize downtime in case of failure

(5) Design Constraints

Security: Prioritize security measures to protect sensitive data encryption

Scalability: design systems to accommodate future growth and increasing user demand

Integration: Consider time integration points with other library systems and external sources

(2) non functional attributes

Usability: conduct usability test to ensure system is interactive and easy to use

Maintainability: use clear coding practices, documentation and modular design

Interoperability: ensure compatibility with industry standards

(3) preliminary schedule and Budget

Month 1: requirements gathering and system design

Month 2: core module development

Month 3: Integrate, searching, borrowing and return features

Month 4: testing and QA

Month 5: development and user training

Budget Estimation: The project approach will cost \$ 1000 which includes testing, development and initial support costs

Phased approach: Prioritizing core features and addressing additional requirements in subsequent phase

Stock Maintenance System

Problem Statement:

Businesses often struggle to maintain accurate inventory levels, leading to stock shortages or excess stock, which affects operational efficiency. Without a proper system to track stock levels and generate alerts, companies can suffer from lost sales or overstocking costs. A Stock Maintenance System is required to automate stock tracking, alert staff about low stock levels, and generate reports for effective inventory management.

1. Introduction:

- **1.1 Purpose:**
This SRS defines a Stock Maintenance System (SMS) to manage business inventory.
- **1.2 Scope:**
The system will track stock levels, manage inventory, and generate low stock alerts. Development will take 3 months with a budget of \$25,000.
- **1.3 Overview:**
SMS will help businesses maintain accurate stock records and automate the replenishment process.

2. General Description:

The system will be used by warehouse and inventory managers to monitor stock levels.

3. Functional Requirements:

- Inventory tracking (incoming, outgoing).
- Low stock alerts.

4. Interface Requirements:

GUI for warehouse staff, with integration to supplier management systems.

5. Performance Requirements:

Inventory updates must occur within 1 second of stock change.

6. Design Constraints:

The system must be compatible with existing enterprise resource planning (ERP) tools.

7. Non-Functional Attributes:

- Reliable and scalable.
- Real-time data integrity.

8. Preliminary Schedule and Budget:

Development will take 3 months with a budget of \$25,000.

Problem Statement

Managing inventory effectively is crucial for businesses to ensure the availability of products, minimize holding costs and avoid stockouts. Stock maintenance system aims to streamline inventory management by providing real-time tracking of stock levels, automating order placements, generating insightful reports. This system will enhance operational efficiency, reduce human errors and improve decision-making capabilities.

SRS1. Introduction1.1 Purpose of this document

This document outlines software requirements for the system. It serves as guide for developers, stakeholders and end-users, detailing the system's functionalities, performance, and design constraints.

1.2 Scope of this document

The document covers objectives, functionalities, expected outcomes of Stock maintenance System. It includes an estimation of developments cost and time required, providing valuable insights for outcomes.

1.2 Inventory Stock

The system is designed to automate inventory tracking, manage stock levels, and facilitate order processing. It ensures real-time data accuracy, reduces manual efforts, and supports efficient stock management practices.

Q1 General description

The system aims to assist users in maintaining optimal stock levels, preventing overstocking or stockouts, users will benefit from automated alerts, detailed inventory reports, and user-friendly interface. The primary users include inventory managers, warehouse staff, and procurement teams.

Q2 Functional requirements

- Realtime inventory tracking and updates
- Automated stock alerts when stock levels fall below predefined thresholds
- Detailed reporting on stock levels, order history, and stock movements
- User authentication and role-based access control, showing privilege

- Integration with existing ERP, Symphem for seamless data flow
- Search and filter capabilities for quick access to stock information

(4) Interface Requirements

- A user friendly graphical interface for easy navigation and operation
- API's for integration with external systems
- Data import/export capabilities in standard formats (e.g. CSV, Excel)
- Notification interfaces for email and SMS alerts

(5) Performance Requirements

- Symphem should handle upto 10000 stock entries concurrently
- Real time processing with response time within 2 seconds for user actions
- minimal memory usage to ensure efficient performance
- Error rate below 0.1% for inventory calculations

(6) Design Constraints

- The system must be compatible with databases & languages
- Use MySQL for database management
- Compliance with industry standards for data security and privacy
- Implementation of RESTful APIs for external integration

(7) Non-functional Attributes

Security: Role-based access control, data encryption, secure login, post-ability, system should be accessible via browser and mobile devices

Reliability: 99.9% uptime with regular backups

Scalability: Modular design for easy updates and feature addition

Scalability: ability to handle increased data load and user numbers

(8) Preliminary schedule and budget

Schedule: Requirements Gathering: 2 meets

System Design: 3 meets

Development: 8 meets

Testing: 4 meets

Deployment: 2 meets

Budget: Estimated development cost ¹⁷
\$50000 with additional cost for
maintenance and updates of \$10000 per year
~~and a minimum of 10 hours of
development per month for maintenance of
existing or new analysis modules
and possibly extra development and
new analysis of existing modules
would require yet more time and cost
and also more (and growing) effort
allowing for concurrent new analysis
and use of modified regulations is
another option~~

Conclusion 1

forward with the results of
current analysis of active modules and
existing modules based on a
of draft regulations as it never
and the two available results
would consider the analysis products
obtained from the various
modules and for each of
the various products with the new
steps taken to generate better
and simpler for regulatory as well as
client products between and the
various and different

Passport Automation System

Problem Statement:

The manual passport application process is often slow, cumbersome, and prone to delays, causing frustration for applicants. Long waiting times for document verification and application status updates lead to inefficiency in passport issuance. There is a need for a Passport Automation System that simplifies the application process, automates status tracking, and integrates with document verification to ensure timely passport issuance.

1. Introduction:

- 1.1 Purpose:**

The document outlines the SRS for Passport Automation System (PAS), automating the passport application process.

- 1.2 Scope:**

PAS will handle passport applications, document verification, and appointment scheduling. Development will take 6 months with a budget of \$50,000.

- 1.3 Overview:**

The system will streamline passport applications, from submission to approval and issuance.

2. General Description:

The system will be used by applicants and passport authorities to manage applications.

3. Functional Requirements:

- Applicant registration and document submission.
- Status tracking and passport issuance.

4. Interface Requirements:

Integration with government databases for verification.

5. Performance Requirements:

The system must process application statuses within 2 seconds.

6. Design Constraints:

The system must adhere to government regulations for passport issuance.

7. Non-Functional Attributes:

- Secure and reliable.
- Scalable for high user loads.

8. Preliminary Schedule and Budget:

Development will take 6 months with a budget of \$50,000.

Passport automation system

Problem Statement

A manual process of passport application and issuance is time consuming, prone to errors and inefficient. A passport automation system aims to streamline and automate entire process from application submission to passport issuance. The system will enhance operational efficiency, reduce processing time, ensure data accuracy, empower user experience by providing a centralized platform for all passport related activities.

1. Introduction

1.1 Purpose of this document

The document outlines the software requirements for the passport automation system. It serves as a comprehensive guide for developer stakeholders and end-users, detailing system's functionalities, performance criteria and design constraints.

1.2 Scope of this document

Document covers the objectives, functionalities, expected outcomes of Passport automation system. It includes an estimation of development costs and how automated processing will provide valuable insights for customers.

2. System Description

System aims to assist users in submitting passport applications, tracking application status and issuing passports efficiently. Users will benefit from automated notifications, detailed application tracking, and user friendly interface. The primary users include applicants, passport officers and administrative staff.

3. Functional Requirements

- Online application submission and document upload.
- Automated validation of application details and document.
- Real-time tracking of application status.
- User authentication and role-based access control.
- Integration with government database for data verification.
- Notification system for updates via SMS and email.
- Appointment system for interview and document verification.

4. Interface Requirements

- A user friendly graphical interface for easy navigation and operation
- APIs for integration with email (System can use dB)
- data export (export capabilities on Standard format (e.g. PDF, XML))
- notification interface for email and SMS alerts

5. Performance Requirements

- The system should handle 1000 users
- Real-time processing with response times within 3 seconds for user actions
- minimal memory usage to ensure efficient performance
- Error rate below 0.1% for data processing and transmission

6. Design Requirements

- The system must be compatible with Windows, Mac, Linux OS
- Use of PostgreSQL SQL for database mgmt
- Compliance with government regulations for data security and privacy

- Implementation of RESTful APIs for software integration

7. Non Functional Attributes

Security: Rule-based access control, data encryption and secure login

Reliability: System should be accessible via web browsers and mobile devices

Reliability: 99.9% uptime with regular backups

Reliability: Modular design for easy updates and feature additions

Scalability: Ability to handle increased data loads and user numbers

8. Preliminary Schedule and Budget

Schedule: Requirements Gathering: 2 weeks

System Design: 4 weeks

Development: 10 weeks

Testing: 5 weeks

Deployment: 3 weeks

Budget: Estimated development cost is

\$ 100,000. Additional cost for maintenance and updates cost upto \$ 20,000/year

2. CLASS MODELLING

Hotel Management System

Classes:

- **Hotel:** Represents the overall hotel, with attributes like name, location, etc.
- **Room:** Represents individual rooms in the hotel with attributes like room number, type, price, and availability.
- **Guest:** Represents customers staying at the hotel, with attributes like name, contact information, and booking details.
- **Booking:** Represents the booking details, containing the check-in date, check-out date, and guest information.
- **Service:** Represents services provided by the hotel (e.g., room service), with attributes like service type and cost.
- **Invoice:** Represents billing details, with attributes like total amount, payment method, etc.

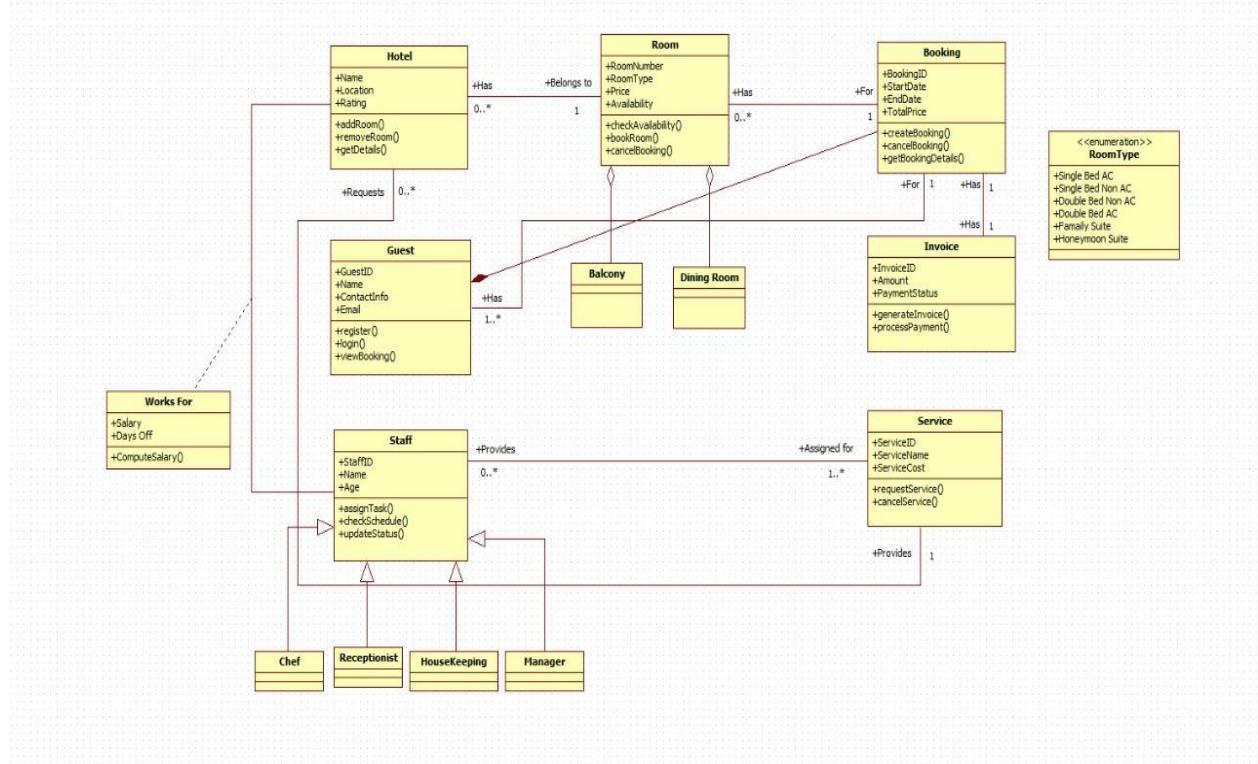
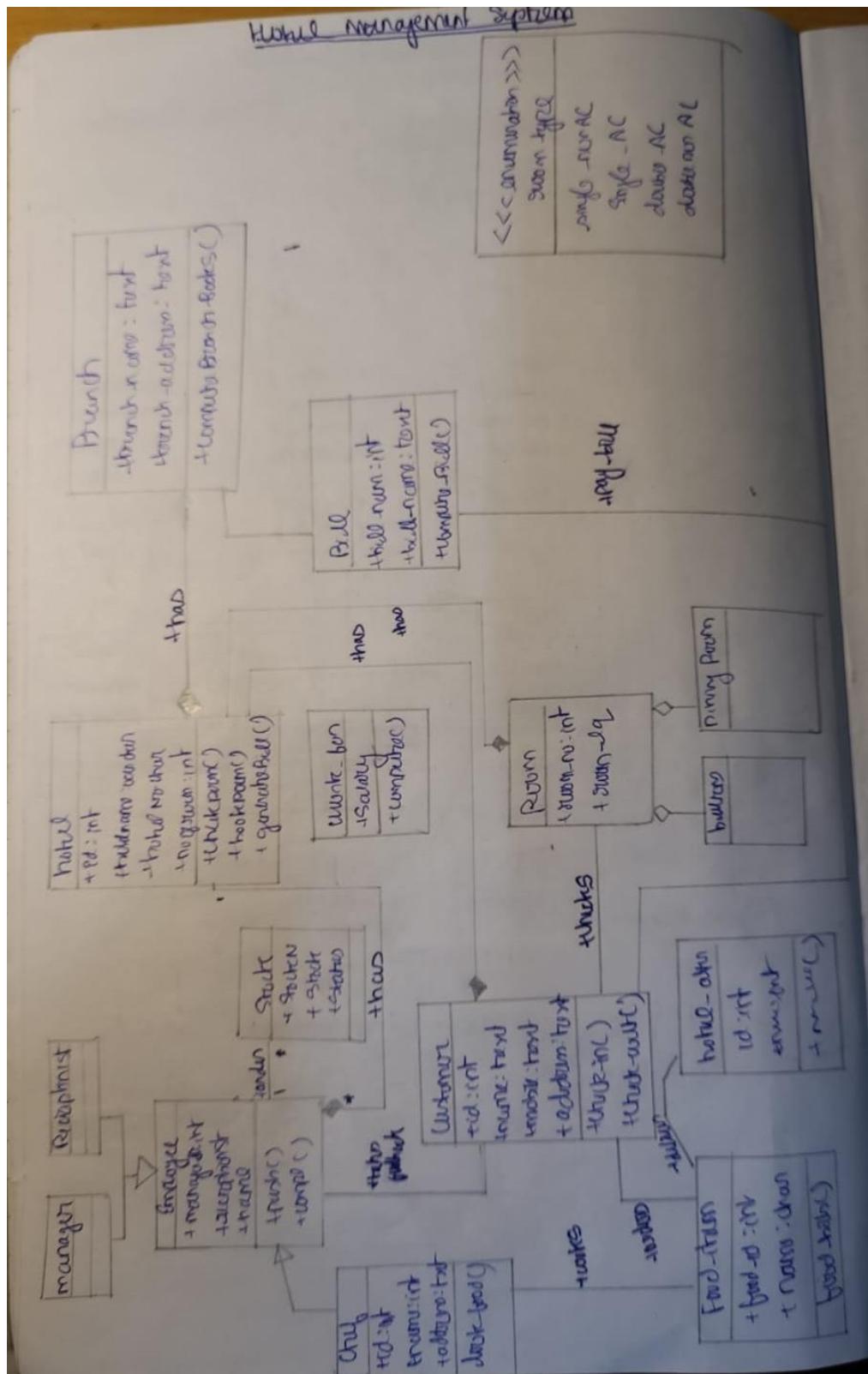


Figure 2.1 - HMS Class Diagram



Credit Card Processing System

Classes:

- **Customer:** Represents the cardholder, with attributes like name, card number, and billing address.
- **Transaction:** Represents individual credit card transactions, with attributes like transaction ID, amount, date, and status.
- **Merchant:** Represents a business or vendor accepting payments, with attributes like merchant ID, name, and location.
- **Bank:** Represents the bank issuing the credit card, with attributes like bank name and account details.
- **Payment Gateway:** Represents the system responsible for processing payments, with attributes like gateway ID and API details.

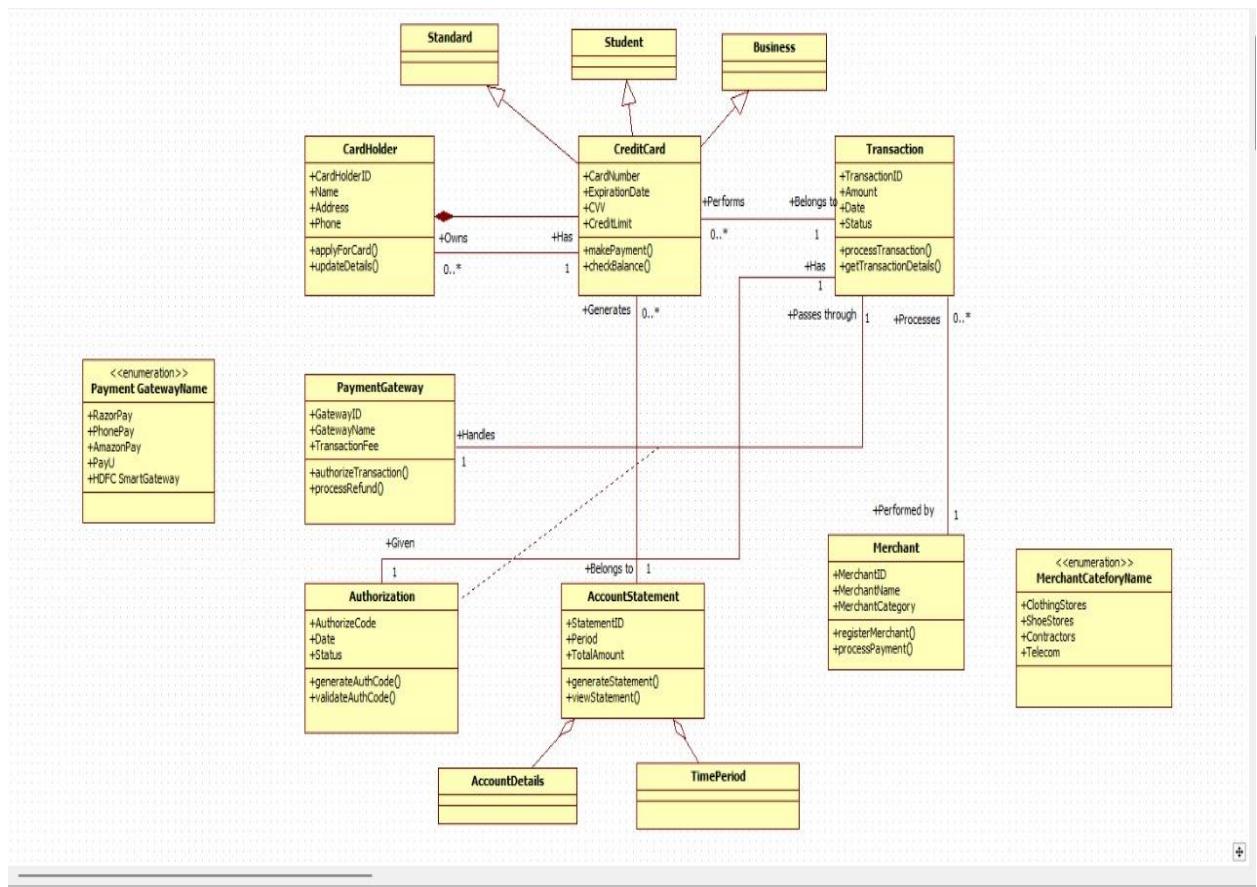
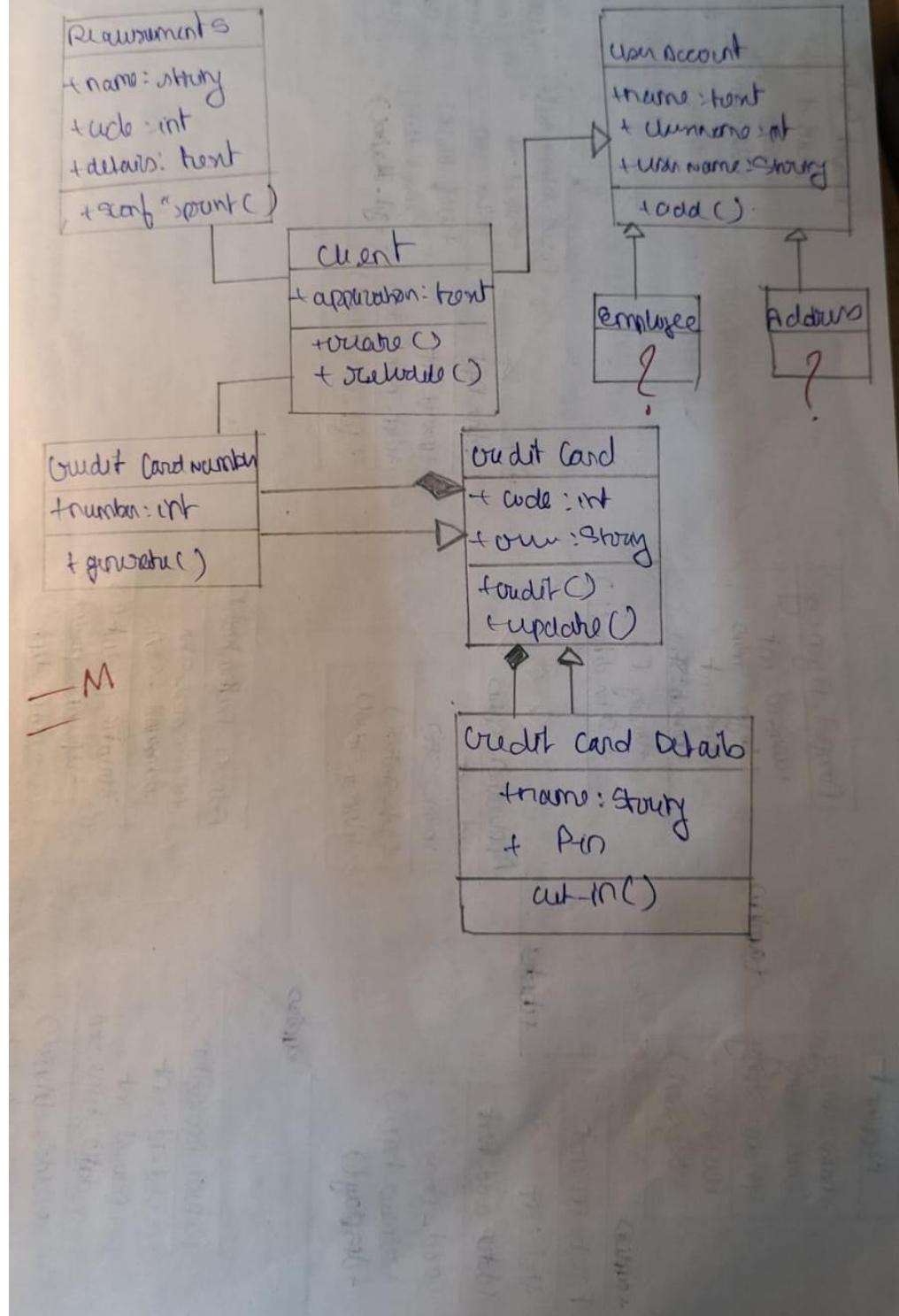


Figure 2.2 – CCPS Class Diagram

Credit Card Processing System



Library Management System

Classes:

- **Library**: Represents the overall library system, with attributes like name, location, and catalog.
- **Book**: Represents individual books in the library, with attributes like title, author, ISBN, and availability.
- **Member**: Represents a library member, with attributes like name, membership ID, and contact details.
- **Loan**: Represents the loan of a book, with attributes like loan ID, issue date, and due date.
- **Fine**: Represents a fine for overdue books, with attributes like fine amount and payment status.

