

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:

MANAV KUMAR

1BM22CS348

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 Lab work entitled "**OBJECT ORIENTED MODELING**" was carried out by **Manav Kumar (1BM22CS348)**, who is a Bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-2025. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Modeling- (23CS5PCOOM)** work prescribed for the said degree.

Spoorthi D M

Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. KAVITA SOODA

Professor and Head
Department of CSE
BMSCE, Bengaluru

Github Link: https://github.com/Manav-Kumar123/MANAV_1BM22CS348_OOM

Table Of Contents

Sl.no.	Experiment Title	Page no.
1.	System Requirements Specifications	4-21
2.	Class Modelling	22-31
3.	State Modelling	32-41
4.	Interaction Modelling: Use Case Models	42-51
5.	Interaction Modelling: Sequence Models	52-61
6.	Interaction Modelling: Activity Models	62-71

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.

~~2023~~ 23/09/2024

B- SRS document for Hotel Management System:-

→ ① Introduction:

1.1. Purpose:

This document outlines the requirements for a hotel management system (HMS) which serves the purpose of guiding the developers.

1.2 Scope of this document:

This document covers all and small functionality of a HMS, and it emphasizes its goals and the values it offers users such as streamlined operations. It also includes estimates for development costs and timelines.

1.3 Overview:

The HMS is designed to automate and streamline hotel operations, including reservations, check-ins, billing and customer relationship management, with the aim of improving efficiency and customer satisfaction.

② General Description:-

The HMS will assist hotel staff and guests, featuring key user roles such as front desk personnel, managers and guests. Key functionalities include room booking, billing, reporting, and inventory management.

③ Functional Requirements:

- **User Registration/ Login:** Secure account creation and login for users.
- **Room - Reservation:** Search and book rooms based on availability.
- **Check-in / Check-out:** Automated guest check-in and check-out processes.
- **Billing System:** Invoice generation is based on utilized services.

- Reportings: Generate reports on occupancy, revenue, and customer feedback.

④ Interface Requirements:

- User Interface: A web-based interface accessible to guests and staff.
- API Integration: Interfaces for external payment system, location services and other third party services.
- Data Exchange: To interchange data between the system modules.
- Room Table Reservation: Provides a user friendly web-table reservation interface that shows availability & pricing of rooms & time slots for table booking.

⑤ Performance Requirements:

- Response Time: System responses should occur within 1 second for most of operations.
- Data Handling: Efficient management of upto 100000 records.

⑥ Design constraints:

- Technology: Used specified frameworks (React, Node.js)
- Database: Use relational database (MySQL).

⑦ Non-functional Attributes:

- Security: Protect user data with encryption.
- Scalability: Support for future system expansion.
- Reliability: Guarantee of 99.99% uptime.
- Usability: Intuitive user interface for ease of use.
- Efficiency: Ensure optimized resource utilization, reducing latency and minimizing memory usage while processing large datasets or concurrent users.

⑧ Preliminary Schedule & Budget:

- Estimated duration: Six months for development & testing.
- Projected cost: £10000000 which covers development, testing and deployment.

~~20/9~~

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

Q. SRS document for credit card processing system

(1) Introduction:

1.1 Purpose of this document: This document specifies the requirements for credit card processing (CCPS) aimed at processing, authorizing and settling credit card transactions efficiently and securely.

1.2 Scope of this document:

The system will manage transactions between customers, merchants, and banks including payment authorization, transaction processing, fraud detection & reporting.

1.3 Overview:

CCPS will streamline credit card payments and ensure secure transactions while providing real-time processing and data reporting.

(2) General Description:

The system will serve merchants and cardholders by offering secure and fast credit card processing. It will support multiple card types and banking regulations.

(3) Functional Requirements:

- Transaction authorization
- Payment processing and settlement.
- Fraud detection and prevention.
- Reporting and transaction history.

(4) Interface Requirements:

The system will communicate with external banking networks and user interfaces via secure API's and standard data exchange formats.

⑤ Performance Requirements:-

The system must process transactions within 1 second and handle peak loads with low latency and high accuracy.

⑥ Design constraints:

The system must process transactions within seconds and handle peak loads with low latency and high accuracy.

⑦ Non-Functional Attributes:

- Security: High-level encryption for data protection.
- Reliability: 99.99% uptime.
- Scalability: Ability to handle high transaction volumes.
- Portability: Support for multiple platforms.

⑧ Preliminary Schedule & Budget:

- Estimated duration: six months for development & testing.
- Estimated cost: £1000000 for development, testing & deployment.

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.

LAB-2

Manual
Date _____
Page 05

Date: 30/09/2024:-

Q. SRS document for Library Management System (LMS):

① Introduction:

1.1 Purpose of this document:

This document outlines the requirement for LMS, serving as a guide for developers to ensure, alignment, performance and overall objectives.

1.2 Scope of this document:

The LMS will facilitate efficient management of library operations including book borrowing, returns, cataloging and user management. Development cost is ₹ 1500000 with a timeline of 6 months for project completion.

1.3 Overview:

The LMS is designed to streamline library processes, ensuring easy track of library resources and efficient member management.

② General Description:

The LMS will help streamline the management of book inventory, member information and the issuance and return of books. Users will be able to search for books, check their availability, and manage borrowing. Administrators will have full access to manage members and books, including adding new books, removing when necessary.

③ Functional Requirements:

3.1 Book Management:

- Add new books, delete books, track book availability.

3.2 Member Management:

- Register a new member, delete an existing member and assign a unique membership id to each member.

3.3 Issue and Return Books:

- Librarians and issue book, members can return books.
- The system calculates overdue fines automatically.

3.4 Search functionality:

- Users can search for books by title, author, ISBN or genre.

(4) Interface Requirements:4.1 Software Interfaces:

- The LMS will communicate with an internal database to store all information related to books, members and transactions.

4.2 User Interfaces:

- Login Screen: Fields for username & password to authenticate users.
- Book Search Interface: A search bar for searching books by title, author or ISBN.
- Issue/Return Interface: Dropdown lists for selecting books and members when issuing or returning books.
- Member Registration Interface: Form for entering member details.

(5) Performance Requirements:

- The system should be able to handle upto 10,000 users.
- The response time for a search query should not exceed 3 seconds.
- The system should be able to manage up to 1,000,000 books records without performance degradation.
- Fine calculation for overdue books should occur in real time.

⑥ Design constraints:

- The system must be implemented using a relational database such as MySQL.
- The interface should be built using web technologies to ensure cross-platform compatibility (e.g. HTML, CSS, JS).
- The system must support scalability for future upgrades and extensions.
- The user interface must be accessible and responsive across different screen sizes.

⑦ Non-Functional Attributes:

- **Security:** The system must ensure that only authorized users can access sensitive features.
- **Reliability:** The system should operate with minimal downtime and have an uptime of 99.99%.
- **Scalability:** The system must handle and increase in users, books, and transactions without a loss in performance.
- **Portability:** The system should run on various OS.
- **Maintainability:** The codebase must be modular and well-documented to allow for future maintenance & upgrades.

⑧ Preliminary Schedule and Budget:

- **Development Time:** The project will take approximately 6 months from the initial design phase to the final deployment.
- **Estimated cost:** The total cost for development, including hardware and software, is estimated at ₹ 50000000. This includes cost of designs, development, testing and deployment.

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.

Q. SRS document for Stock Maintenance System (SMS) :-

① Introduction :

1.1 Purpose: This document describes the requirements for the stock maintenance system (SMS), a software solution for tracking inventory, managing orders, and monitoring stock levels in warehouse or retail outlets.

1.2 Scope: The SMS will automate inventory management, including stock entry/updates, order tracking, and low stock alerts. It will improve efficiency and reduce manual errors, catering to businesses of all sizes.

1.3 Overview: The system will enable users to manage stock efficiently through features like real-time stock updates, supplier management, order tracking & report generation.

② General Description:

The SMS is designed to provide real-time information on stock-levels, orders and suppliers. It allows users to manage stock efficiently by providing a user-friendly interface to input, update and monitor stock data. Users can also generate reports to analyse stock trends and make informed decisions on ordering more products or managing supplier relations.

③ Functional Requirements:

3.1 Stock Management:

- Add and update stock with product details.
- Track stock availability and generate low stock alerts.

3.2 Order & supplier Management:

- Place, track and manage orders.
 - Manage supplier information and track delivery performance

3.3 Report Generation:

- Generate reports on stock usage, levels and supplier efficiency.

④ Interface Requirements:

9.1 Use Interfaces:

- Login: Authentication for users.
 - Stock Management: Manage stock details and availability.
 - Order and Supplier Management: Track orders & manage suppliers.

⑤ Performance Requirements:

- The system should handle up to 10,000 transactions.
 - Low stock alerts and updates must occur in real time.

⑥ Design constraints

- The system will use a relational database and be compatible with mobile devices.
 - Secure communication protocols (e.g. HTTPS) will ensure data protection.

⑦ Non-functional Attributes:

- Security: Access control for authorized users.
 - Scalability: Supports increased stock levels and users.
 - Reliability: 99.9% uptime and fast response time for transactions.

⑧ Preliminary Schedule & Budget:

- Development Time: estimated at 6 months.
 - Budget: 600 000 000 option for design / development & deployment.

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.

Q. SRS document for Passport Automation System (PAS) :-

① Introduction:

1.1 Purpose: The purpose of this document is to outline the requirements for the PAS. The system will automate the process of passport application, verification and issuance, reducing manual intervention & streamlining the workflow for both applicants and officials.

1.2 Scope: The PAS will manage the complete passport application lifecycle, from submission and verification to issuance and renewal. It will also provide real-time status updates, track application progress, and integrate with various government databases for validation purposes.

1.3 Overview: PAS will allow users to submit applications online, verify details, upload required documents, and track the status of their application. Government officials will have tools for verifying information, managing the approval process & issuing passports.

② General Description:

PAS is a web-based system that automates the traditional, paper-based passport process. It interacts with national identification, address verification and immigration database to streamline data validation. The system is designed to replace the manual handling of passport applications with an automated, user-friendly interface.

③ Functional Requirements:

3.1 Application Management:

- Applicants can create accounts, fill out passport applications, and upload documents.
- Officials can review, approve or reject applications based on provided information.

3.2 Document verification:

- The system will validate uploaded documents against national databases.
- Officials will manually check flagged documents that require further investigation.

3.3 Passport issuance:

- After approval, the system generates a passport for printing, and applicants are notified of the issuance.
- The system will also handle passport renewals and cancellations.

3.4 Status Tracking and Notifications:

- Applicants can check the status of their application in real-time.
- Email/SMS notification will be sent at key stages.

④ Interface Requirements:

- Applicant portal: enables applicants to register, fill out applications, upload documents and track status.

- Official portal: enables government officials to review applications, perform verifications and approve/reject requests.

- Admin Portal: Provides system administrators with the ability to manage user roles, monitor system performance and generate reports.

(5) Performance Requirements:

- The system should support up to 10000 simultaneous applications.
- Real-time tracking and notification features must update within 5 seconds.
- The system should issue within 1 business day after application approval.

(6) Design Constraints:

- The system must be integrated with national databases for identity verification and criminal background checks.
- Secure communication protocols (HTTP) & encryption of sensitive data (e.g. passport details) are mandatory.

(7) Non-functional Attributes:

- Security: The system must ensure data confidentiality and restrict access to authorized users.
- Scalability: PAs should accommodate future growth, including an increase in users and applications.
- Reliability: The system must have 99.9% uptime and handle large-scale user access without performance issues.

(8) Preliminary Schedule & Budget:

- Development time: Estimated development time is 8 months.
- Estimated Budget: The project is estimated to cost 1500000/- covering design, development, testing & deployment.

21/10/2023

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.

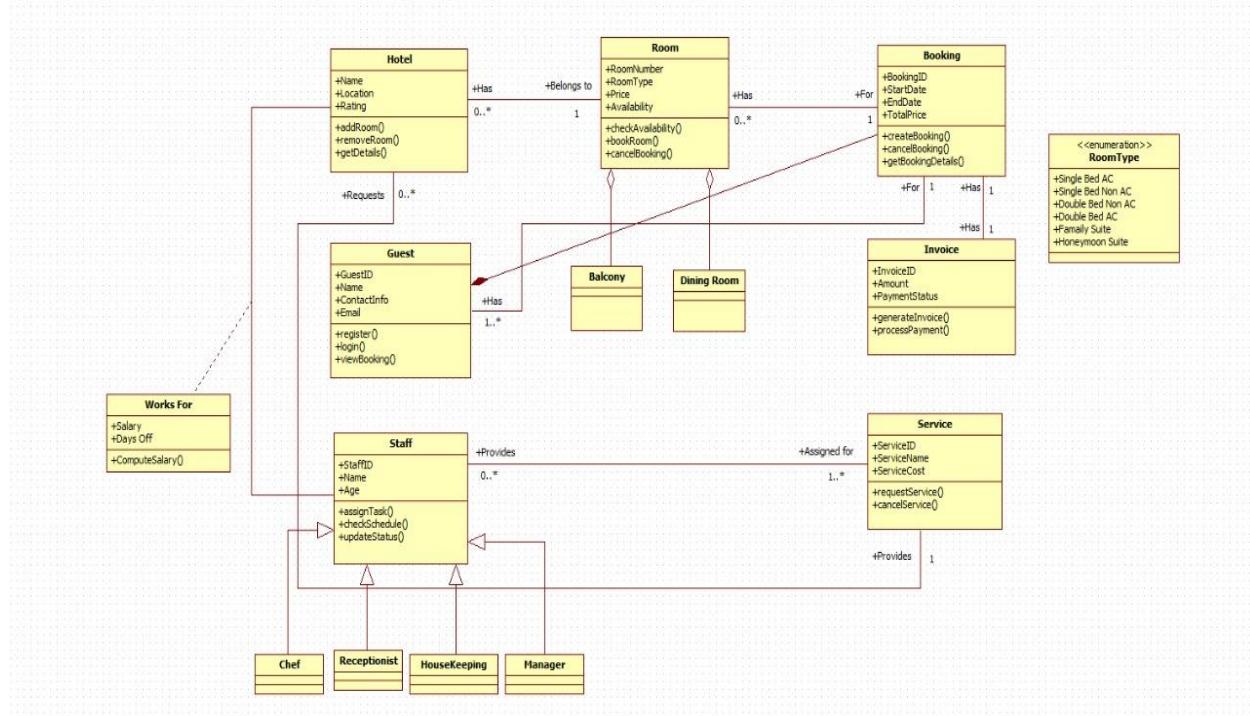


Figure 2.1 - HMS Class Diagram

18.

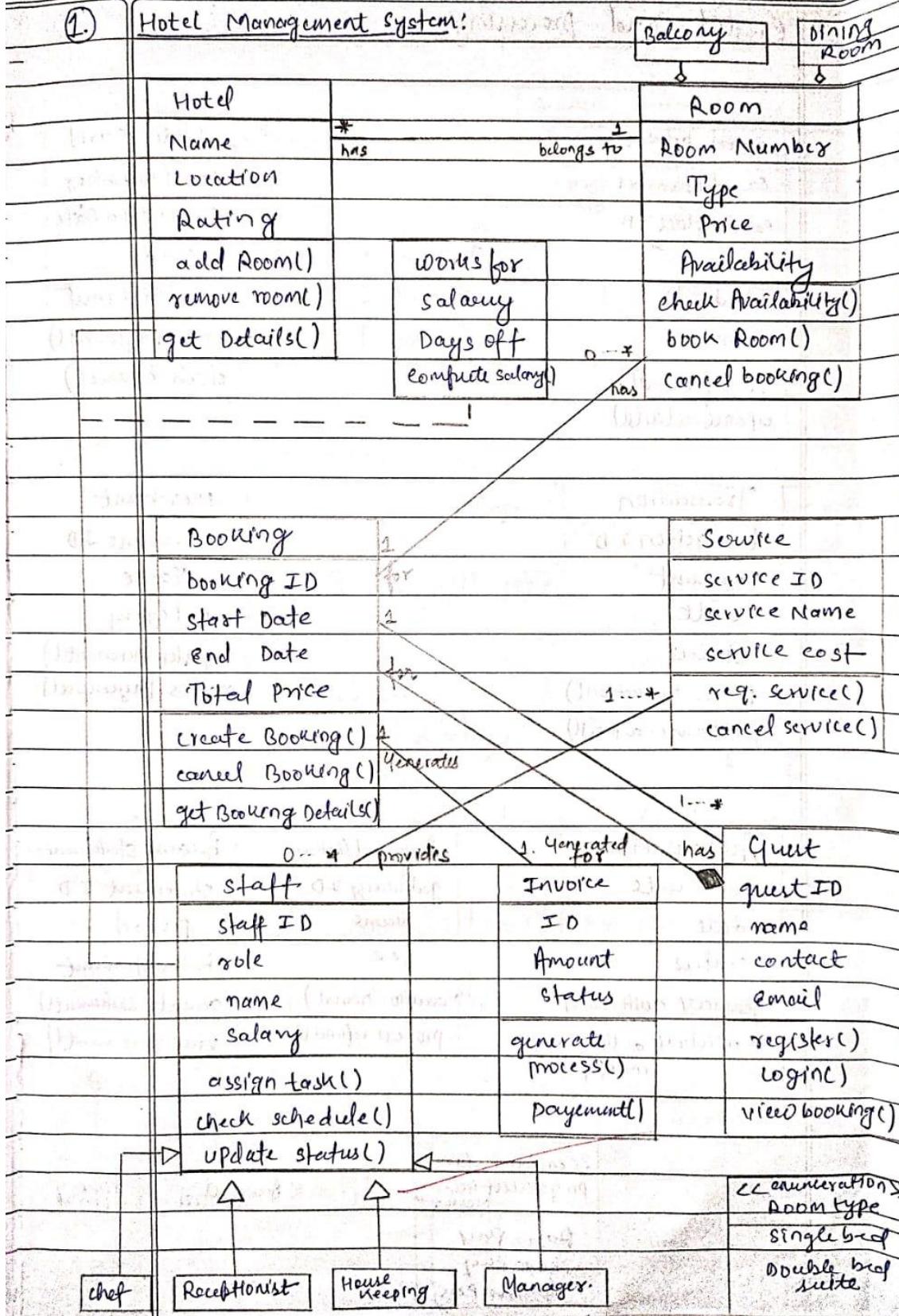
Class Diagrams (Advanced)

LAB-4

SURYA Gold

Date _____ Page _____

(1)

Hotel Management System:

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.

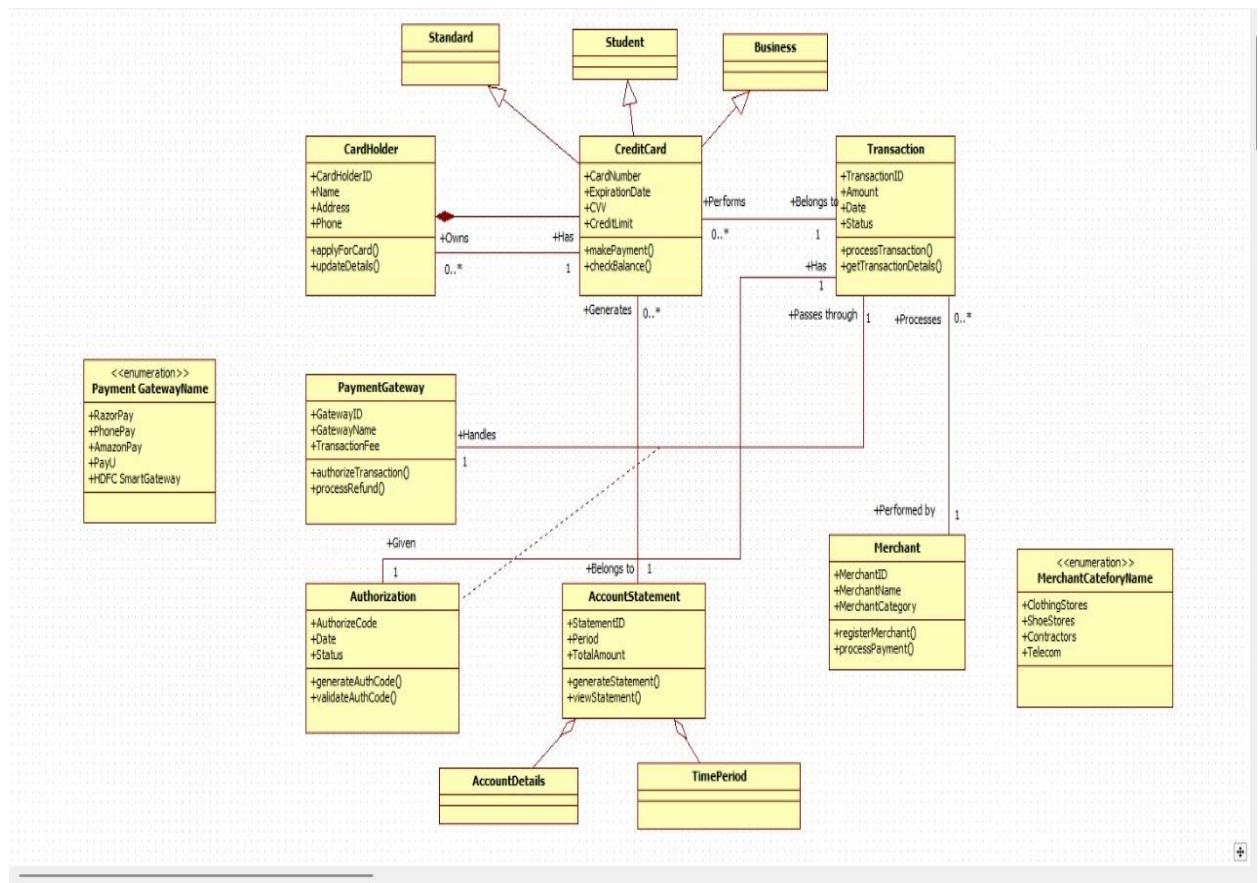
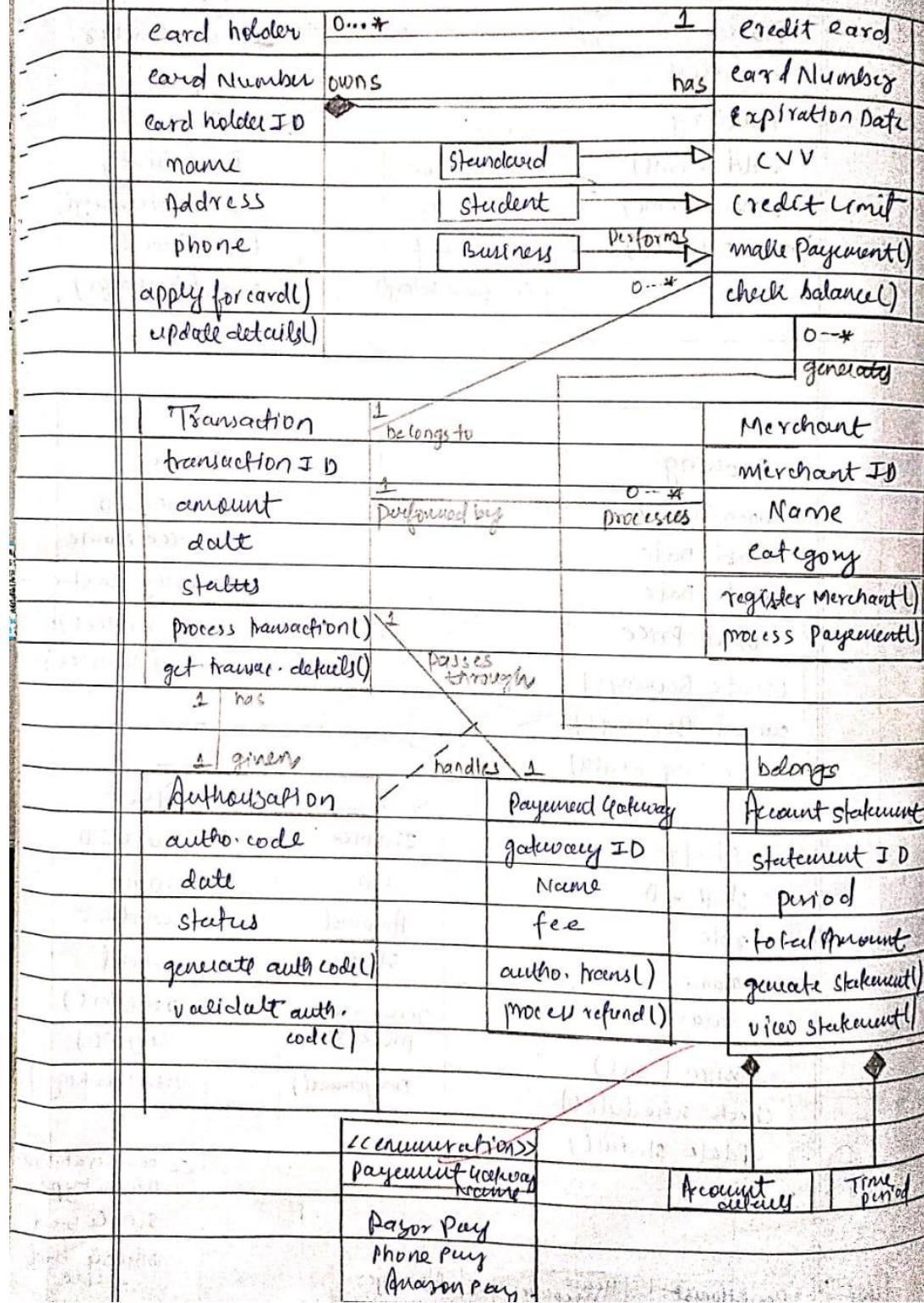


Figure 2.2 – CCPS Class Diagram

2. Credit Card Processing:-



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.

