

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



Lab Record

Object-Oriented Modeling – 23CS5PCOOM

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Nidhi D B

1BM23CS211

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

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND

ENGINEERING



CERTIFICATE

This is to certify that the Object-Oriented Modeling(23CS5PCOOM) laboratory has been carried out by Nidhi D B (1BM23CS211) during the 5th Semester August 2025-December 2025

Signature of the Faculty Incharge:

Sunayana S
Assistant Professor

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

Table of Contents

S no	Experiment Title	Pg no
1	Hotel Management System	4
2	Credit Card Processing	12
3	Library Management System	20
4	Stock Maintenance System	28
5	Passport Automation System	26

Github link: <https://github.com/Nidhidb4/OOM>

Chapter 1: Hotel Management System

Problem Statement and SRS:

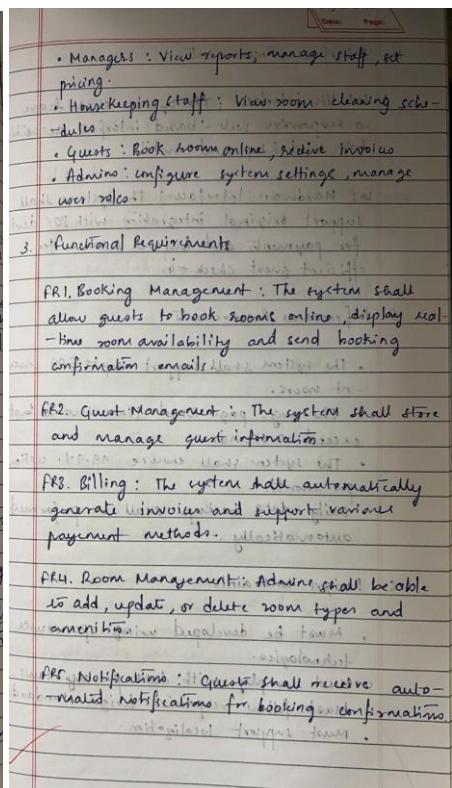
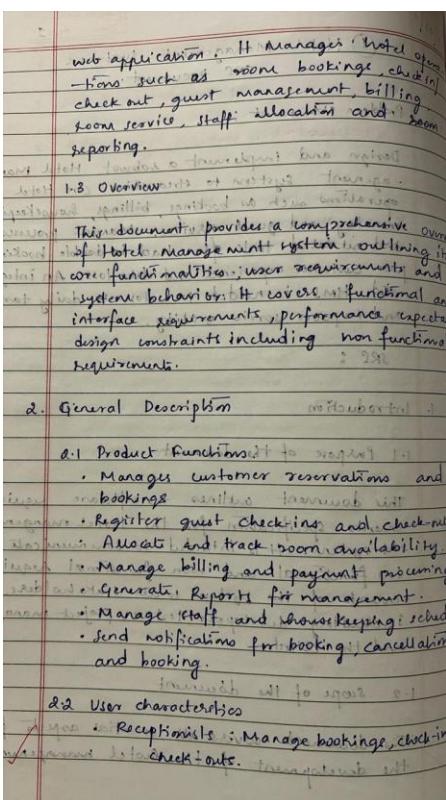
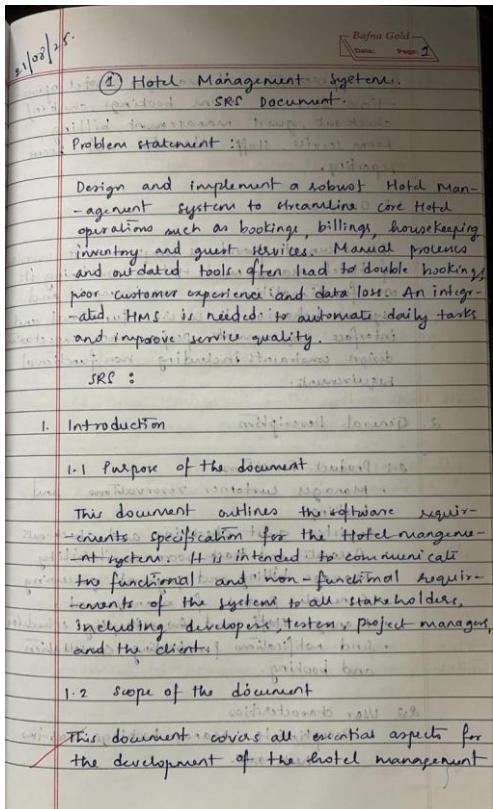


Fig 1.1

Fig 1.2

Fig 1.3

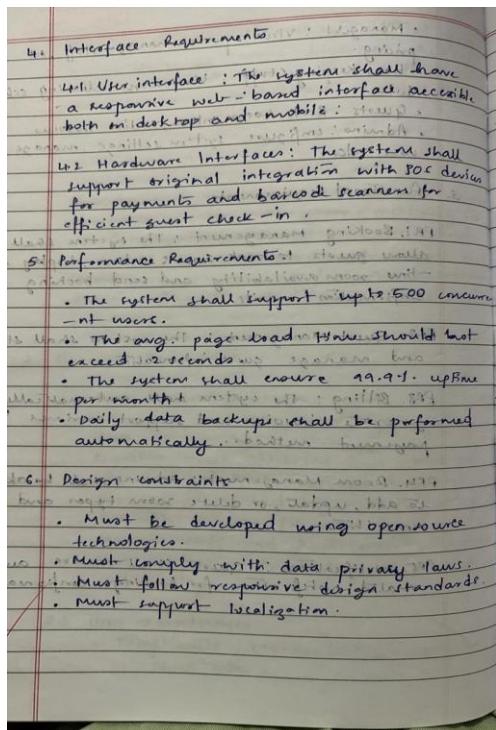


Fig 1.4

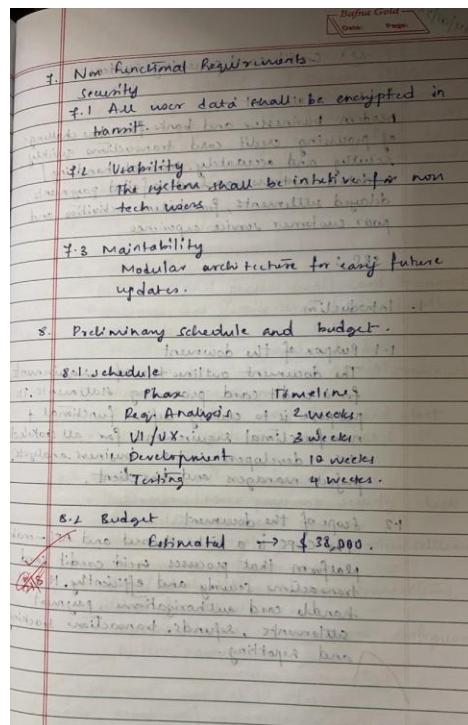


Fig 1.5

Class Diagram:

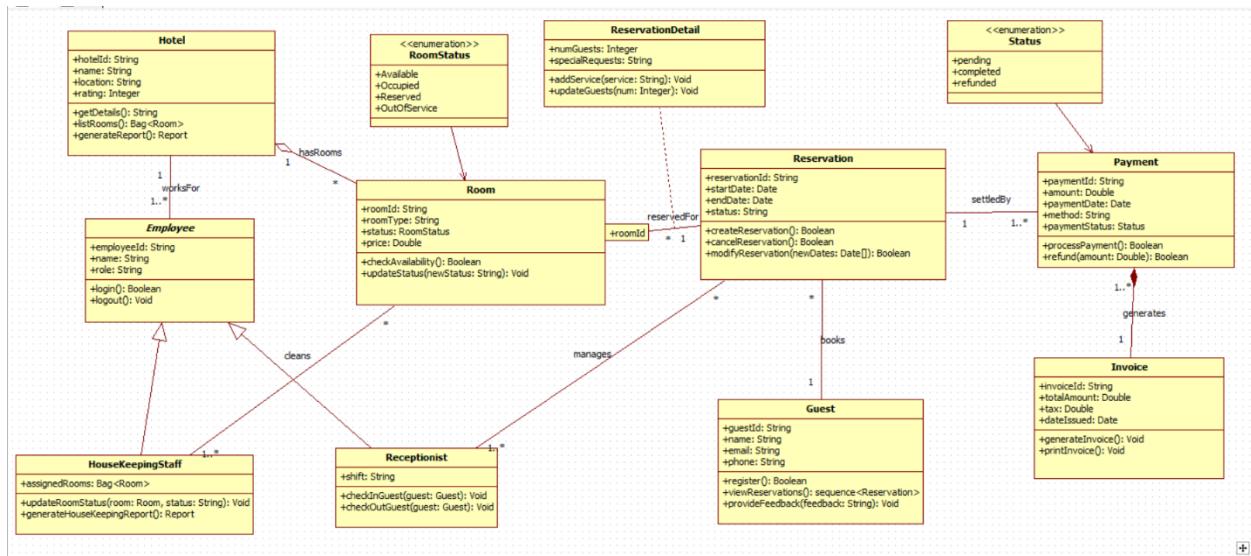


Fig 1.6 Hotel Management System advanced class diagram

This diagram is a UML class diagram for a hotel management system. It models the core entities and their relationships, including classes for Hotel, Room, Reservation, Guest, Employee, and supporting classes for payments, invoices, and housekeeping. The diagram shows how rooms are managed with status and assignments, how reservations are created and linked to guests and rooms, and how payment and invoicing are processed. It uses associations to indicate responsibilities (such as which staff manage rooms, or which guest books a reservation) and enumeration types to manage room and payment statuses. This design provides a structured and object-oriented view of the major operations and data flow in a hotel management application.