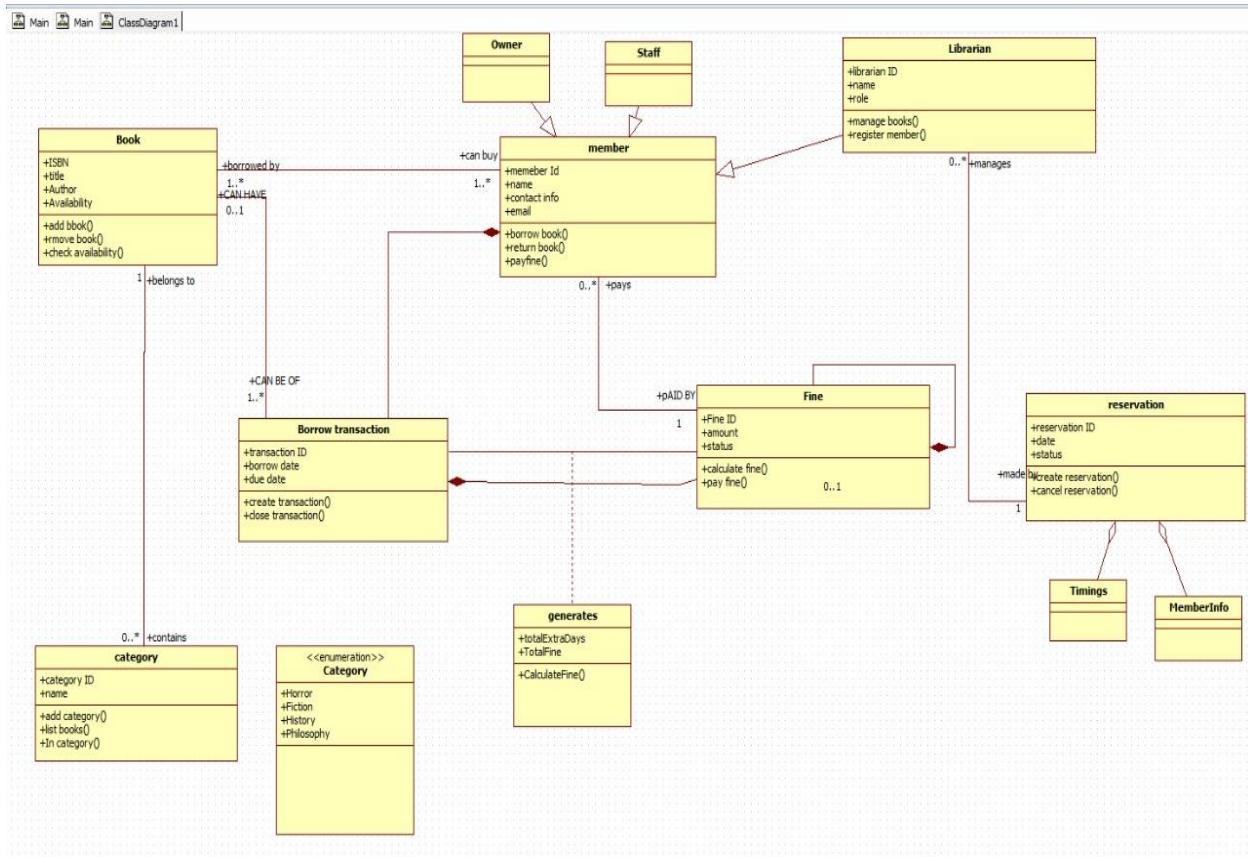
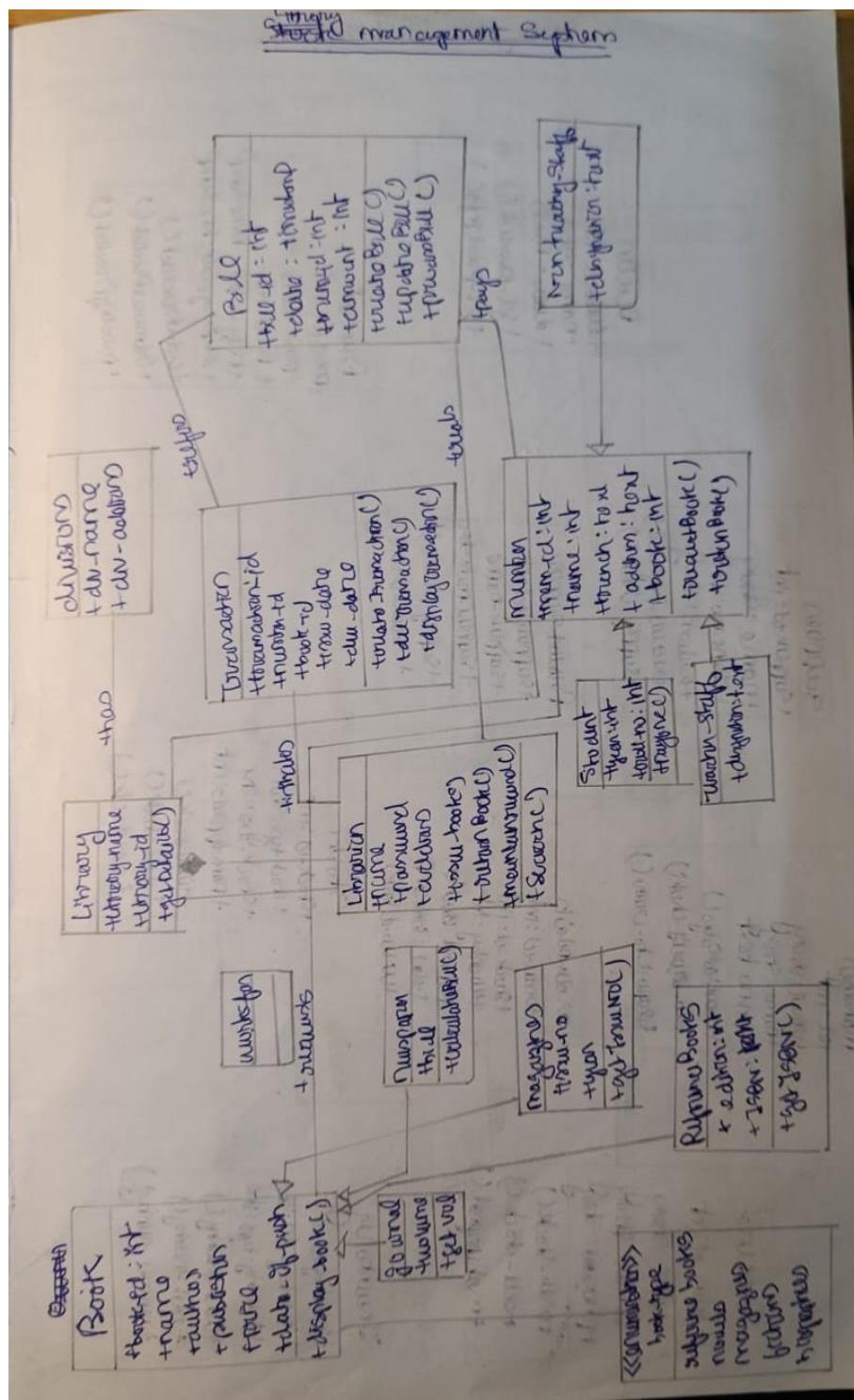


Figure 2.3 – LMS Class Diagram



Stock Maintenance System

Classes:

- **Warehouse**: Represents the storage location for stock items, with attributes like warehouse ID and location.
- **Item**: Represents individual stock items, with attributes like item ID, name, quantity, and price.
- **Supplier**: Represents the entity supplying items, with attributes like supplier name and contact details.
- **Order**: Represents orders placed for restocking, with attributes like order ID, date, and items ordered.
- **Inventory**: Represents the overall stock, with attributes like total stock levels and low stock alerts.

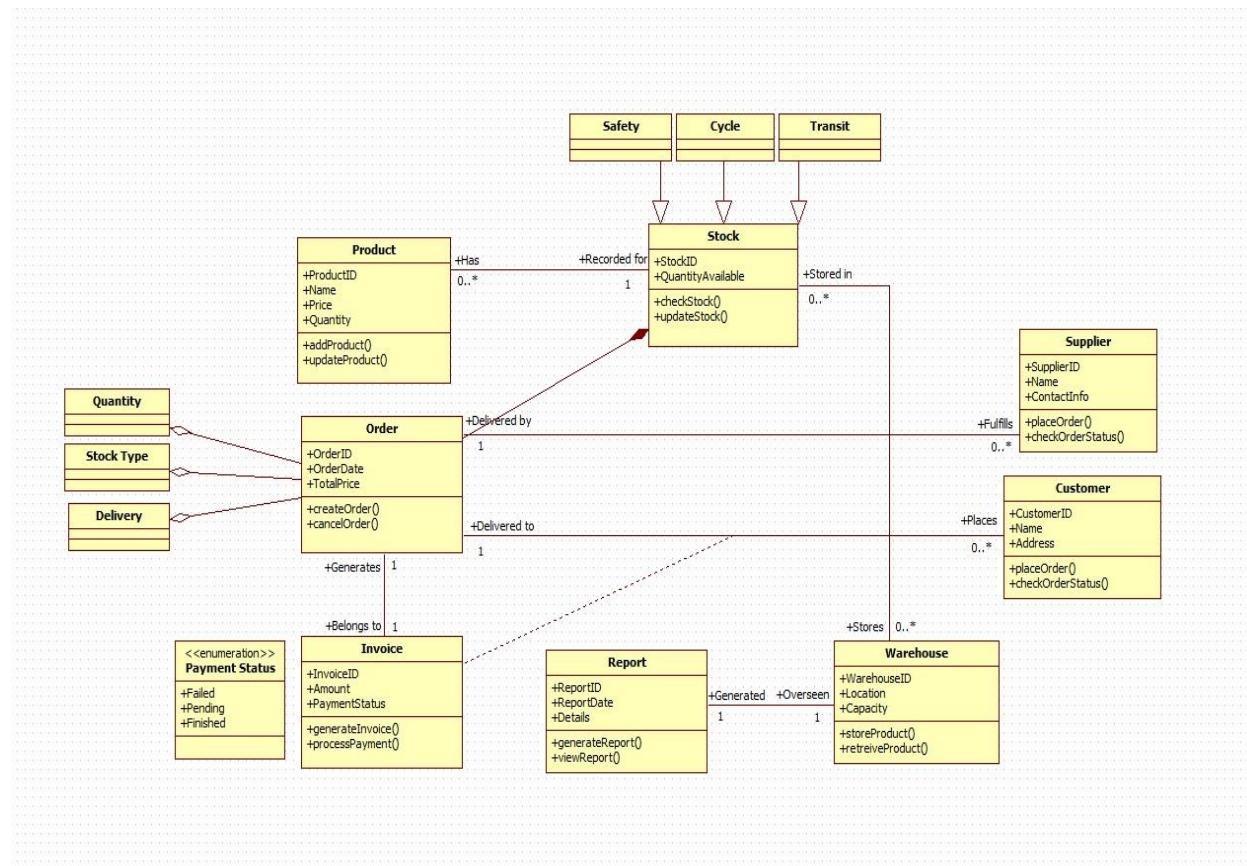
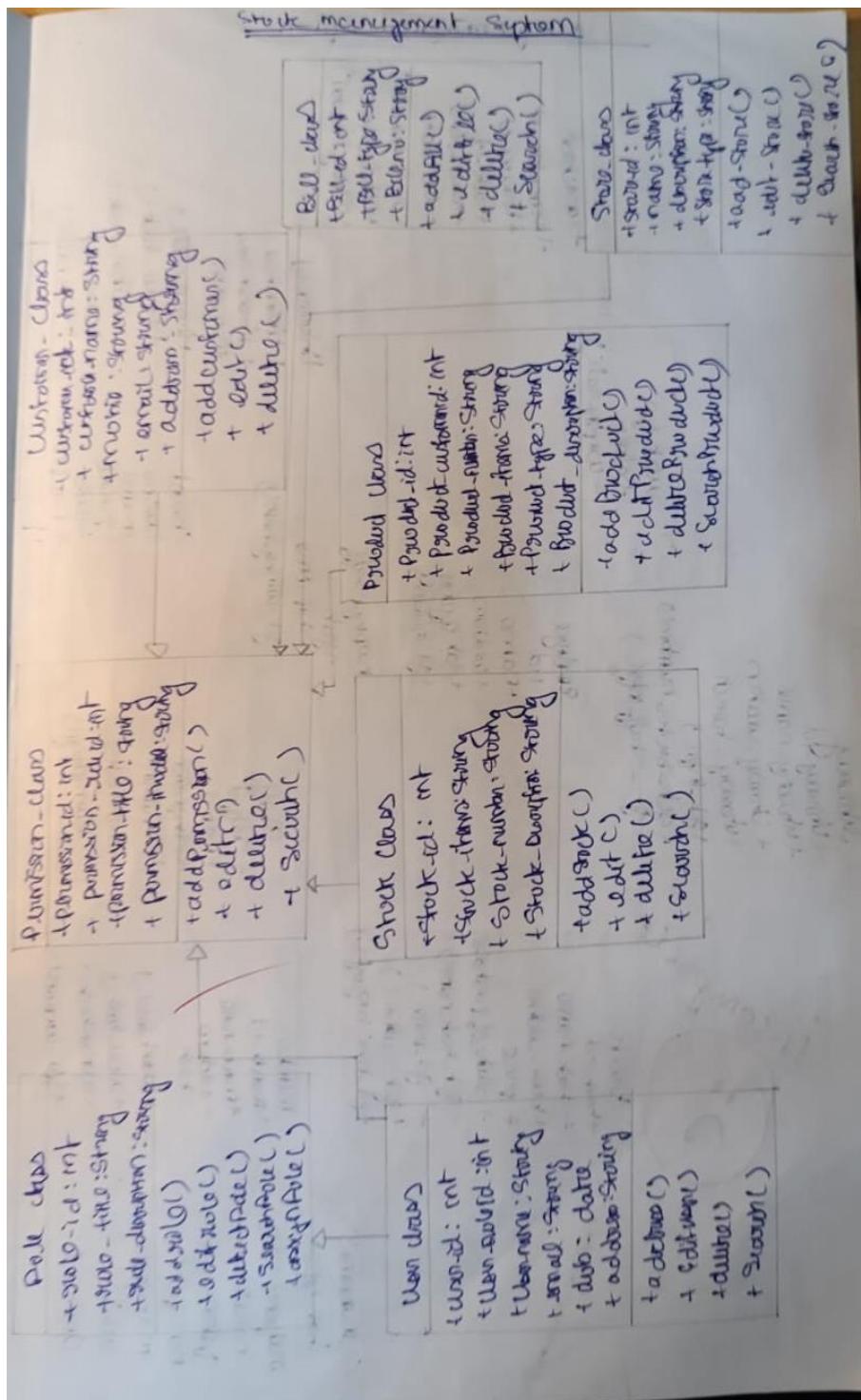


Figure 2.4 – SMS Class Diagram



Passport Automation System

Classes:

- **Applicant:** Represents an individual applying for a passport, with attributes like name, address, and ID proof.
- **Application:** Represents the passport application, with attributes like application ID, status, and submission date.
- **Appointment:** Represents an appointment for document verification, with attributes like appointment ID, date, and time.
- **Document:** Represents documents submitted by the applicant, with attributes like document ID, type, and status.
- **Passport:** Represents the issued passport, with attributes like passport number, issue date, and expiration date.

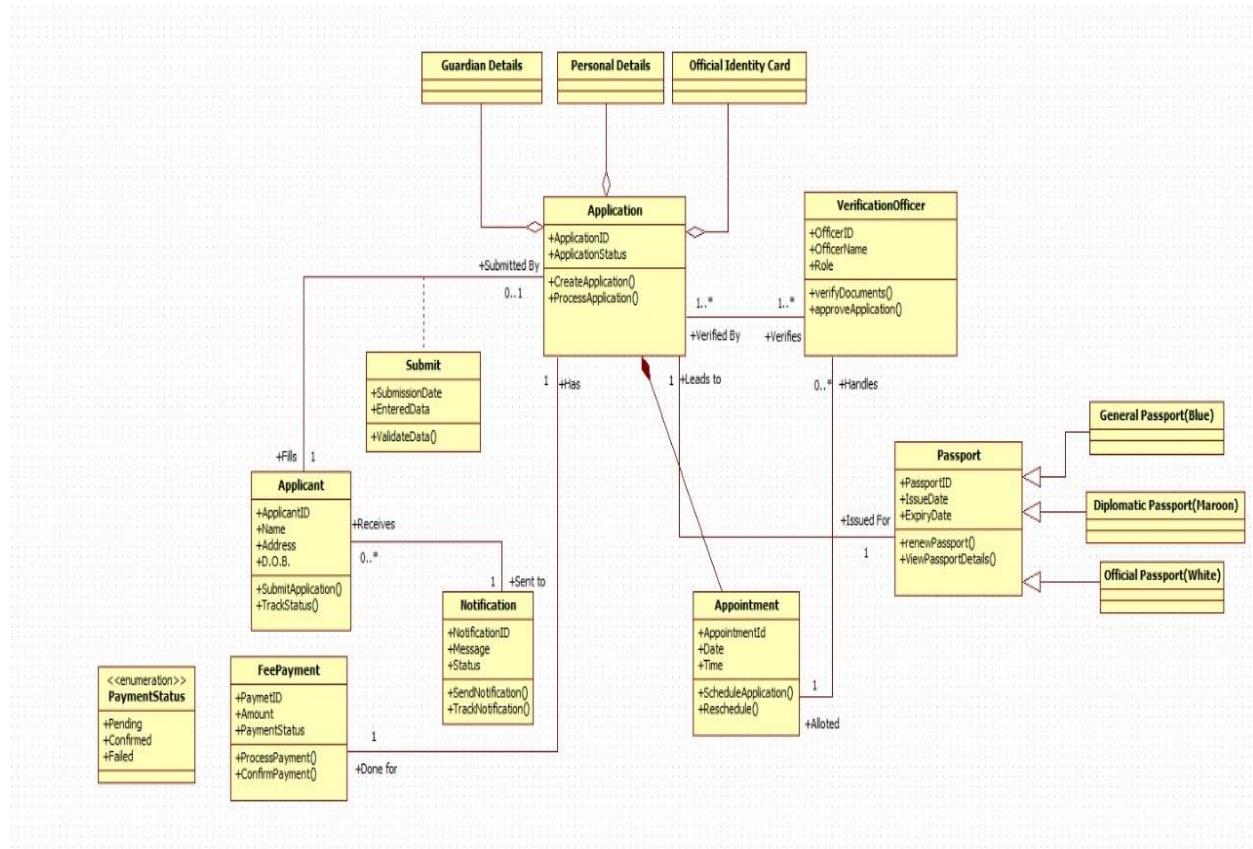
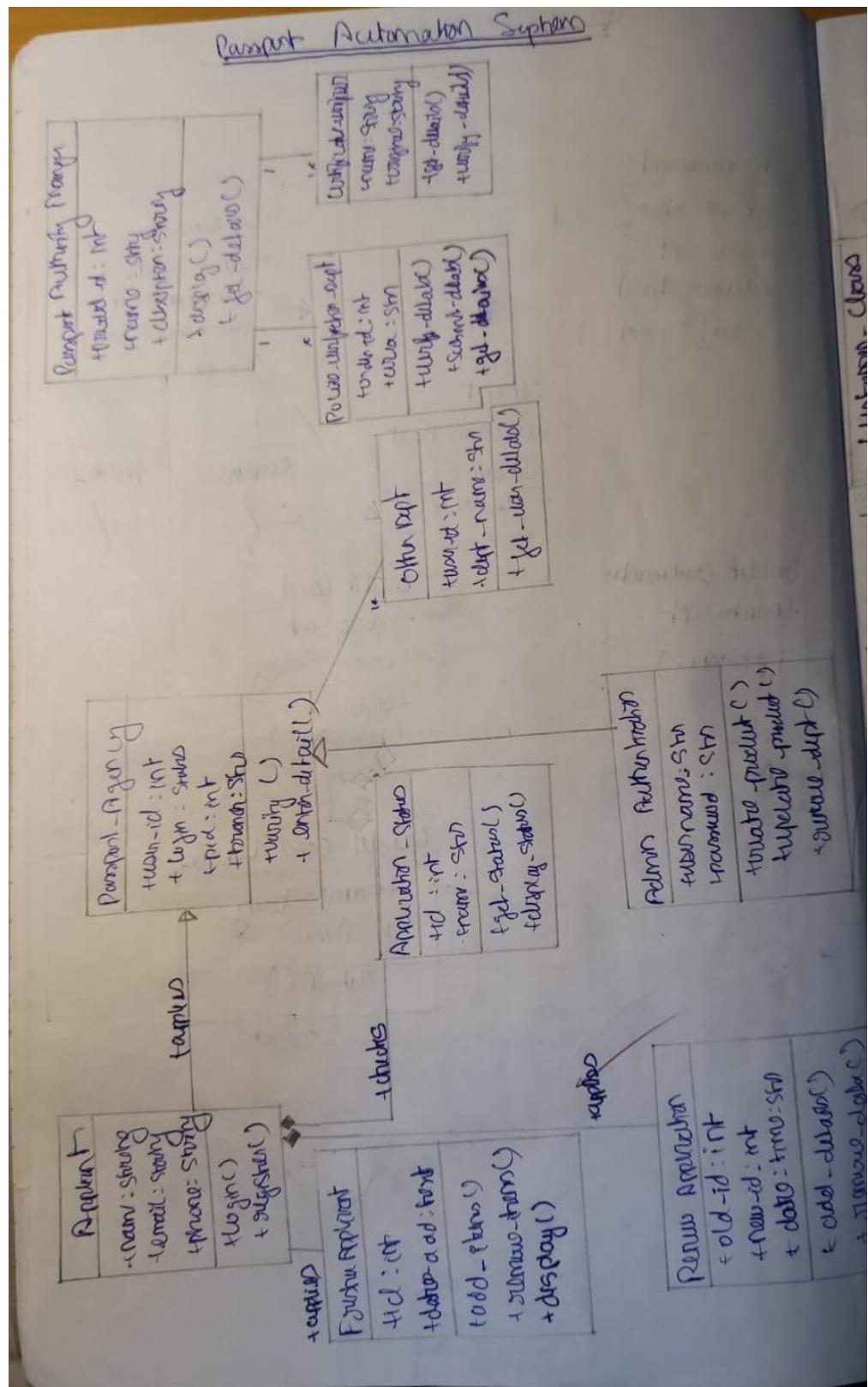


Figure 2.5 - PAS Class Diagram



3. STATE MODELLING

Hotel Management System

States:

- **Initial:** Start of booking process.
- **Room Selected:** Room selection completed.
- **Guest Details Entered:** Guest details are entered.
- **Payment Processed:** Payment is completed.
- **Booking Confirmed:** Booking is confirmed.
- **Checked In:** Guest has checked into the room.
- **Checked Out:** Guest has checked out, ending the booking process.