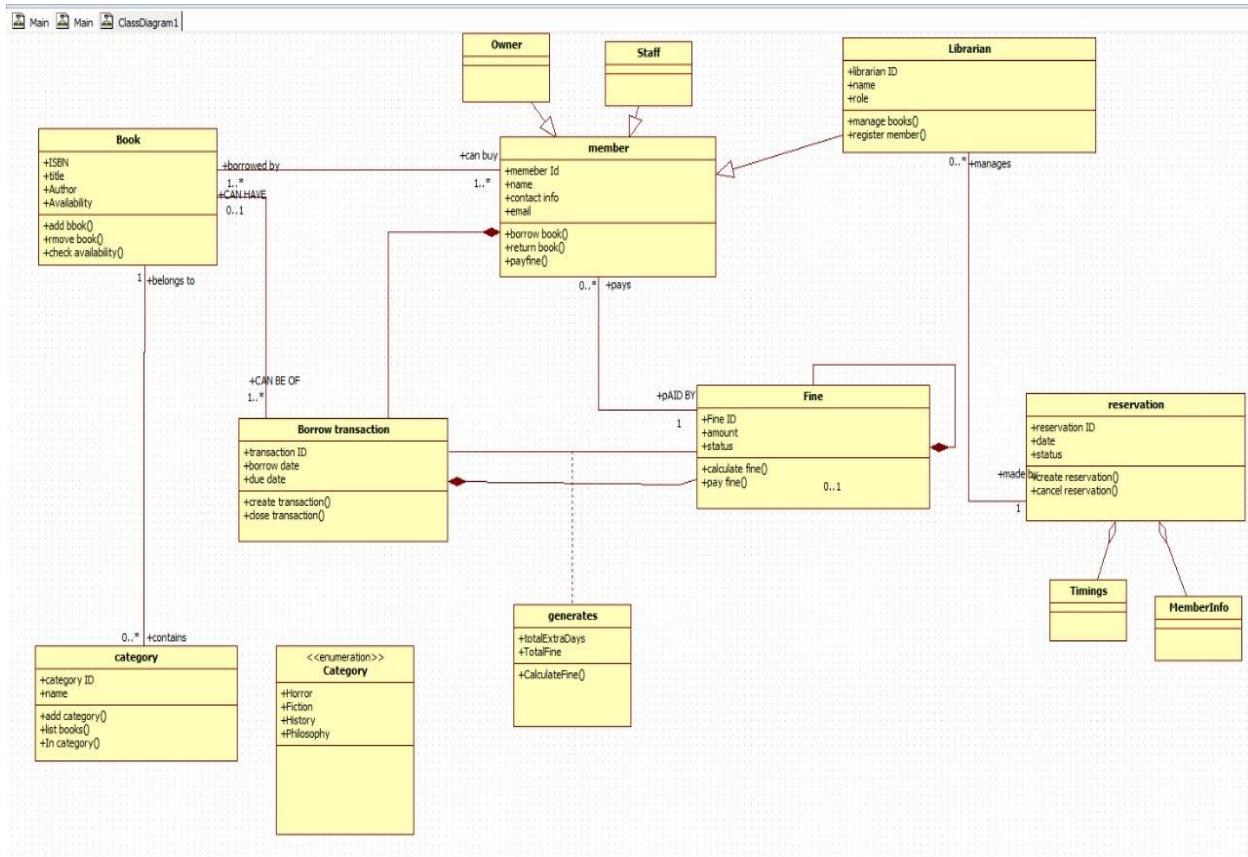
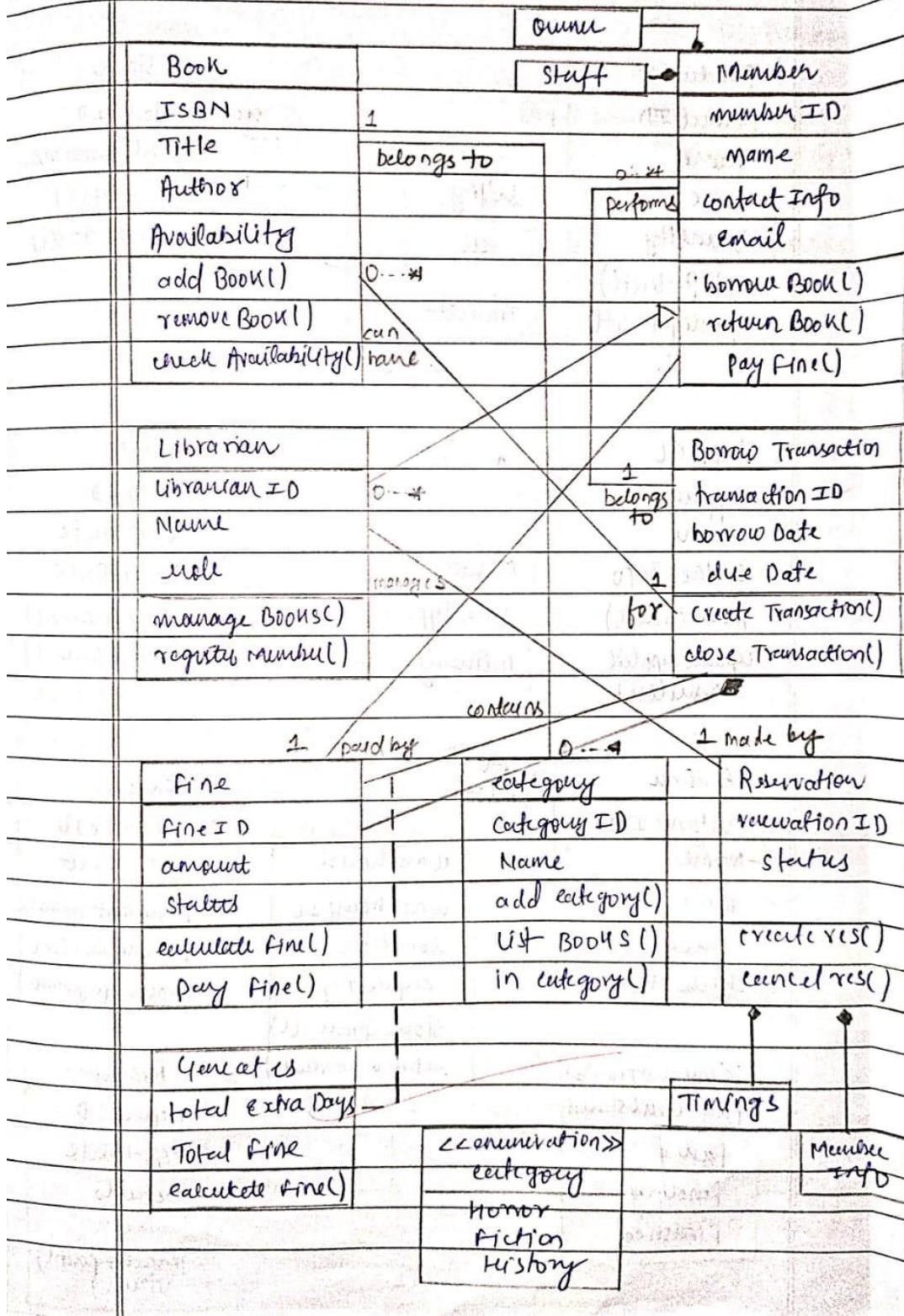


Figure 2.3 – LMS Class Diagram

3

Library Management System:-



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.

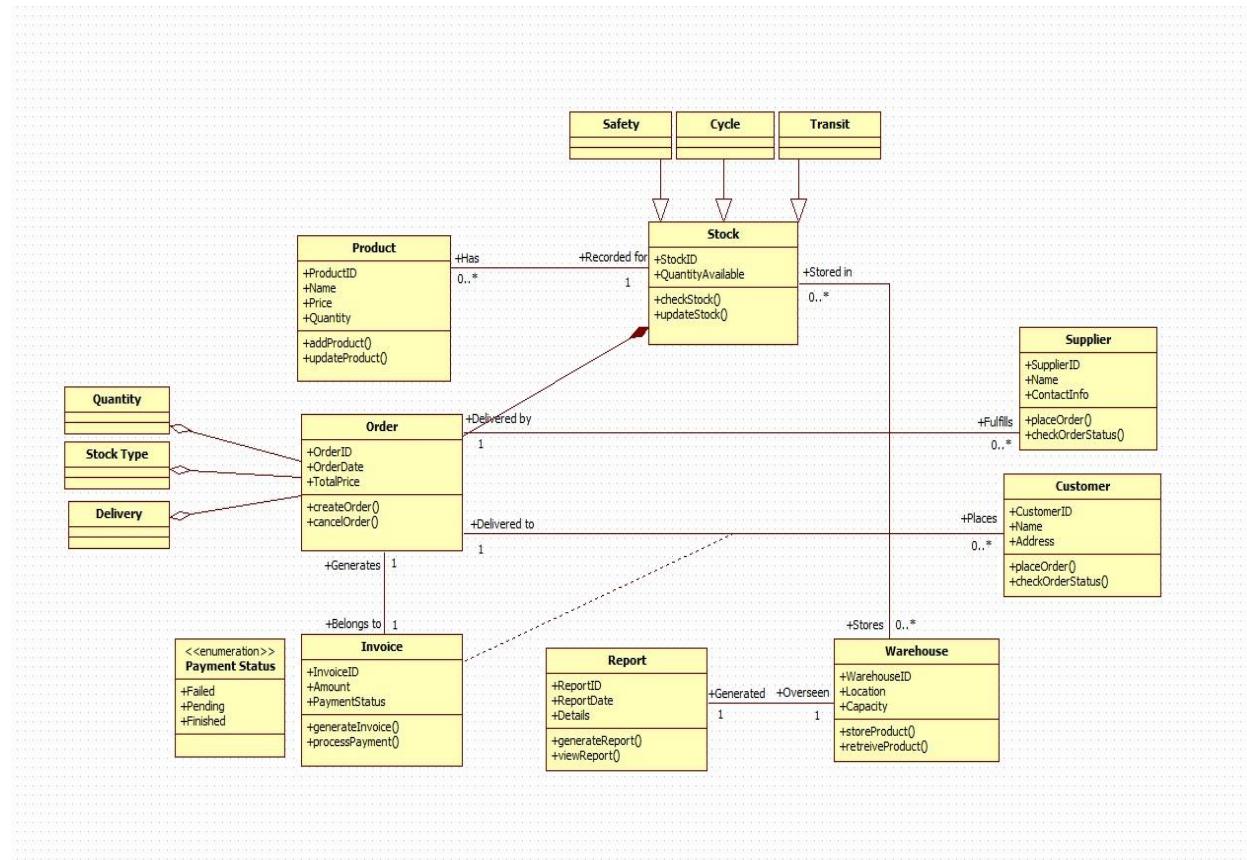


Figure 2.4 – SMS Class Diagram

(1) Stock Maintenance System:

product	0...*	1	Stock
product ID	has	recorded	stock ID
name		for	quantity available
price	safety	0...*	check stock()
quantity	cycle	stored in	update stock()
add Product()			
update Product()	Transit		

Supplier	0...*	1	Order
Supplier ID	fulfills	delivered by	order ID
Name			Order Date
Contact Info	Quantity	1	total price
place order()	Stock Type		create order
update supplier details()	Delivery		cancel order

Customer	0...*	places		Invoice
Customer ID	0...*	Stores		Invoice ID
Name		Warehouse		amount
Address		Warehouse ID		payment status
place order()		location		generate invoice
check status()		capacity		process payment
		store product()		
		retrieve product()		

<Enumeration>			Report
Payment status	1	Unseen	Report ID
Failed			Report Date
Pending		Generated	Details
Finished.			generate report() view()

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.

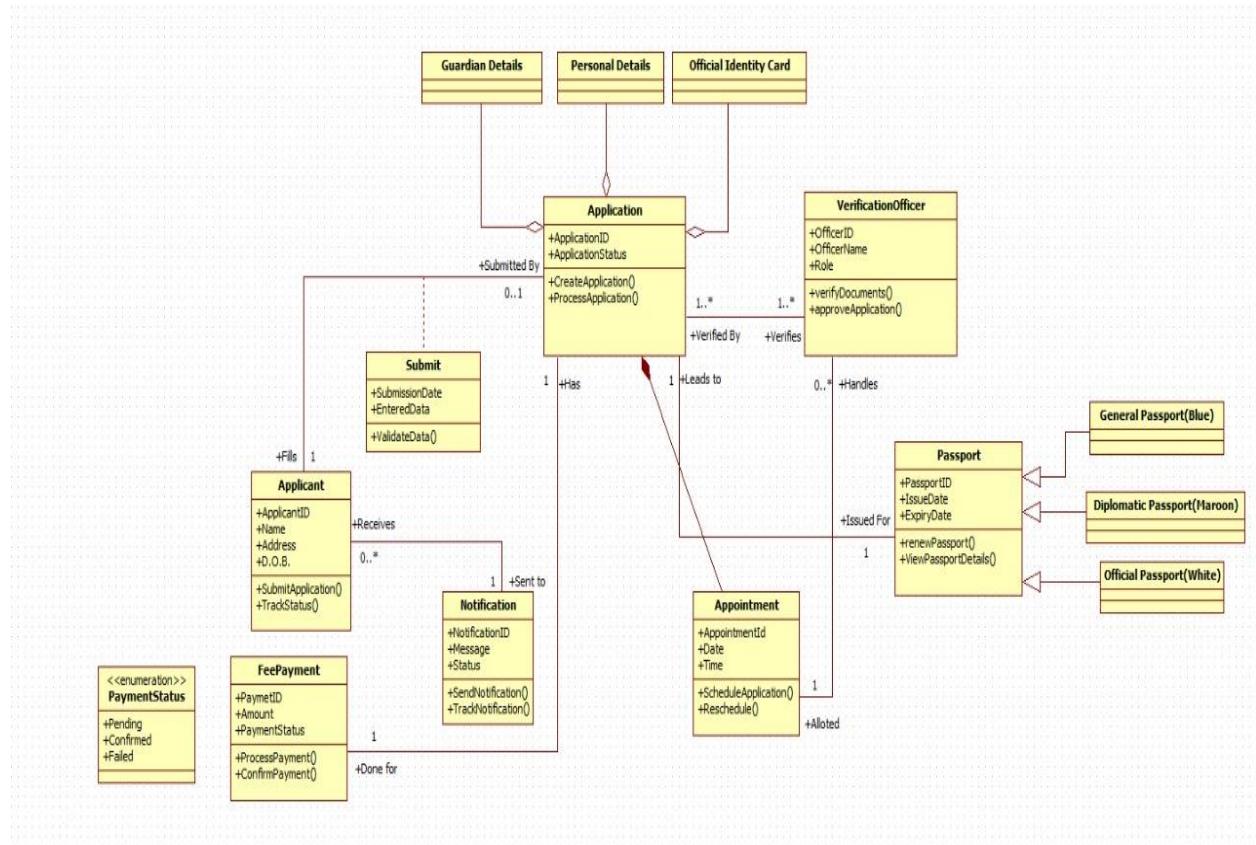
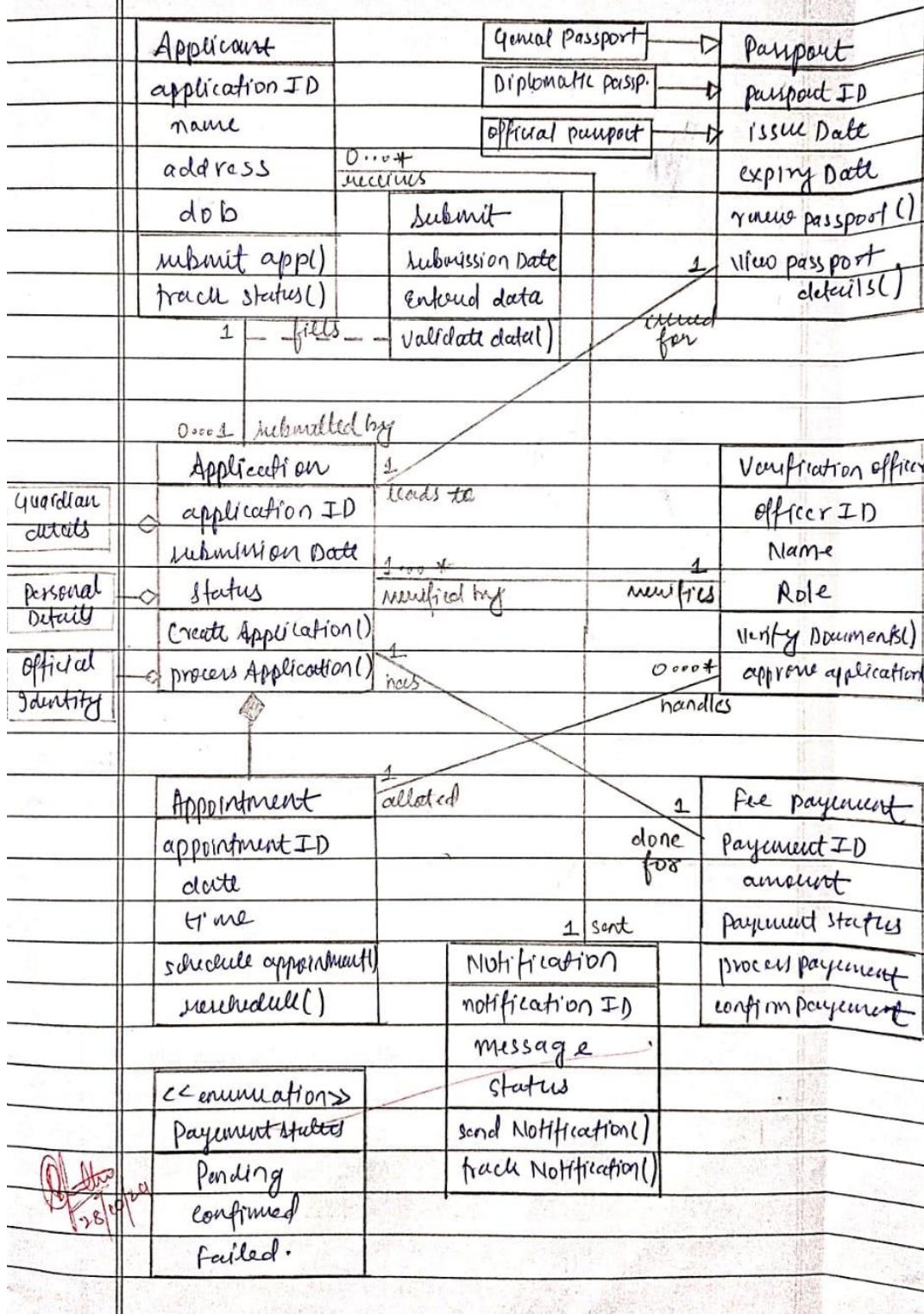


Figure 2.5 - PAS Class Diagram

5. Passport Automation System:-



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

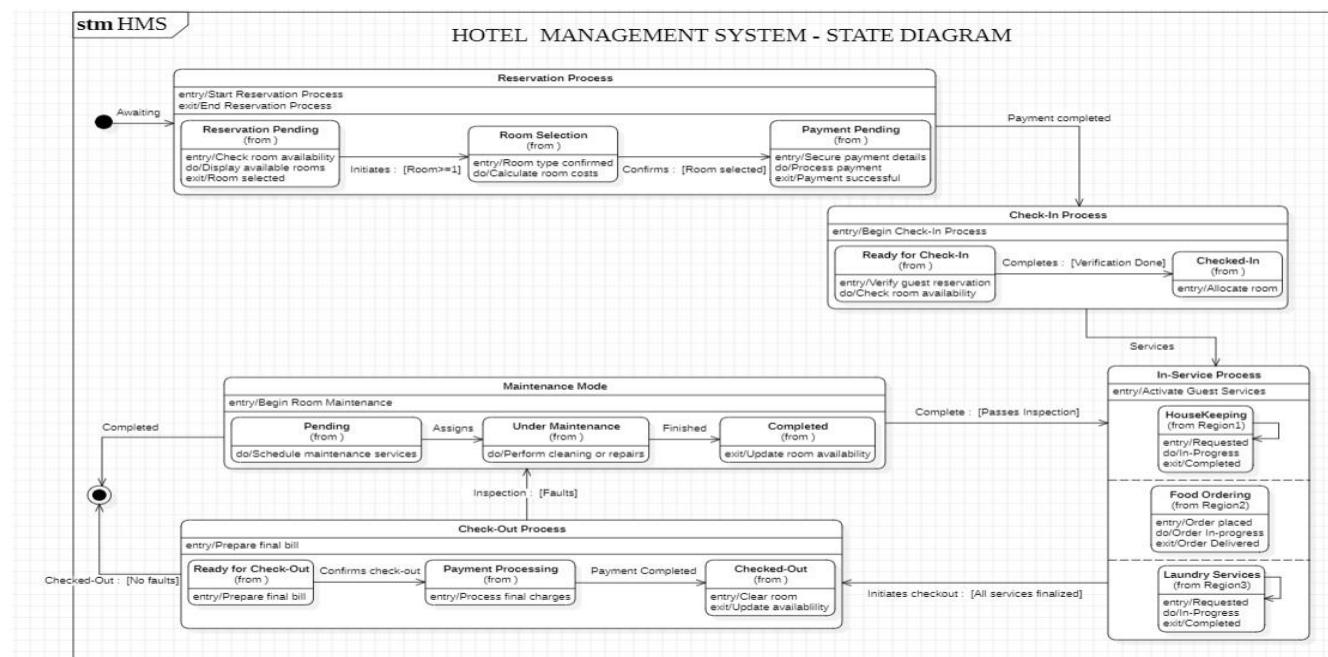


Figure 3.1 – HMS State Diagram

LAB-5

Manual

Date:

Page:

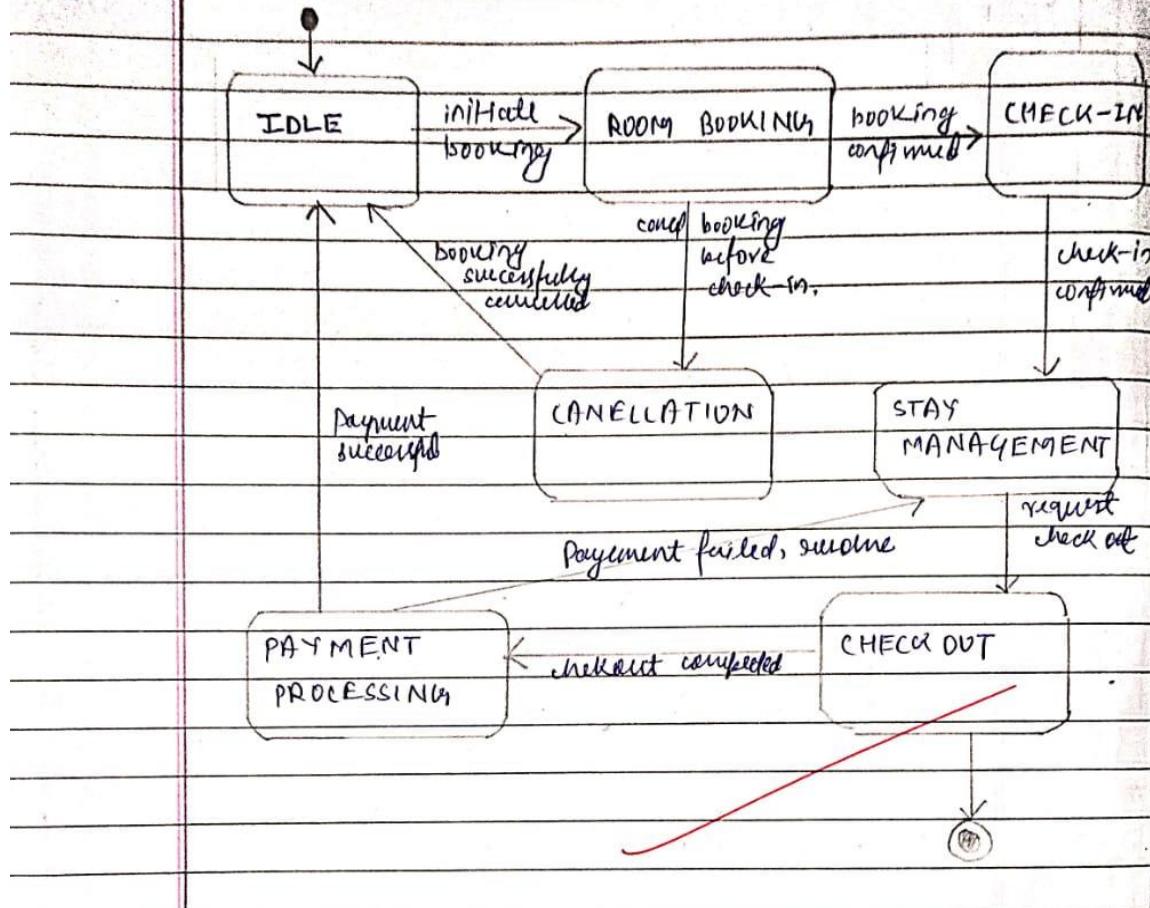
23.

Date: 11/11/2024:

④ State Diagrams:-

① HMS: (Hotel management system)

→ Hotel Management System



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

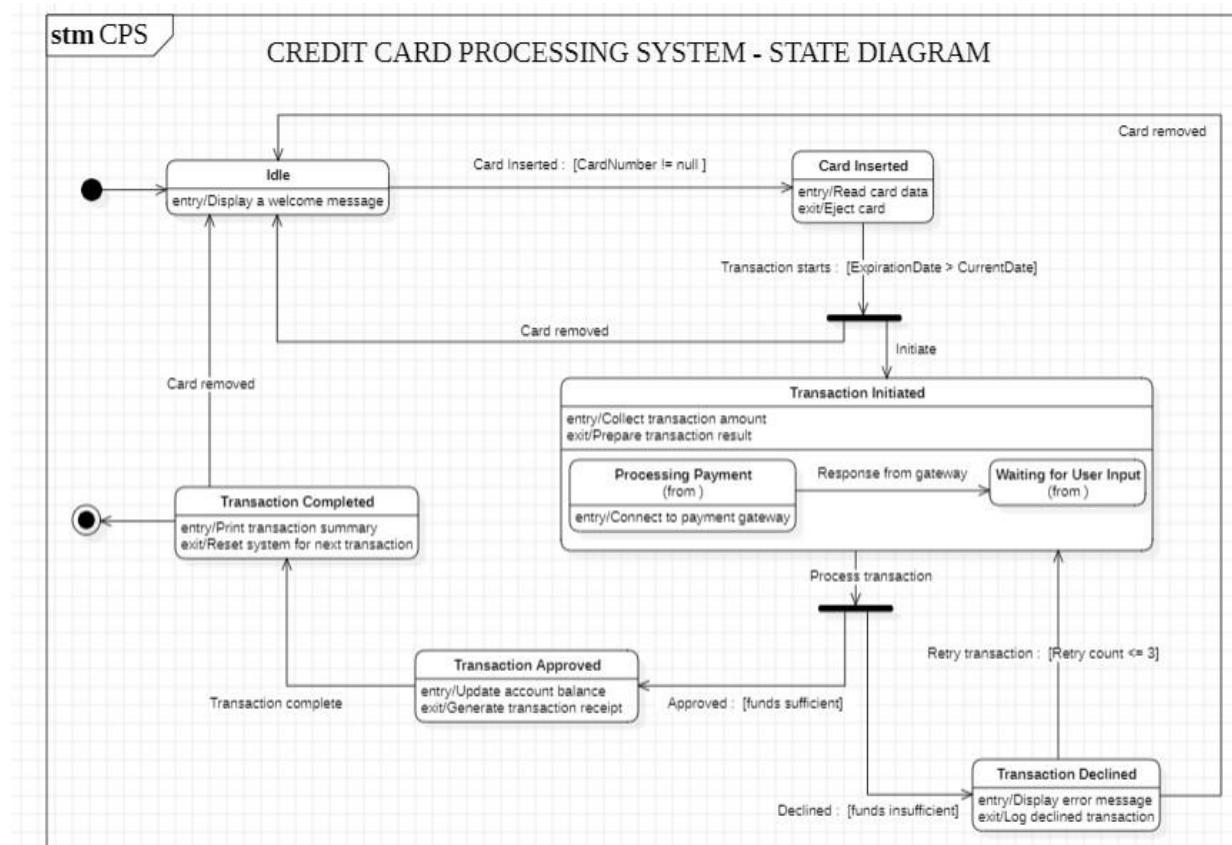
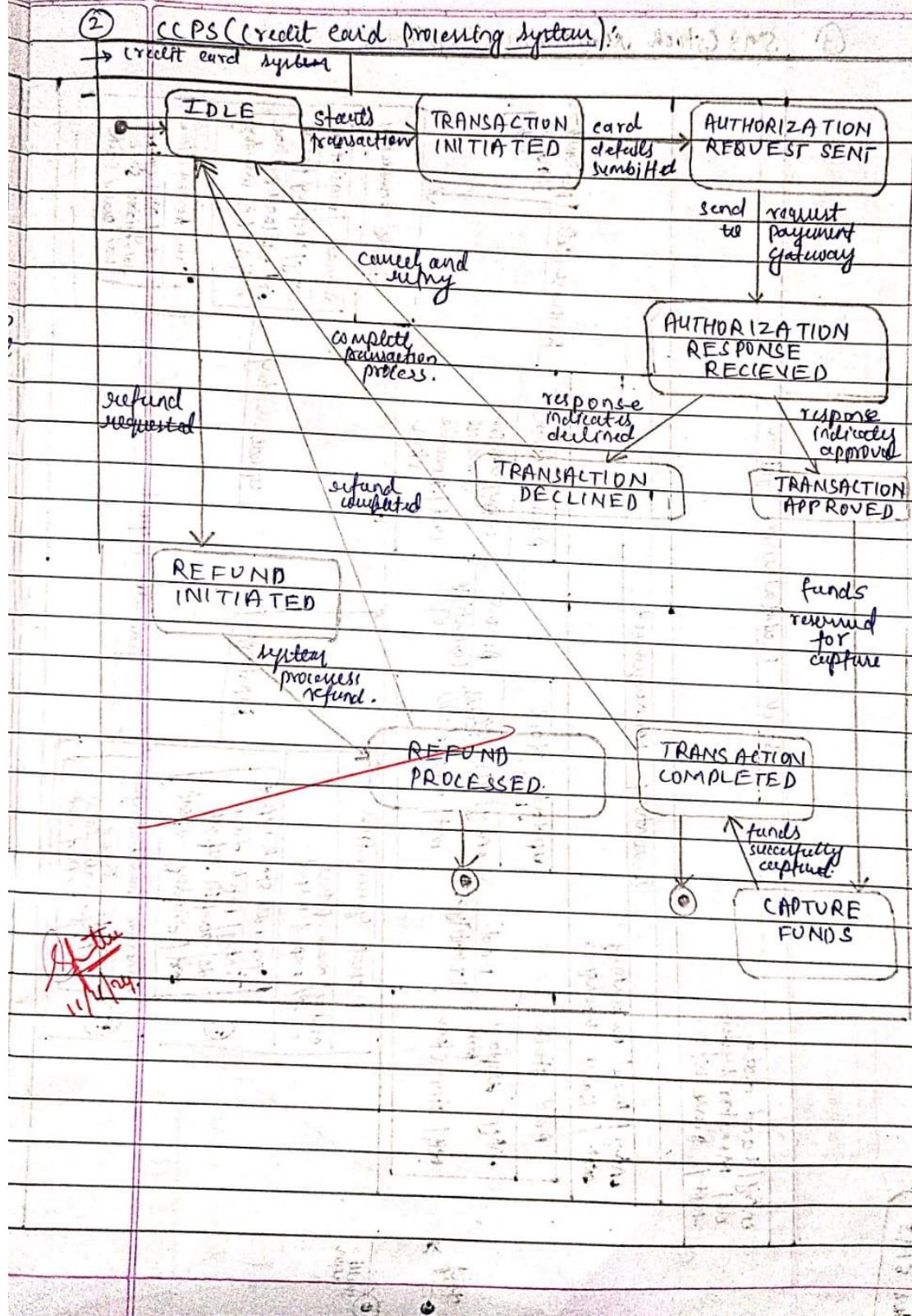


Figure 3.2 – CCPS State Diagram



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

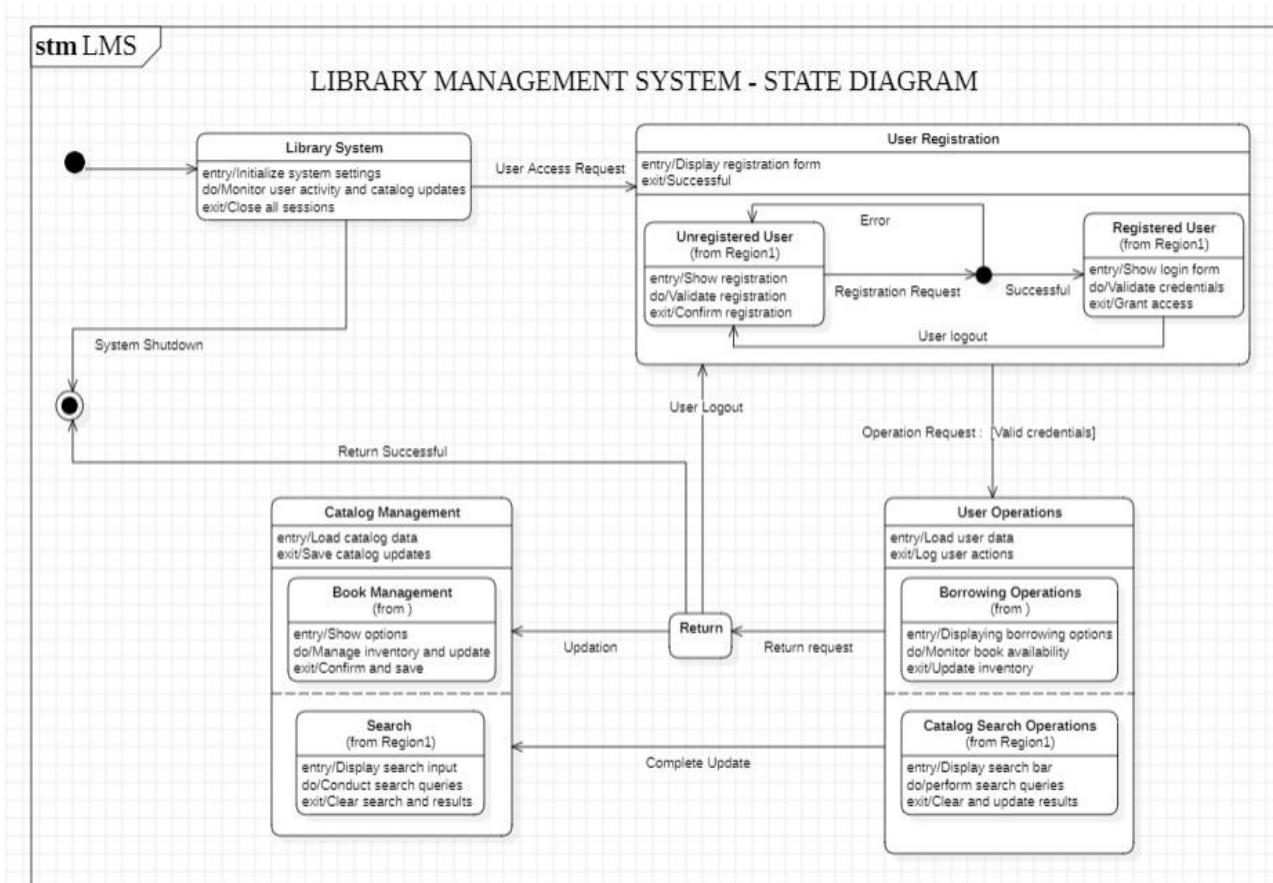


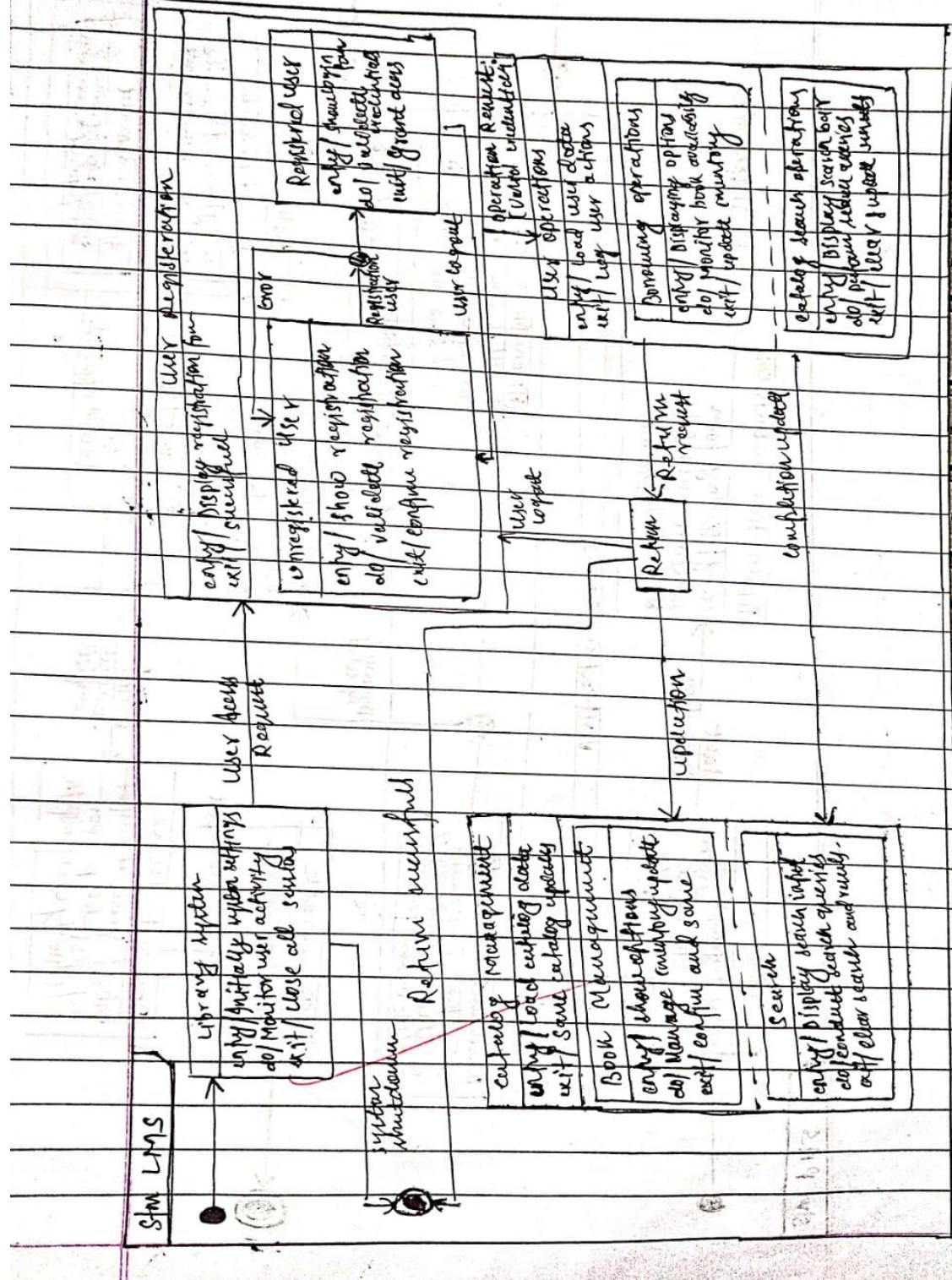
Figure 3.3 – LMS State Diagram

④ LMS (Library Management System)!

Mangal

date 2/15

26.



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

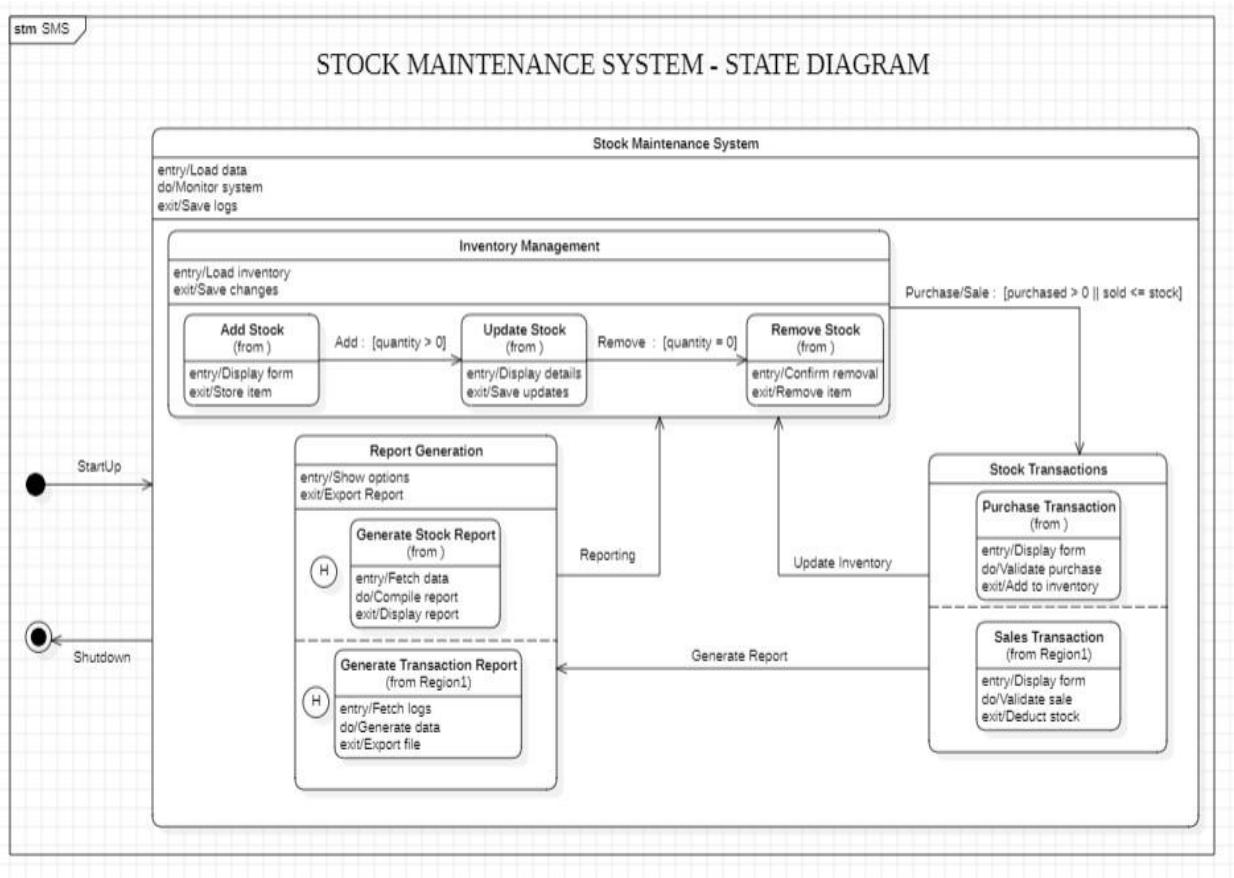
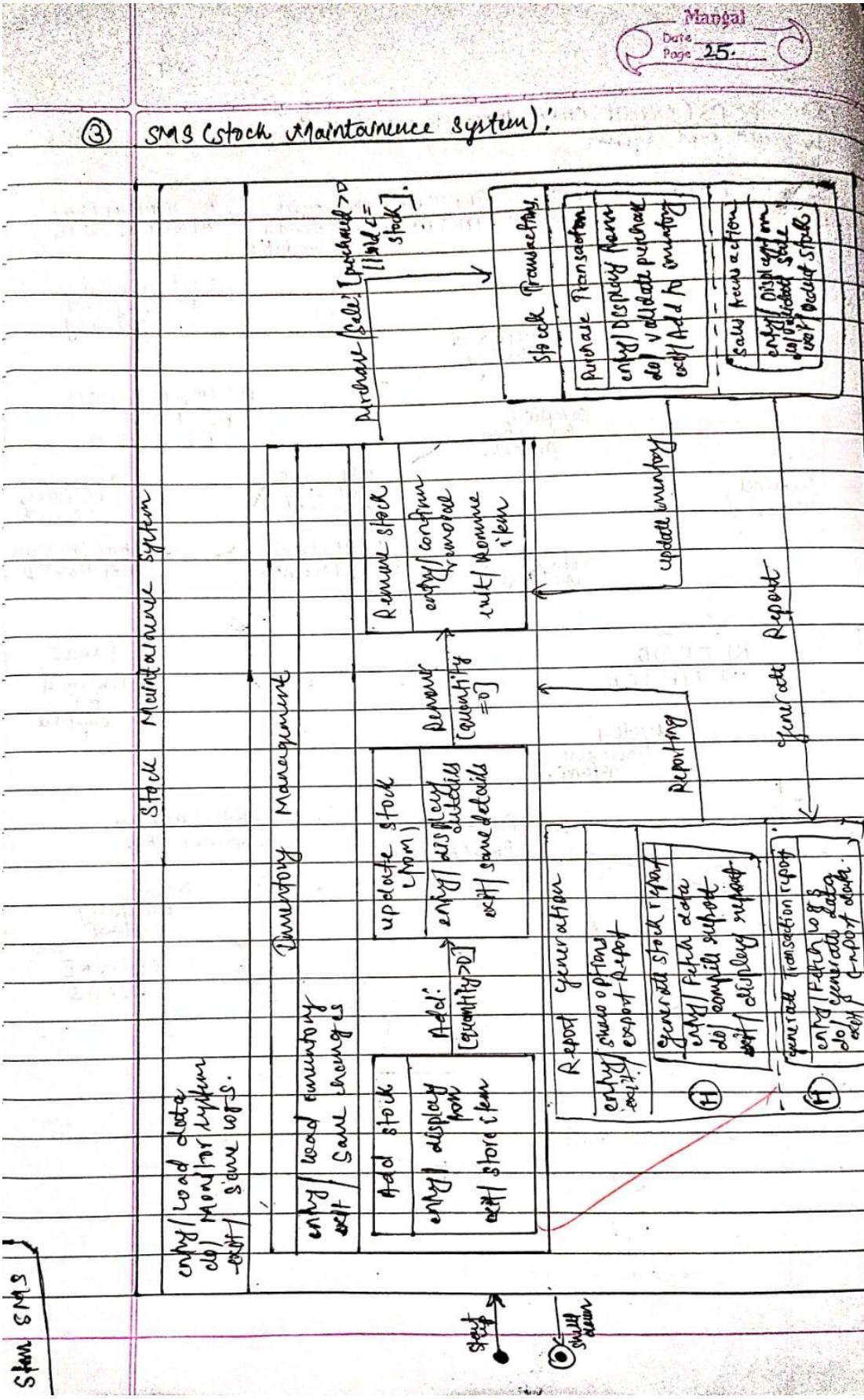