State Diagram (Simple and Advanced):

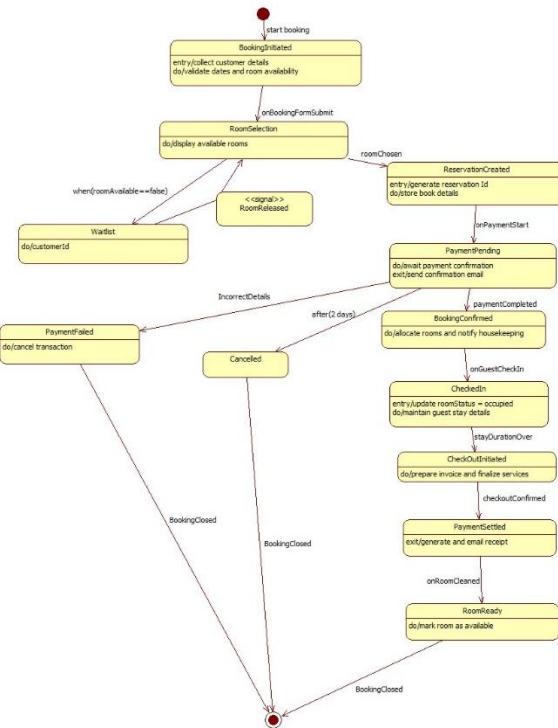


Fig 1.7 HMS simple state diagram

This diagram is a state diagram representing the hotel booking process. It visualizes the main workflow from booking initiation, room selection, payment, and check-in, through to successful check-out or potential cancellation. The diagram includes branches for successful, failed, and waiting states, and clearly shows how the process closes based on different outcomes.

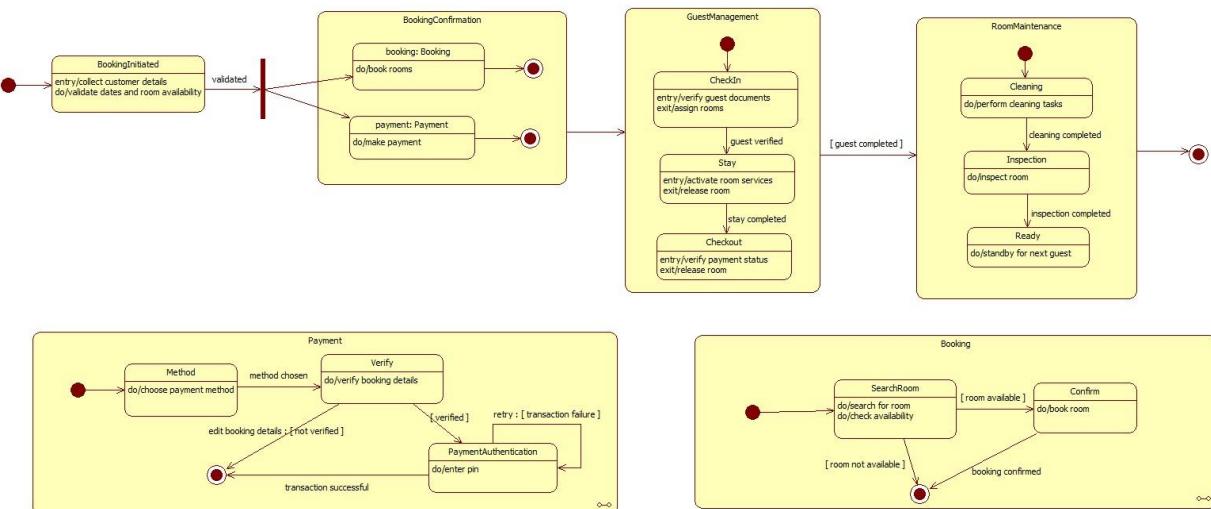


Fig 1.8 HMS advanced state diagram

This diagram shows the major processes in hotel management. It depicts the flow for booking confirmation, payment processing, guest management (check-in, stay, check-out), and room maintenance, detailing how each stage transitions based on actions and outcomes. The diagram provides a clear visual of how bookings and supporting operations progress from initiation to completion or cancellation.

Use Case Diagram (Simple and Advanced):

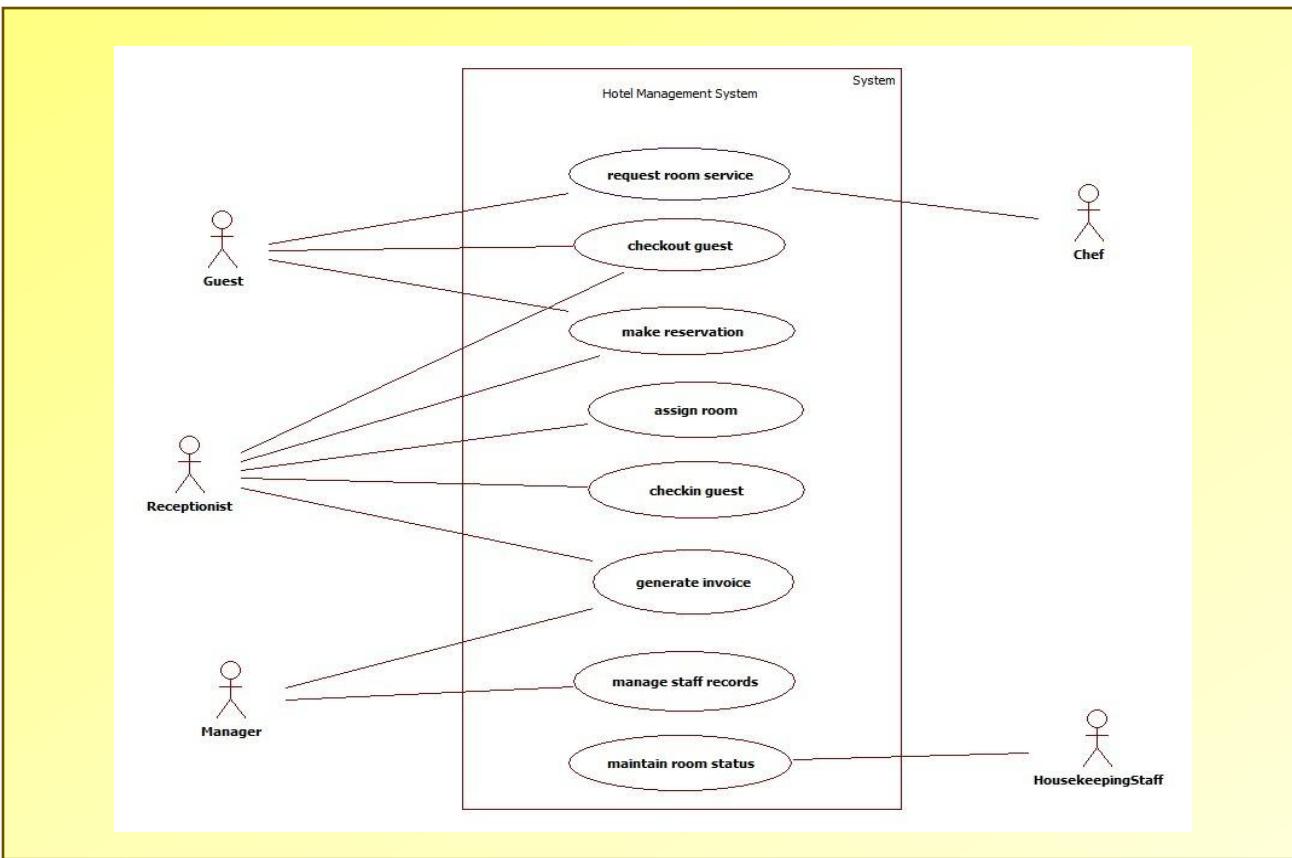


Fig 1.9 HMS simple use case

This use case diagram illustrates the main functions of a hotel management system and the interactions between different types of users, such as guests, receptionists, managers, chefs, and housekeeping staff. It shows how each role can perform actions like making reservations, managing room status, checking in guests, generating invoices, and handling room service requests within the system.