Transitions:

- **Select Room:** Initial → Room Selected
- **Enter Guest Details:** Room Selected → Guest Details Entered
- **Process Payment:** Guest Details Entered → Payment Processed
- **Confirm Booking:** Payment Processed → Booking Confirmed
- **Check In:** Booking Confirmed → Checked In
- **Check Out:** Checked In → Checked Out

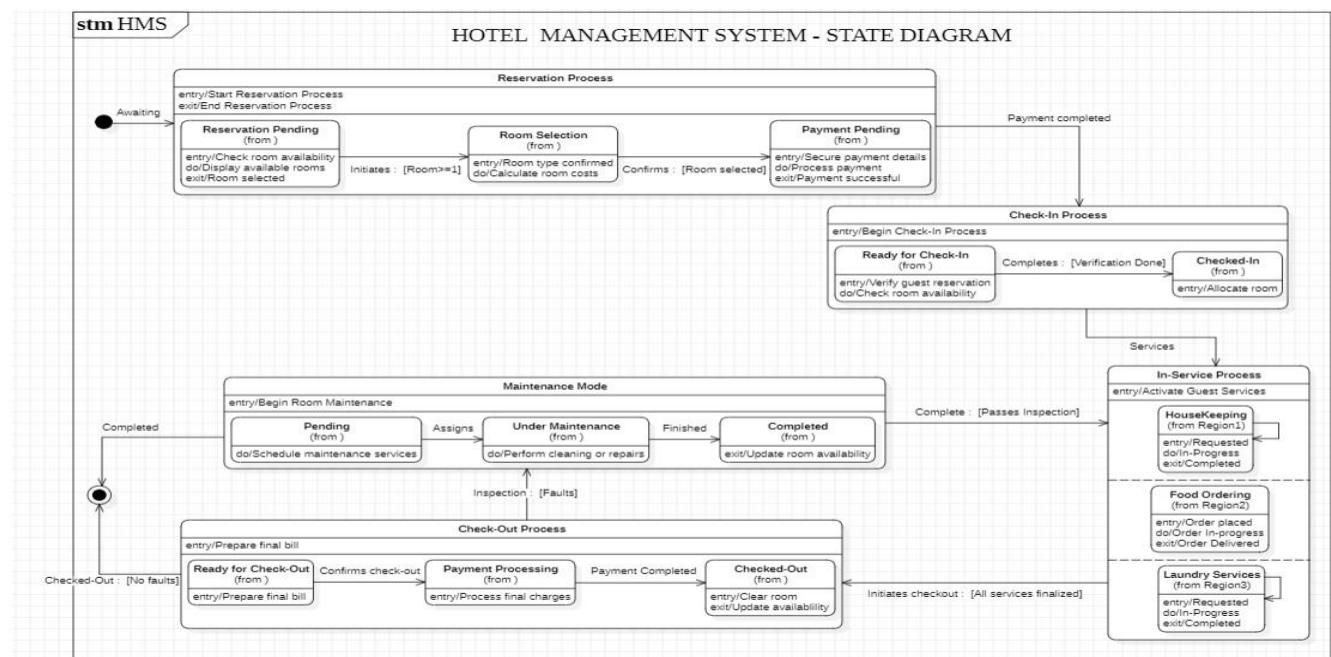
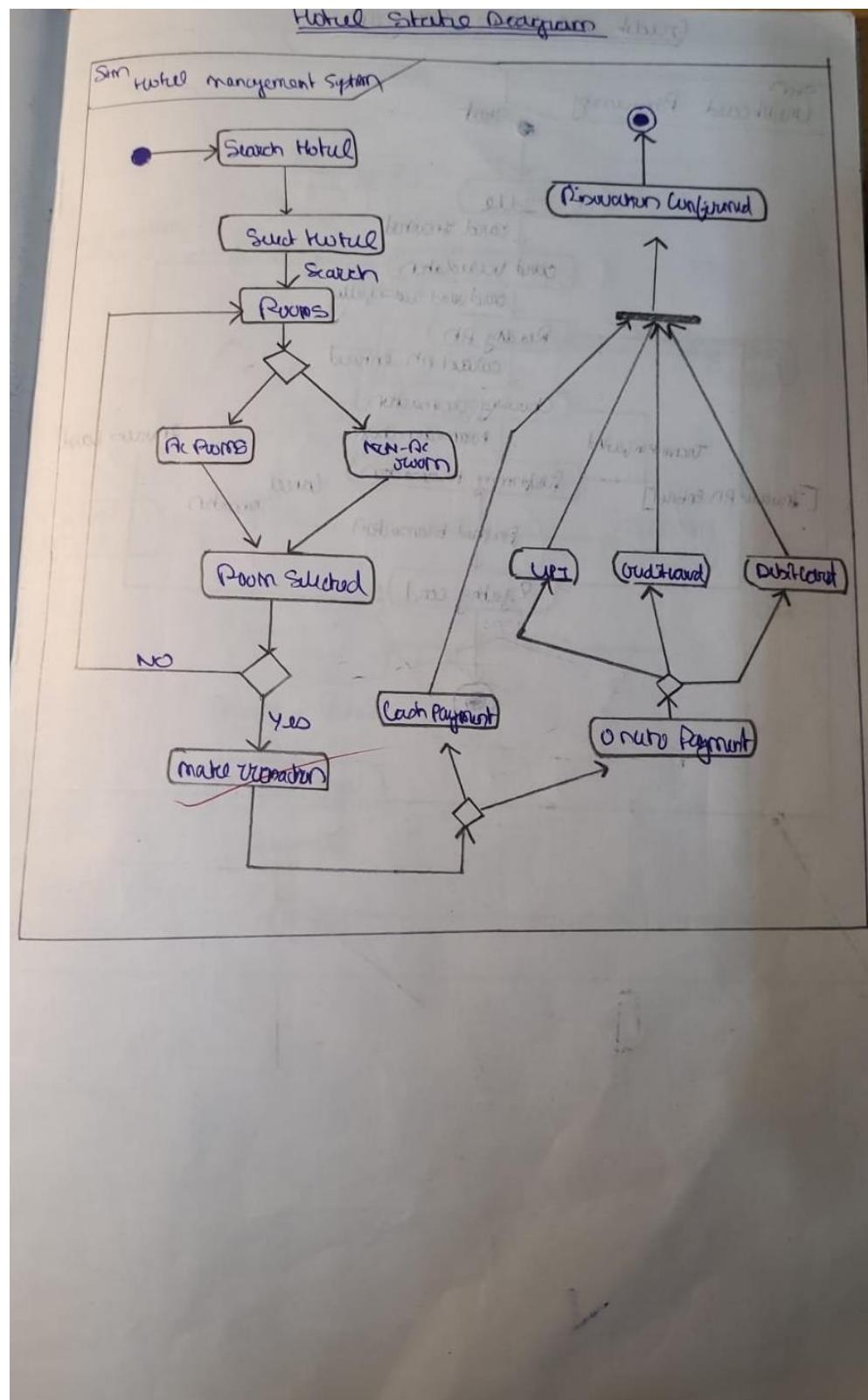


Figure 3.1 – HMS State Diagram



Credit Card Processing System

States:

- **Idle:** Ready to initiate a transaction.
- **Transaction Initiated:** Transaction has started.
- **Authorized:** Transaction authorized by the bank.
- **Processed:** Transaction completed.
- **Failed:** Transaction failed.
- **Refund Processed:** Refund issued (if applicable).

Transitions:

- **Initiate Transaction:** Idle → Transaction Initiated
- **Authorize:** Transaction Initiated → Authorized
- **Process Payment:** Authorized → Processed
- **Fail Transaction:** Transaction Initiated → Failed
- **Process Refund:** Processed → Refund Processed

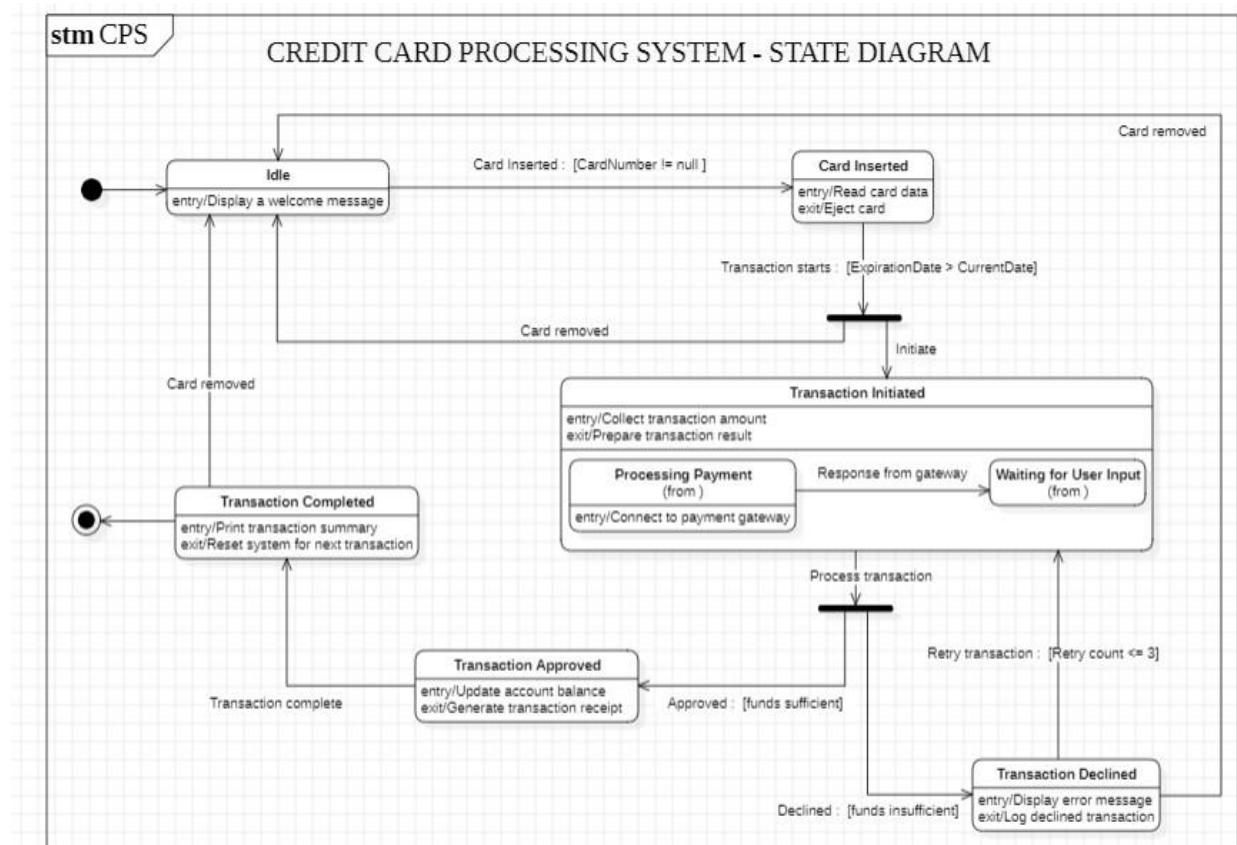
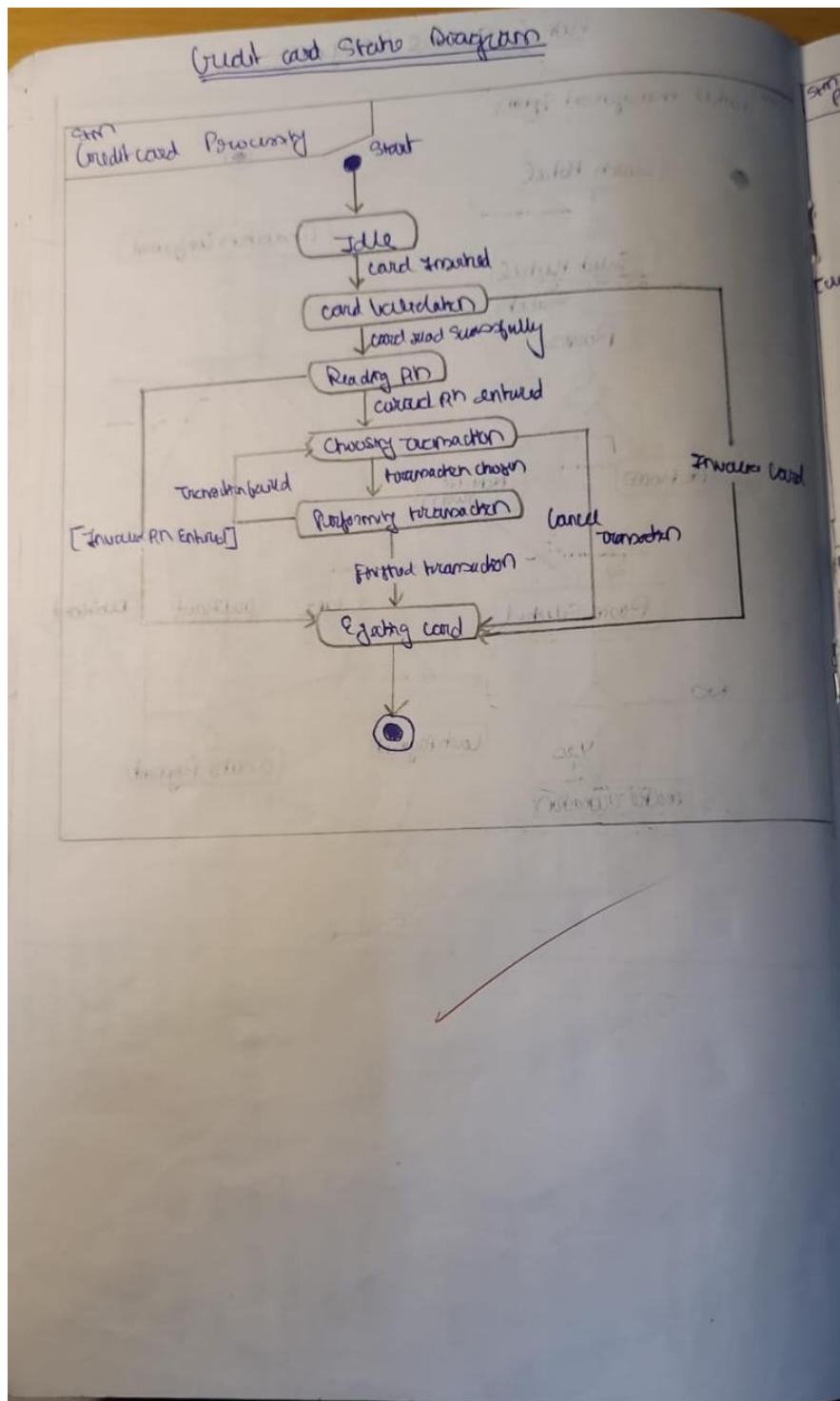


Figure 3.2 – CCPS State Diagram



Library Management System

States:

- **Available:** Book is available in the library.
 - **On Loan:** Book is borrowed by a member.
 - **Overdue:** Book loan is overdue.
 - **Returned:** Book is returned to the library.
 - **Damaged:** Book is returned but marked as damaged.

Transitions:

- **Loan Book:** Available → On Loan
 - **Mark Overdue:** On Loan → Overdue
 - **Return Book:** On Loan → Returned
 - **Return Damaged:** On Loan → Damaged
 - **Return Overdue Book:** Overdue → Returned

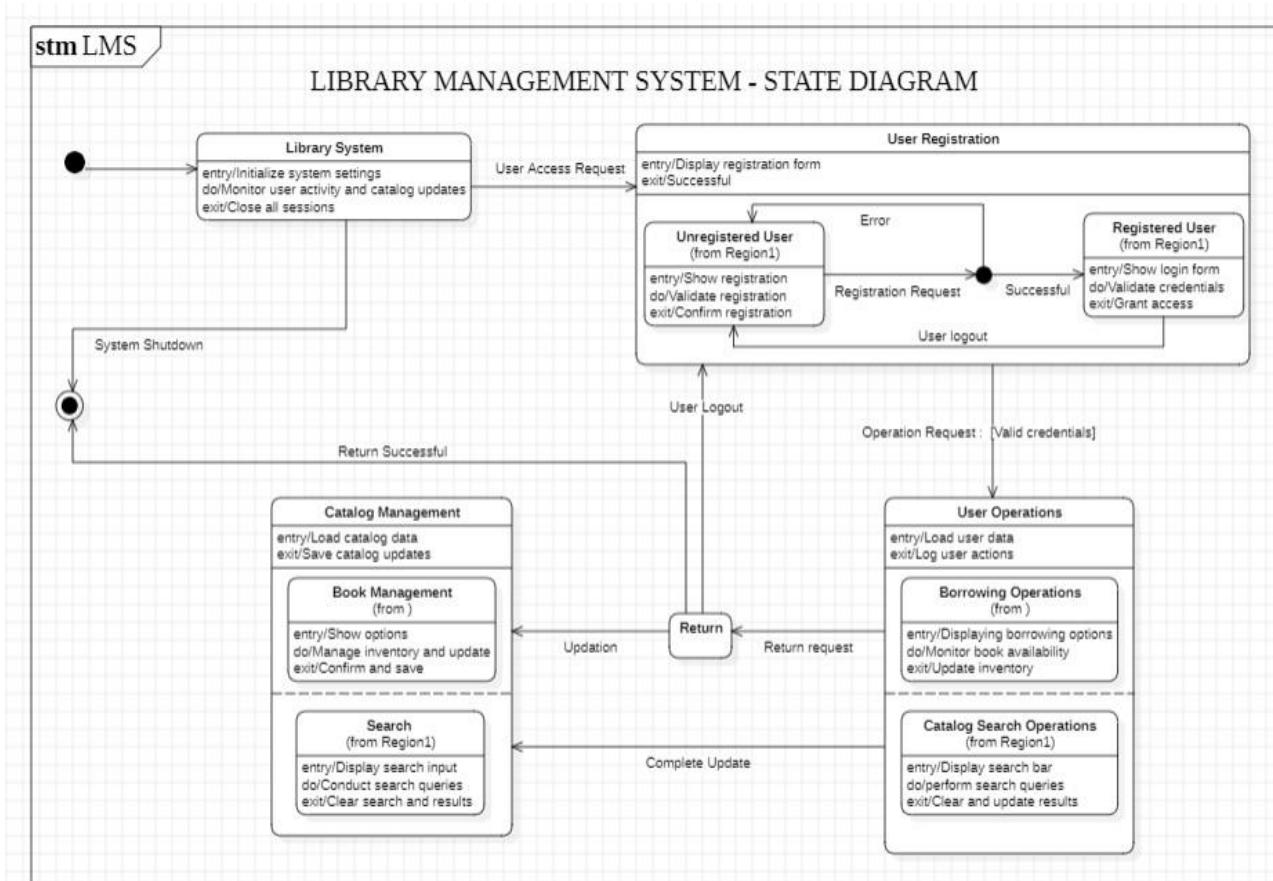
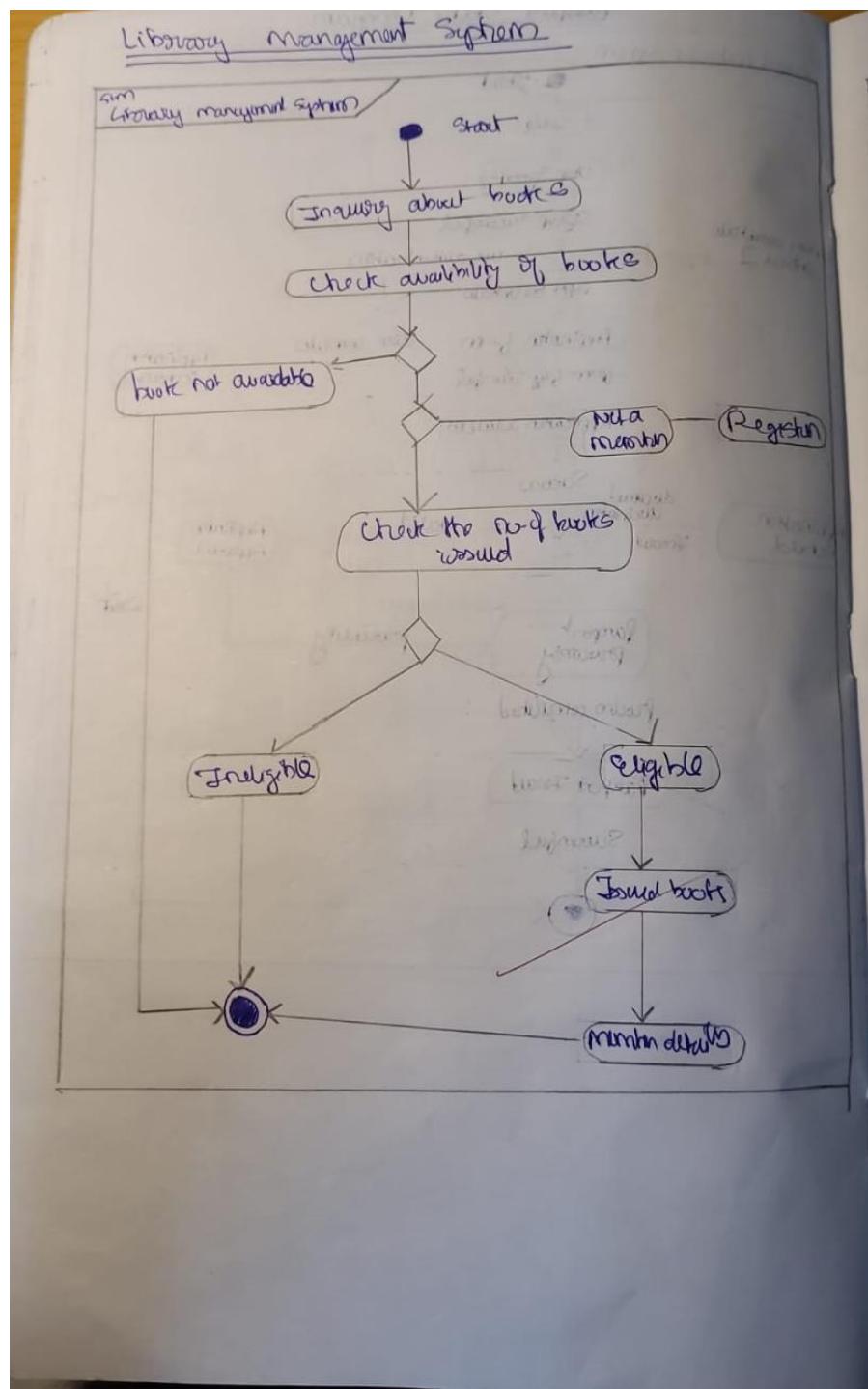


Figure 3.3 – LMS State Diagram



Stock Maintenance System

States:

- **In Stock:** Item is available in inventory.
 - **Low Stock:** Stock level is low.
 - **Out of Stock:** Item is out of stock.
 - **Reordered:** Item has been reordered.
 - **Restocked:** Item is restocked and available again.