Figure 3.4 – SMS State Diagram



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

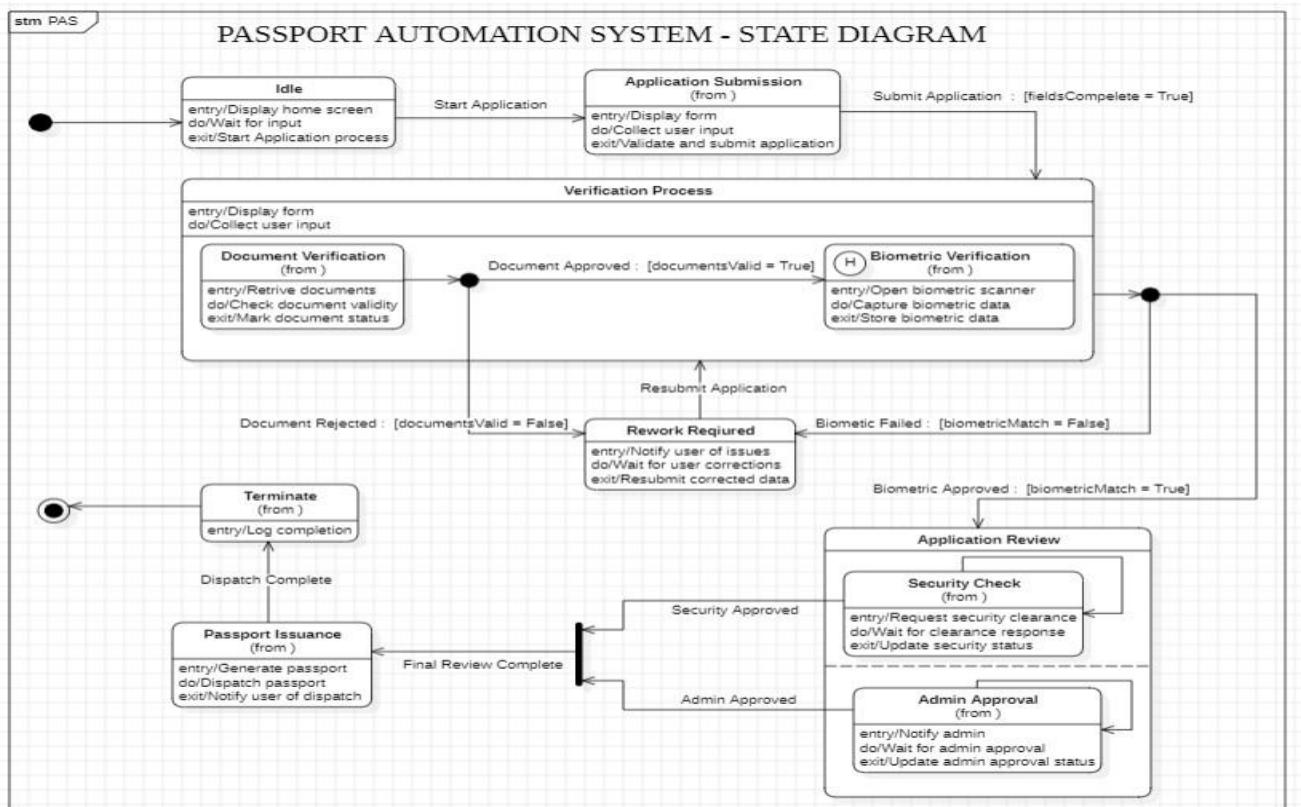
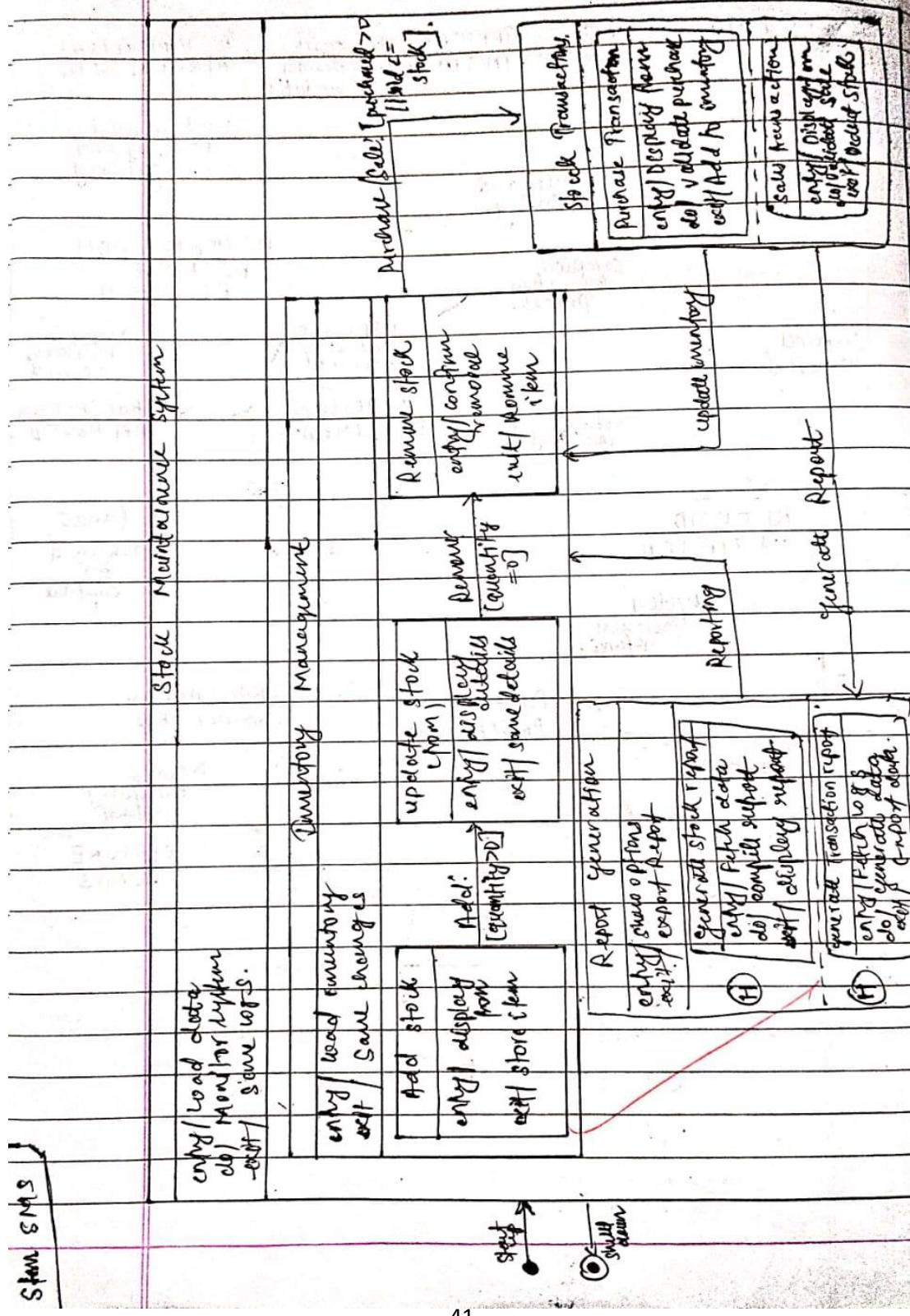


Figure 3.5 – PAS State Diagram

③ SMS (Stock Maintenance System):



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.

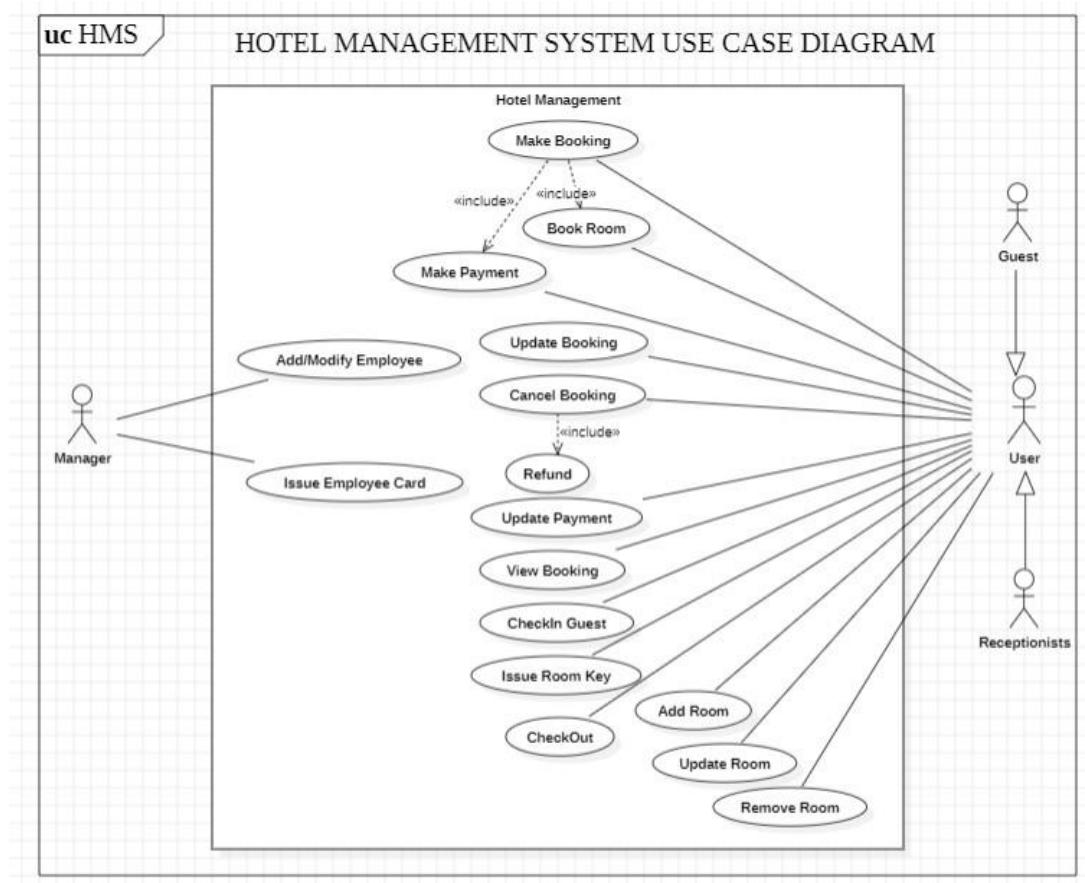
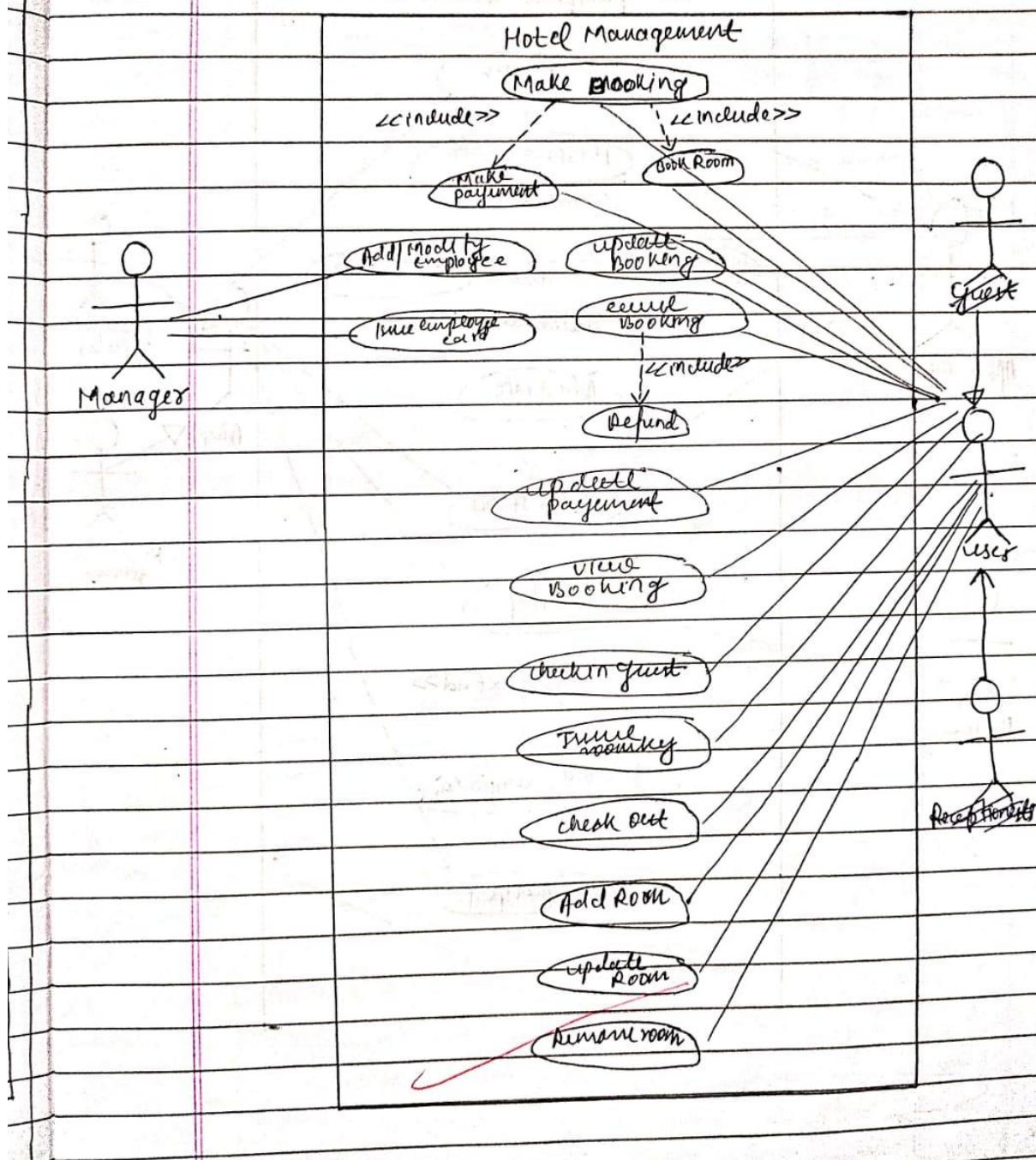


Figure 4.1 – HMS Use Case Diagram

(1). Use case Diagrams!

(1) HMS : (Hotel Management System).



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.

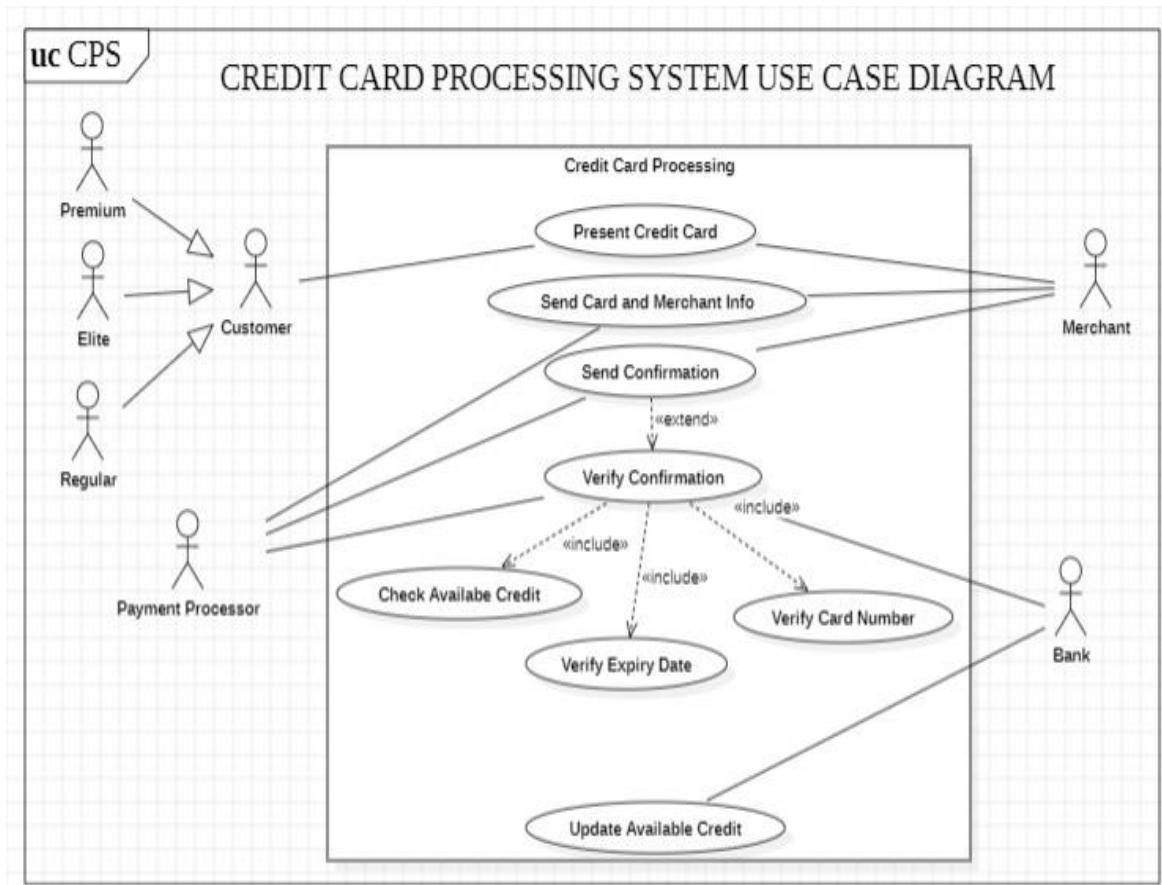
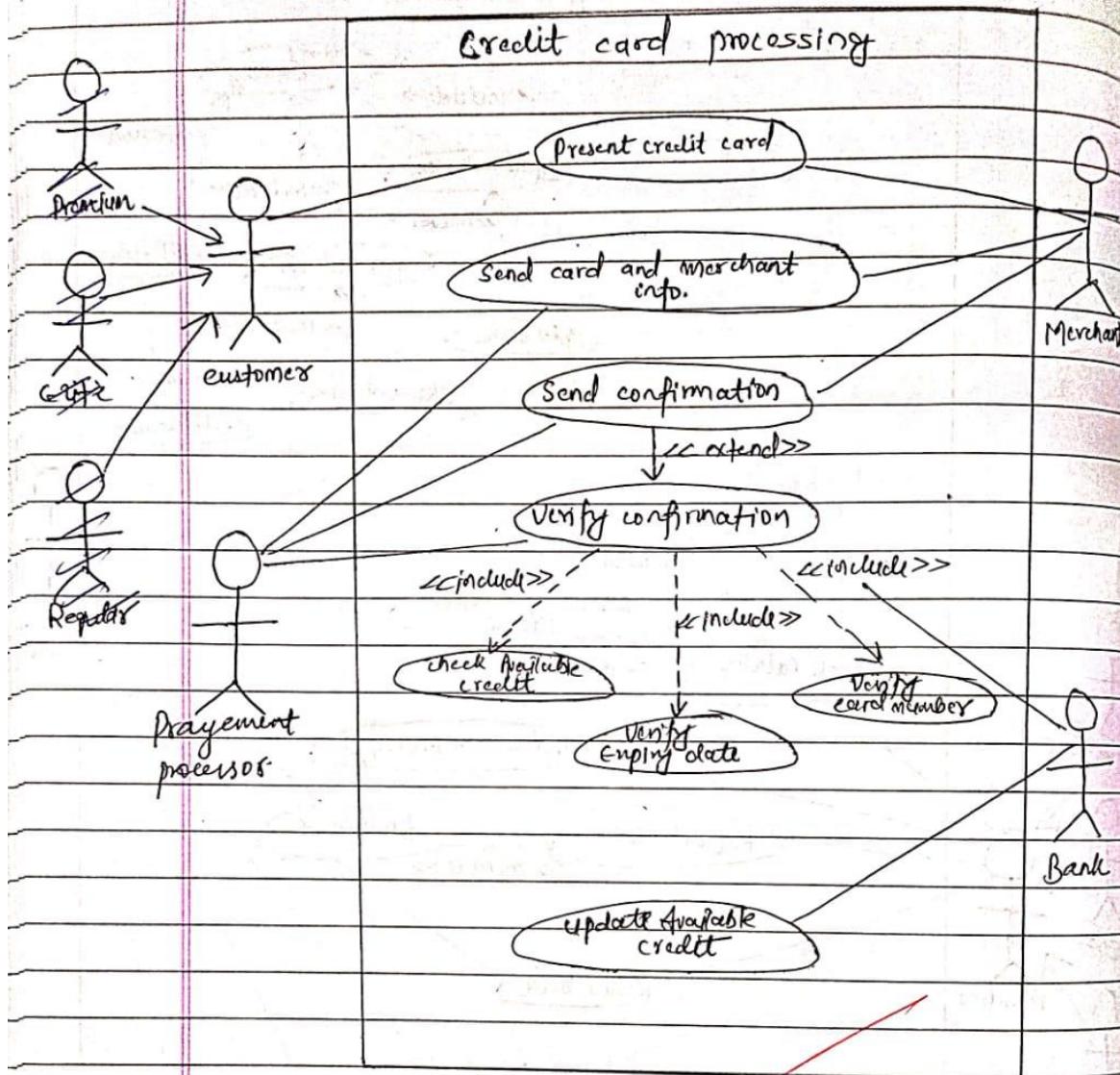


Figure 4.2 – CCPS Use Case Diagram

(4) CCPS (Credit card processing system) ! -



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.