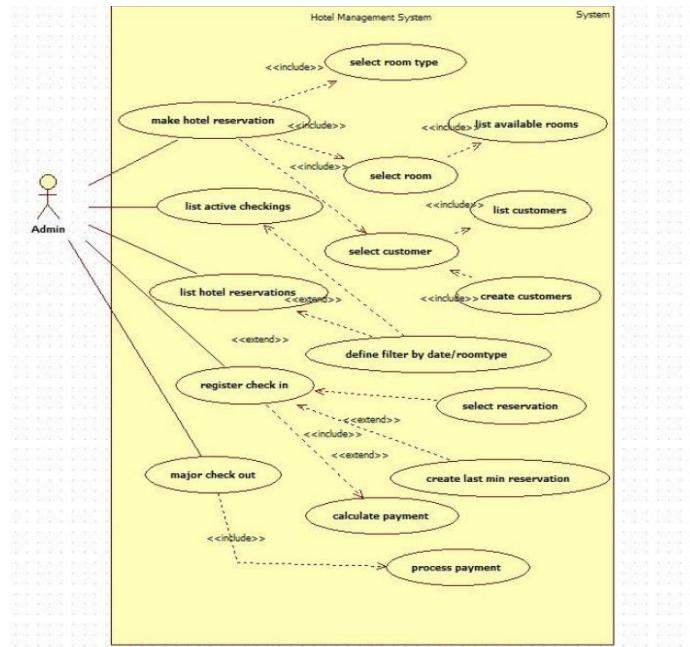


Fig 1.10 HMS advanced use case

This is a use case diagram for a hotel management system, showing how an admin can perform key tasks such as making reservations, registering check-ins, checking out guests, managing customers, and processing payments. It outlines the major system functions and demonstrates how they are related through "include" and "extend" relationships for a complete reservation and management workflow.

Sequence Diagram (Simple and Advanced):

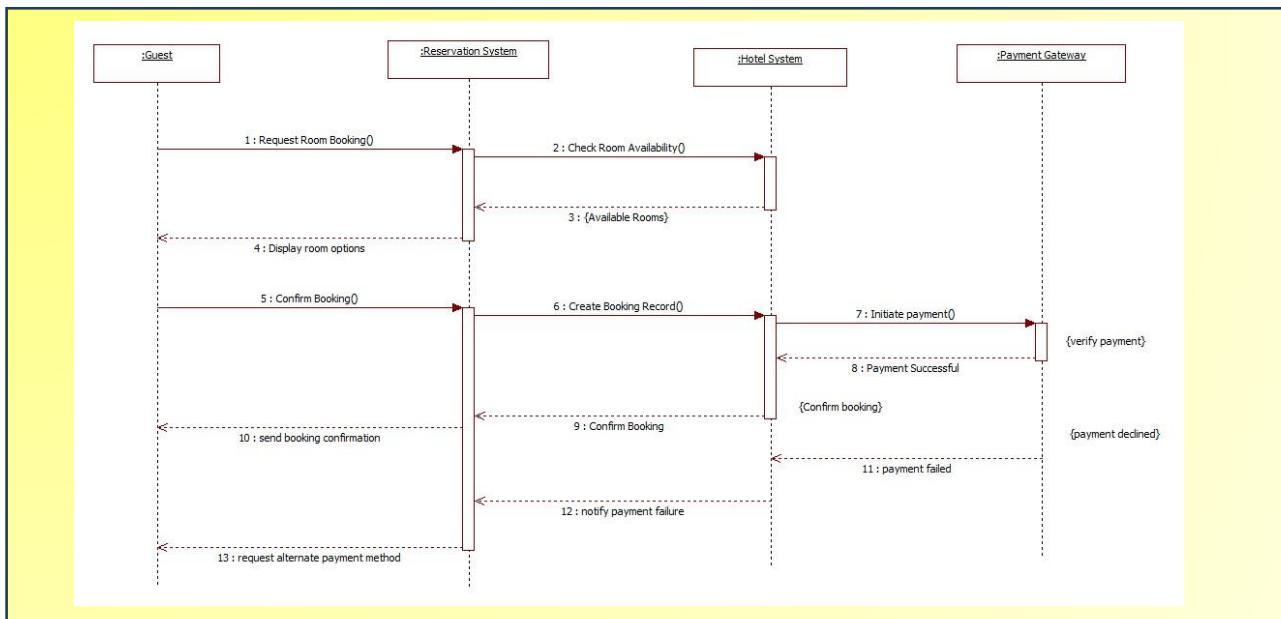


Fig 1.11 HMS Simple sequence

This diagram is a sequence diagram describing the hotel room booking process. It shows how a guest interacts with the reservation system, hotel system, and payment gateway: from requesting a booking, checking room availability, confirming the reservation, to handling payment success or failure, including notifications and possible requests for alternate payment if needed.

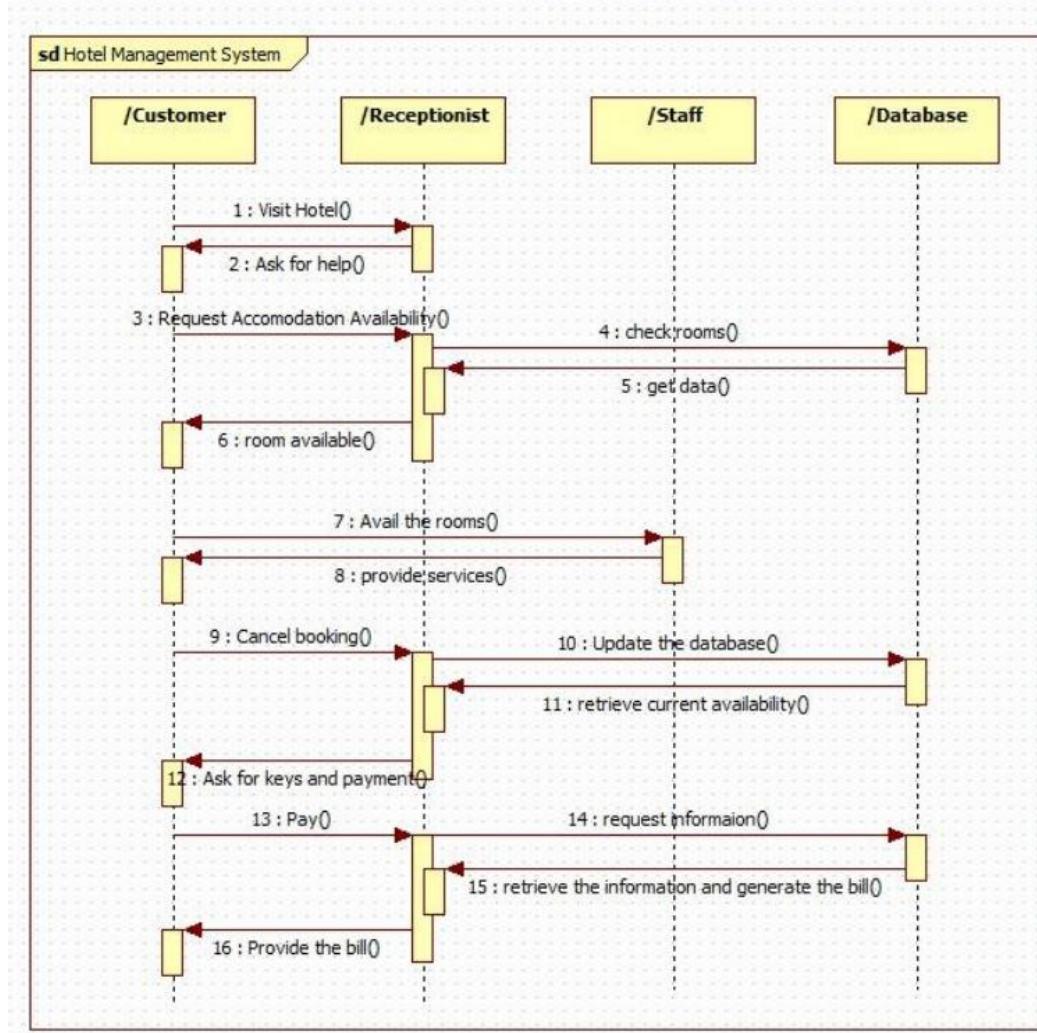


Fig 1.12 HMS Advanced Sequence

This sequence diagram describes the process flow in a hotel management system. It shows how a customer, receptionist, staff, and database interact for tasks like checking room availability, booking, cancelling, payment, and generating the bill. Each message represents a step in the hotel service cycle, from the customer's visit to the final payment and billing.

Activity Diagram (Simple and Advanced):

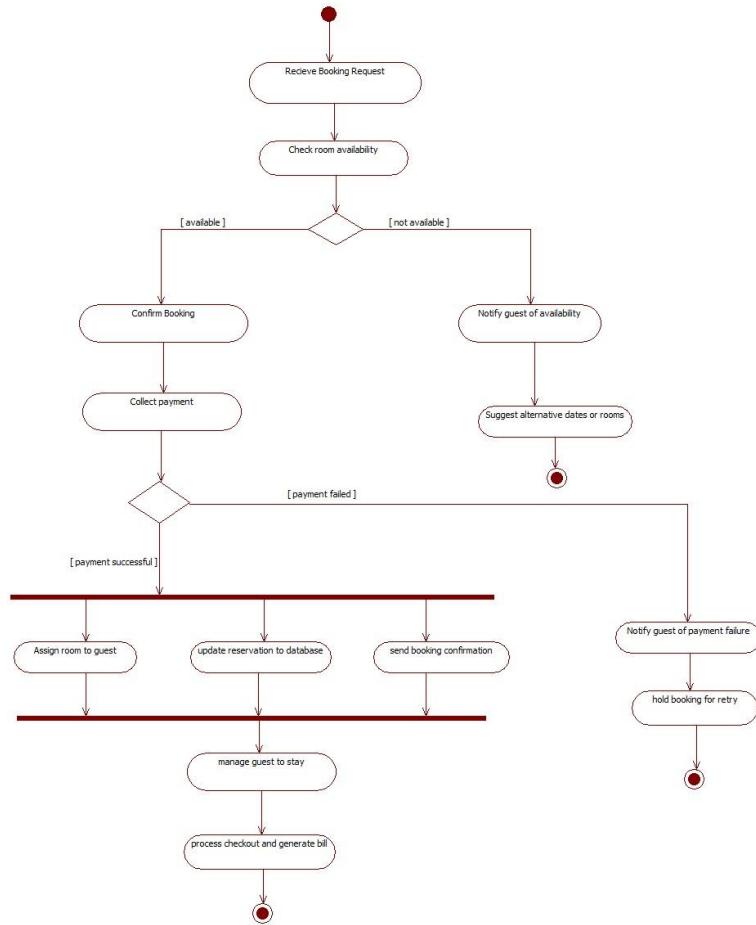


Fig 1.13 HMS simple activity diagram

This diagram shows the end-to-end process of booking a hotel room. It starts from receiving a booking request, checks room availability, confirms the booking, handles payment, and assigns the room. It also details steps for payment failure, offering alternatives or retry, and ends with guest management and bill generation.