Transitions:

- **Reduce Stock:** In Stock → Low Stock
 - **Stock Exhausted:** Low Stock → Out of Stock
 - **Place Order:** Out of Stock → Reordered
 - **Receive Stock:** Reordered → Restocked
 - **Replenish Stock:** Low Stock → In Stock

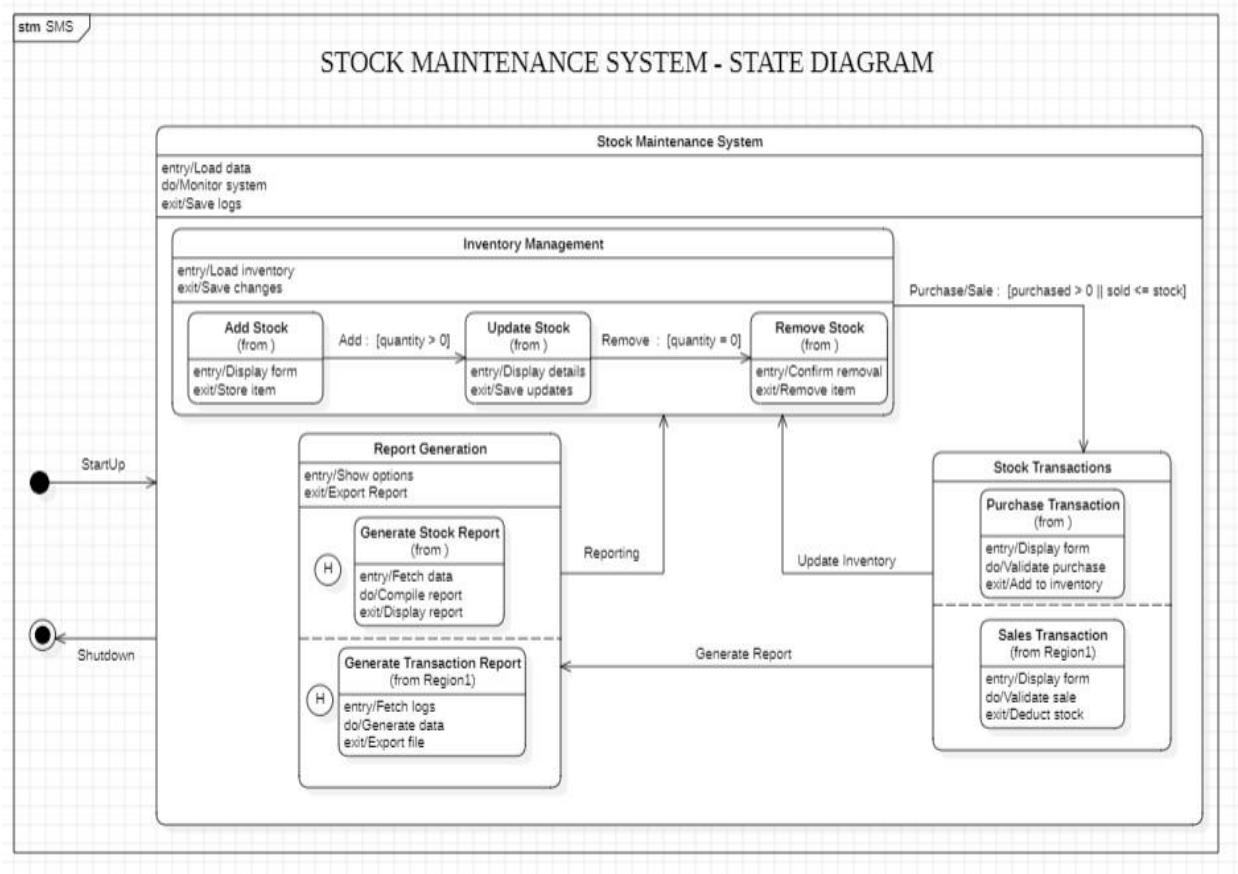
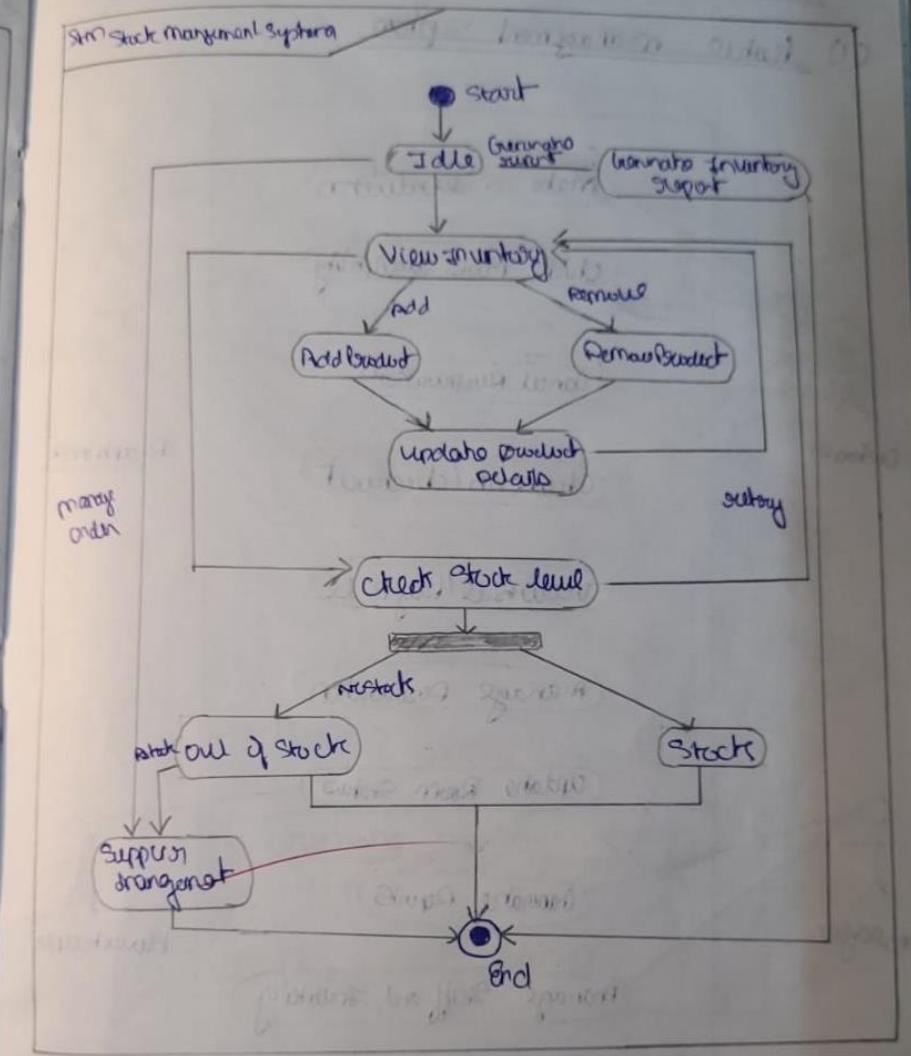


Figure 3.4 – SMS State Diagram

Stock management System



Passport Automation System

States:

- **Application Submitted:** Application submitted by applicant.
- **In Review:** Application is being reviewed.
- **Document Verification:** Documents are verified.
- **Approved:** Application is approved.
- **Rejected:** Application is rejected.
- **Passport Issued:** Passport is issued.

Transitions:

- **Submit Application:** Start → Application Submitted
- **Review Application:** Application Submitted → In Review
- **Verify Documents:** In Review → Document Verification
- **Approve Application:** Document Verification → Approved
- **Reject Application:** Document Verification → Rejected
- **Issue Passport:** Approved → Passport Issued

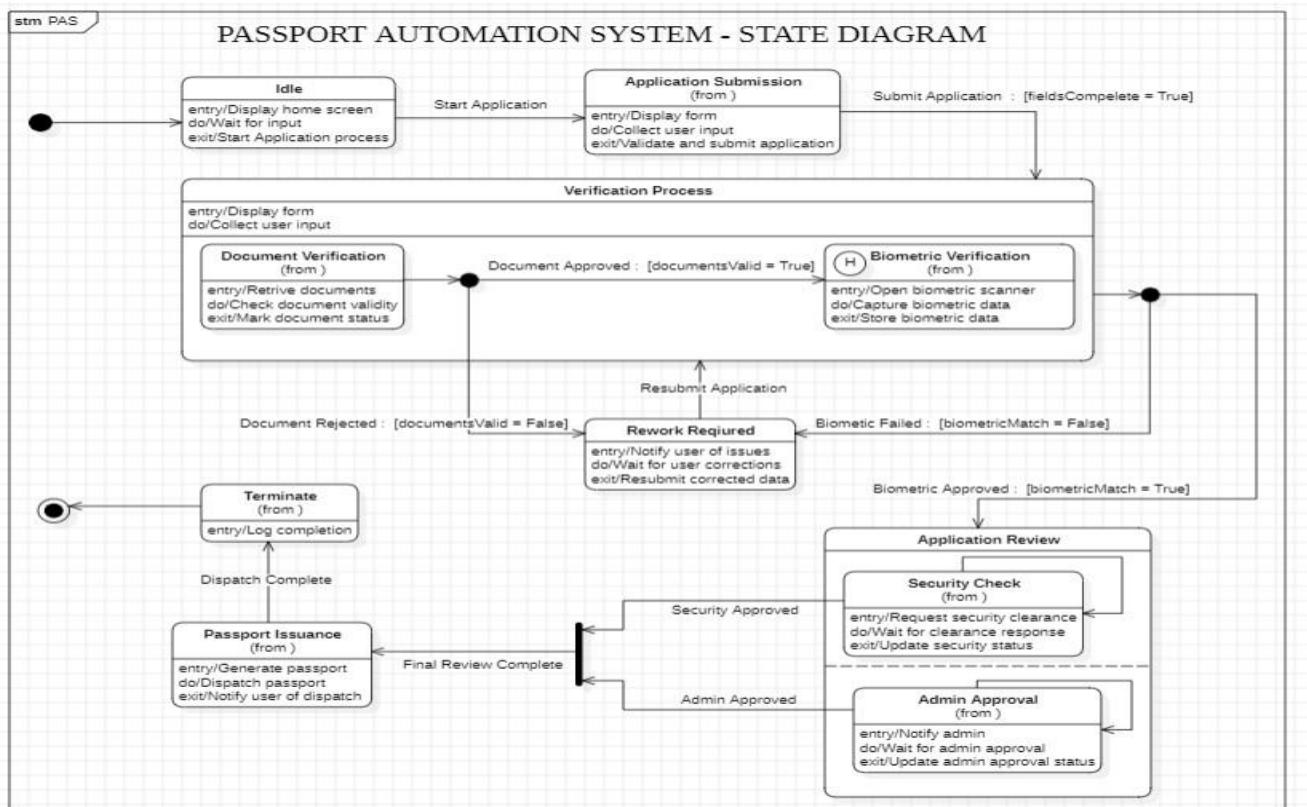
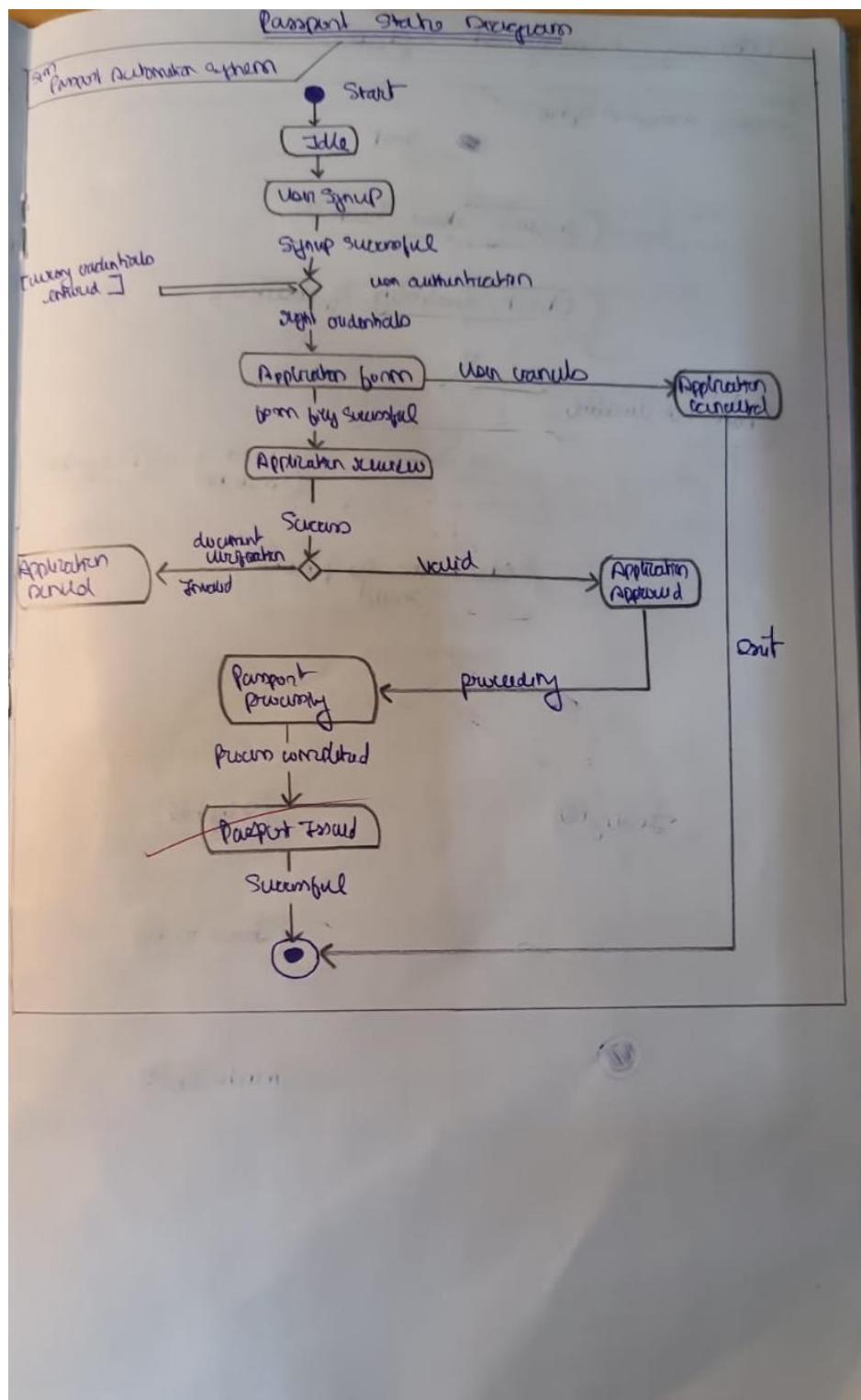


Figure 3.5 – PAS State Diagram



4. INTERACTION MODELLING: USE CASE MODELS

Hotel Management System

Actors:

- **Guest:** Interacts with the system to book rooms, avail services, and make payments.
- **Receptionist:** Manages guest details, room allocation, and service requests.
- **Housekeeping:** Handles room cleaning requests and status updates.

Use Cases:

- **Book Room:** Allows guests to reserve rooms.
- **Make Payment:** Handles payment processing for bookings and services.
- **Check-In:** Completes the check-in process for guests.
- **Check-Out:** Handles guest departures and generates invoices.
- **Request Room Service:** Enables guests to request additional services.

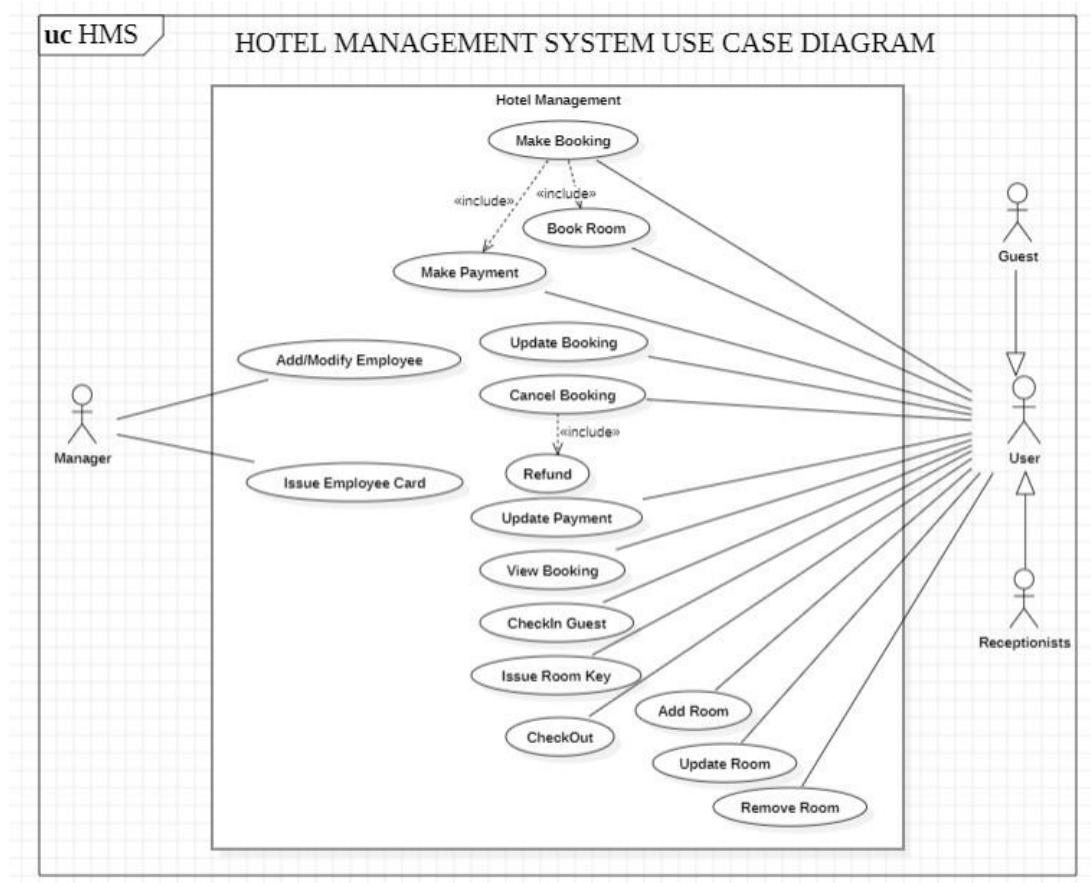
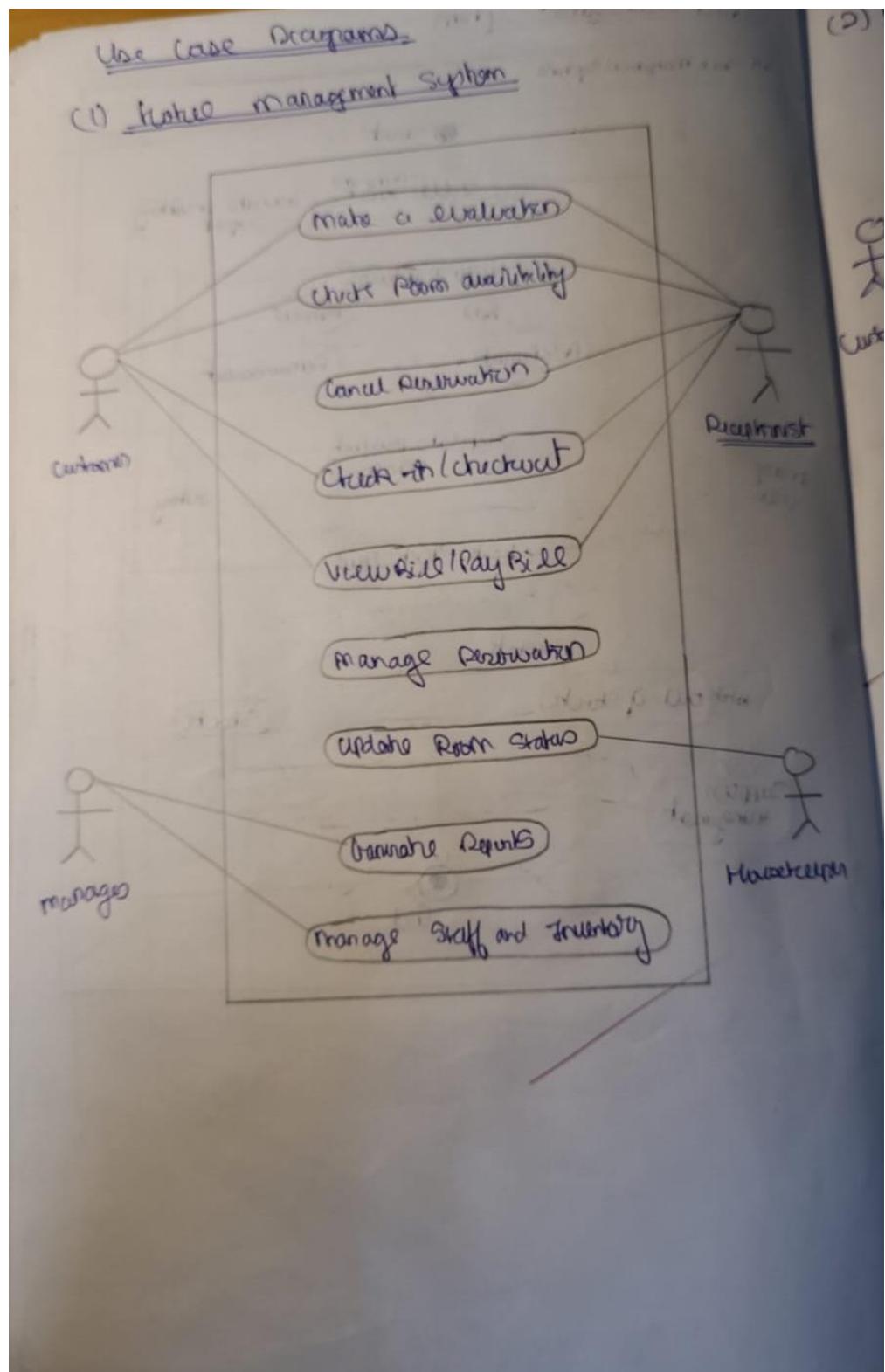


Figure 4.1 – HMS Use Case Diagram



Credit Card Processing System

Actors:

- **Cardholder:** Uses the credit card for transactions.
- **Merchant:** Accepts credit card payments.
- **Bank:** Processes and authorizes transactions.
- **Payment Gateway:** Facilitates secure communication between systems.

Use Cases:

- **Initiate Transaction:** Begins the transaction process.
- **Authorize Payment:** Verifies the transaction with the issuing bank.
- **Process Payment:** Completes the payment process.
- **Issue Refund:** Handles refunds for failed or disputed transactions.

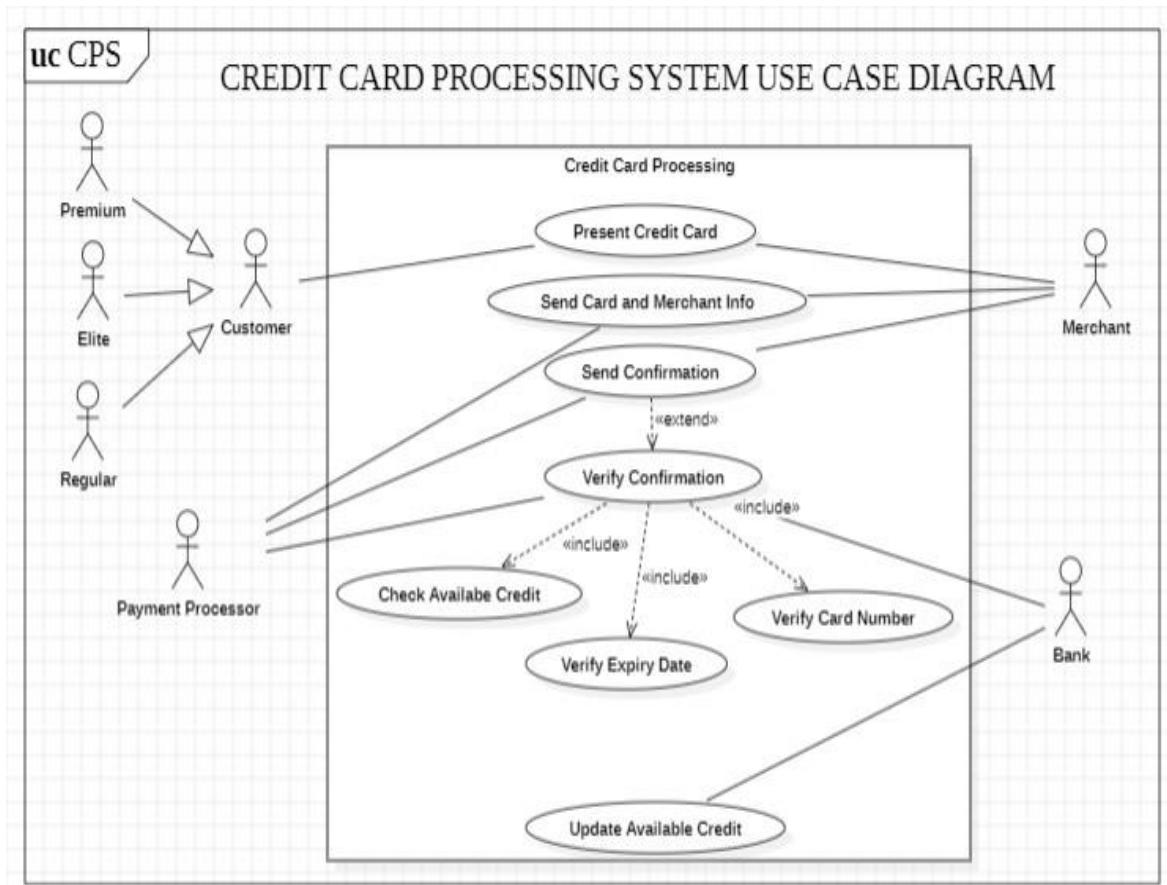
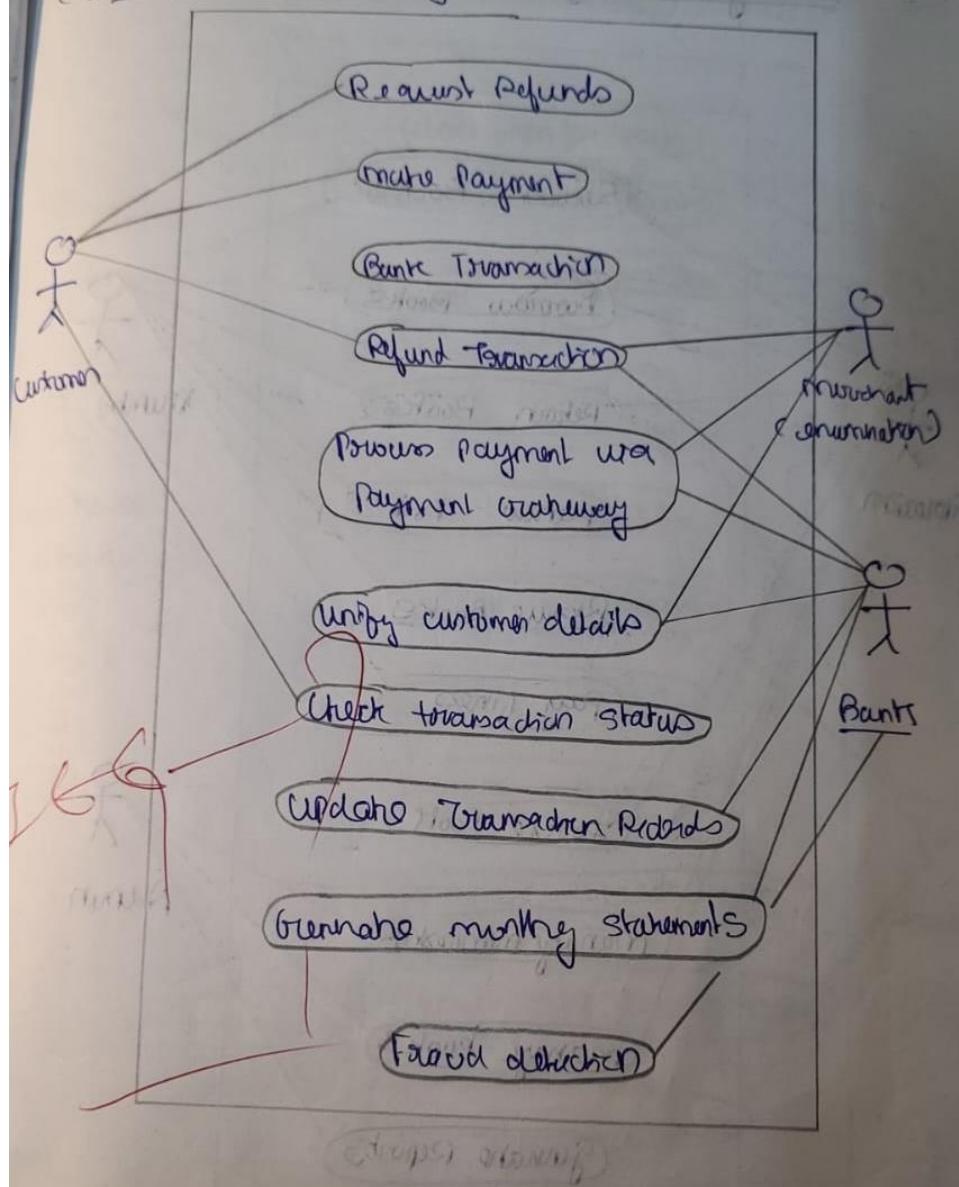


Figure 4.2 – CCPS Use Case Diagram

(2) Credit Card Processing



Library Management System

Actors:

- **Member:** Borrows, returns, and renews books.
- **Librarian:** Manages inventory, loans, and member registrations.