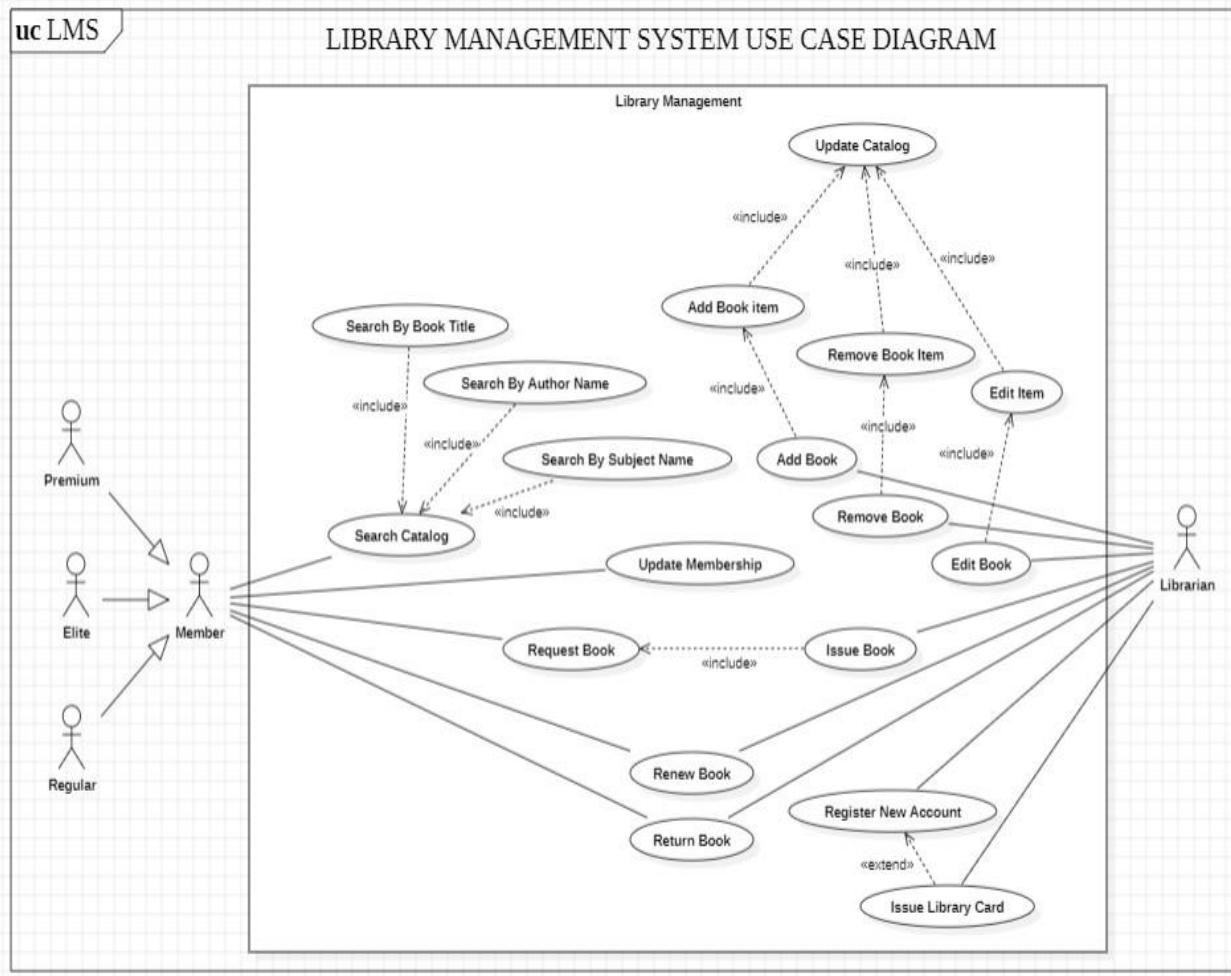
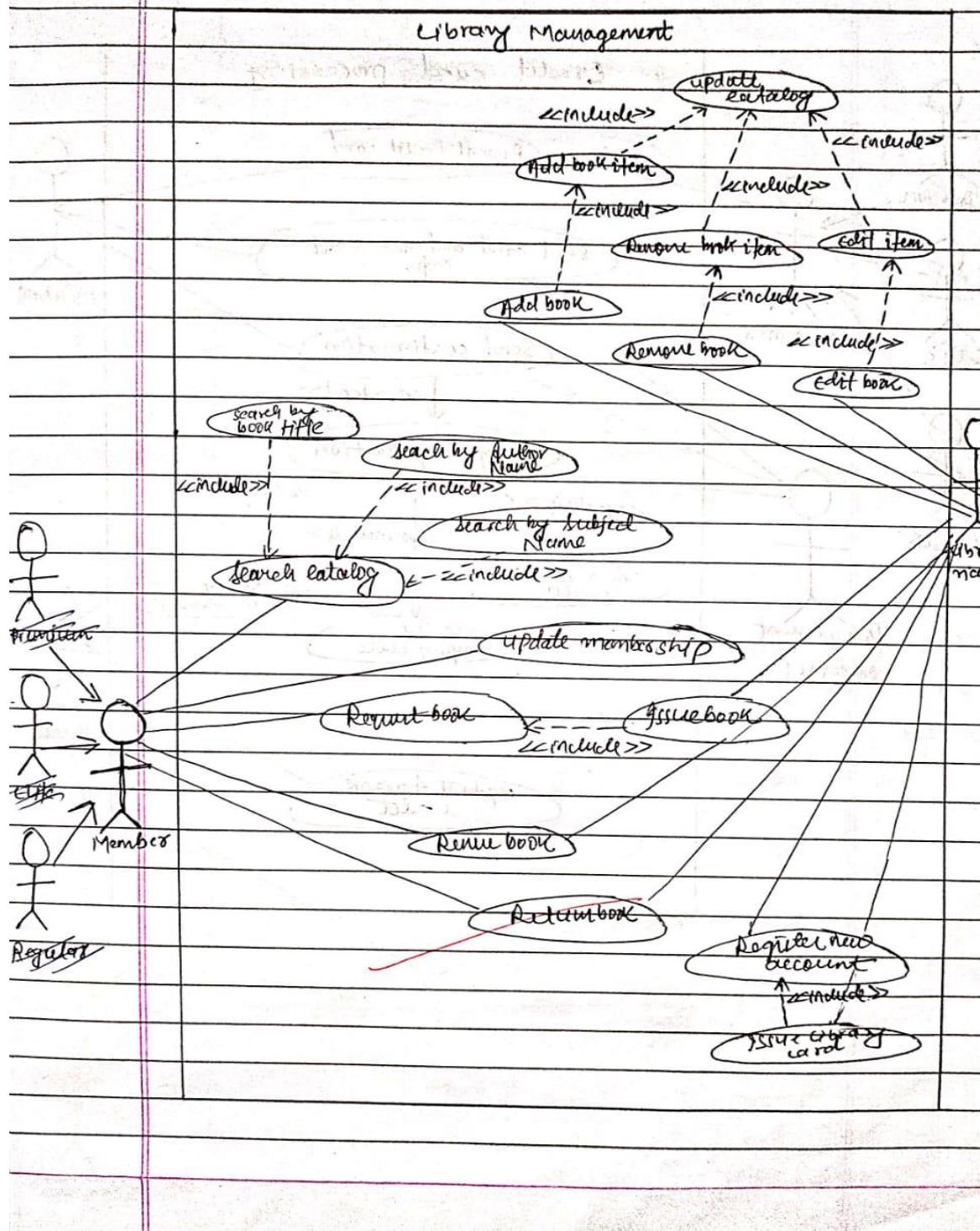


Figure 4.3 – LMS Use Case Diagram

(3) LMS (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.

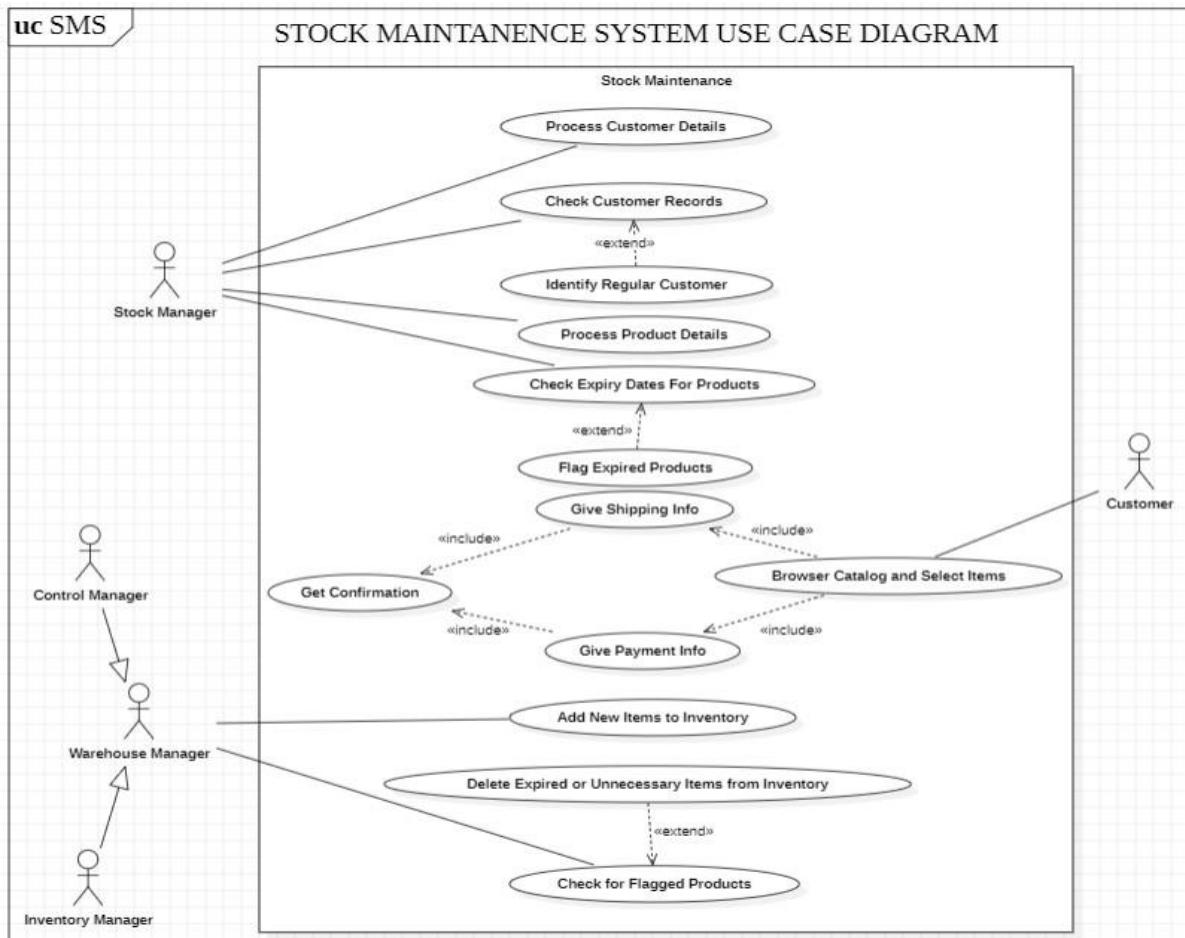
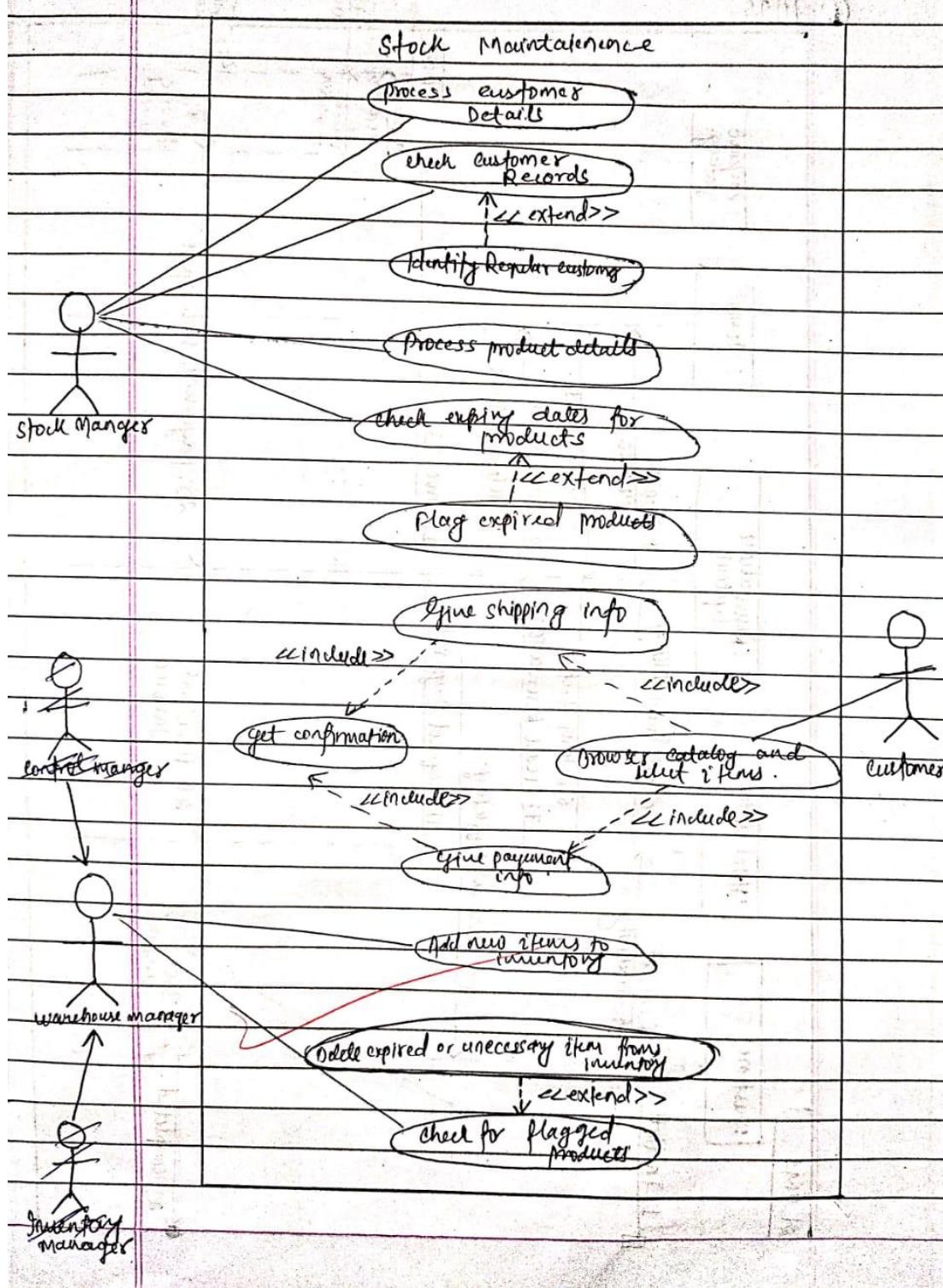


Figure 4.4 – SMS Use Case Diagram

(5) SMS (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.

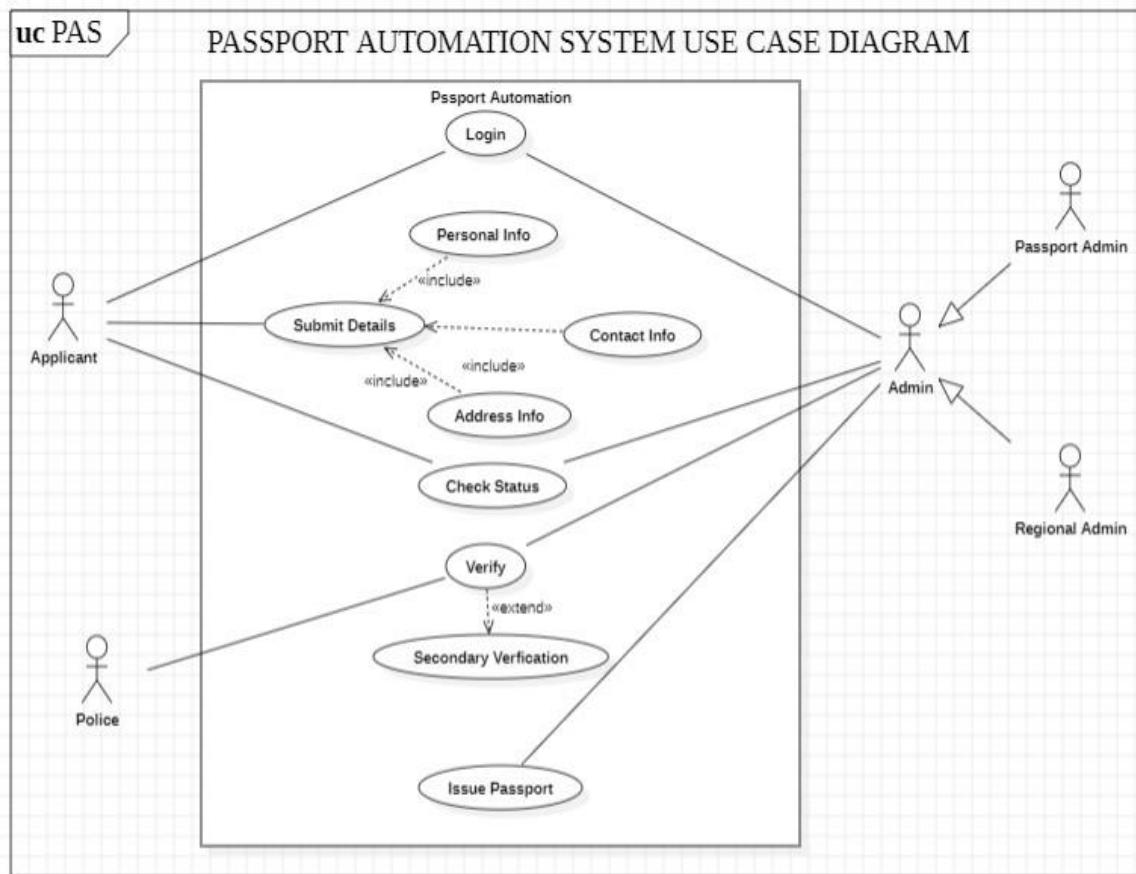
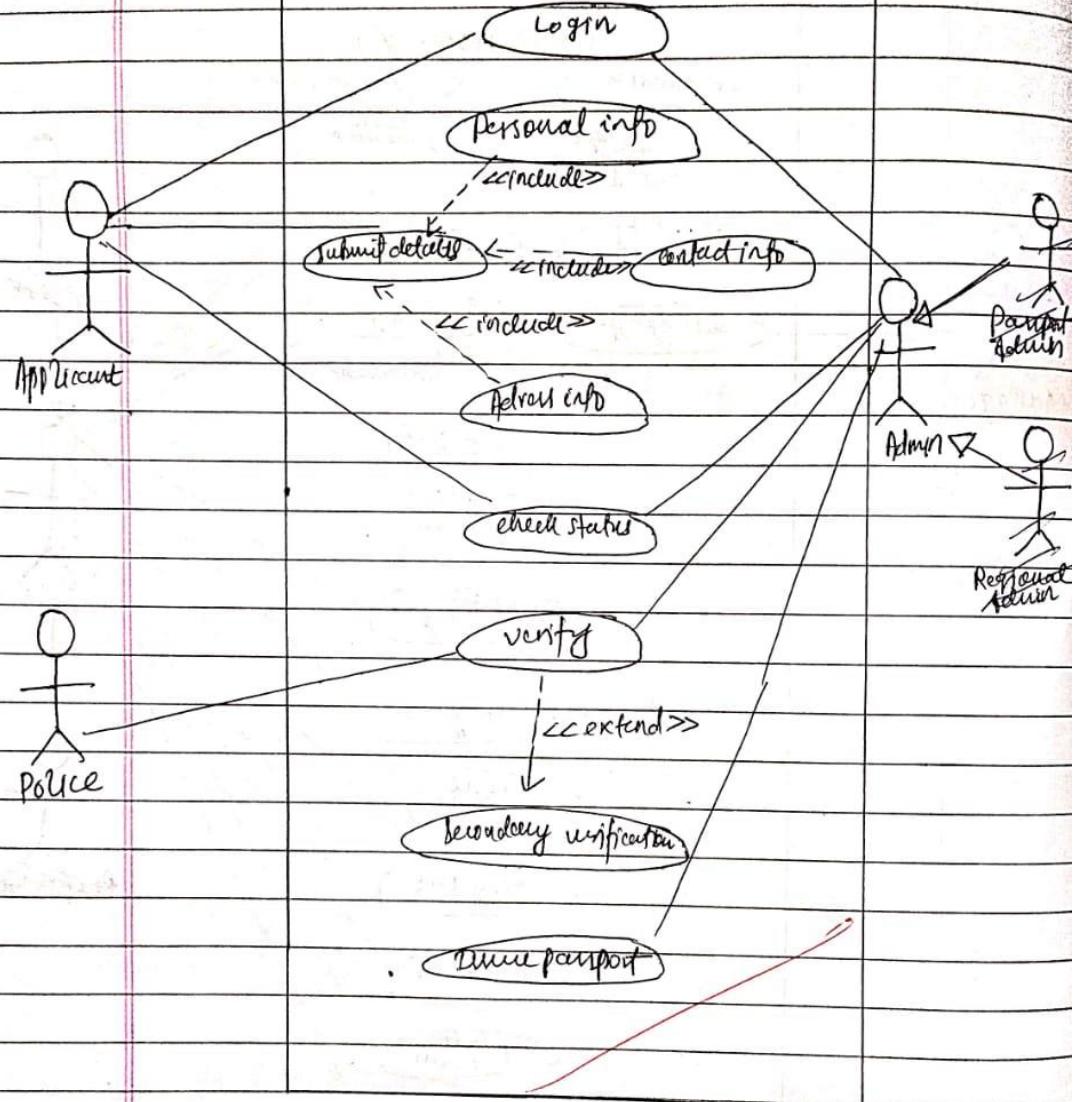


Figure 4.5 – PAS Use Case Diagram

(2)

PAS (Passport Automation System).

Passport Automation



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.