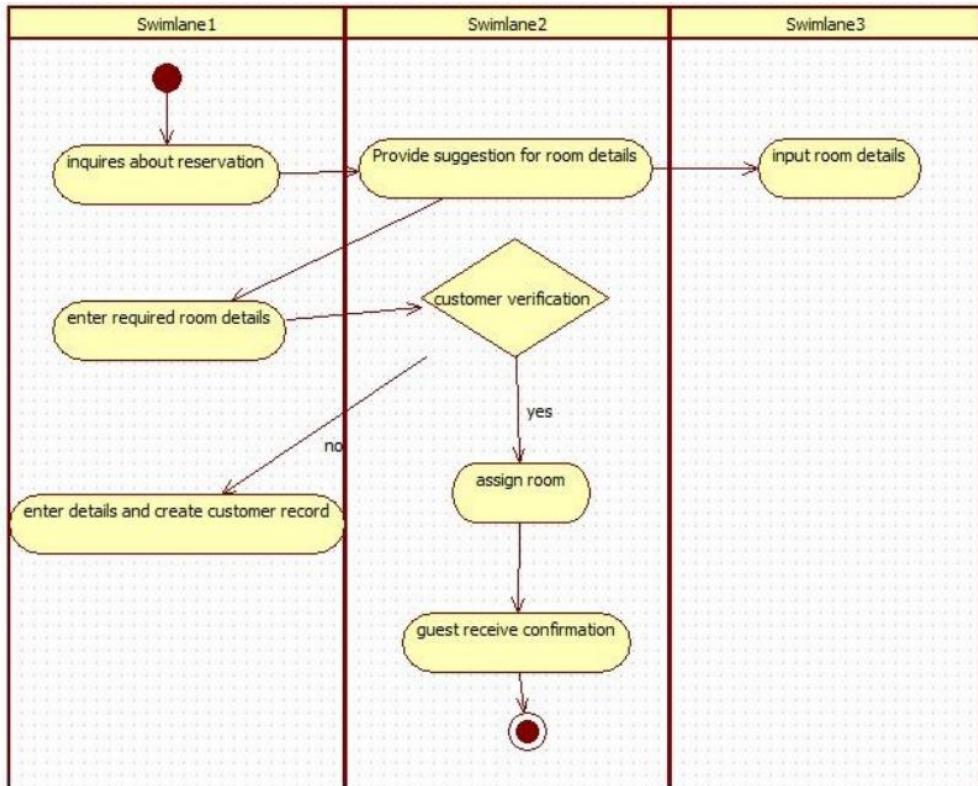


Fig 1.14 HMS Advanced activity

This swimlane diagram illustrates the reservation process in a hotel management system. It divides the workflow across three roles, showing how a customer inquires and enters details, the system suggests and verifies information, and once verified, a room is assigned and a confirmation is sent to the guest.

Chapter 2: Credit Card Processing

Problem statement and SRS:

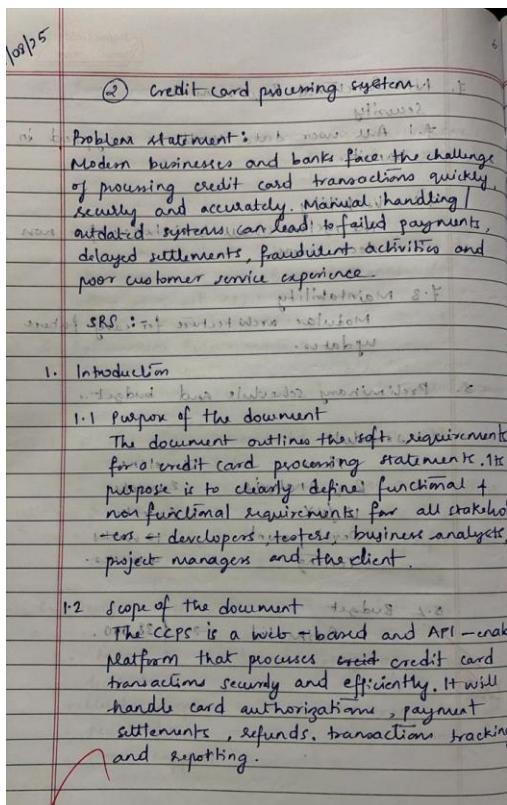


Fig 2.1

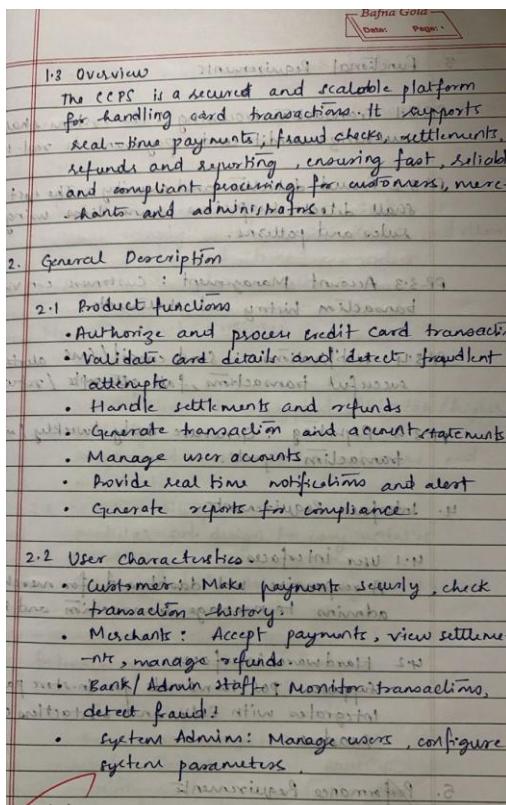


Fig 2.2

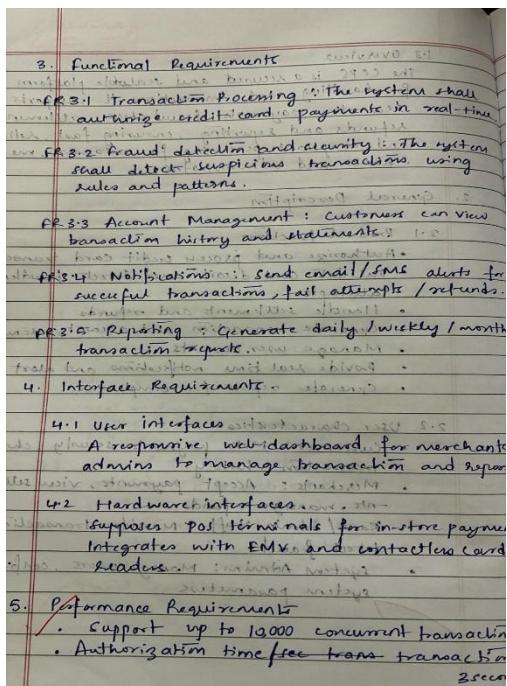


Fig 2.3

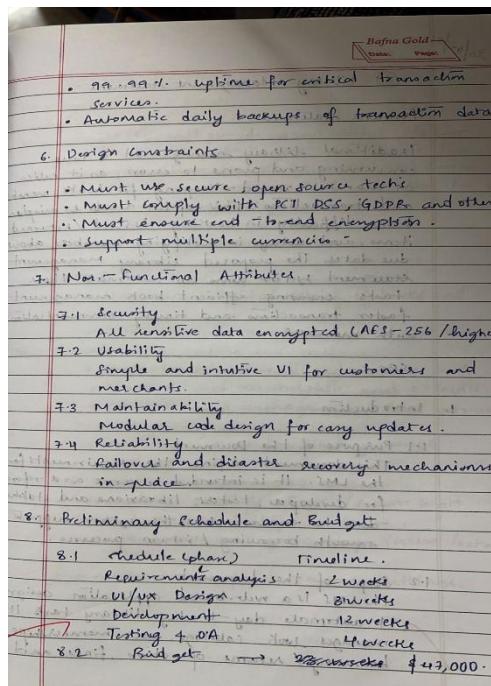


Fig 2.4

Class diagram:

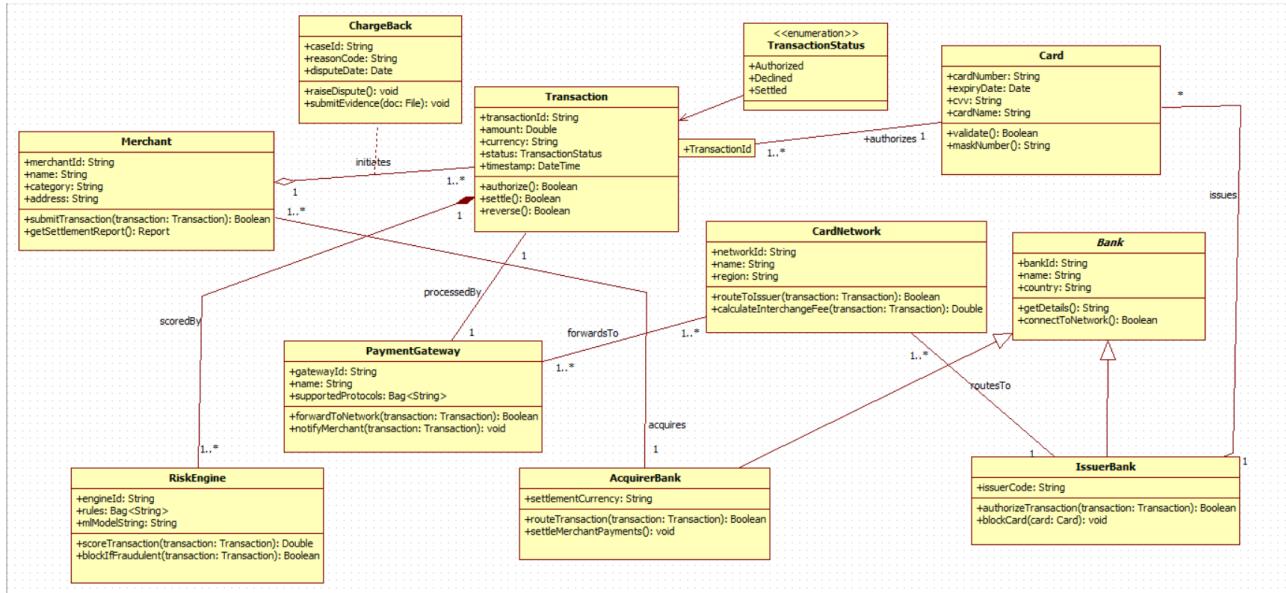


Fig 2.5 CCP class diagram

This UML class diagram models the payment and transaction process in an electronic payment system. It defines how merchants, payment gateways, card networks, risk engines, banks (issuer and acquirer), and cards interact to process, authorize, and settle transactions. The diagram also includes support for chargebacks, risk assessment, and different transaction statuses, outlining the full transaction life cycle from initiation to settlement or dispute.

State Diagrams (Simple and advanced):

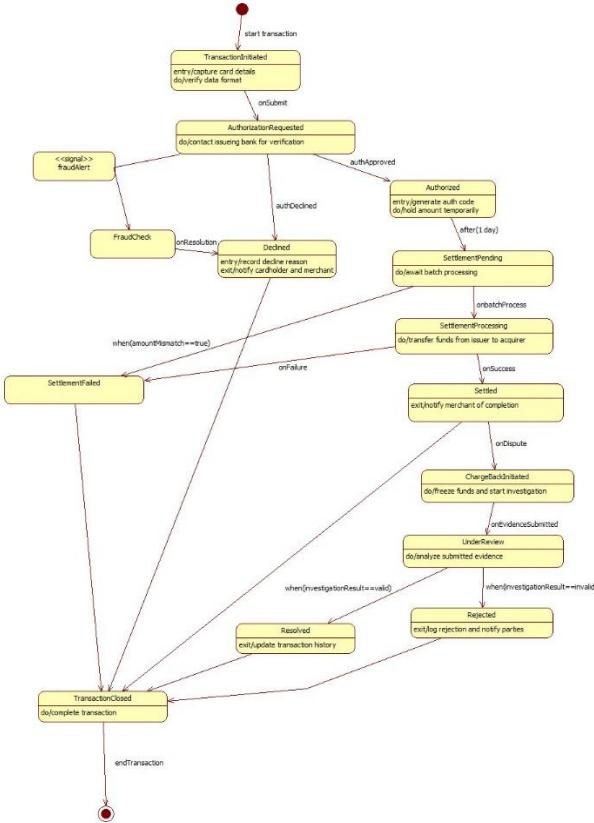


Fig 2.6 CCP simple state diagram

This state diagram illustrates the lifecycle of a financial transaction. It tracks each stage from initiation, authorization, clearing batch processing, potential decline or fraud check, settlement, and chargeback, through to transaction closure. The diagram shows how errors, disputes, or successful completions determine the path to closing the transaction.

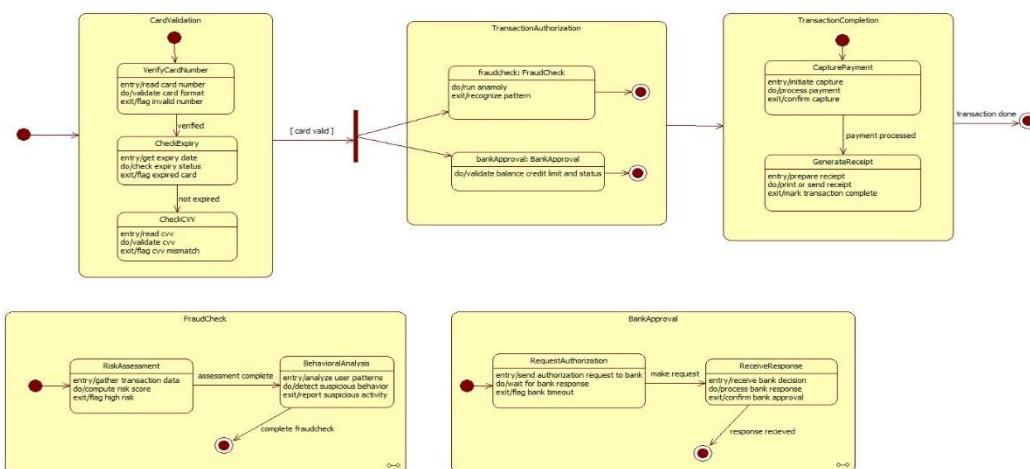


Fig 2.7 CCP advanced state diagram

This diagram details the stages of a card payment transaction. It covers card validation, transaction authorization (with fraud and behavioral checks, plus bank approval), and completion (including capture and settlement). The flow highlights key security checks and decision points before a transaction is finalized.

Use Case diagrams (Simple and Advanced):

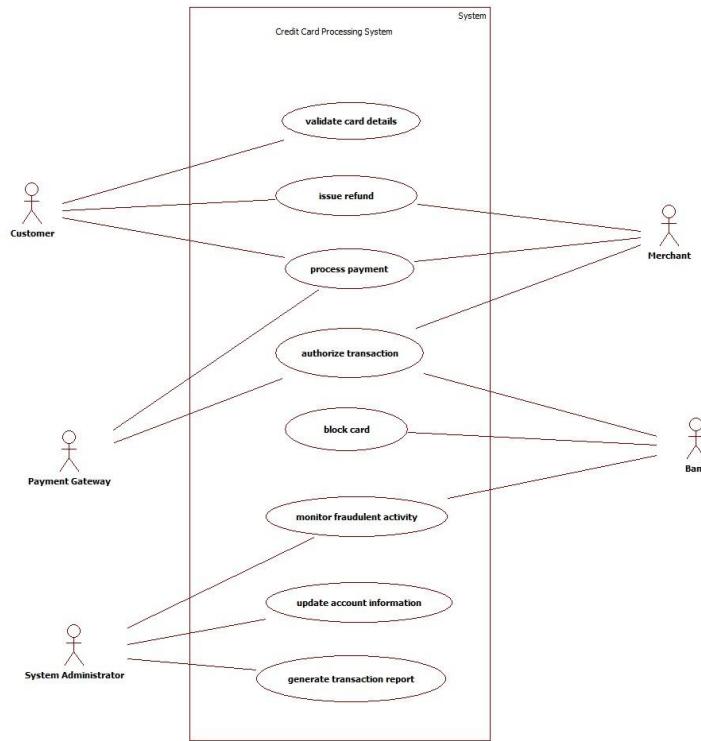


Fig 2.8 CCP simple use case

This use case diagram shows the main operations in a credit card processing system, including validating card details, processing payments, authorizing transactions, issuing refunds, blocking cards, and monitoring fraudulent activity. It highlights how customers, merchants, payment gateways, banks, and system administrators interact with the system to complete financial tasks and manage security.