Use Cases:

- **Borrow Book:** Allows members to borrow books from the library.
- **Return Book:** Handles the return of borrowed books.
- **Renew Book:** Extends the loan period for books.
- **Add New Book:** Enables librarians to update the catalog.
- **Pay Fine:** Handles overdue fine payments.

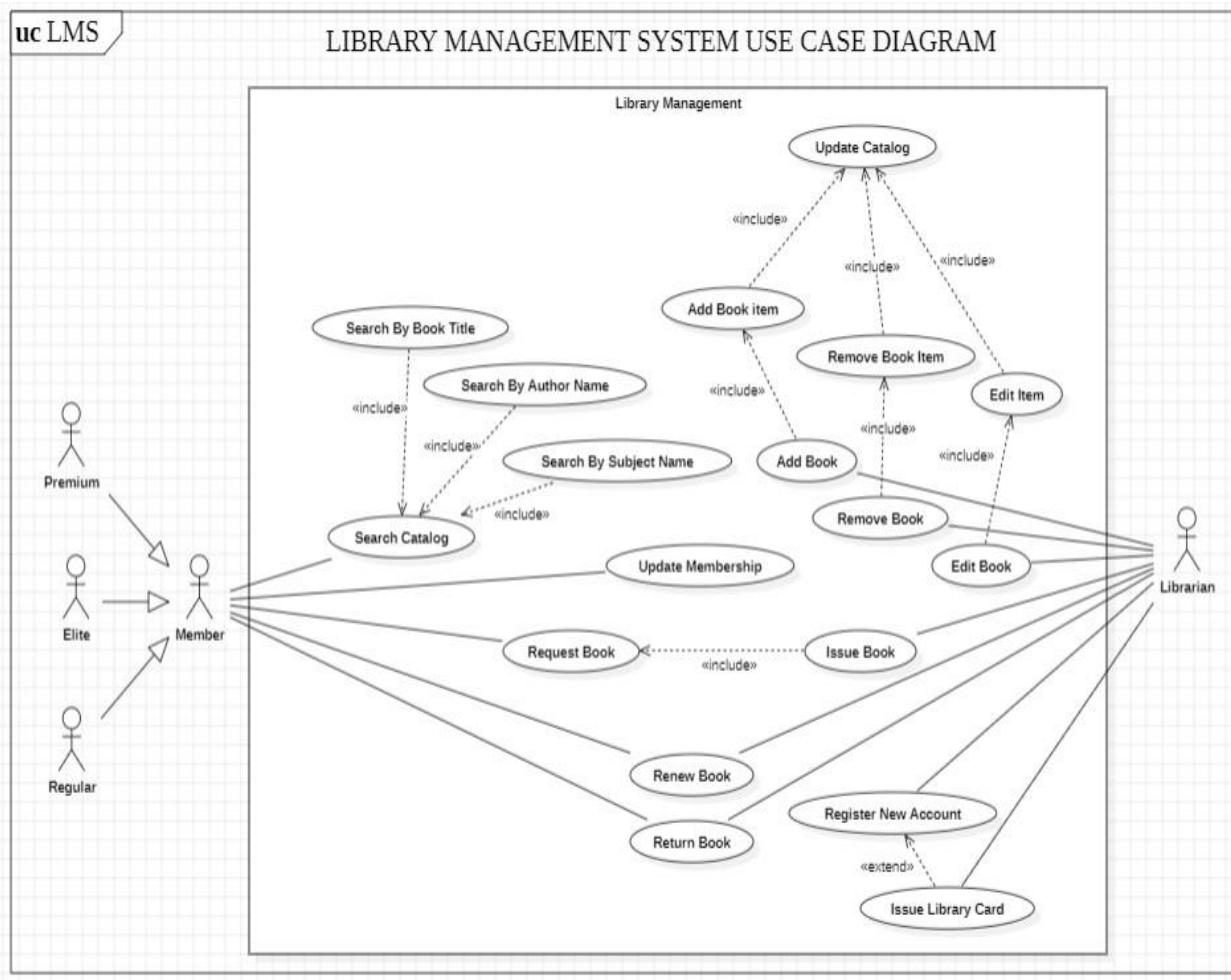
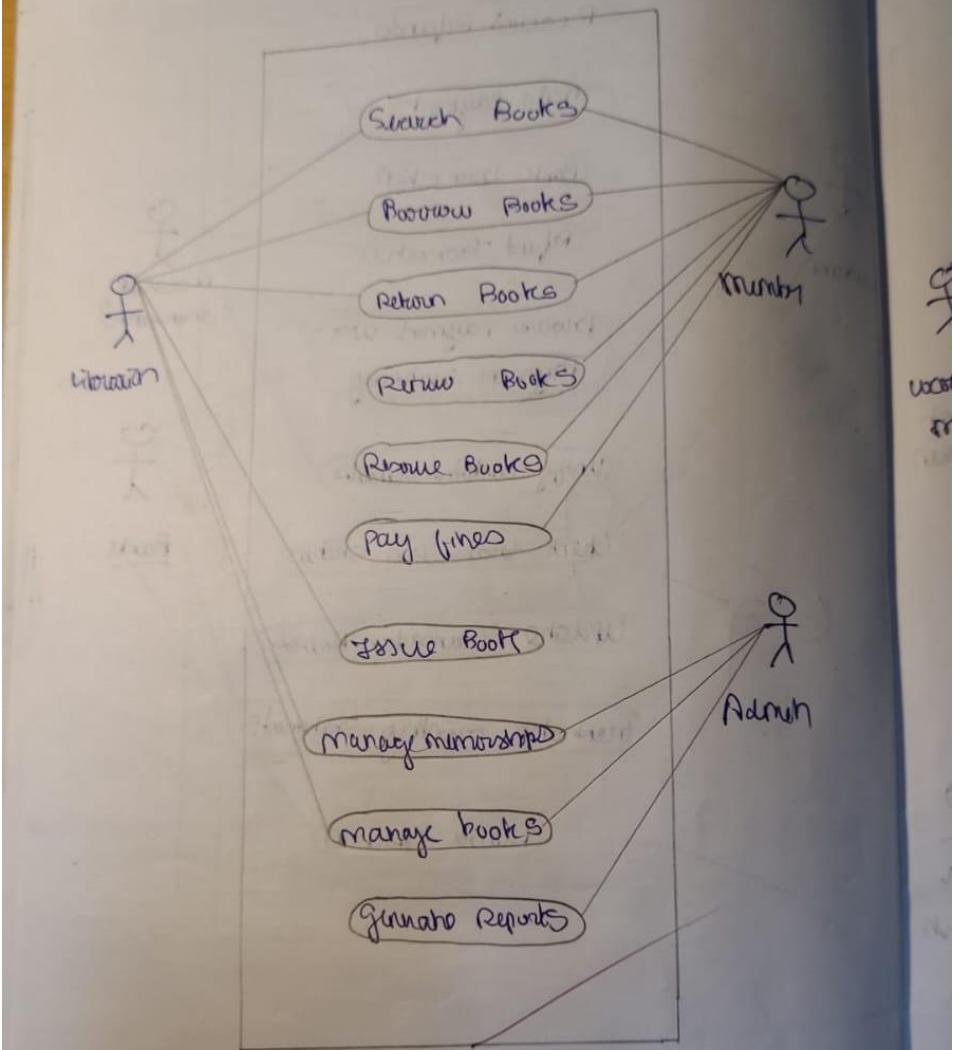


Figure 4.3 – LMS Use Case Diagram

(3) Library Management System



Stock Maintenance System

Actors:

- **Warehouse Manager:** Tracks inventory and places orders.
- **Supplier:** Supplies items to the warehouse.
- **System:** Monitors stock levels and triggers alerts.

Use Cases:

- **Monitor Stock:** Tracks current inventory levels.
- **Place Order:** Initiates the process of replenishing stock.
- **Receive Stock:** Updates inventory after new stock arrives.
- **Generate Low Stock Alert:** Notifies the manager when inventory is low.

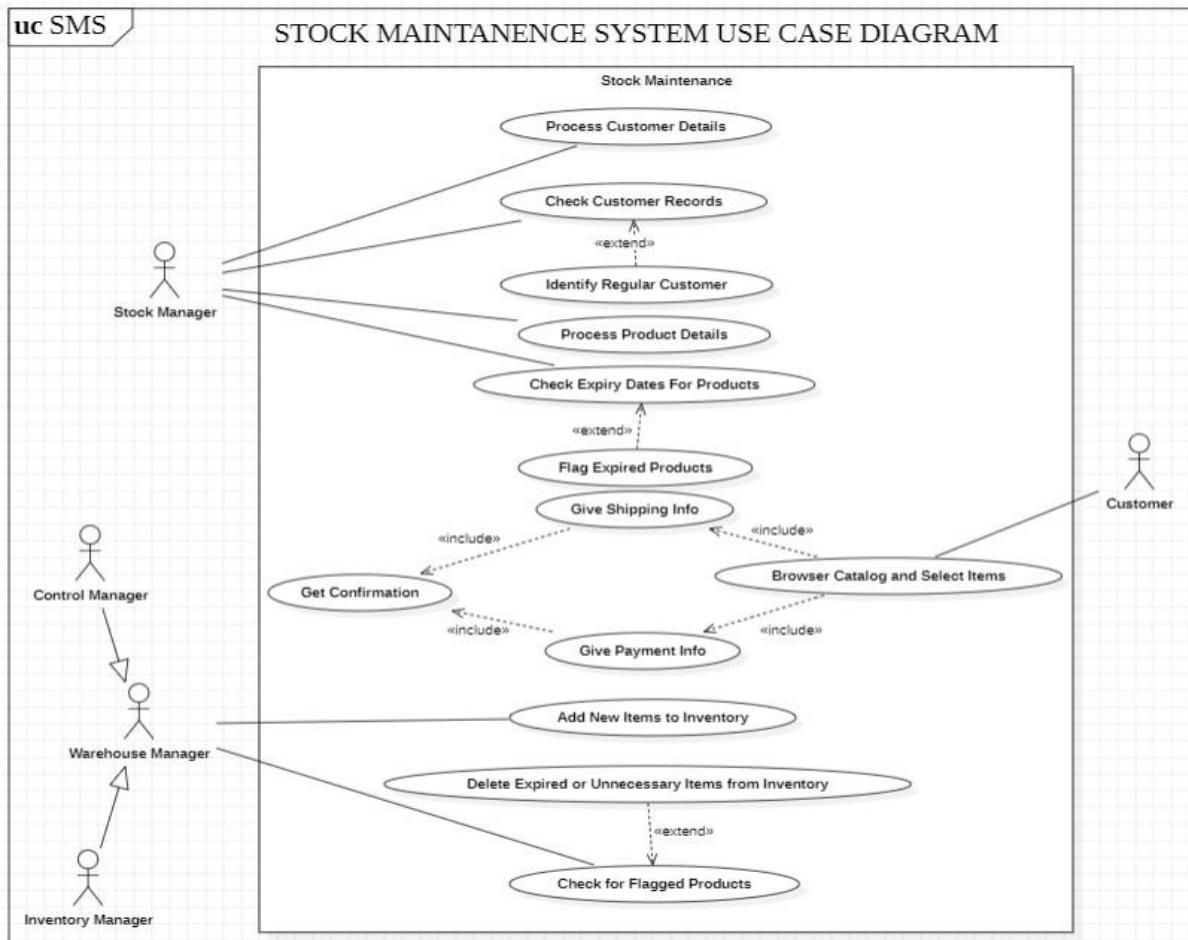
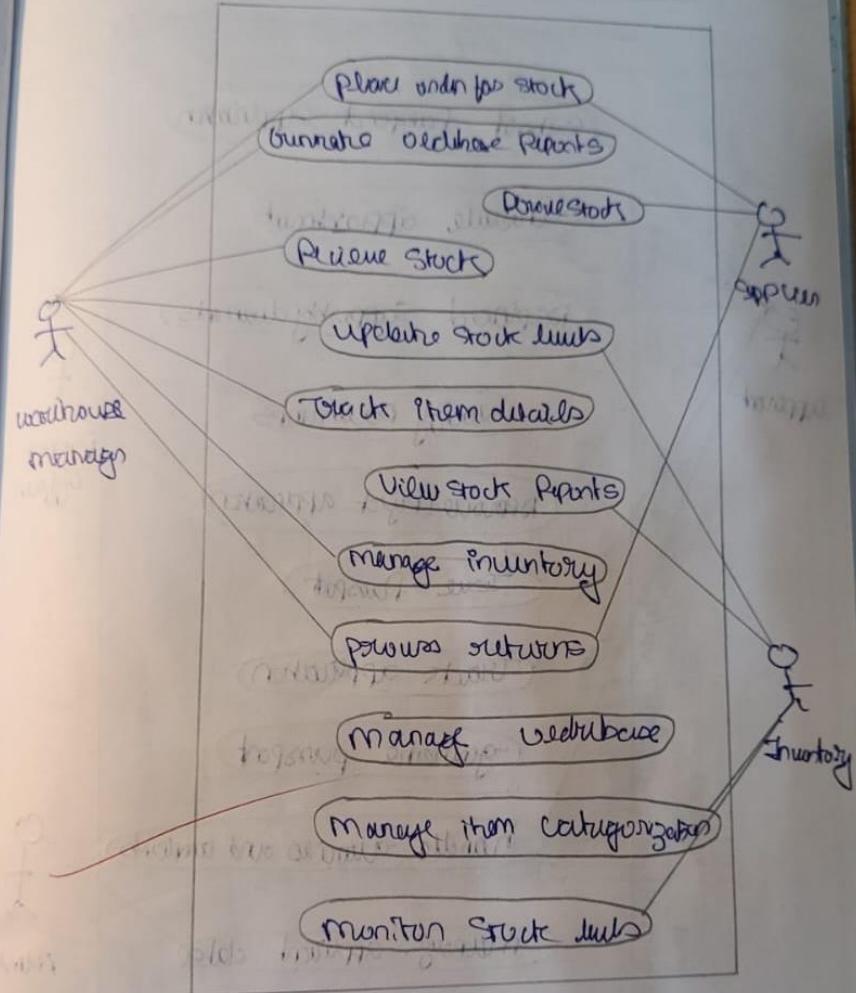


Figure 4.4 – SMS Use Case Diagram

(u) Stock maintenance System



Passport Automation System

Actors:

- **Applicant:** Submits applications and schedules appointments.
- **Officer:** Reviews applications and verifies documents.
- **System:** Automates notifications and application tracking.

Use Cases:

- **Submit Application:** Allows applicants to apply for passports.
- **Schedule Appointment:** Enables applicants to book verification appointments.
- **Verify Documents:** Officers review and validate submitted documents.
- **Approve Application:** Approves applications after review.
- **Issue Passport:** Completes the application process by generating passports.

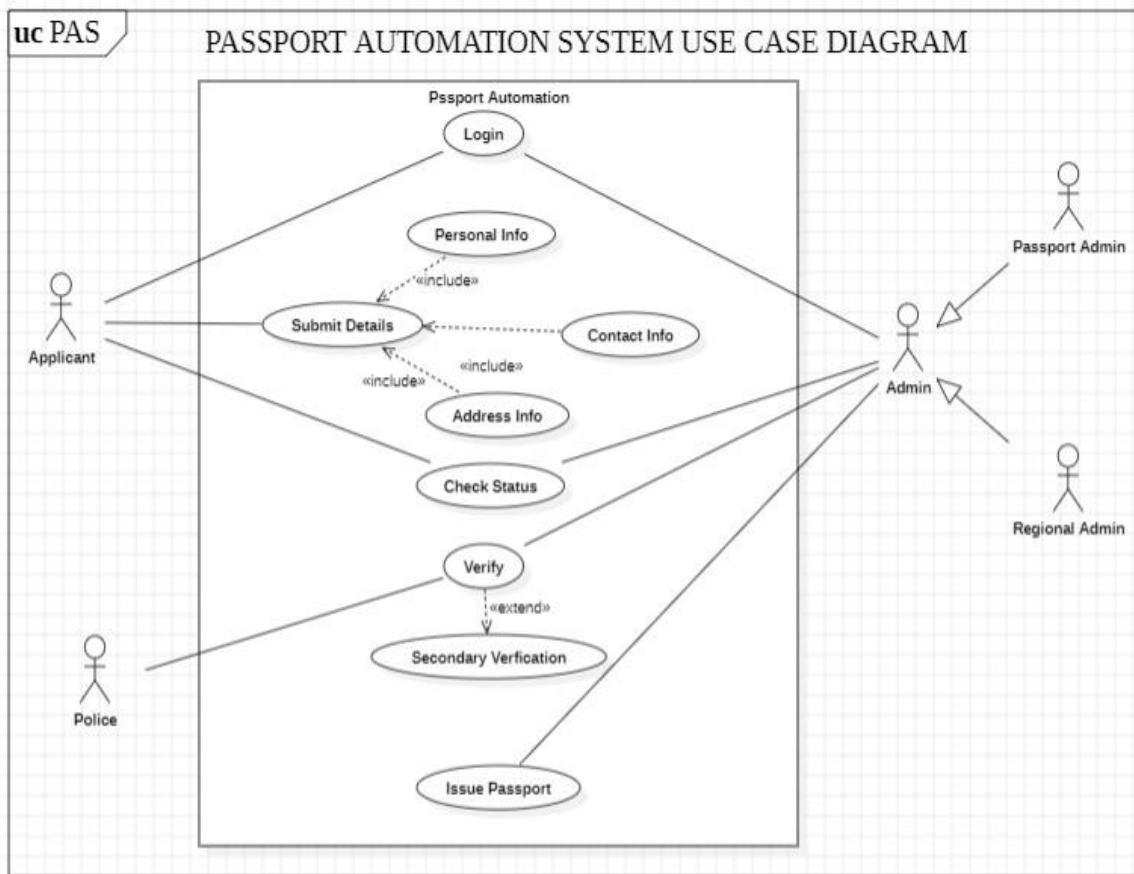
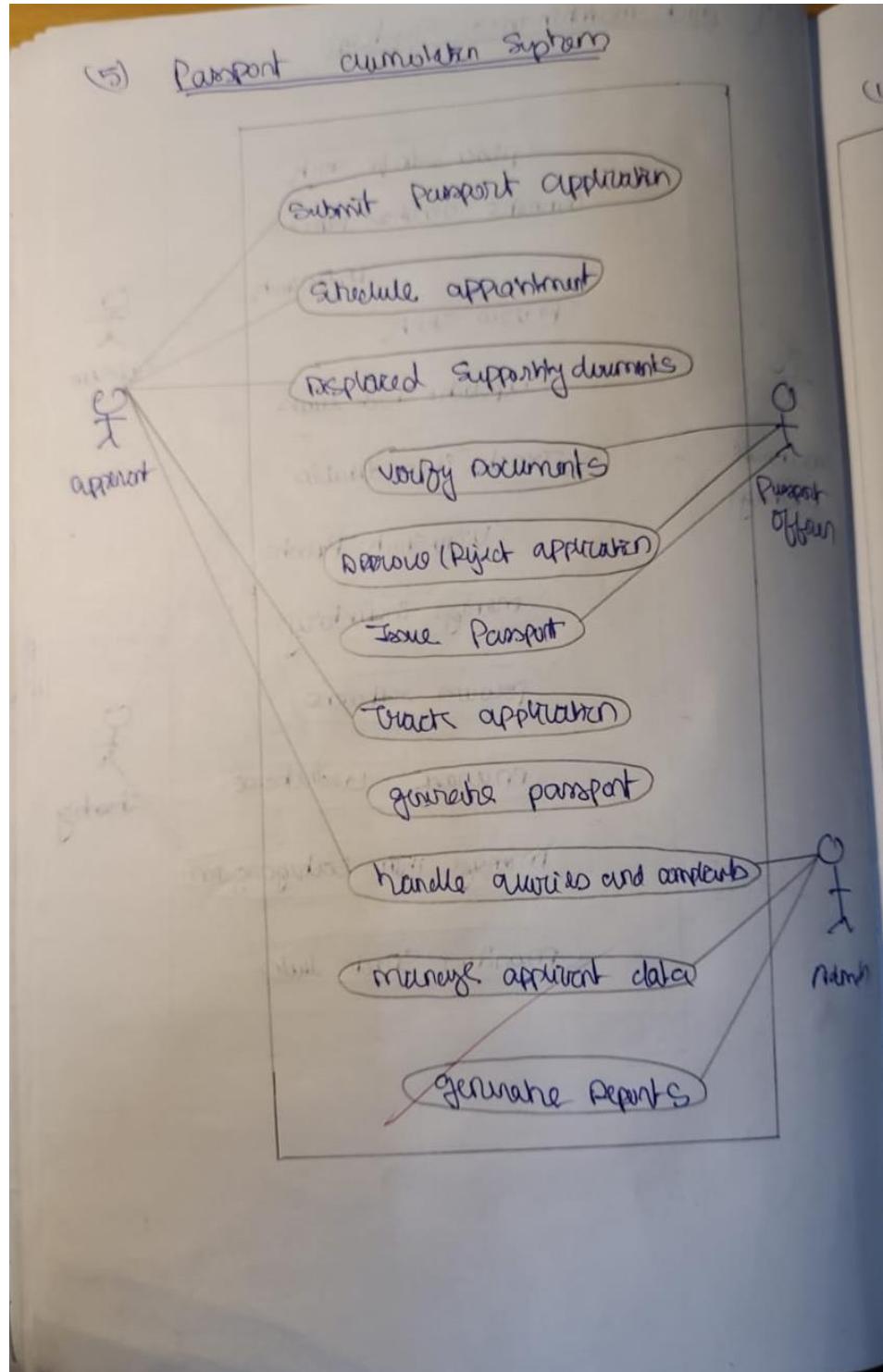


Figure 4.5 – PAS Use Case Diagram

⑤ Passport Application System



5. INTERACTION MODELLING: SEQUENCE MODELS

Hotel Management System

Use Case: Book Room

Actors: Guest, Receptionist, Hotel System

Objects: Guest, Receptionist, Hotel System, Payment Gateway

Sequence:

1. **Guest → Receptionist:** Request to book a room.
2. **Receptionist → Hotel System:** Search for available rooms.
3. **Hotel System → Receptionist:** Return list of available rooms.
4. **Receptionist → Guest:** Provide room options.
5. **Guest → Receptionist:** Select room and provide details.
6. **Receptionist → Payment Gateway:** Process payment.
7. **Payment Gateway → Hotel System:** Confirm payment status.
8. **Hotel System → Receptionist:** Confirm booking and generate receipt.
9. **Receptionist → Guest:** Provide booking confirmation.