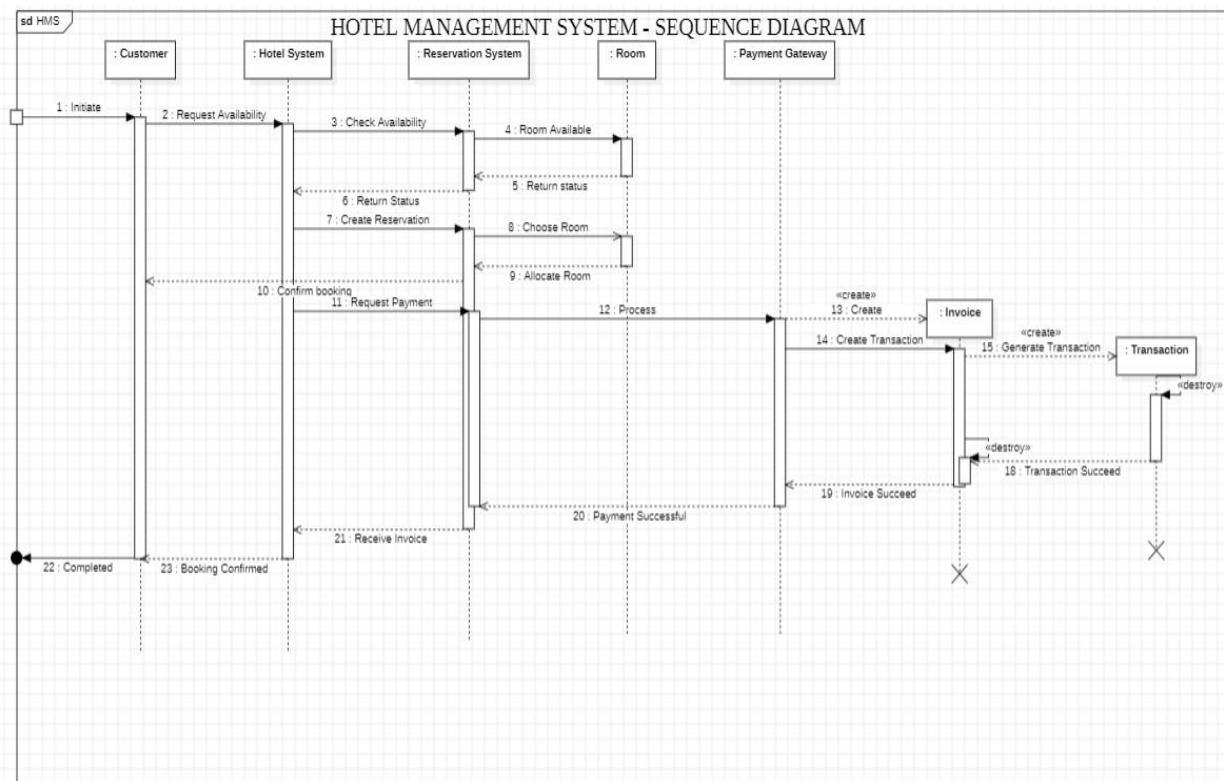
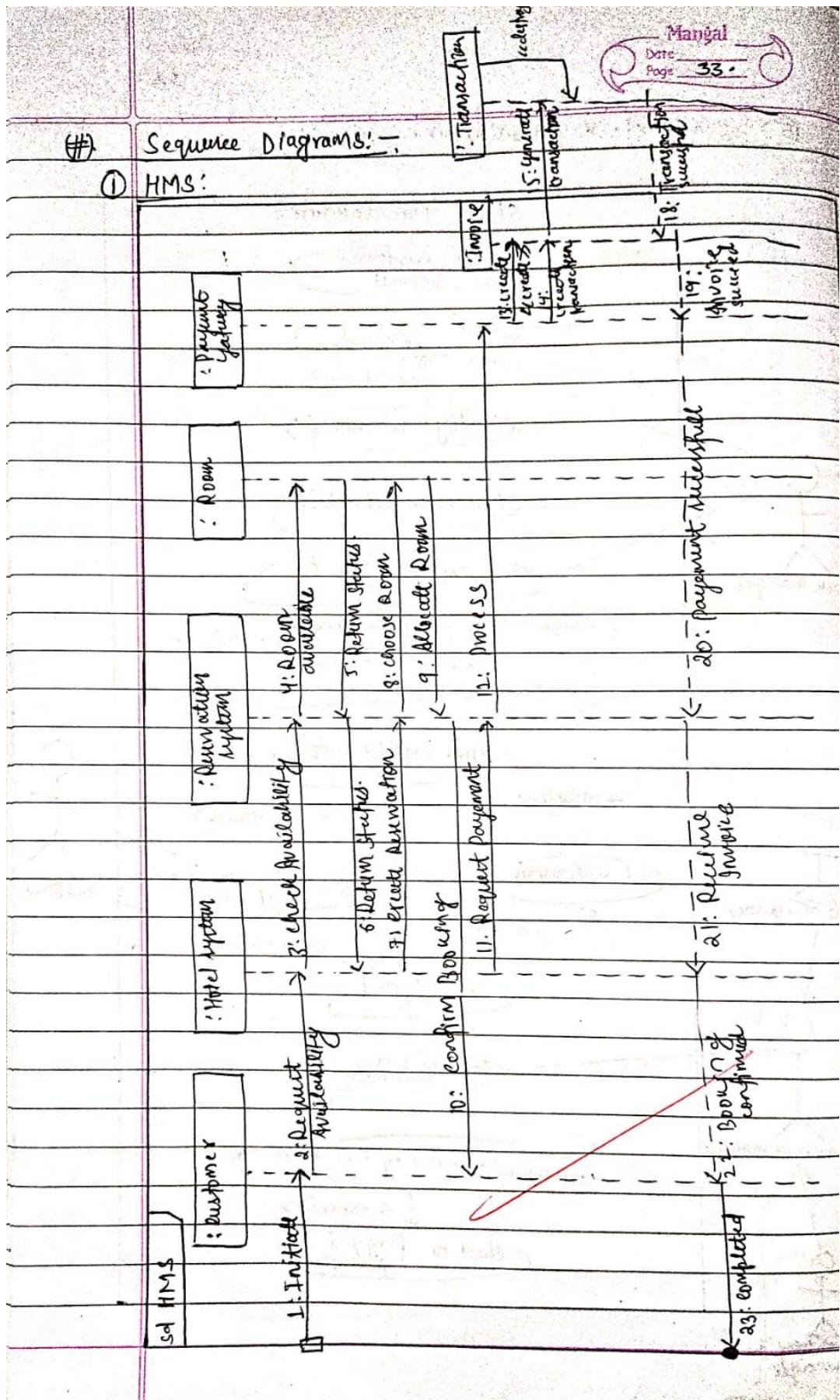


Figure 5.1 – HMS Sequence Diagram



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.

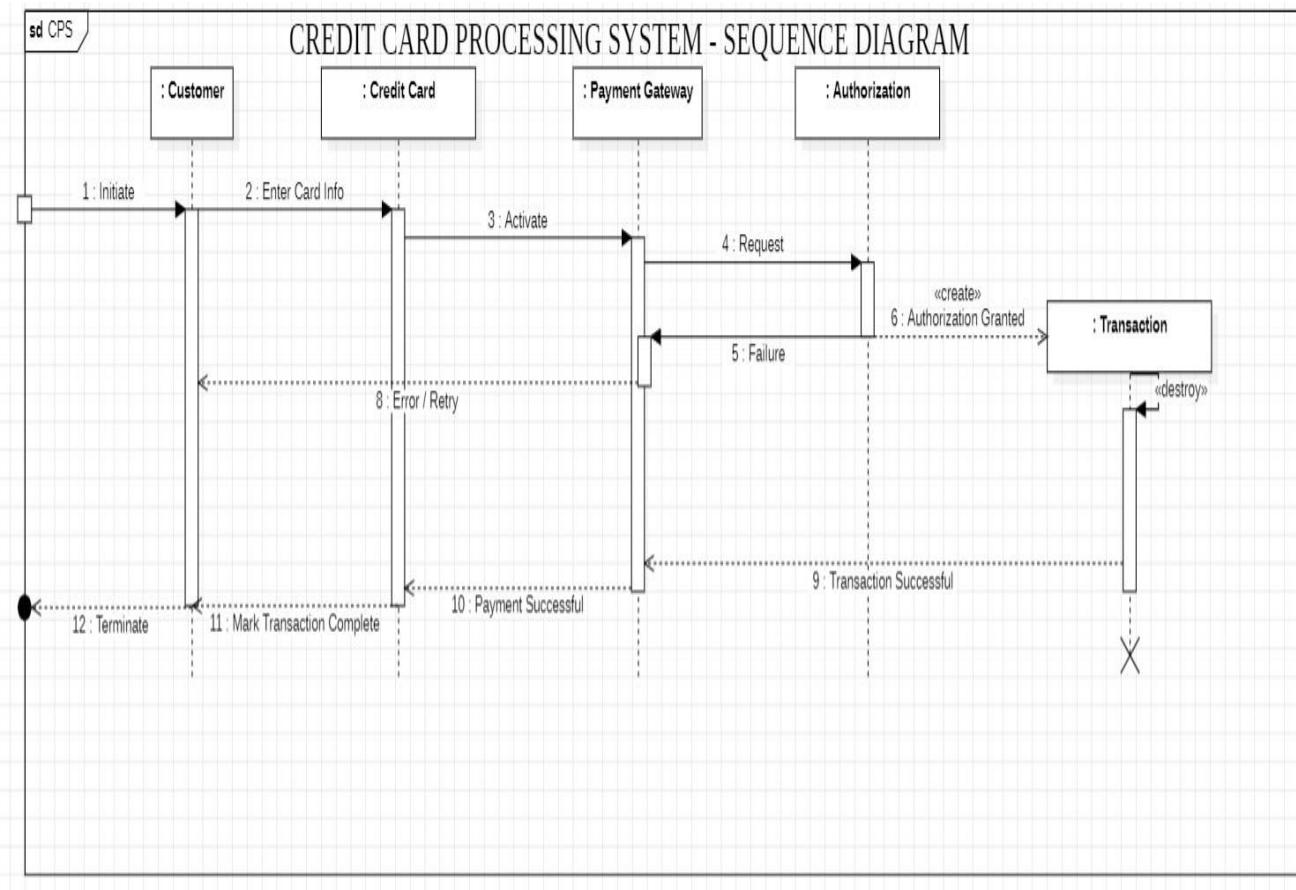
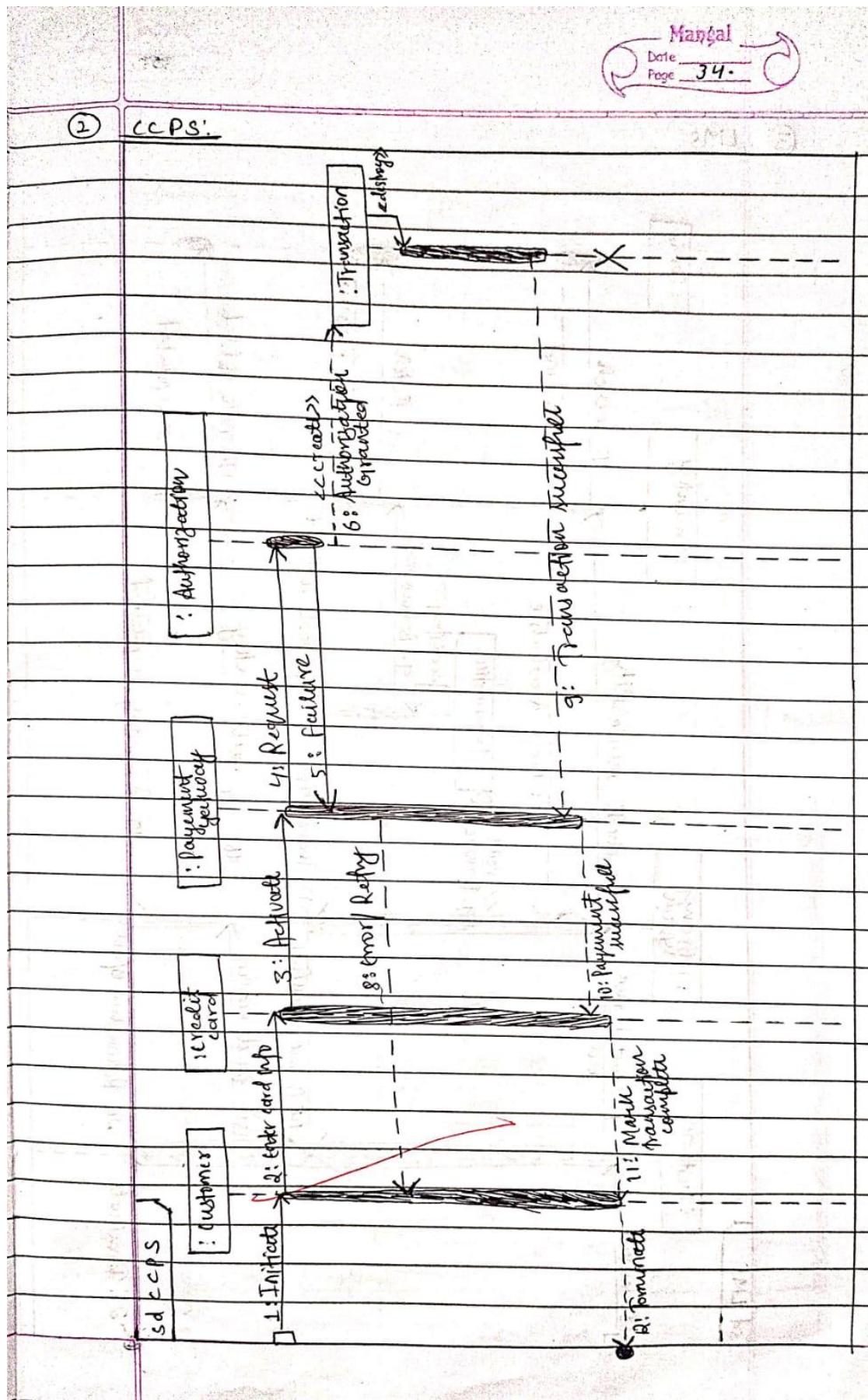


Figure 5.2 – CCPS Sequence Diagram



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.

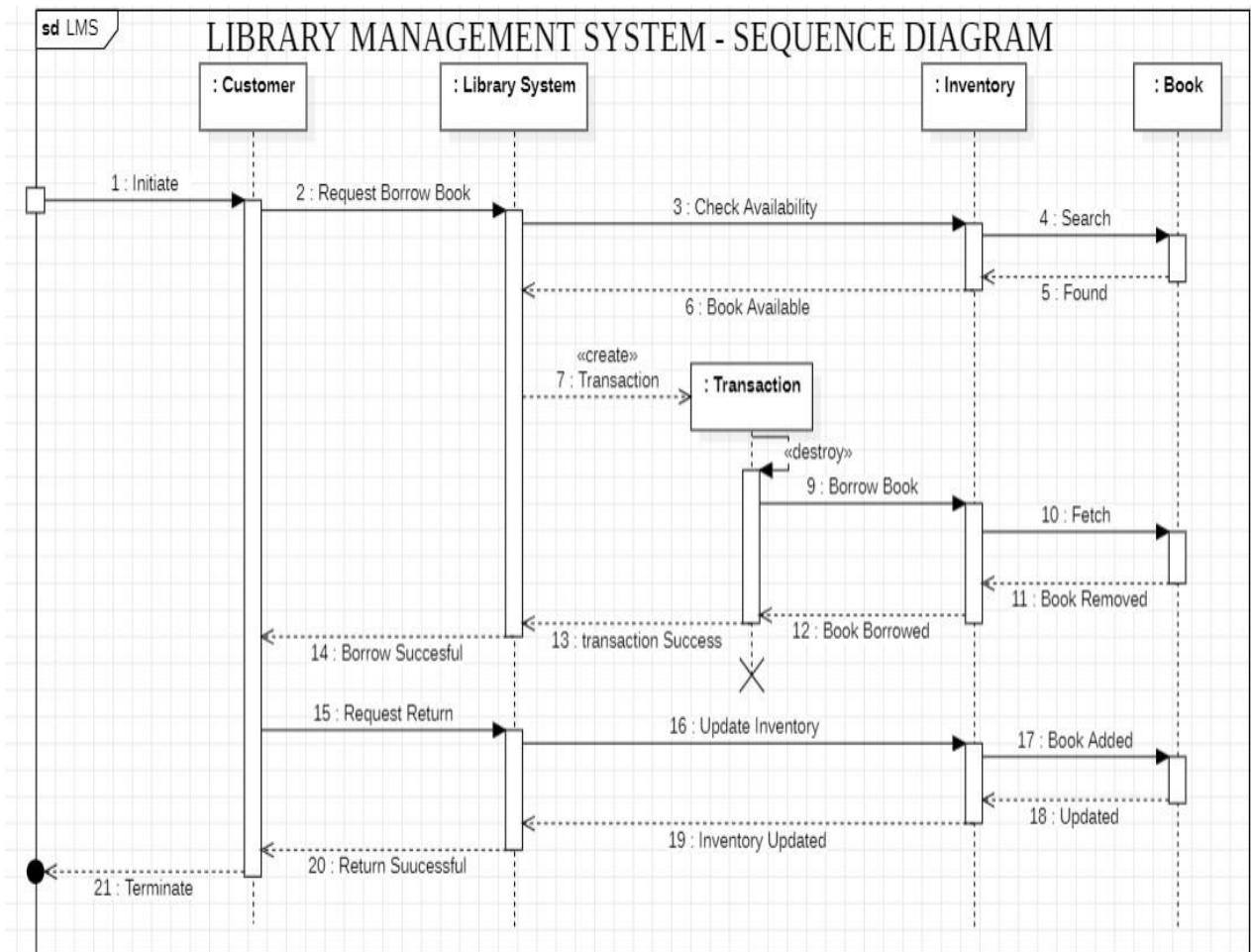
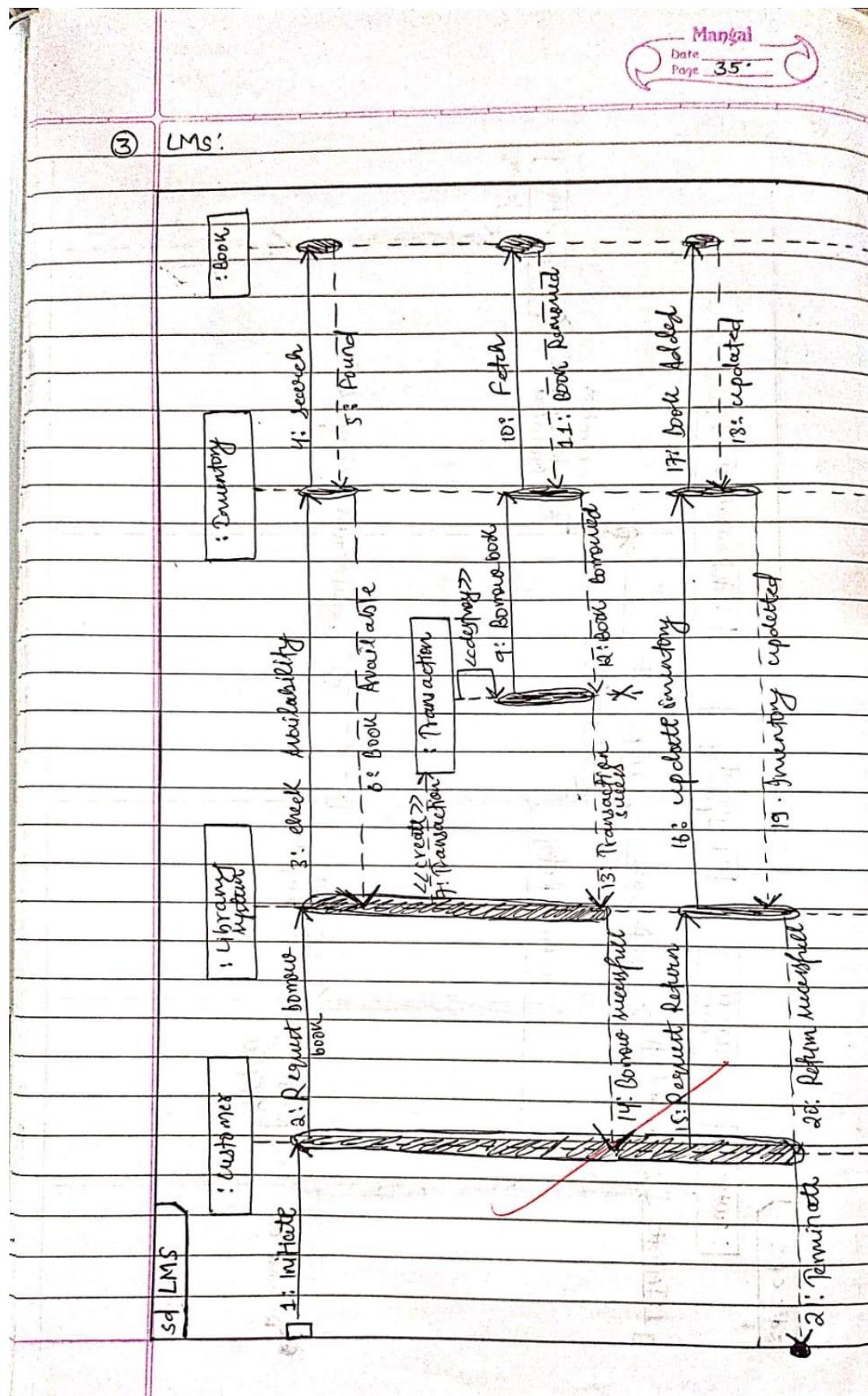


Figure 5.3 – LMS Sequence Diagram



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.