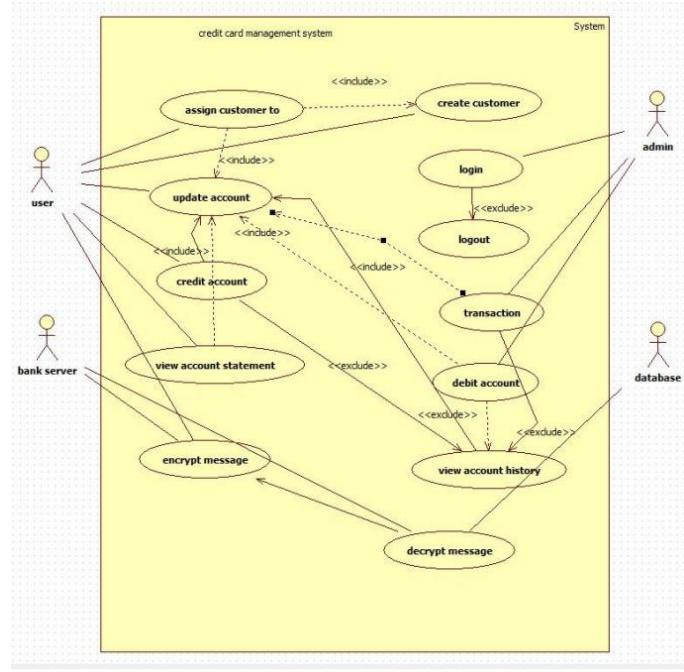


Fig 2.9 CCP advanced use case

This use case diagram models a credit card management system, showing how users, admins, bank servers, and the database interact with the system. It covers core actions like creating and assigning customers, account updates, credit and debit transactions, encrypting/decrypting messages, and viewing account statements and history, with include and exclude relationships linking related use cases.

Sequence Diagrams (Simple and Advanced):

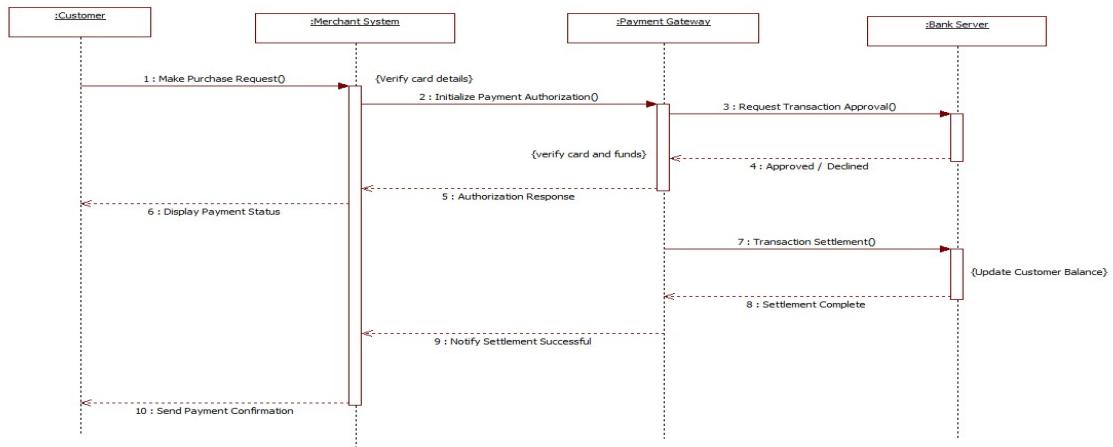


Fig 2.10 CCP simple sequence diagram

This sequence diagram outlines the steps in a credit card payment process. It shows how a customer makes a purchase, the merchant system verifies card details and authorization, the payment gateway and bank server process and settle the transaction, and confirmations are sent at each stage, finishing with payment confirmation to the customer.

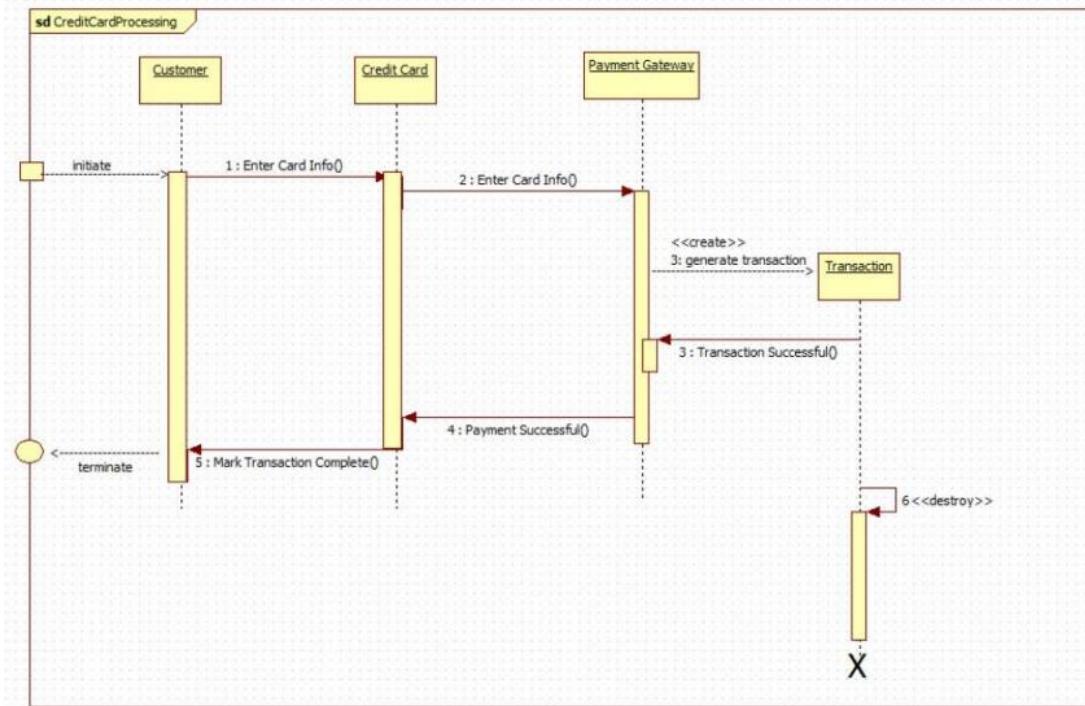


Fig 2.11 CCP advanced sequence diagram

This sequence diagram presents the credit card processing workflow. It depicts how a customer initiates payment by entering card information, which is passed through the credit card and payment gateway to generate a transaction. The process ends with a successful payment acknowledgment and concludes by marking the transaction complete and destroying the transaction instance.

Activity Diagrams (Simple and advanced):

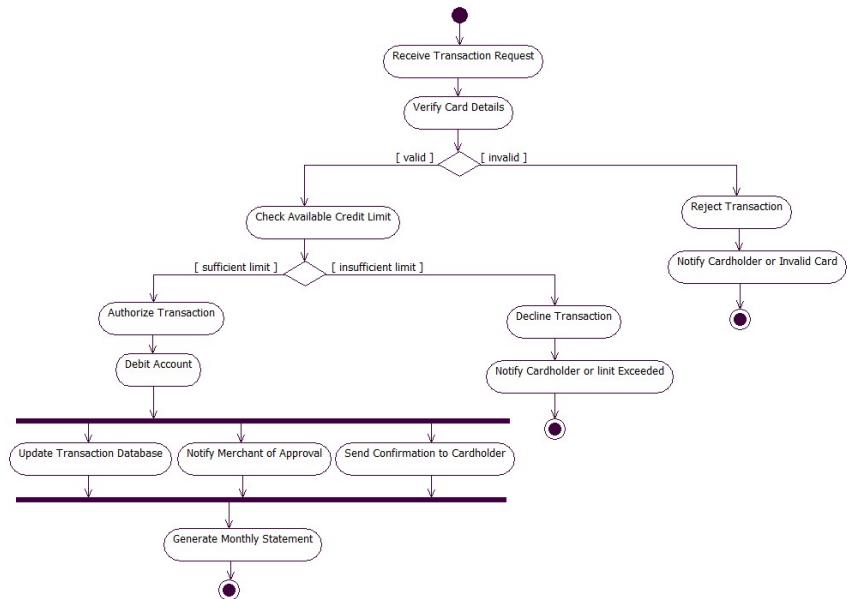


Fig 2.12 CCP simple activity diagram

This diagram visualizes the credit card transaction process. It starts with receiving a transaction request, then verifies card details and the available credit limit. If valid and sufficient, the transaction is authorized and the account is debited, followed by database and notification steps. If invalid or insufficient, the transaction is declined and the user is notified, ending the process.

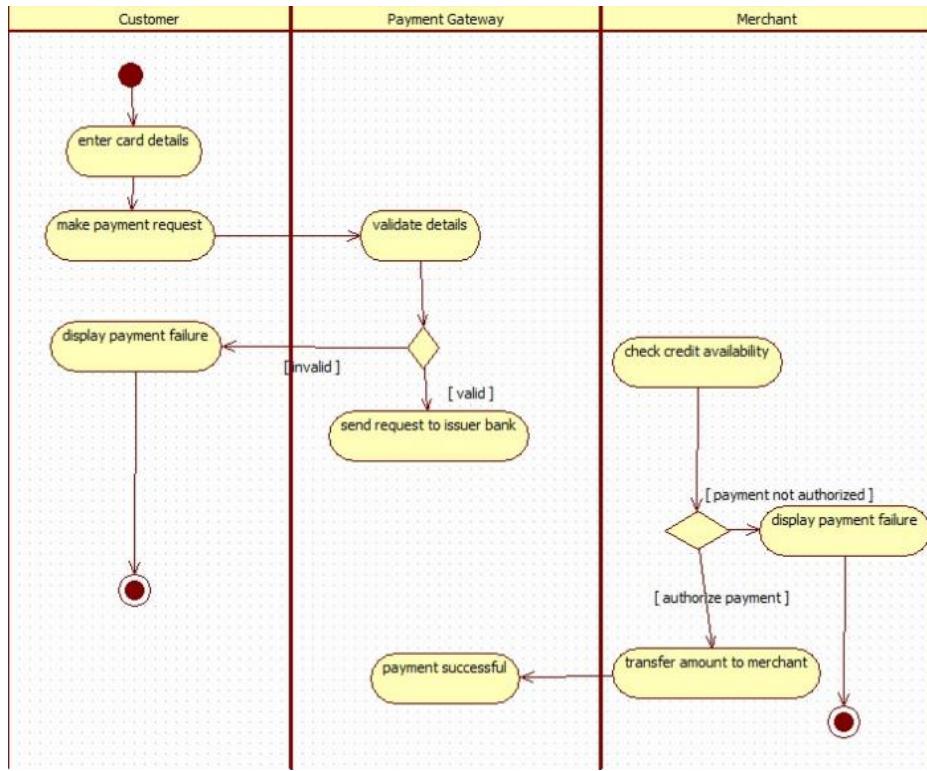


Fig 2.13 CCP advanced activity diagram

This swimlane flowchart illustrates the credit card payment process. It starts with the customer entering card details and making a payment request, then the payment gateway validates the details and communicates with the issuer bank. If the payment is valid and authorized, the merchant receives the funds; otherwise, payment failure is displayed at each stage.