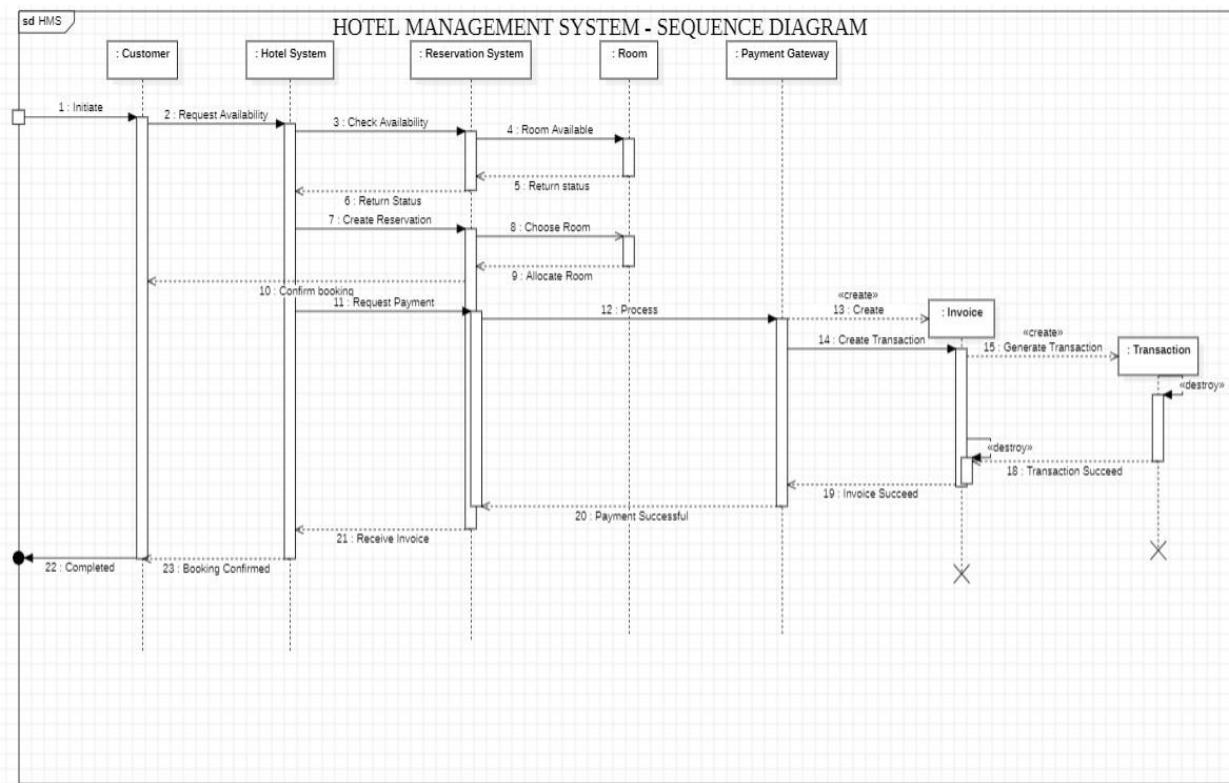
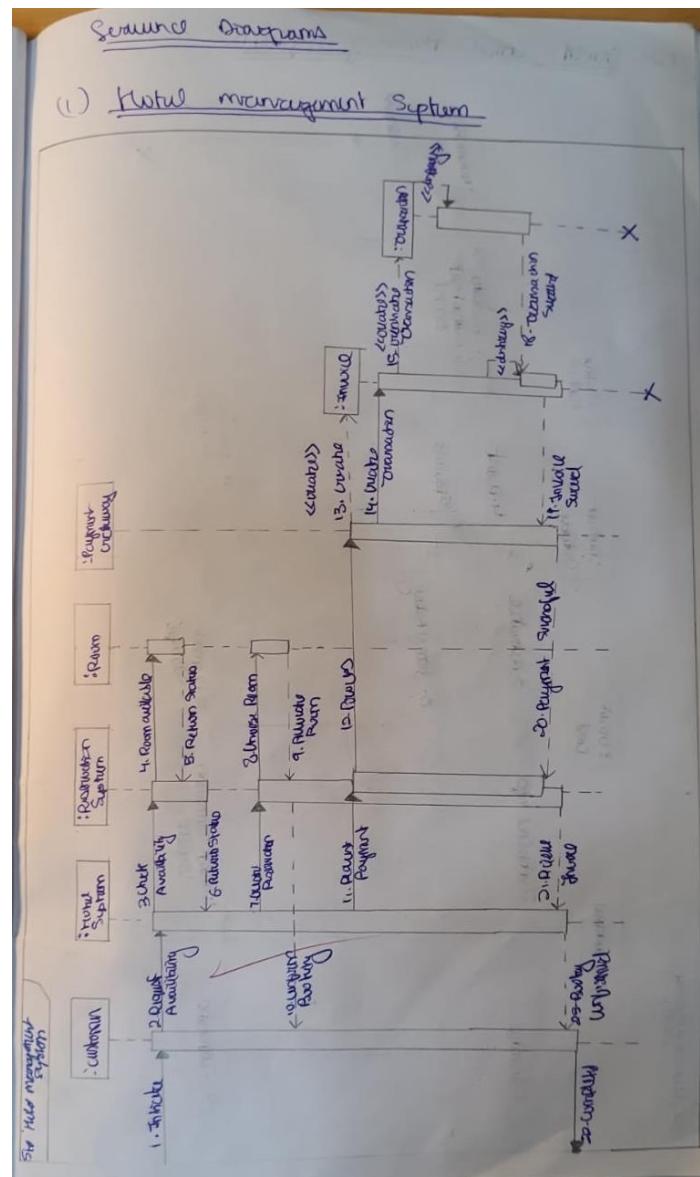


Figure 5.1 – HMS Sequence Diagram



Credit Card Processing System

Use Case: Process Payment

Actors: Cardholder, Merchant

Objects: Cardholder, Merchant, Payment Gateway, Bank

Sequence:

1. **Cardholder → Merchant:** Provide payment details.
2. **Merchant → Payment Gateway:** Forward transaction details.
3. **Payment Gateway → Bank:** Verify card details and funds.
4. **Bank → Payment Gateway:** Return authorization status.
5. **Payment Gateway → Merchant:** Confirm payment.
6. **Merchant → Cardholder:** Notify transaction result.

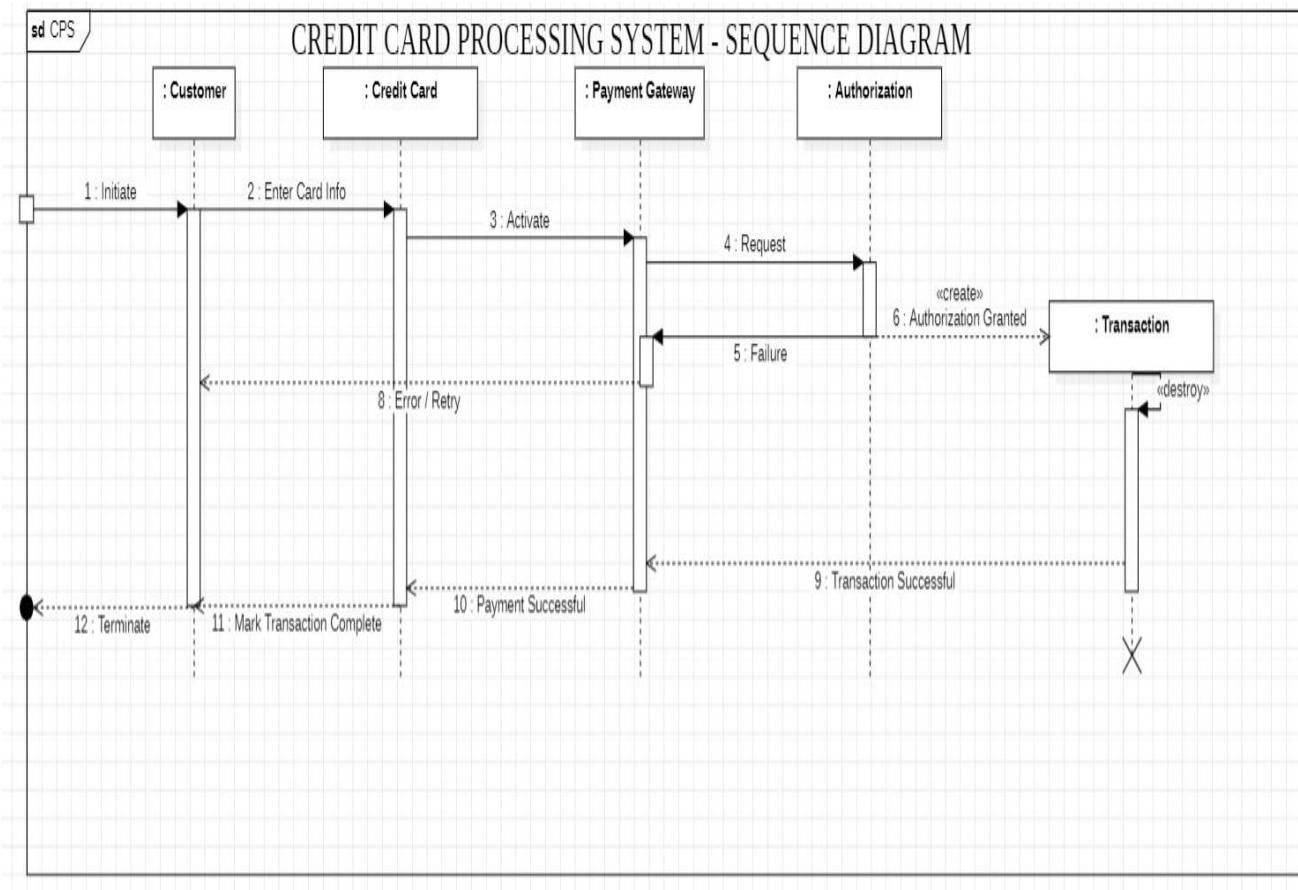
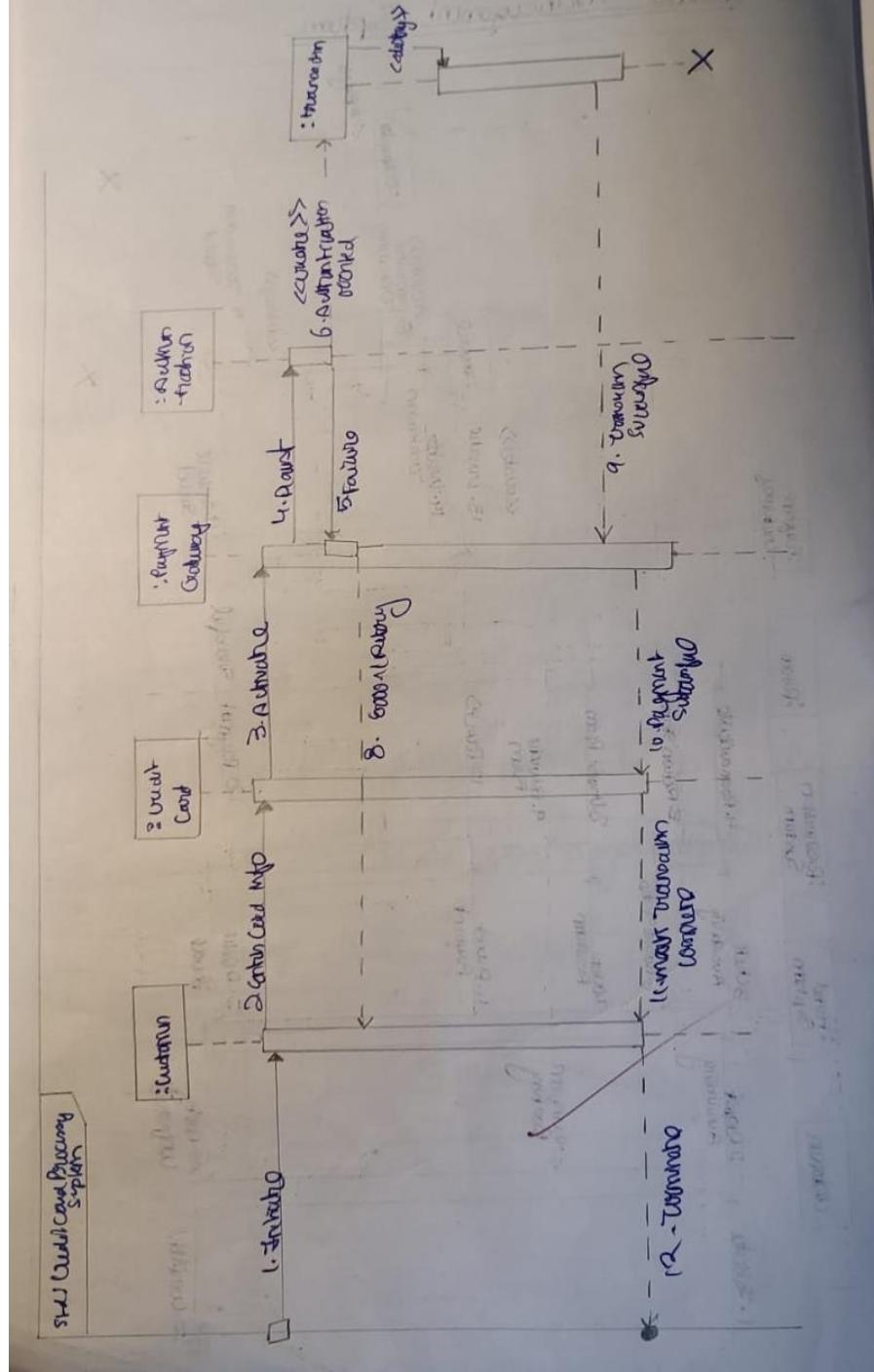


Figure 5.2 – CCPS Sequence Diagram

(2) audit trail Powders Siphon



Library Management System

Use Case: Borrow Book

Actors: Member, Librarian

Objects: Member, Librarian, Library System

Sequence:

1. **Member → Librarian:** Request to borrow a book.
2. **Librarian → Library System:** Check book availability.
3. **Library System → Librarian:** Return availability status.
4. **Librarian → Member:** Confirm loan and record due date.
5. **Library System → Member:** Update inventory and loan record.

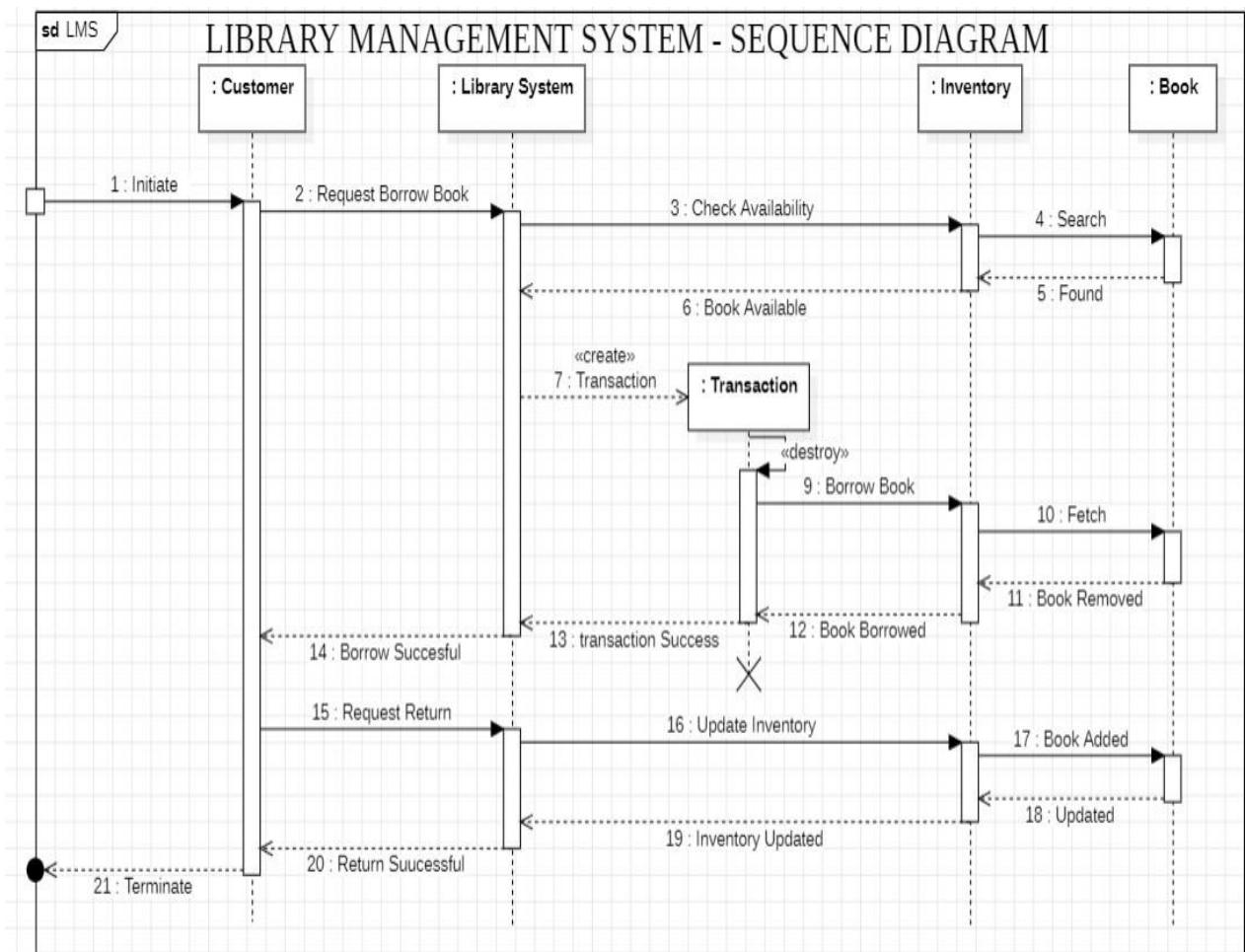
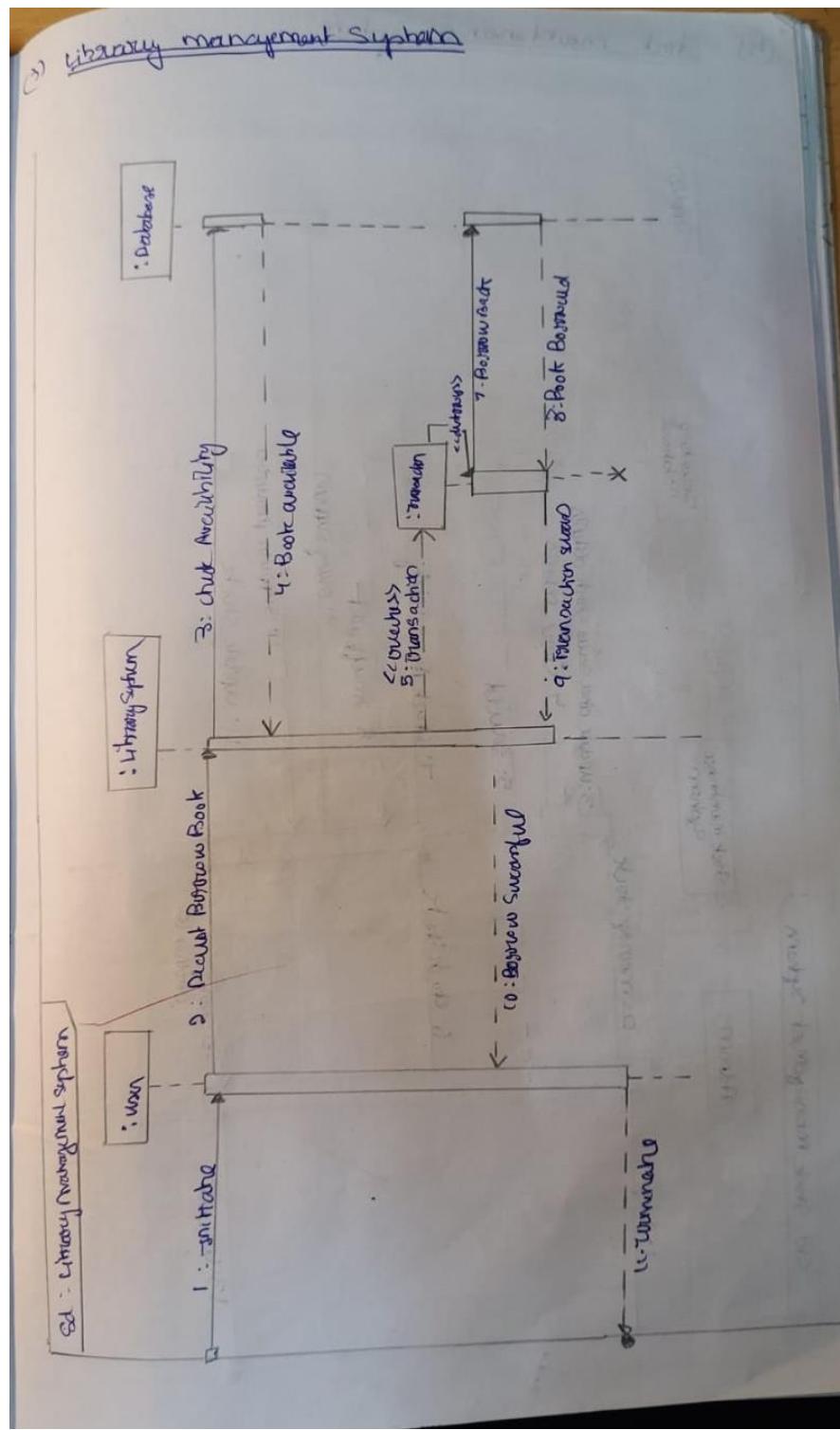


Figure 5.3 – LMS Sequence Diagram



Stock Maintenance System

Use Case: Monitor and Replenish Stock

Actors: Warehouse Manager, Supplier

Objects: Warehouse Manager, Inventory System, Supplier

Sequence:

1. **Inventory System** → **Warehouse Manager**: Notify low stock alert.
2. **Warehouse Manager** → **Supplier**: Place a replenishment order.
3. **Supplier** → **Warehouse Manager**: Confirm order and provide delivery date.
4. **Supplier** → **Inventory System**: Update stock levels after delivery.
5. **Inventory System** → **Warehouse Manager**: Notify successful restocking.