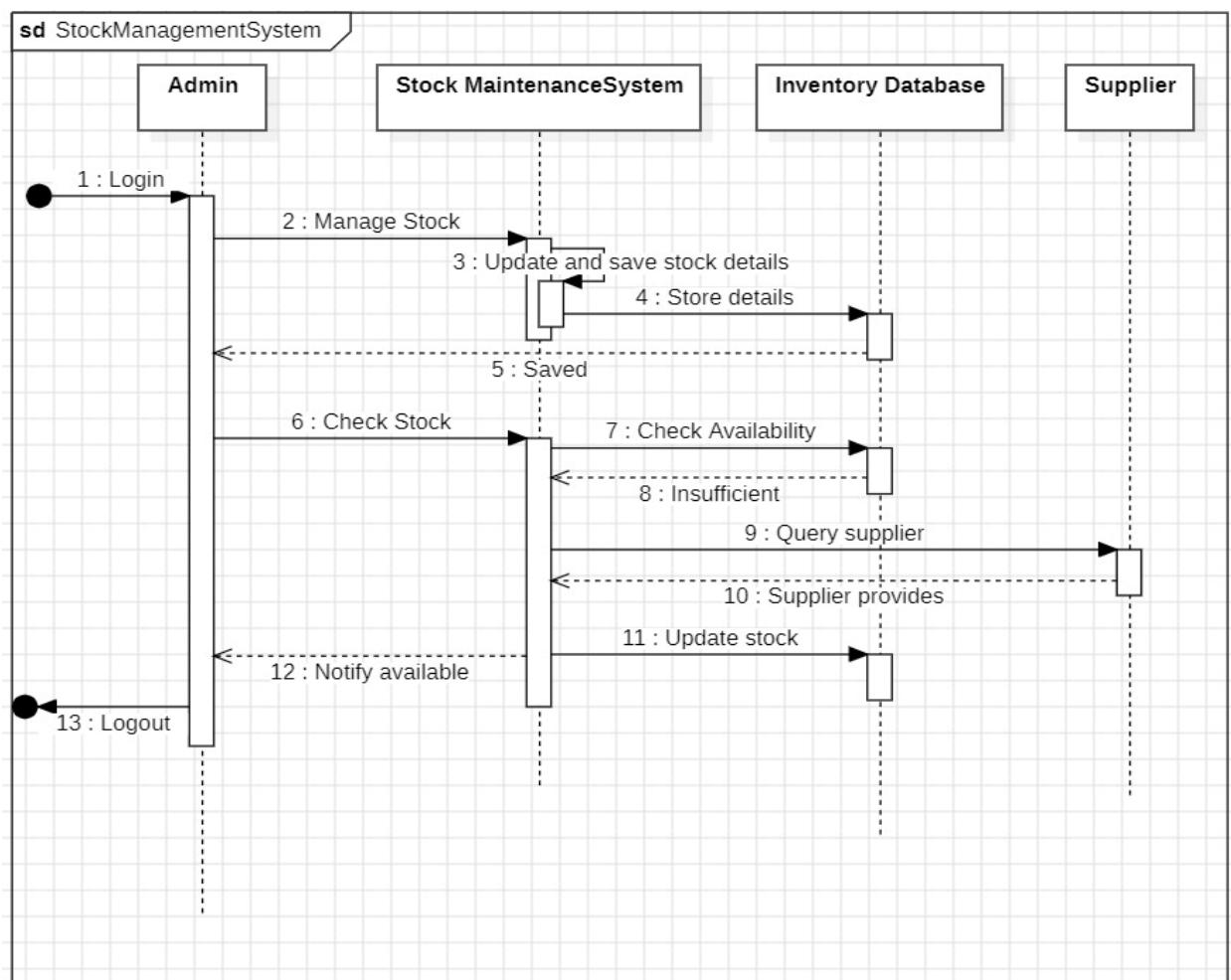
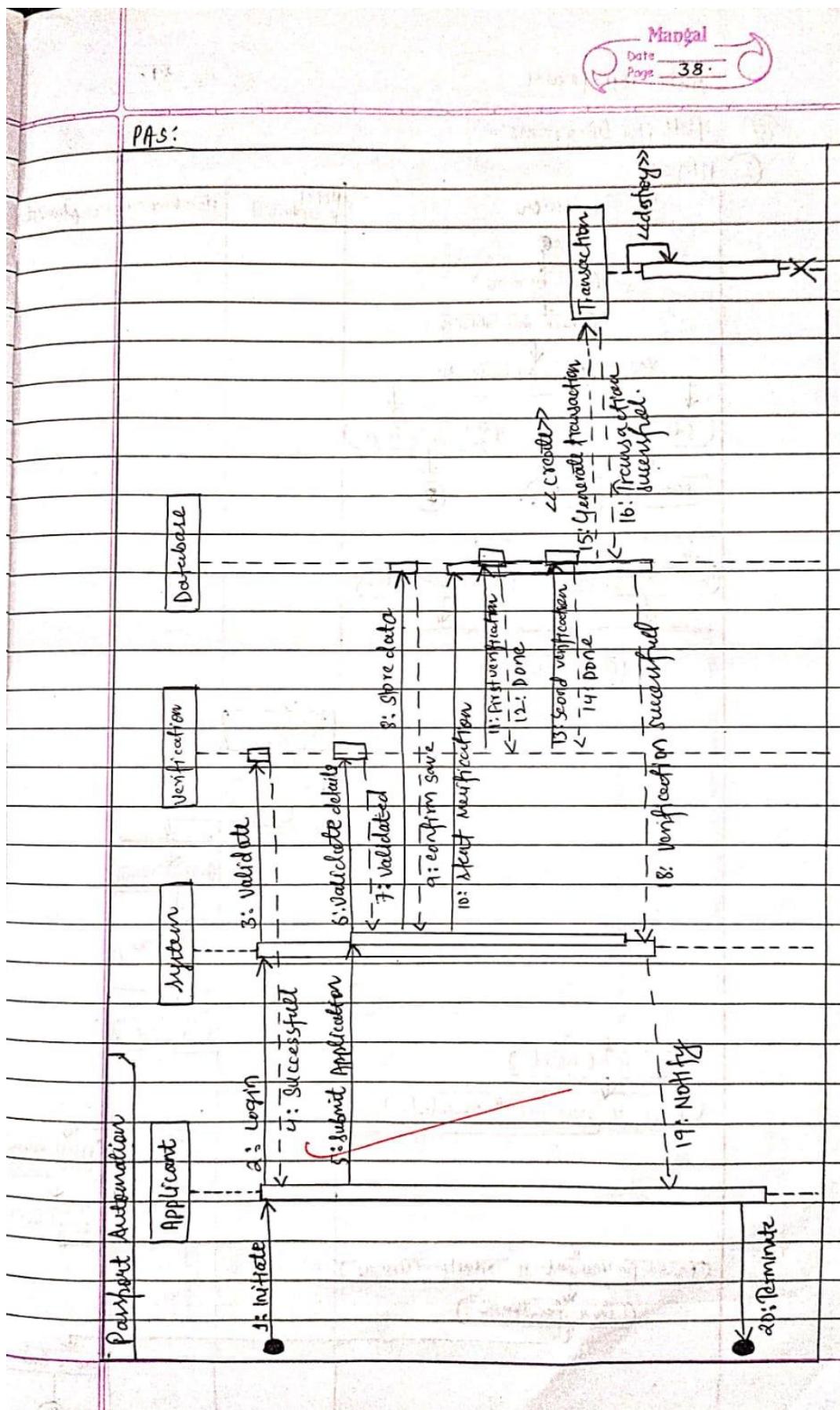


Figure 5.4 – SMS Sequence Diagram



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.

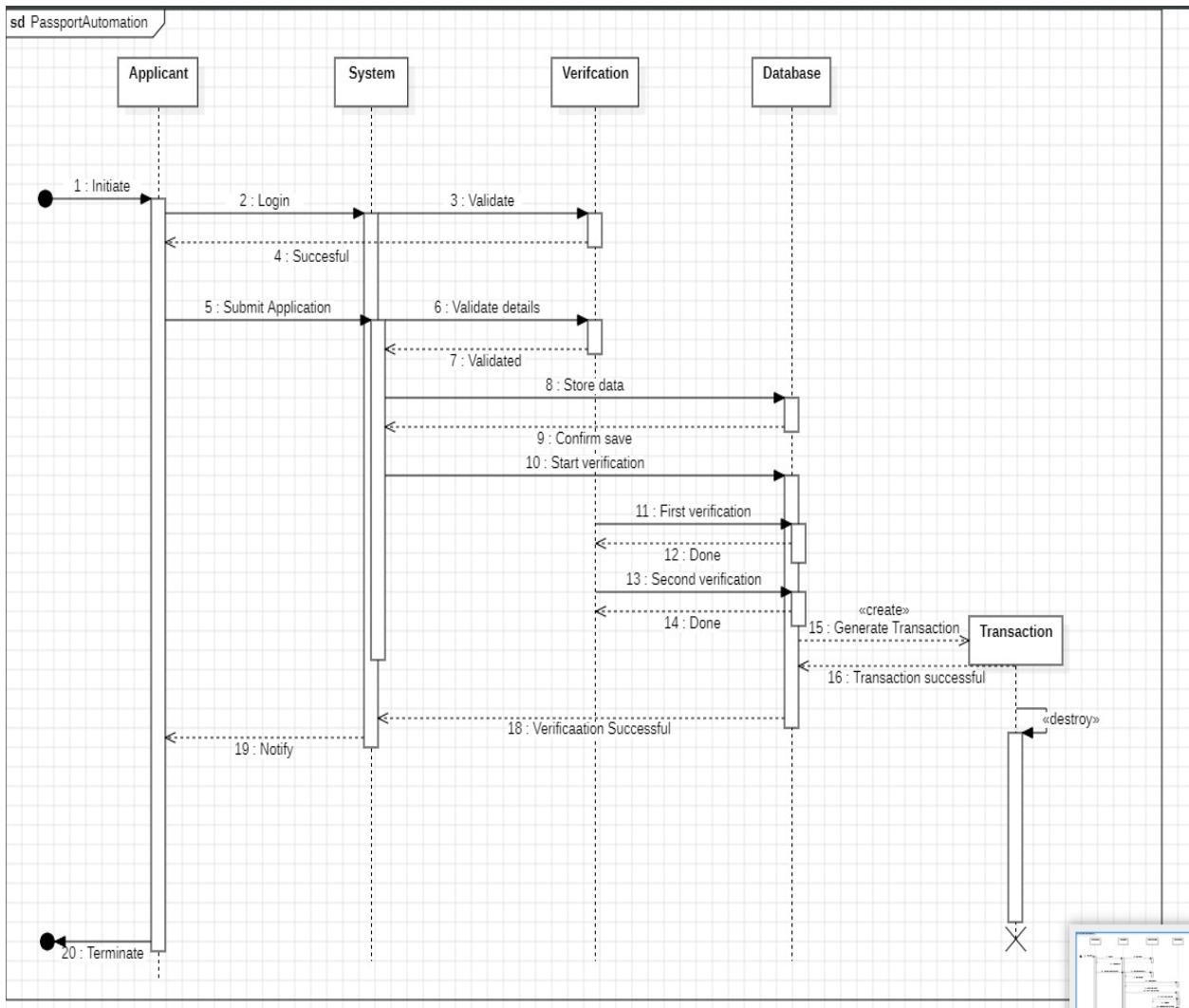
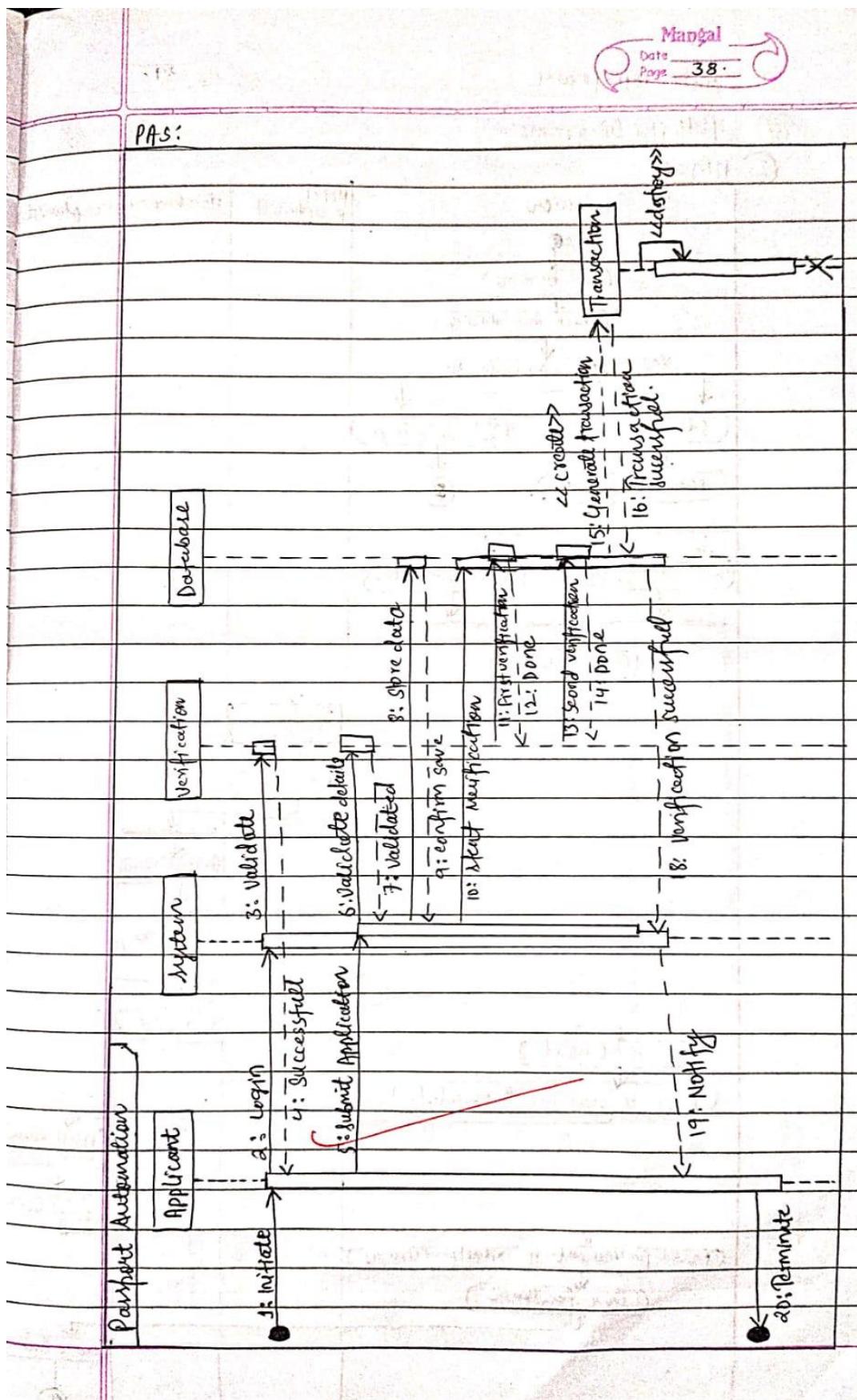


Figure 5.5 – PAS Sequence Diagram



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.

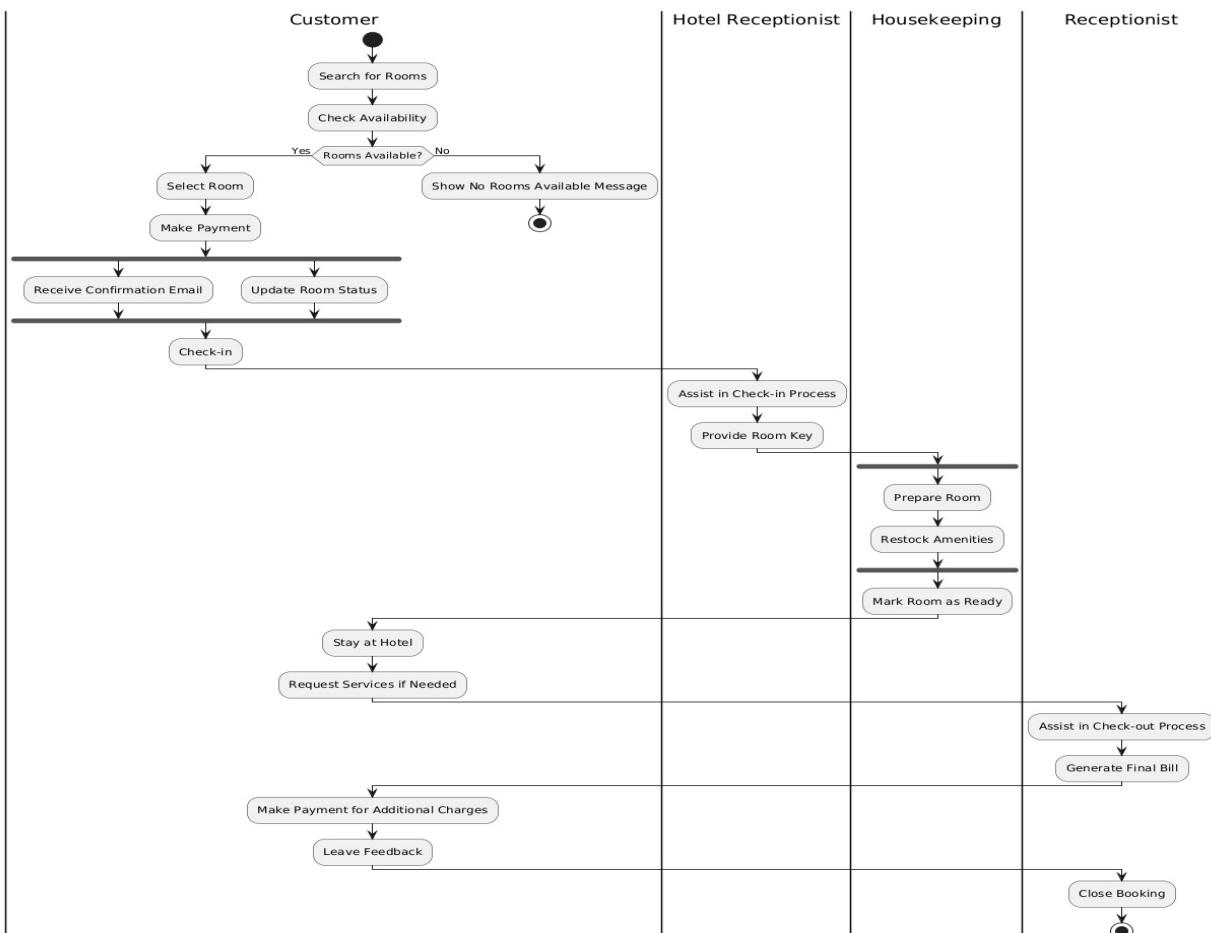
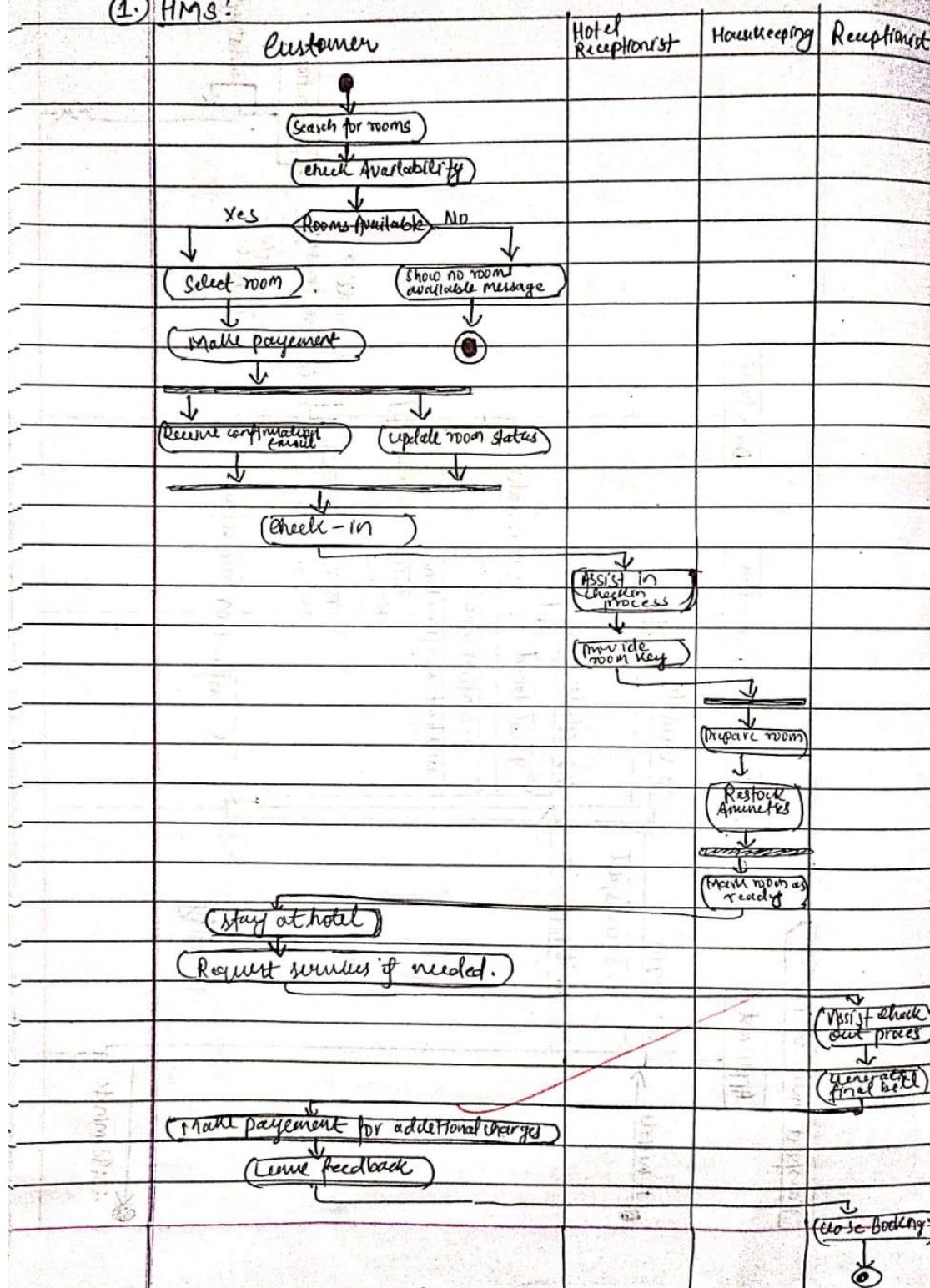


Figure 6.1 – HMS Activity Diagram

Date: 16/12/2024.

(II). Activity Diagrams:-

1. HMS:



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.