Chapter 3: Library Management System

Problem statement and SRS:

<p>(3) Library Management statement :</p> <p>Traditional library management is time-consuming and prone to errors, as it relies on manual record-keeping for books, members and transactions. This often leads to misplaced books, difficulty in tracking issued / returned items, and delays in notifying members about due dates. The proposed library management system aims to automate these tasks, ensuring efficient book management, faster transactions and timely communication with members.</p> <p>SR8 : What are the objectives?</p> <p>1. Introduction</p> <p>1.1 Purpose of the Document</p> <p>This document defines the requirements for the LMS. It is intended to serve as a reference for developers, testers, librarians and stakeholders to ensure that the system supports smooth borrowing / return processes.</p> <p>1.2 Scope of the document</p> <p>The LMS is a web-based application designed to automate day-to-day library tasks. It manages book catalog, user membership, borrowing / return operations, fines and reporting.</p>	<p>1.3 Overview:</p> <p>The LMS will provide a centralized platform for managing books, users, and transactions. It will allow students to search and borrow books online, librarians to handle check-ins and check-outs efficiently, and administrators to manage user access and reporting.</p> <p>2. General Description:</p> <p>2.1 Product Functions</p> <ul style="list-style-type: none"> Maintain a digital catalog of books and resources Allow students to search, reserve and borrow books Track issued, returned, and overdue books Manage user membership and accounts Calculate and manage late fines automatically Generate reports for library usage <p>2.2 User characteristics</p> <ul style="list-style-type: none"> Students / Members: Borrow / return books, check availability, renew loans. Librarians: Manage catalog, issue / return books. Administrators: Configure system settings. 	<p>3. Functional Requirements:</p> <p>FR 3.1 Catalog management: The system shall register new members with personal details.</p> <p>FR 3.2 The system shall allow librarians to add, update, or delete book records.</p> <p>FR 3.3 Students shall be able to search for books by title, author, or category.</p> <p>FR 3.4 The system shall send reminders for due dates and overdue books.</p> <p>FR 3.5 The system shall generate daily / weekly / monthly reports.</p> <p>4. Interface Requirements:</p> <p>4.1 User interfaces</p> <ul style="list-style-type: none"> A clean web dashboard for librarians to manage books and transactions. <p>4.2 Hardware Interfaces</p> <ul style="list-style-type: none"> Supports Barcode / RFID scanners for fast issue / return. <p>5. Performance Requirements:</p> <ul style="list-style-type: none"> Must handle up to 200 concurrent users. Search results must load within 2 seconds. System uptime must be at least 99.9% month. Automatic backups must run daily. <p>6. Design constraints:</p> <ul style="list-style-type: none"> Developed using open-source tech. Must support responsive design for mobile devices.
--	---	---

Fig 3.1

Fig 3.2

Fig 3.3

<p>7. Non-functional Requirements:</p> <p>7.1 Security: Data encrypted during transmission.</p> <p>7.2 Usability: Intuitive UI for both students and librarians.</p> <p>7.3 Maintainability: Modular architecture for easy updates.</p> <p>7.4 Reliability: Handles unexpected shutdowns without data loss.</p> <p>8. Planning Schedule and Budget:</p> <p>8.1 Schedule</p> <table border="1"> <tr> <td>Phase 1: Requirements Analysis</td><td>Timeline: 2 weeks</td></tr> <tr> <td>Phase 2: UI/UX design</td><td>Timeline: 2 weeks</td></tr> <tr> <td>Phase 3: Development</td><td>Timeline: 8 weeks</td></tr> <tr> <td>Phase 4: Testing & QA</td><td>Timeline: 1 week</td></tr> </table> <p>8.2 Budget</p> <p>Total estimated cost: \$ 28,000.</p> <p>Cost breakdown: - Software license: \$ 10,000 - Development: \$ 12,000 - Testing: \$ 4,000 - Training: \$ 2,000</p>	Phase 1: Requirements Analysis	Timeline: 2 weeks	Phase 2: UI/UX design	Timeline: 2 weeks	Phase 3: Development	Timeline: 8 weeks	Phase 4: Testing & QA	Timeline: 1 week
Phase 1: Requirements Analysis	Timeline: 2 weeks							
Phase 2: UI/UX design	Timeline: 2 weeks							
Phase 3: Development	Timeline: 8 weeks							
Phase 4: Testing & QA	Timeline: 1 week							

Fig 3.4

Class Diagram:

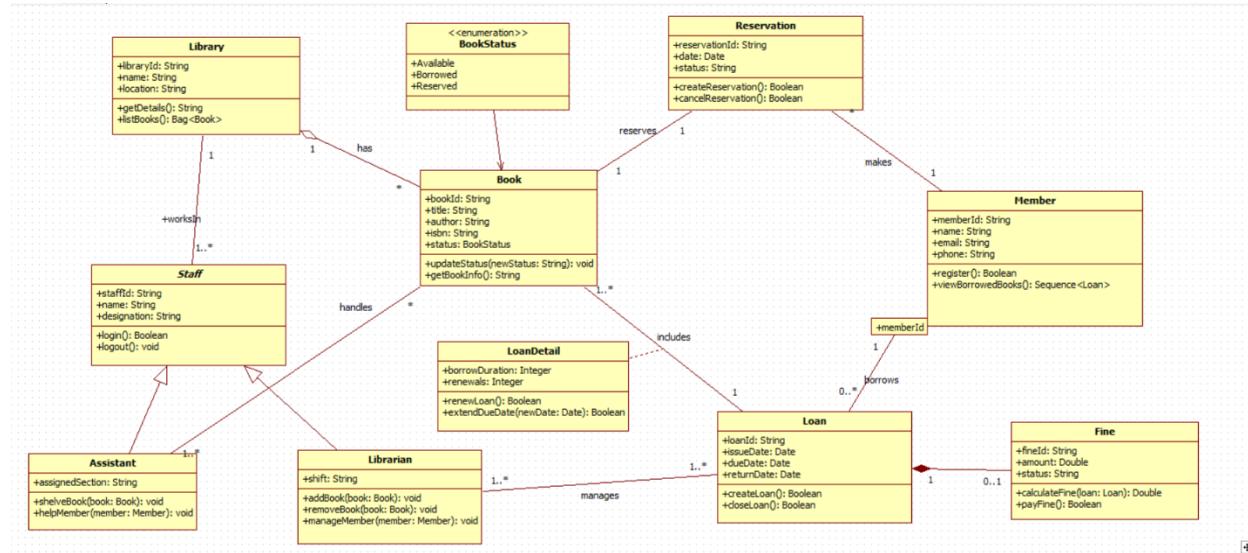


Fig 3.5 LMS advanced class diagram

This UML class diagram represents a library management system. It models entities such as library, book, member, staff, loans, reservations, and fines. The diagram shows their attributes, key operations, and how they interact for borrowing, reserving, and returning books, as well as handling fines and staff duties.

State Diagrams (Simple and Advanced):

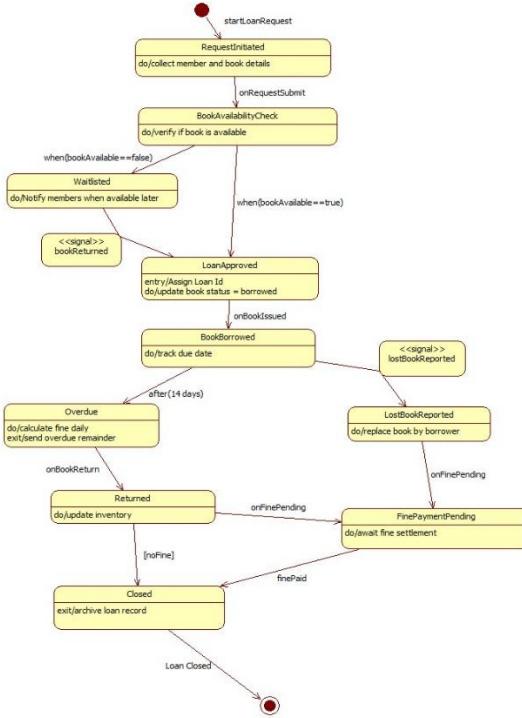


Fig 3.6 LMS simple state diagram

This state diagram models the life cycle of a book loan in a library system. It covers loan requests, book availability checks, waitlisting, approval, book borrowing, overdue handling, lost book reporting, return processing, fine calculations, and loan closure, showing each key state from request to closure.