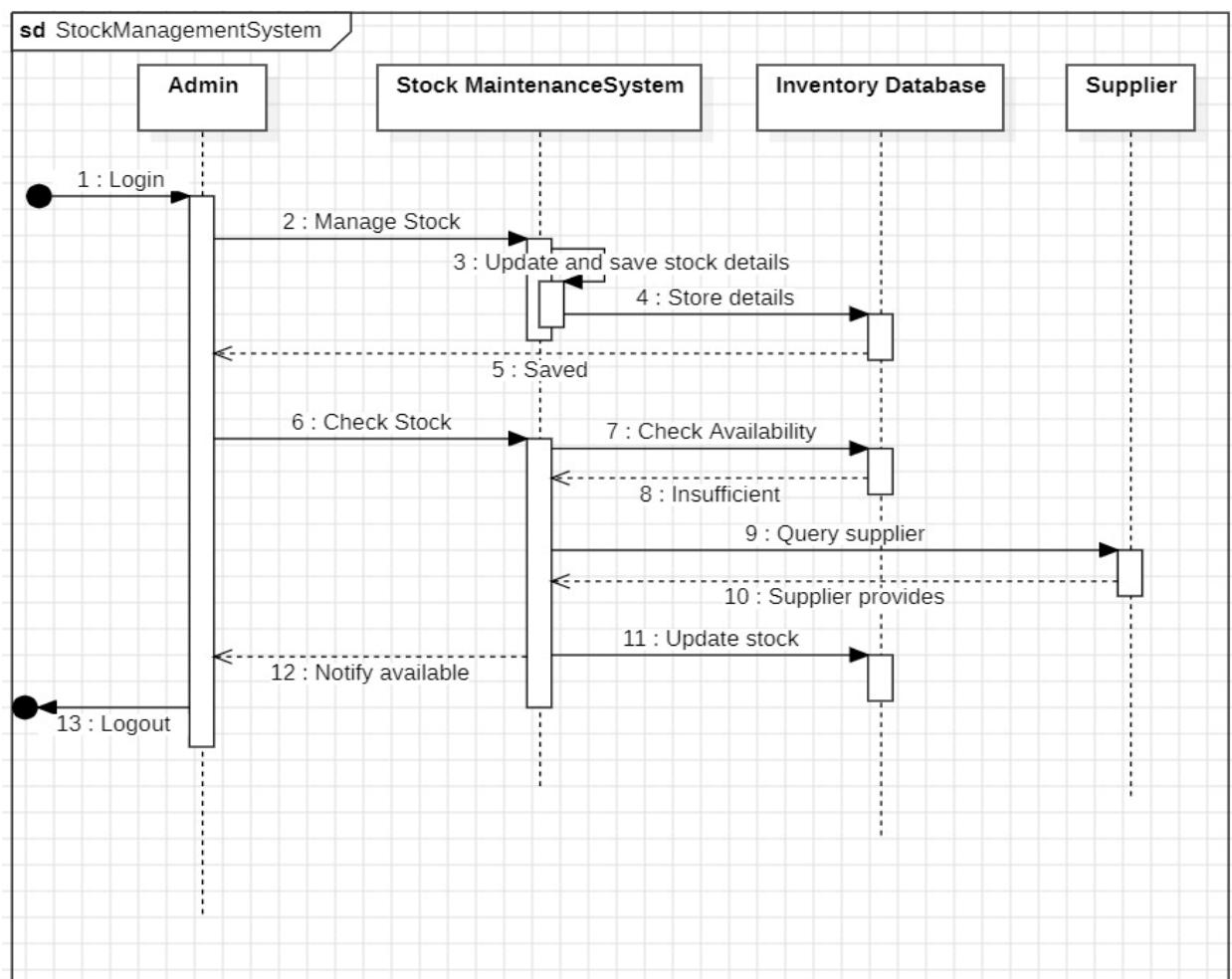
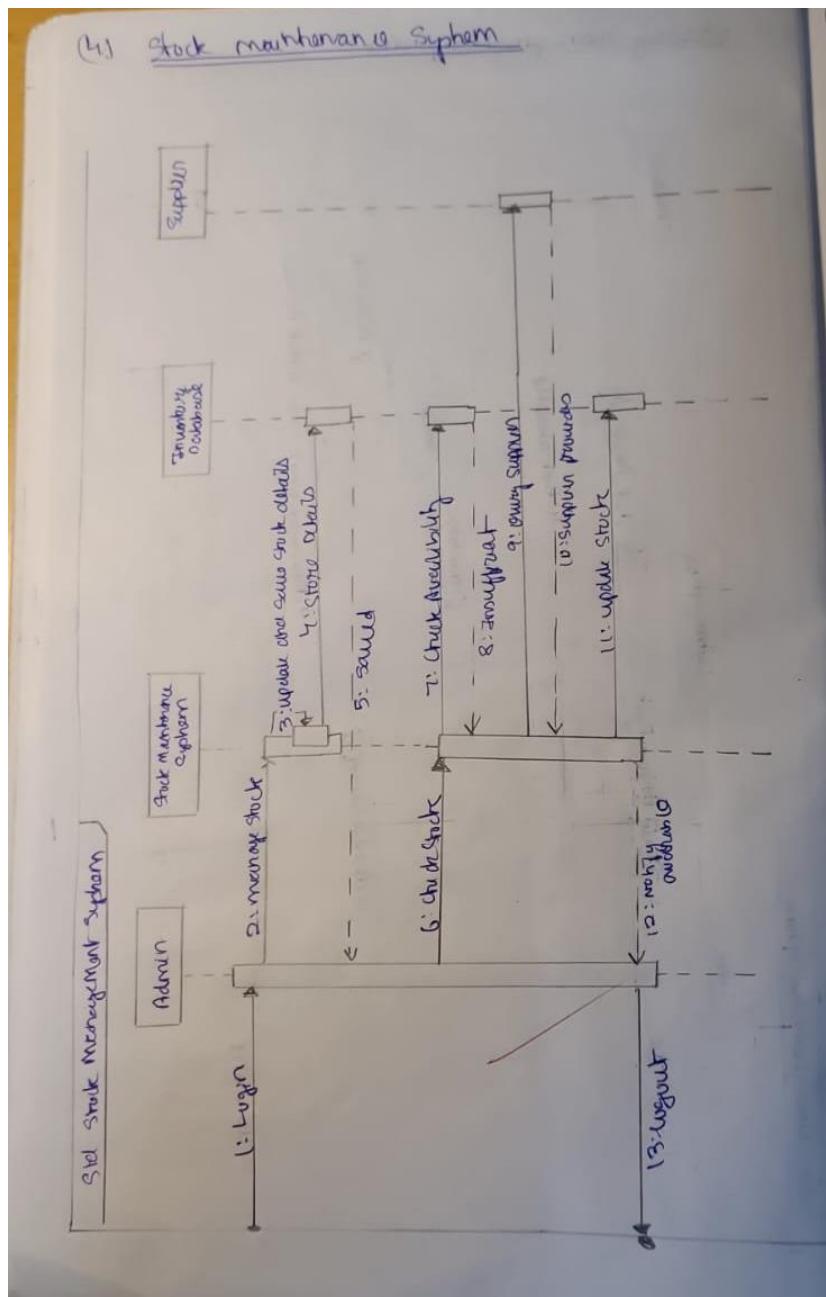


Figure 5.4 – SMS Sequence Diagram



Passport Automation System

Use Case: Submit and Process Application

Actors: Applicant, Officer

Objects: Applicant, Passport System, Officer

Sequence:

1. **Applicant → Passport System:** Submit application and documents.
2. **Passport System → Officer:** Forward application for review.
3. **Officer → Passport System:** Verify documents.
4. **Passport System → Officer:** Return verification status.
5. **Officer → Applicant:** Notify of application approval or rejection.
6. **Passport System → Applicant:** Issue passport if approved.

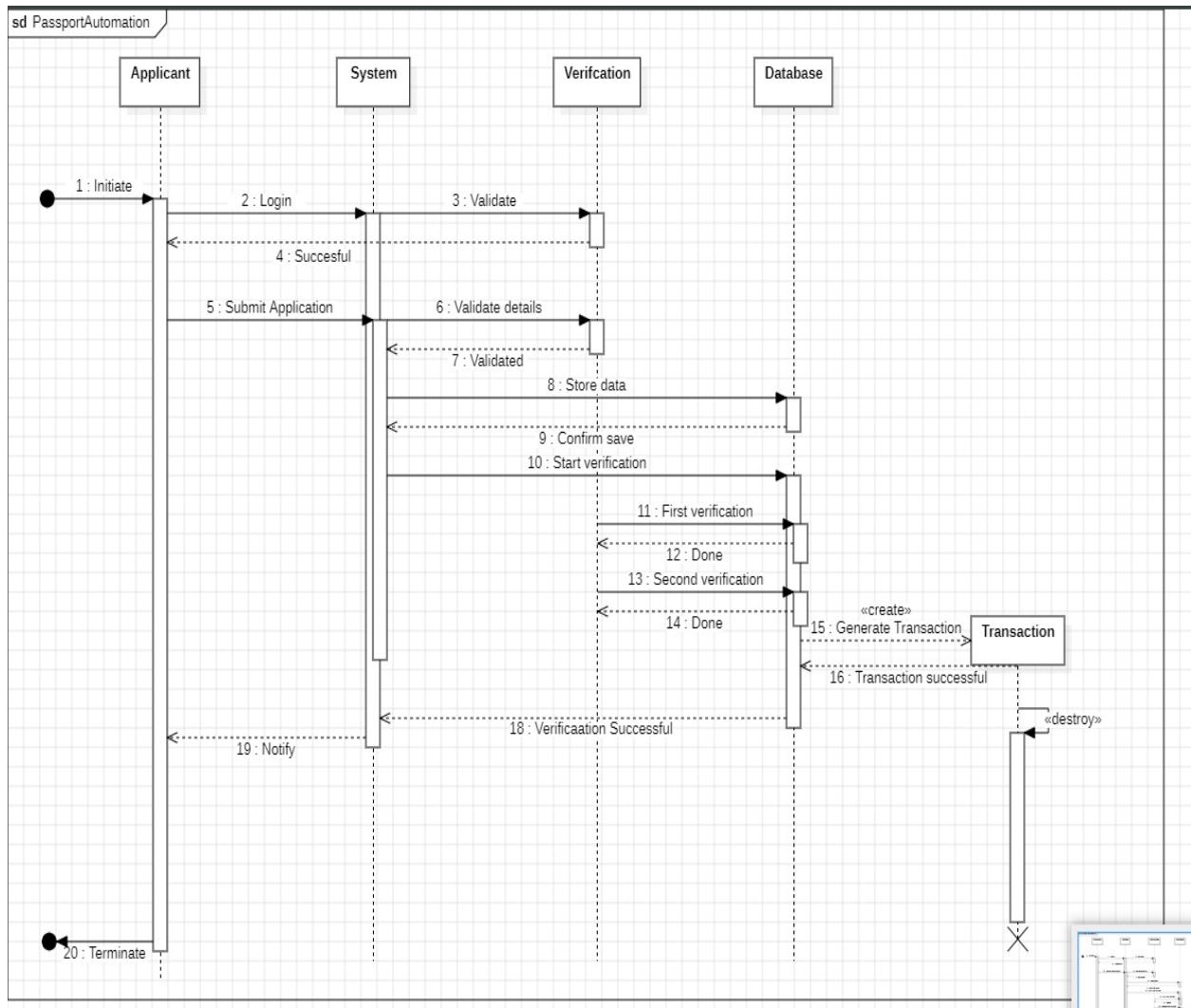
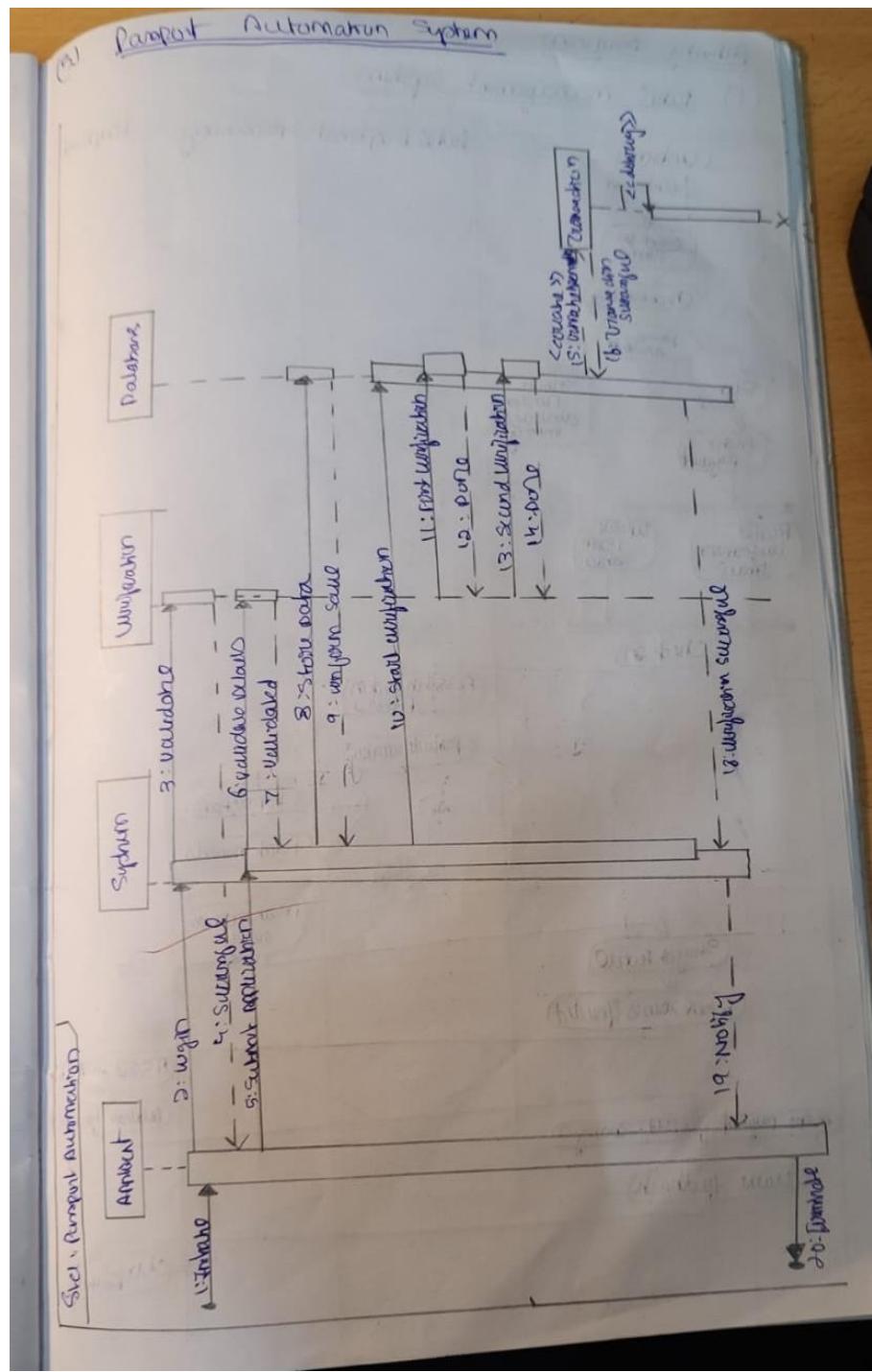


Figure 5.5 – PAS Sequence Diagram



6. INTERACTION MODELLING: ACTIVITY MODELS

Hotel Management System

Key Activities:

1. **Start:** Guest initiates the booking process.
2. **Select Room:** Display available rooms.
3. **Enter Guest Details:** Input personal and contact information.
4. **Process Payment:** Validate and complete payment.
5. **Confirm Booking:** Generate booking confirmation.
6. **Check-In:** Assign room and provide key.
7. **Use Services:** Request and use additional services (optional).
8. **Check-Out:** Generate final bill and mark room as available.
9. **End:** Complete the workflow.

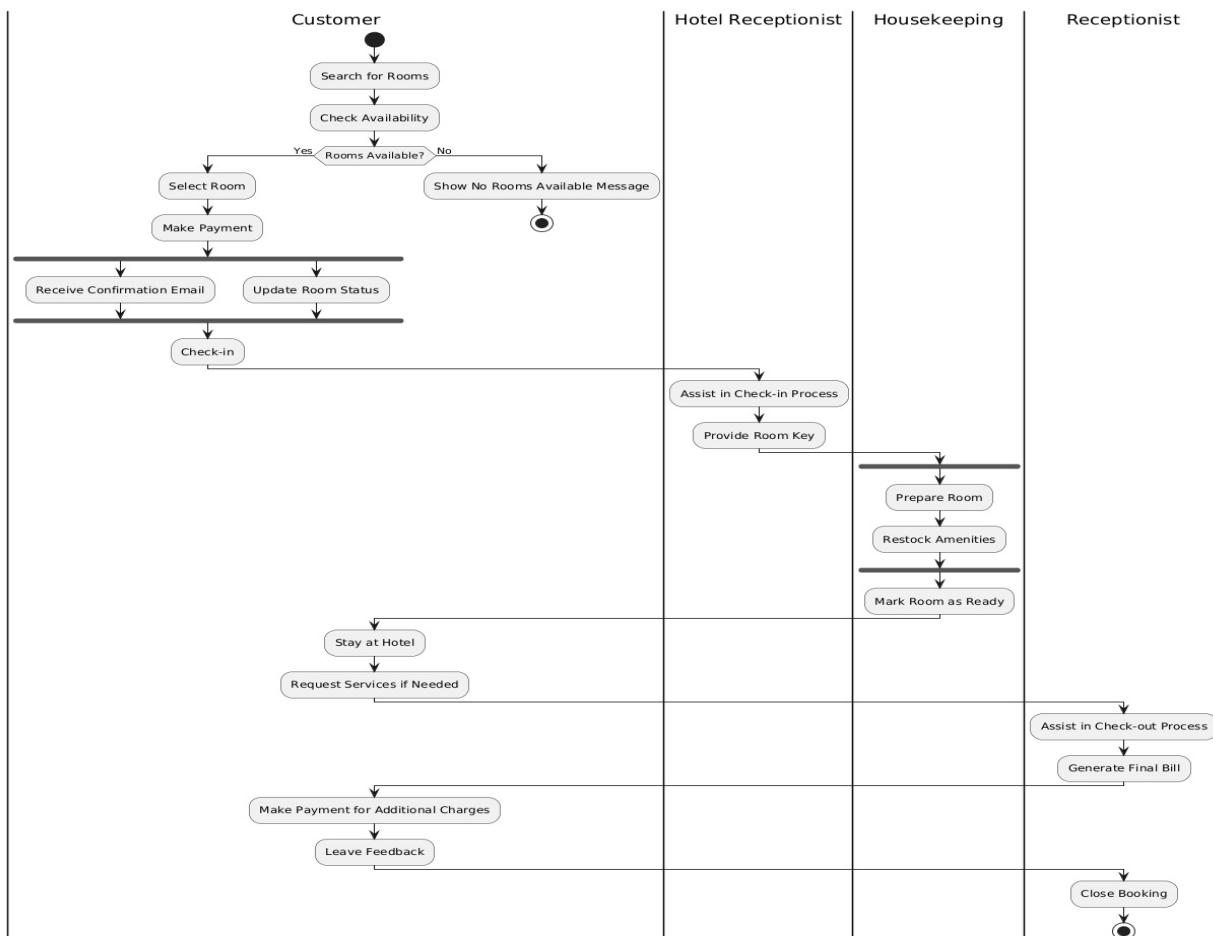
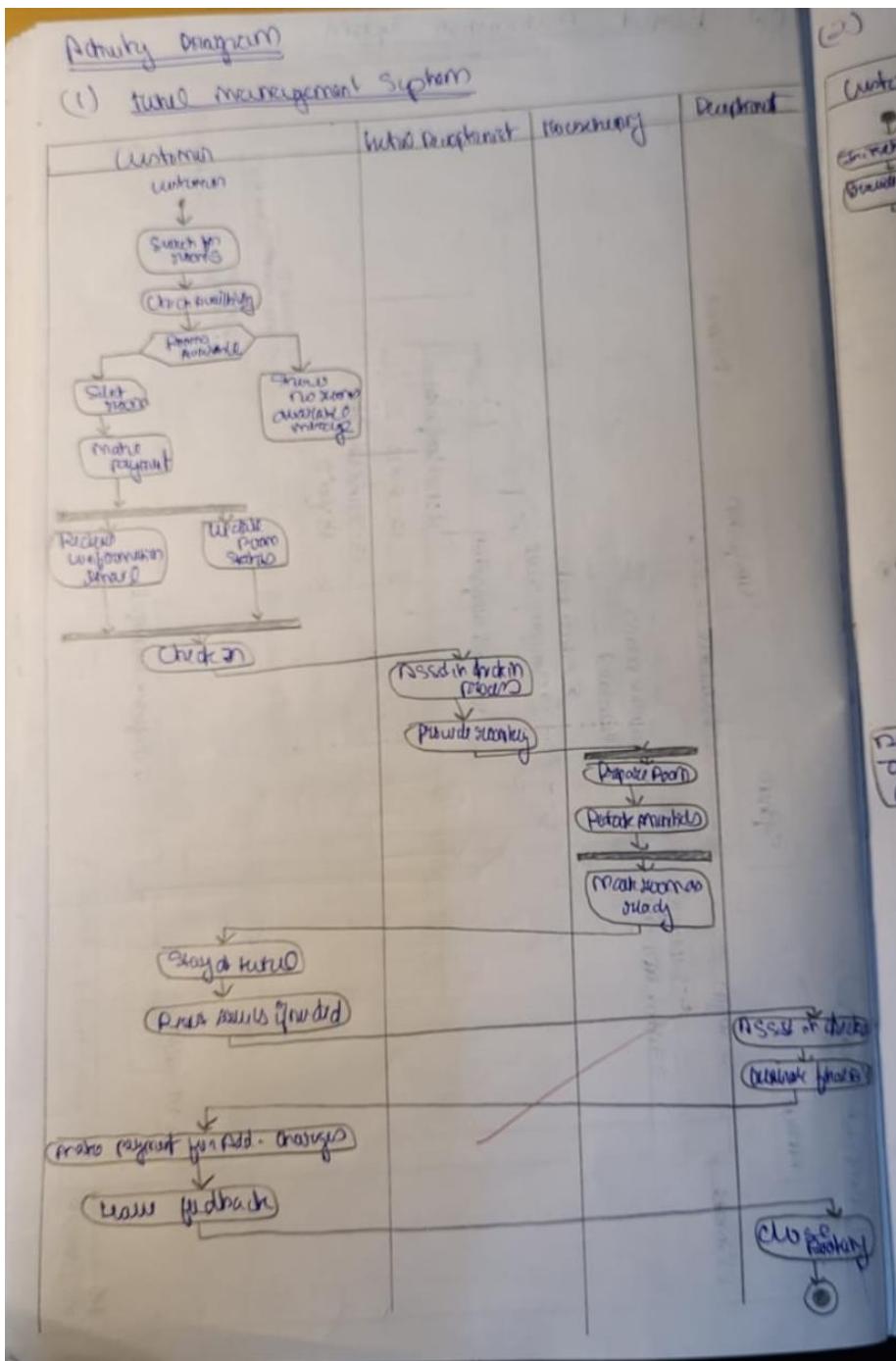


Figure 6.1 – HMS Activity Diagram



Credit Card Processing System

Key Activities:

1. **Start:** Cardholder initiates the transaction.
2. **Input Details:** Enter card and transaction details.
3. **Validate Transaction:**
 - o Check card validity.
 - o Verify funds availability.
4. **Decision:**
 - o **If valid:** Proceed to authorization.
 - o **If invalid:** Reject the transaction.
5. **Authorize Payment:** Secure confirmation from the issuing bank.
6. **Process Payment:** Complete the transaction.
7. **End:** Notify cardholder of transaction status.