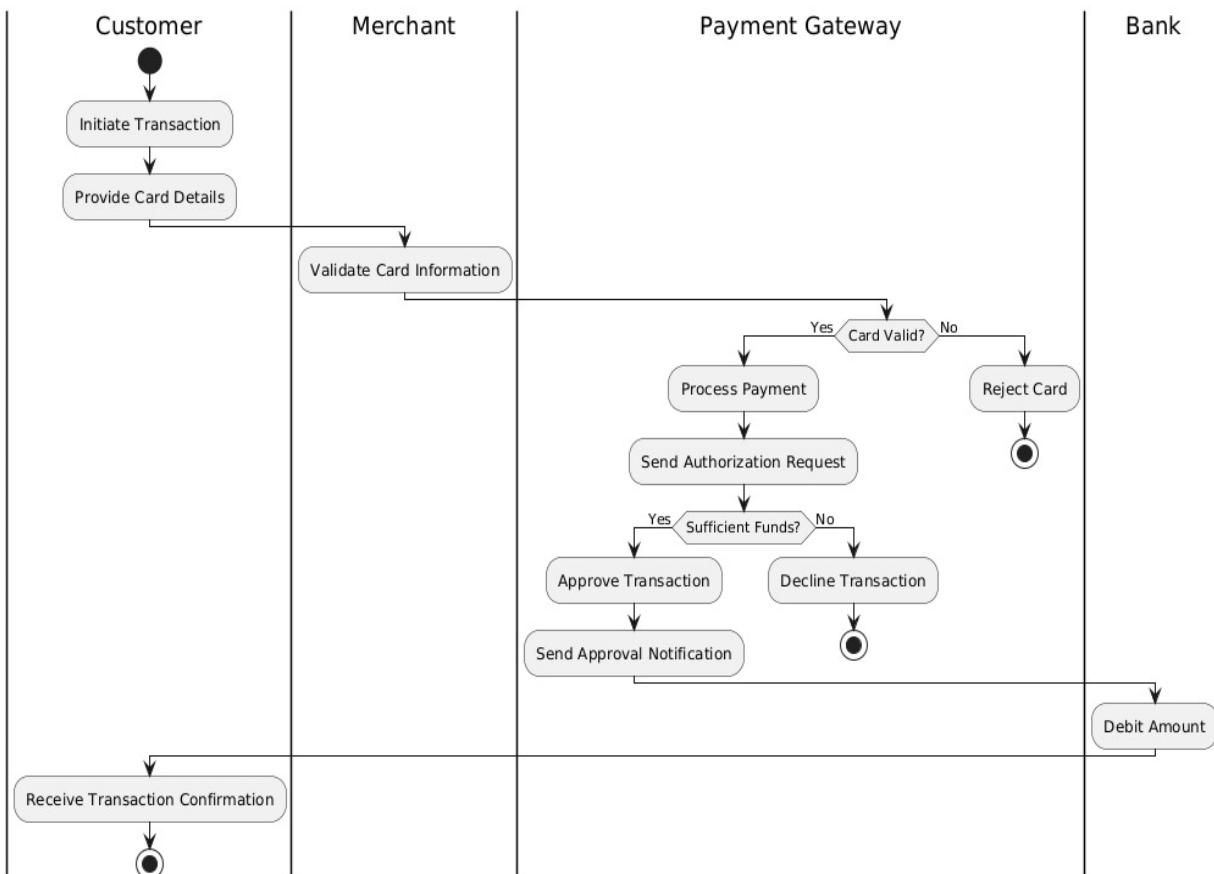
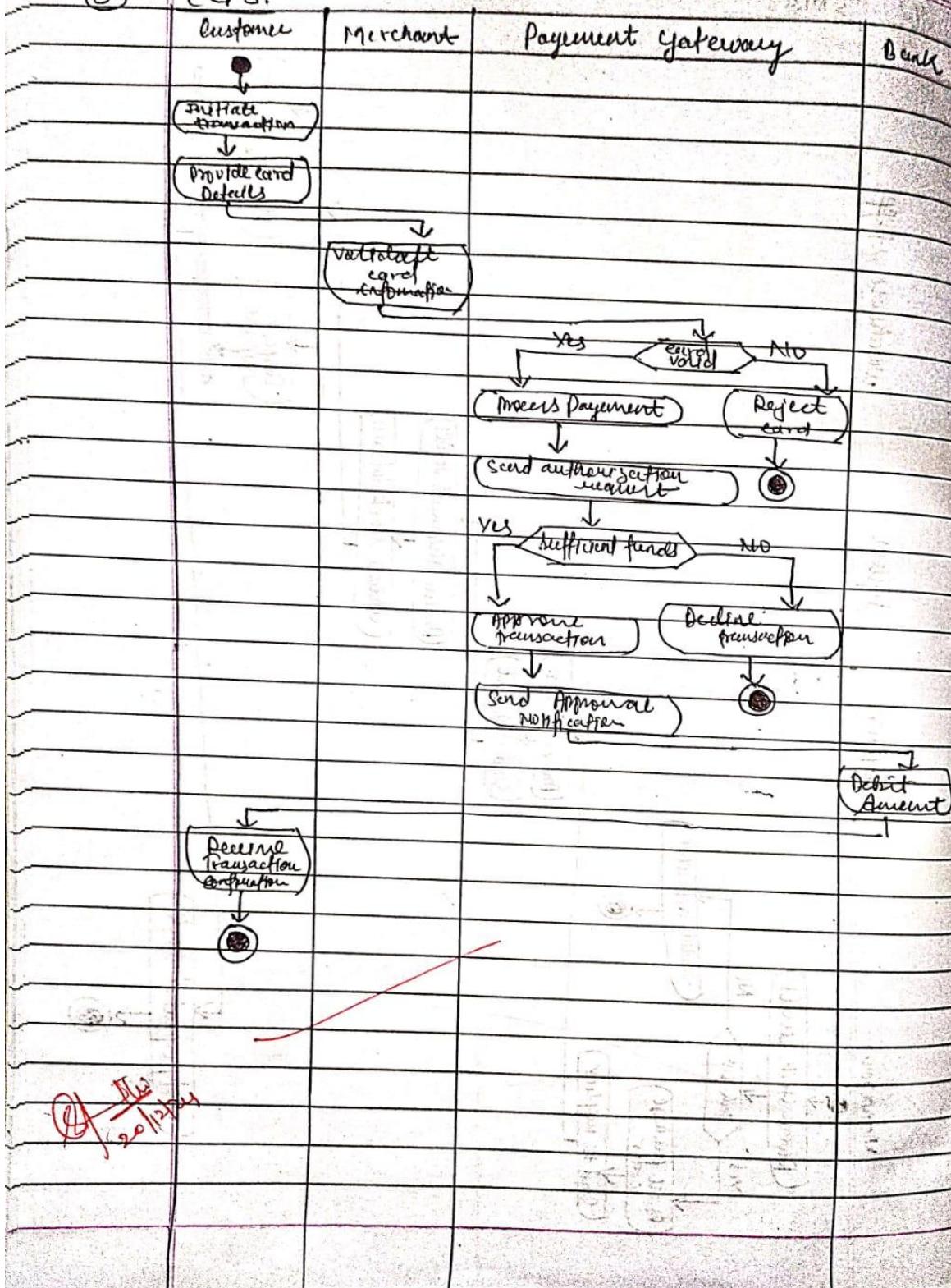


Figure 6.2 – CCPS Activity Diagram

(5)

CCPS:



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.

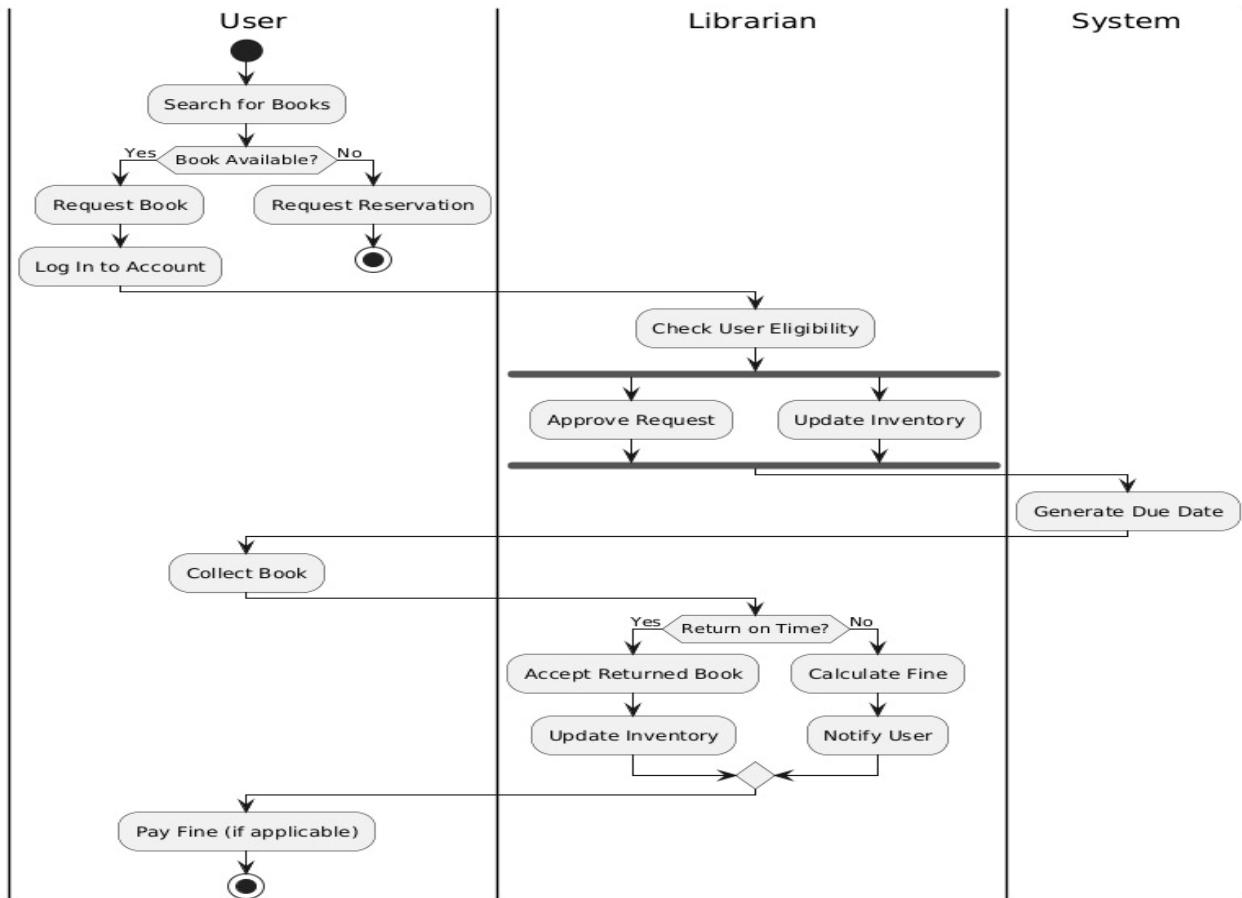
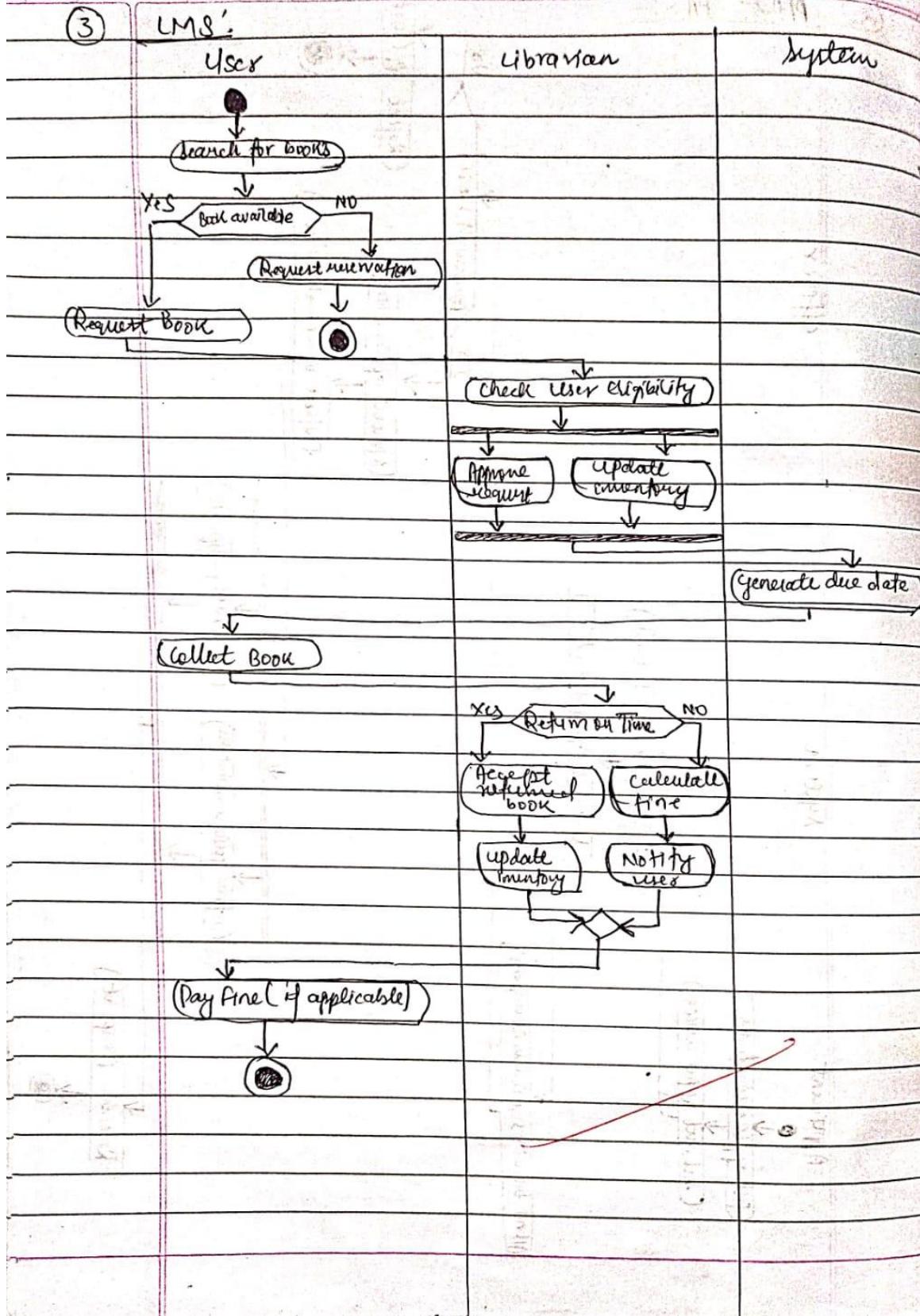


Figure 6.3 – LMS Activity Diagram



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.

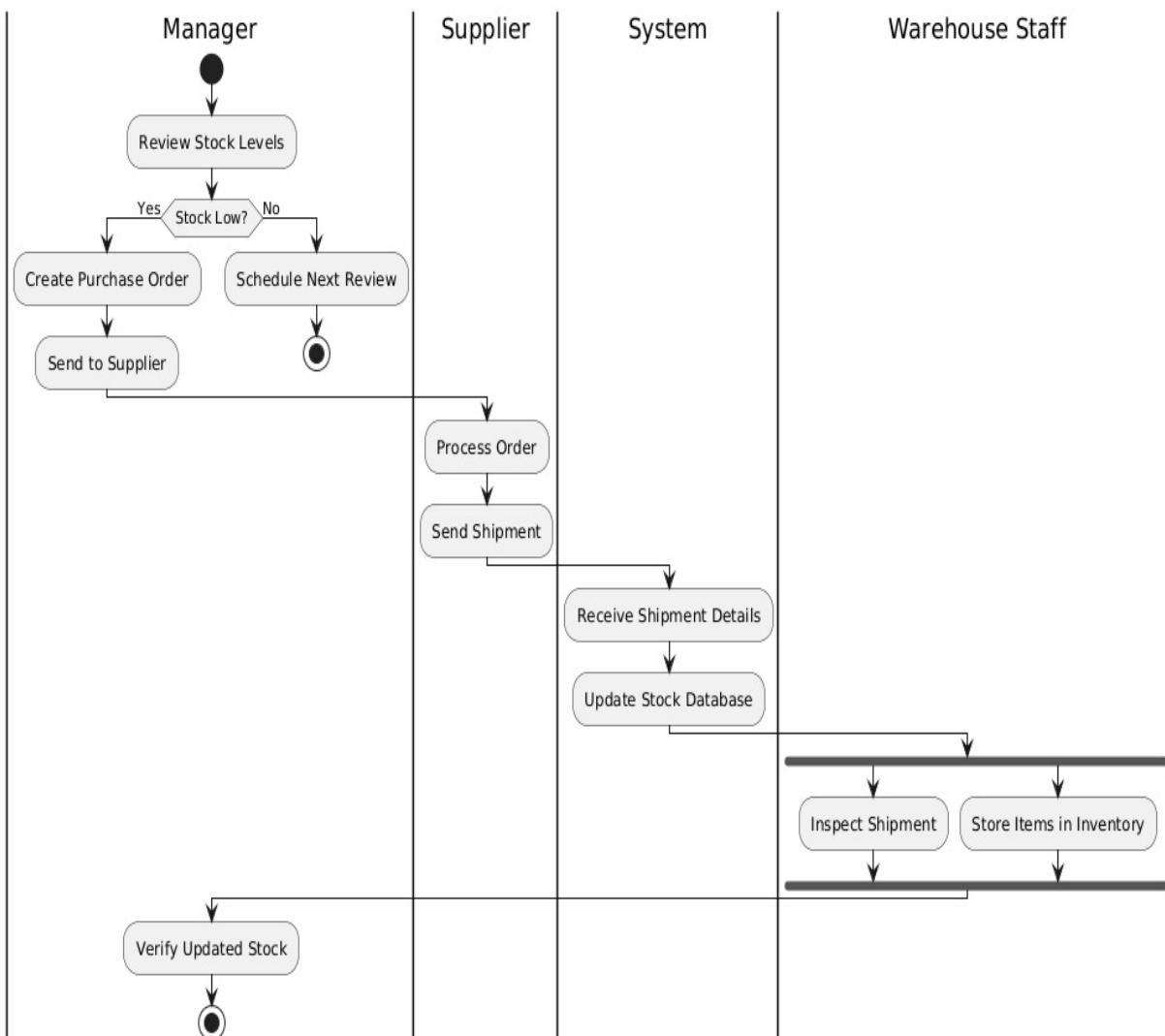
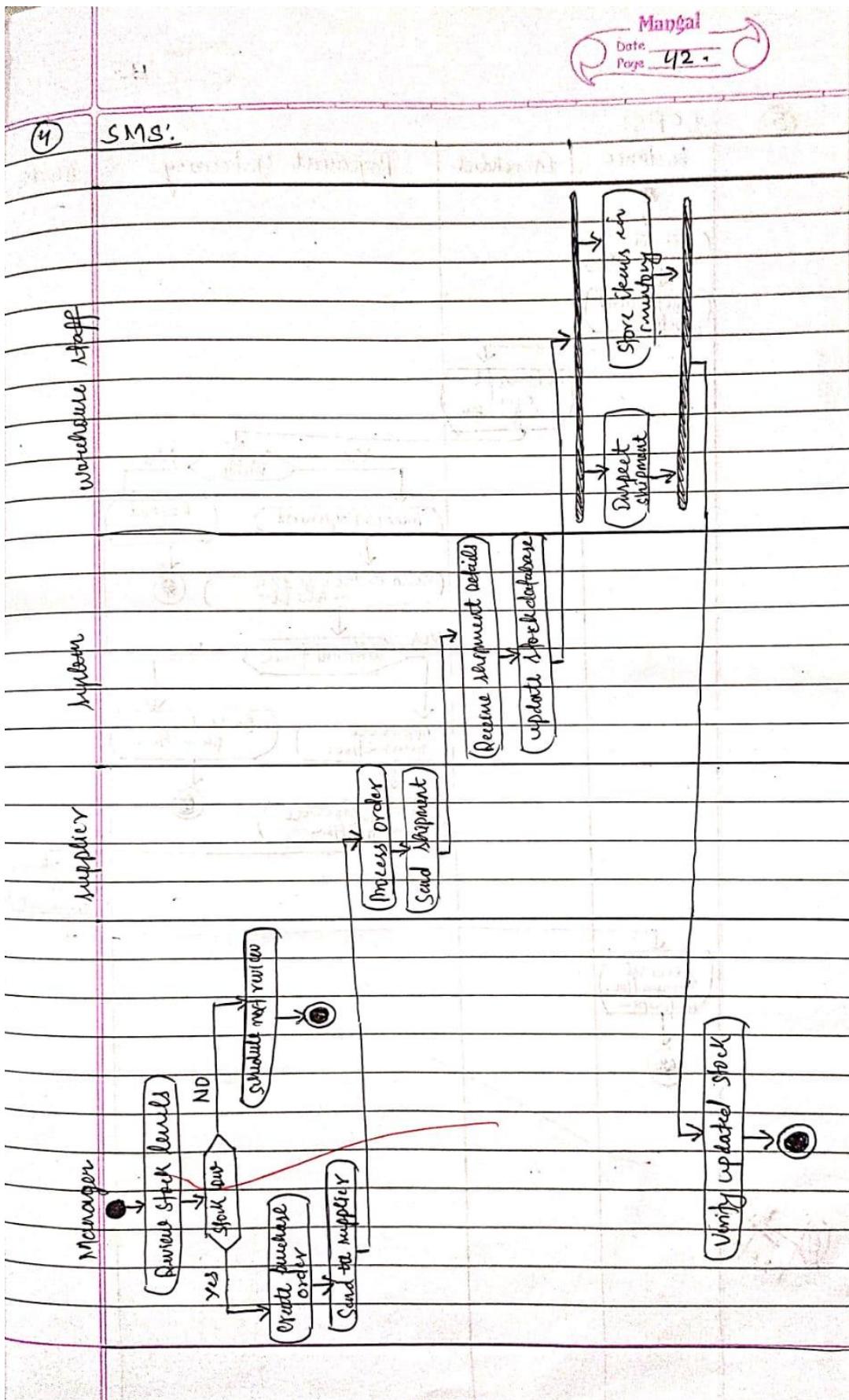


Figure 6.4 – SMS Activity Diagram



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.

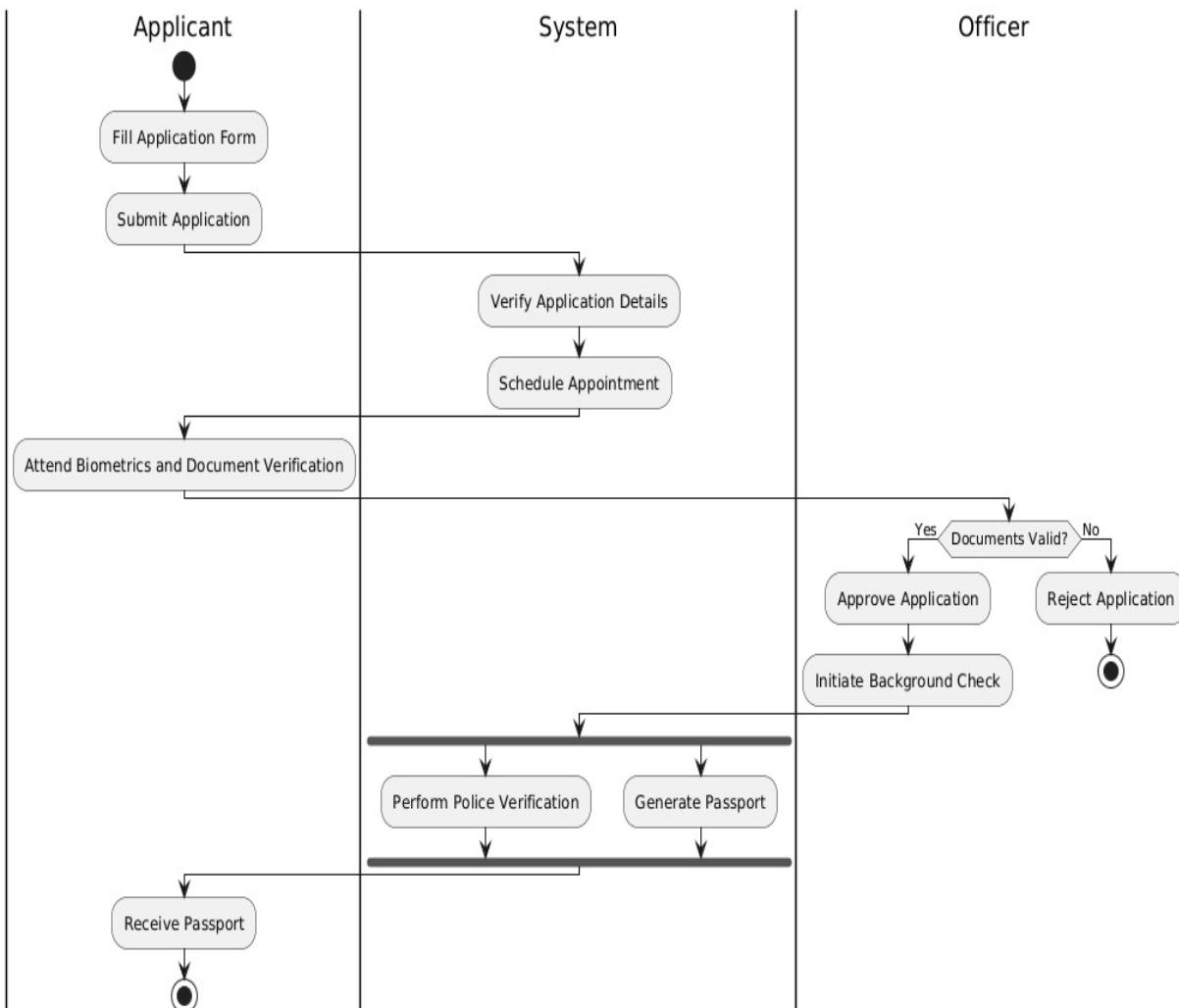


Figure 6.5 – PAS Activity Diagram