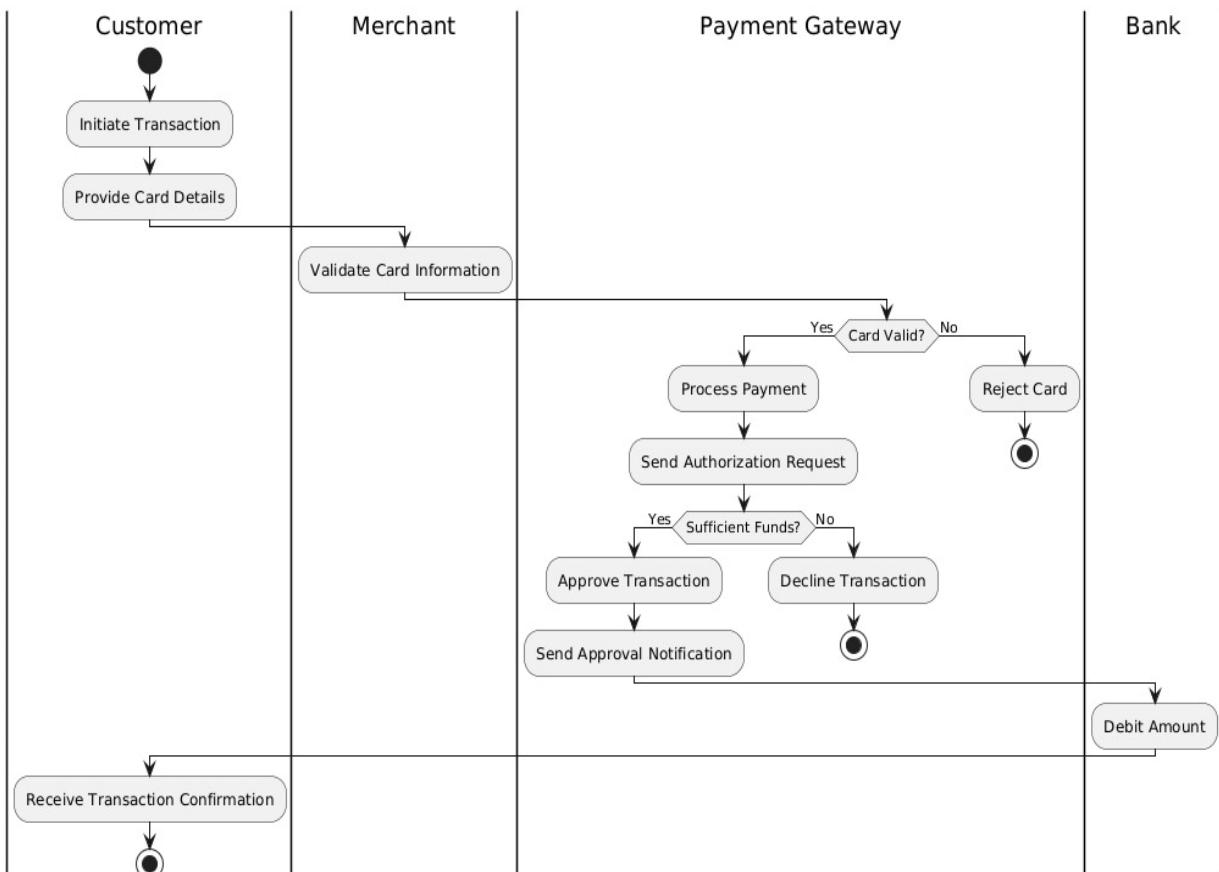
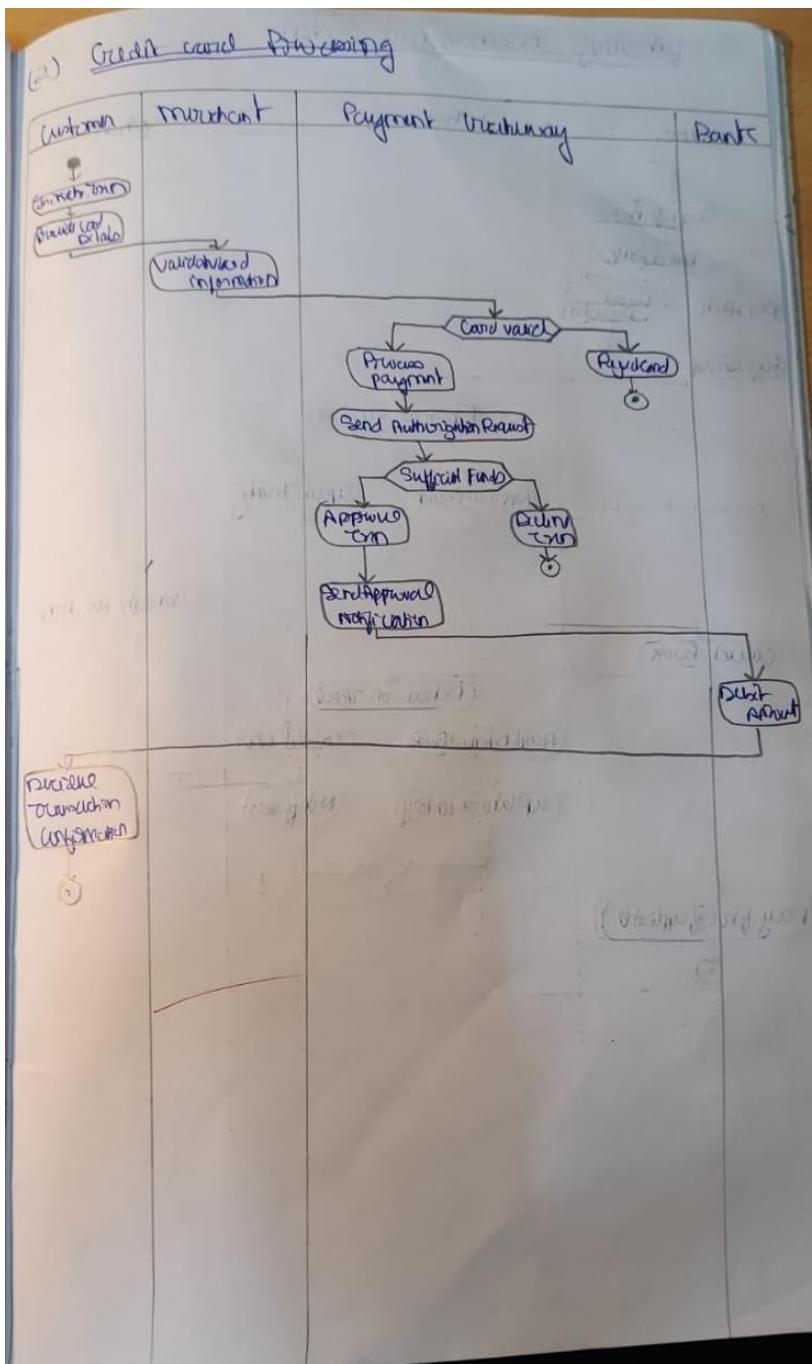


Figure 6.2 – CCPS Activity Diagram



Library Management System

Key Activities:

1. **Start:** Member selects a book.
2. **Check Availability:**
 - o **If available:** Proceed to loan.
 - o **If unavailable:** Notify member.
3. **Borrow Book:** Record loan details and set due date.
4. **Track Loan Duration:** Monitor the loan period.
5. **Decision:**
 - o **If overdue:** Notify member and calculate fine.
 - o **If returned:** Update inventory.
6. **Pay Fine (if applicable):** Member clears overdue charges.
7. **End:** Complete the return process.

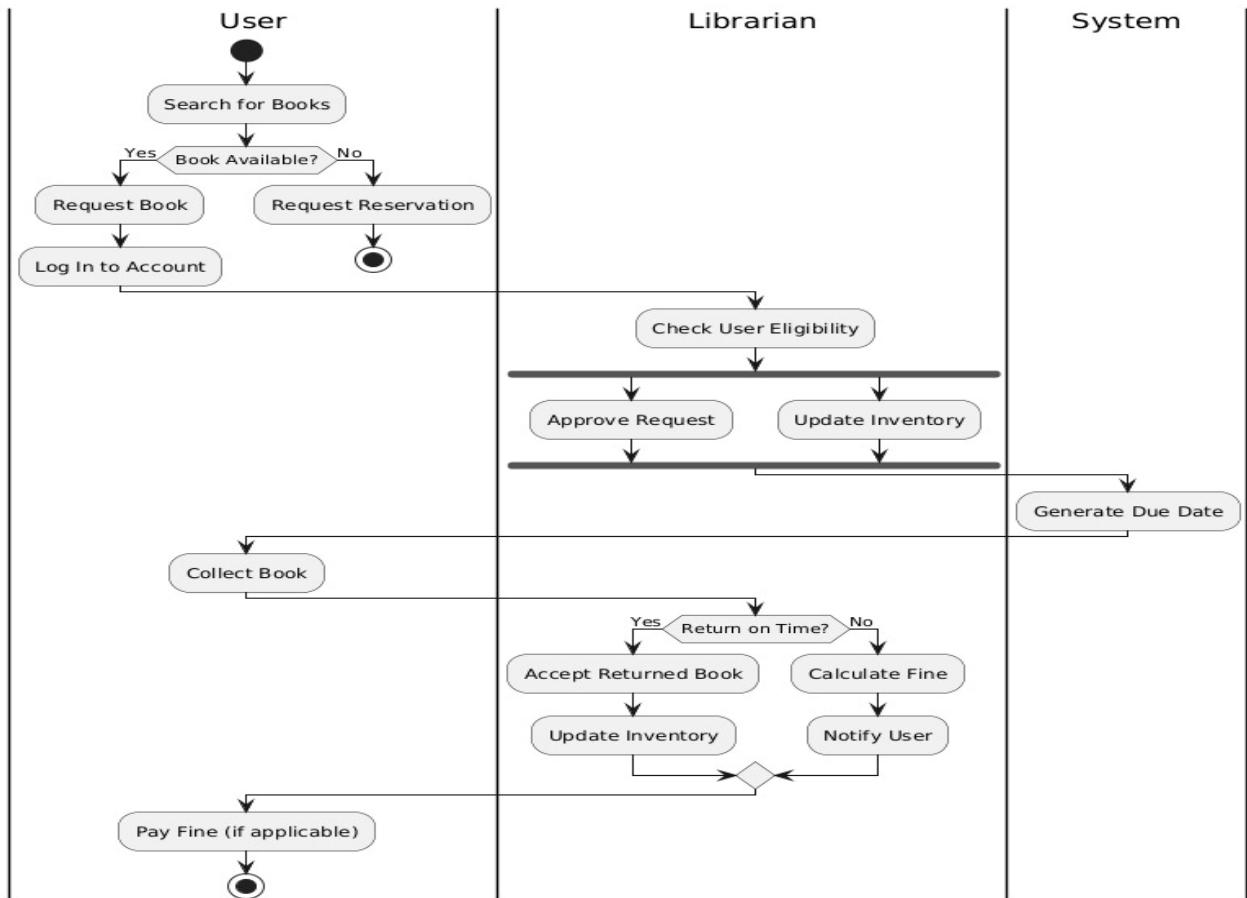
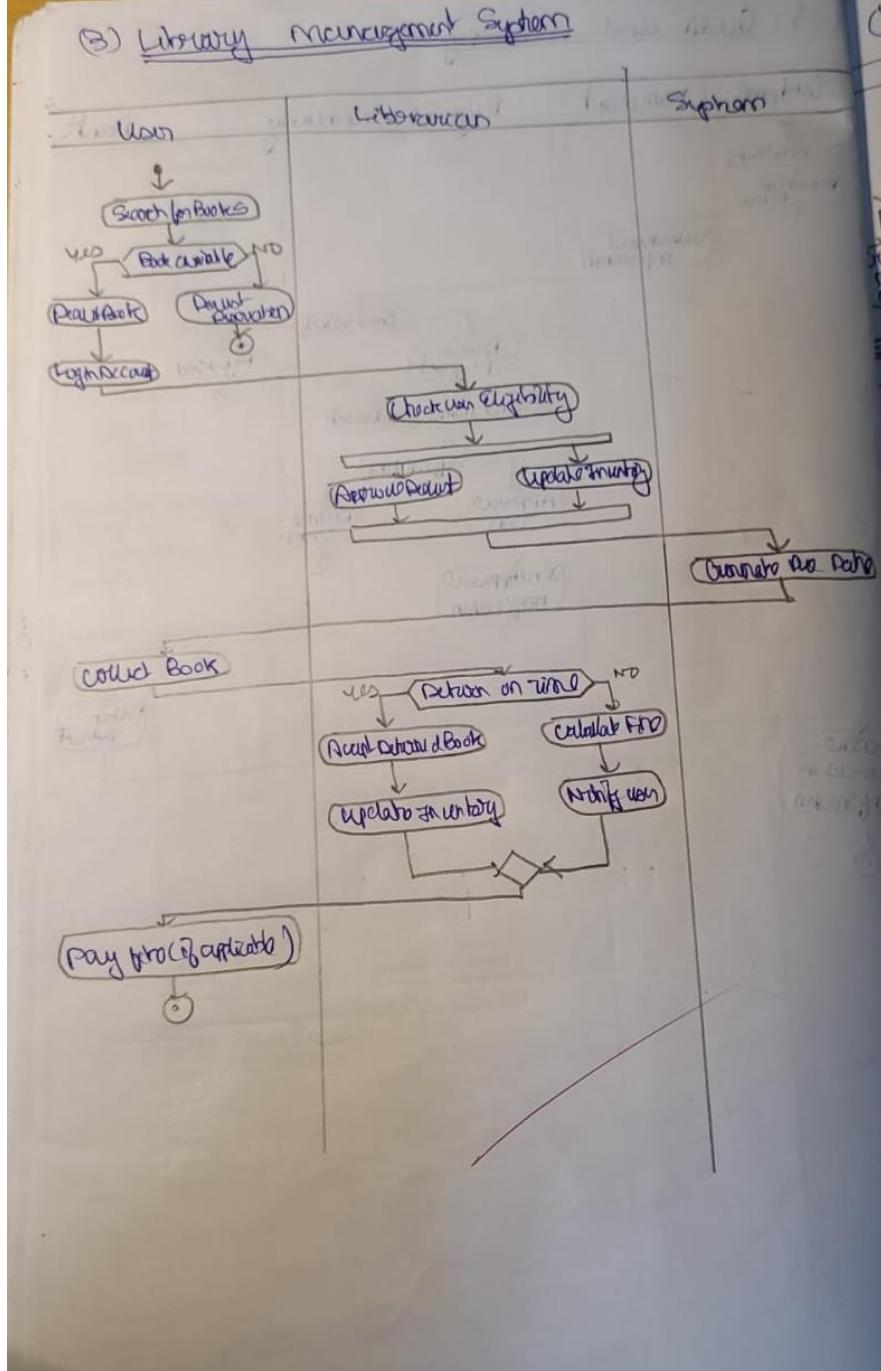


Figure 6.3 – LMS Activity Diagram

(3) Library Management System



Stock Maintenance System

Key Activities:

1. **Start:** Monitor stock levels.
2. **Decision:**
 - o **If sufficient stock:** Continue monitoring.
 - o **If low stock:** Generate an alert.
3. **Place Order:** Initiate a purchase request.
4. **Receive Delivery:** Update stock levels.
5. **Restock:** Replenish inventory and notify stakeholders.
6. **End:** Complete the workflow.

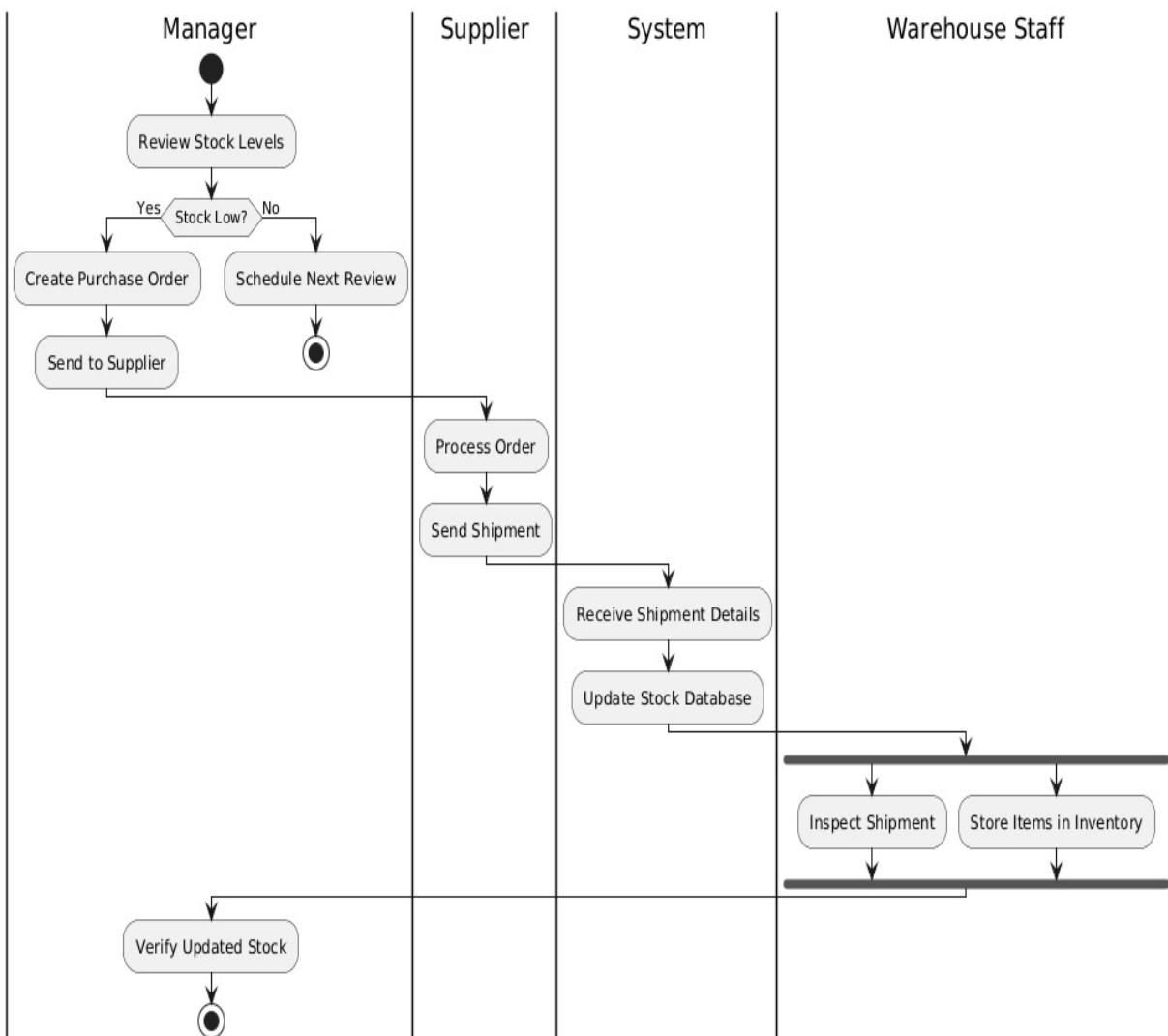
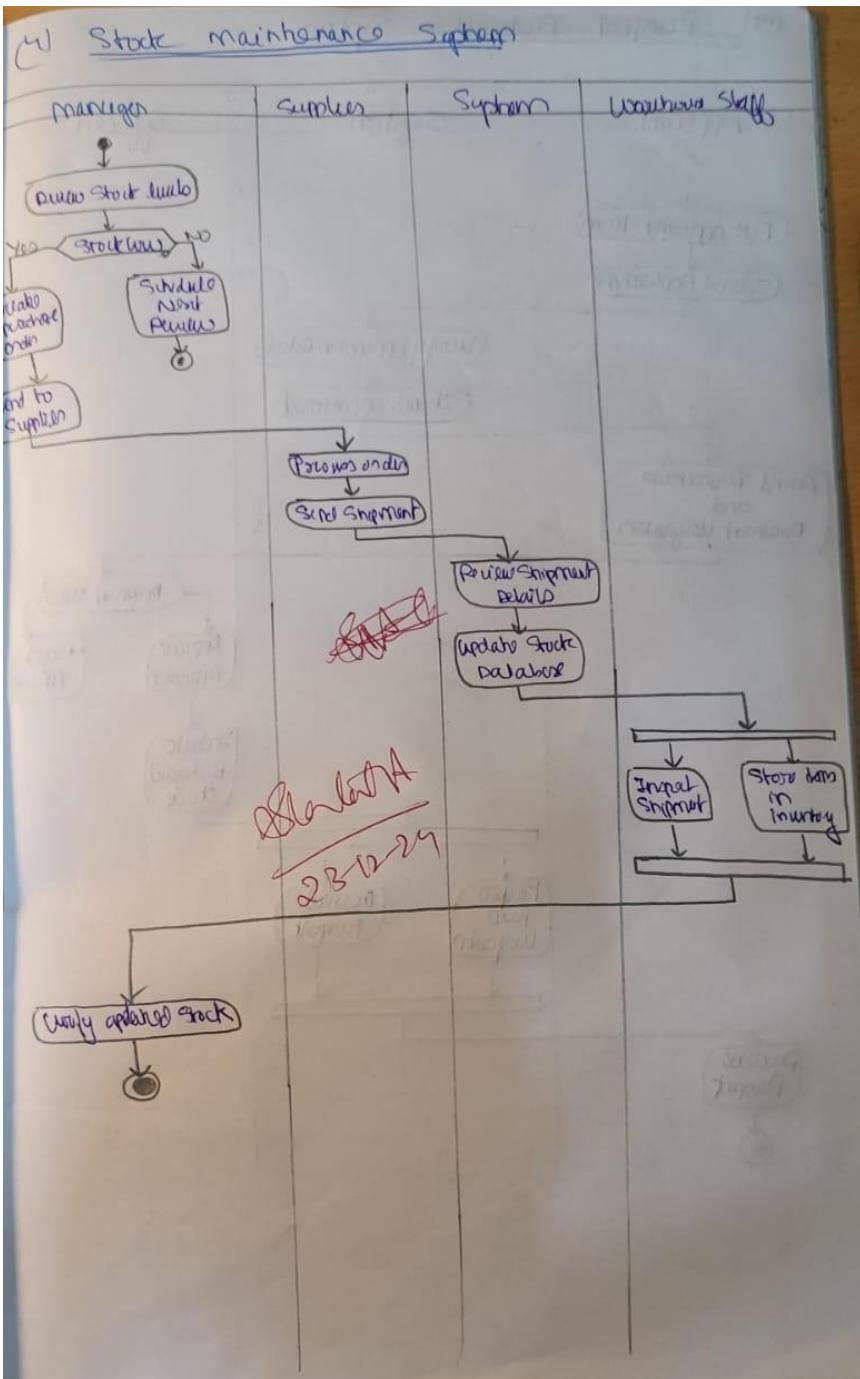


Figure 6.4 – SMS Activity Diagram



Passport Automation System

Key Activities:

1. **Start:** Applicant submits an application.
2. **Schedule Appointment:** Book a verification slot.
3. **Verify Documents:** Check the authenticity of submitted documents.
4. **Decision:**
 - o **If valid:** Approve the application.
 - o **If invalid:** Reject the application.
5. **Issue Passport** (if approved): Generate the passport.
6. **Notify Applicant:** Inform the applicant of the decision.
7. **End:** Complete the process.

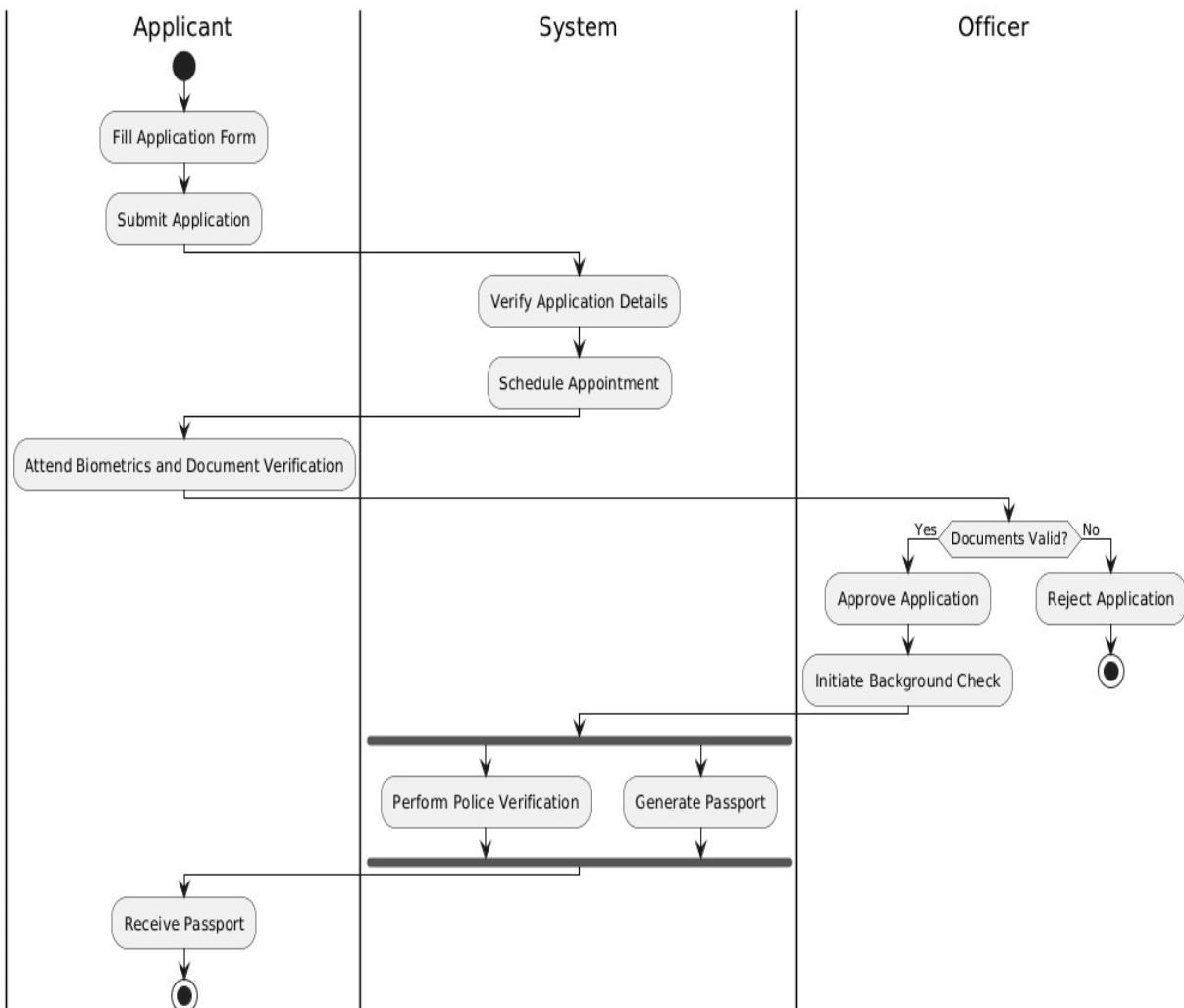


Figure 6.5 – PAS Activity Diagram

(5) Passport Automation System