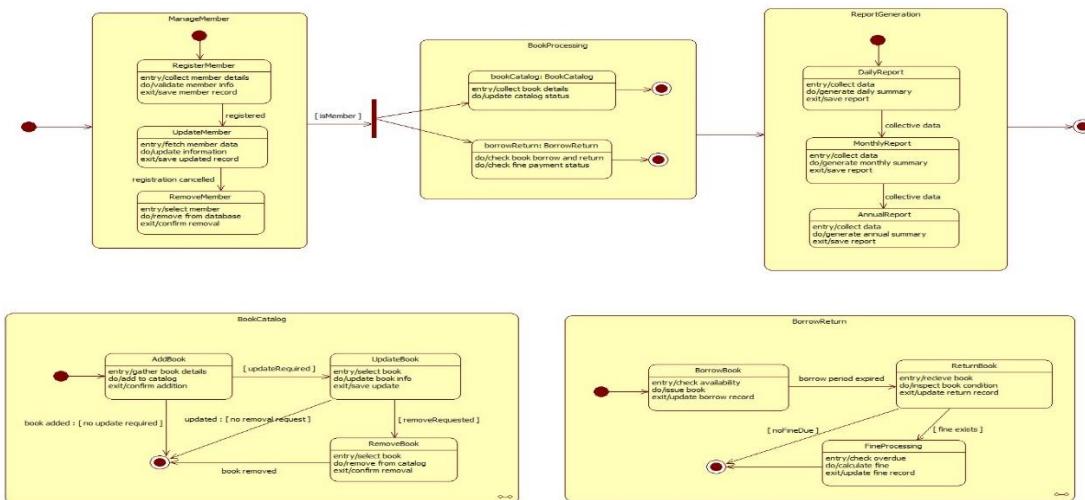


Fig 3.7 LMS Advanced state diagram

This diagram shows the state and activities for library management. It covers processes for member registration, book cataloging, borrowing and returning books, and generating reports. The workflow details how members and books are managed, how borrow/return actions trigger fine checks, and how different types of library reports are created.

Use Case Diagrams (Simple and Advanced):

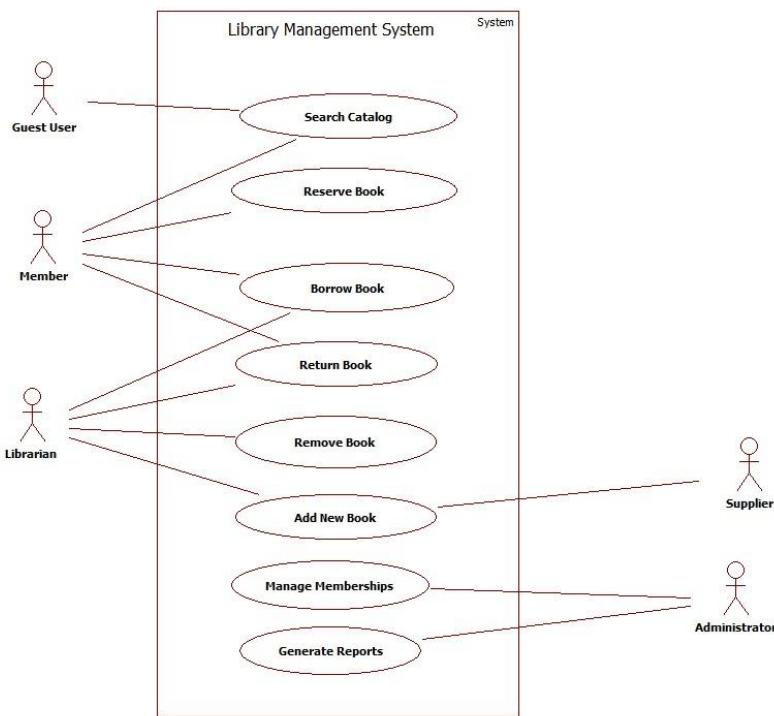


Fig 3.8 LMS simple use case

This use case diagram summarizes the main features of a library management system. It depicts roles such as guest user, member, librarian, supplier, and administrator, and key system functions like searching the catalog, reserving, borrowing, and returning books, as well as managing books, memberships, and generating reports.

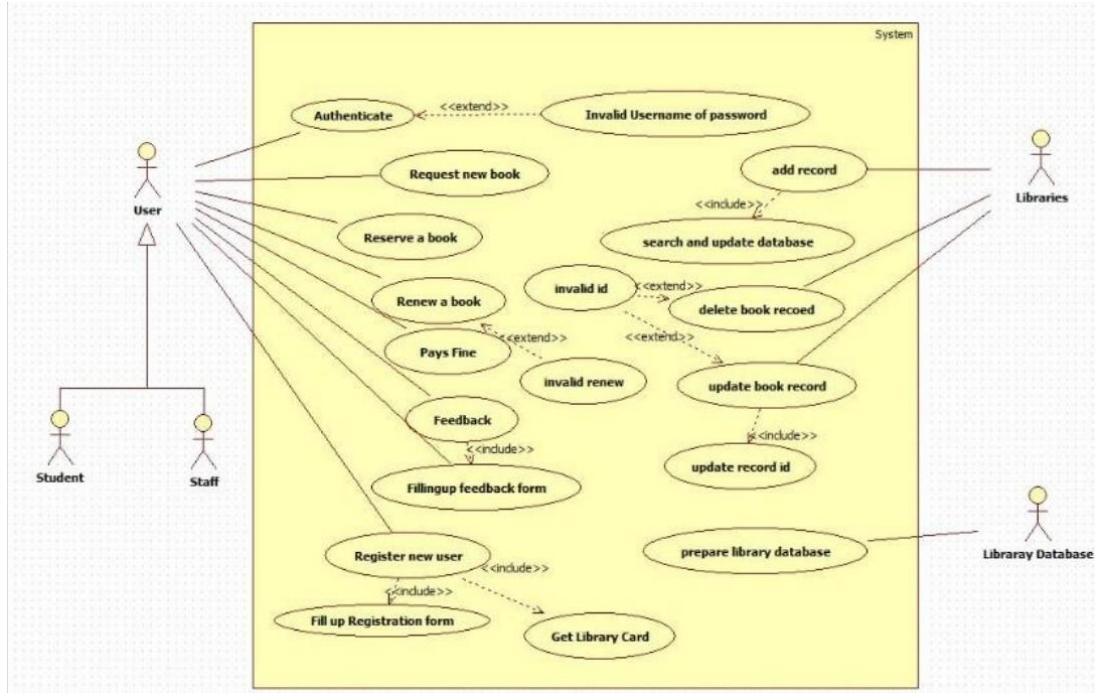


Fig 3.9 LMS advanced use case

This use case diagram depicts various functions in a library management system, involving users (students, staff), libraries, and the library database. It demonstrates actions like authentication, book reservation, renewal, fines, registration, feedback, and database maintenance, including special/exception cases such as invalid records or authentication failures.

Sequence Diagrams (Simple and Advanced):

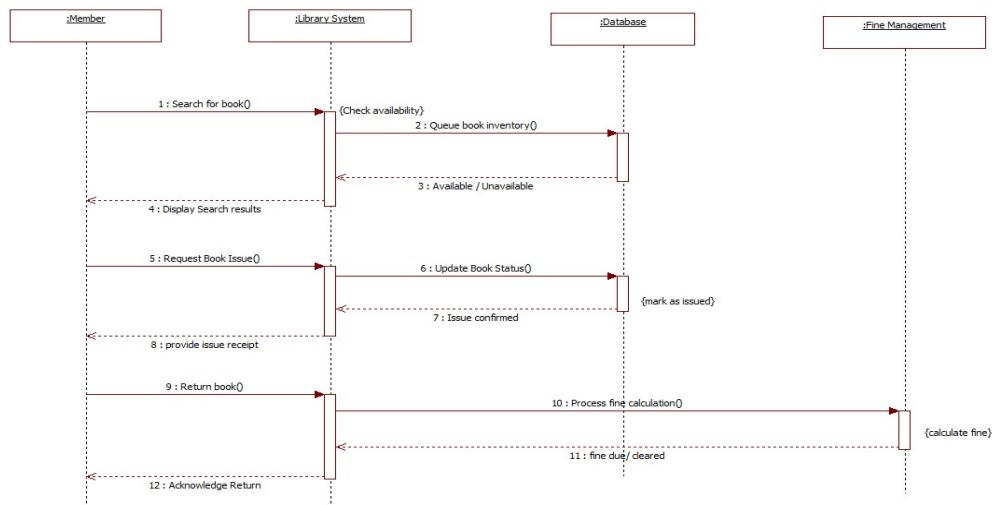


Fig 3.10 LMS simple sequence diagram

This sequence diagram depicts how a library member interacts with the library system to search for a book, request issue, return it, and handle fines. It shows message exchanges between the member, library, database, and fine management system at each step, from searching and issuing to processing returns and clearing fines.

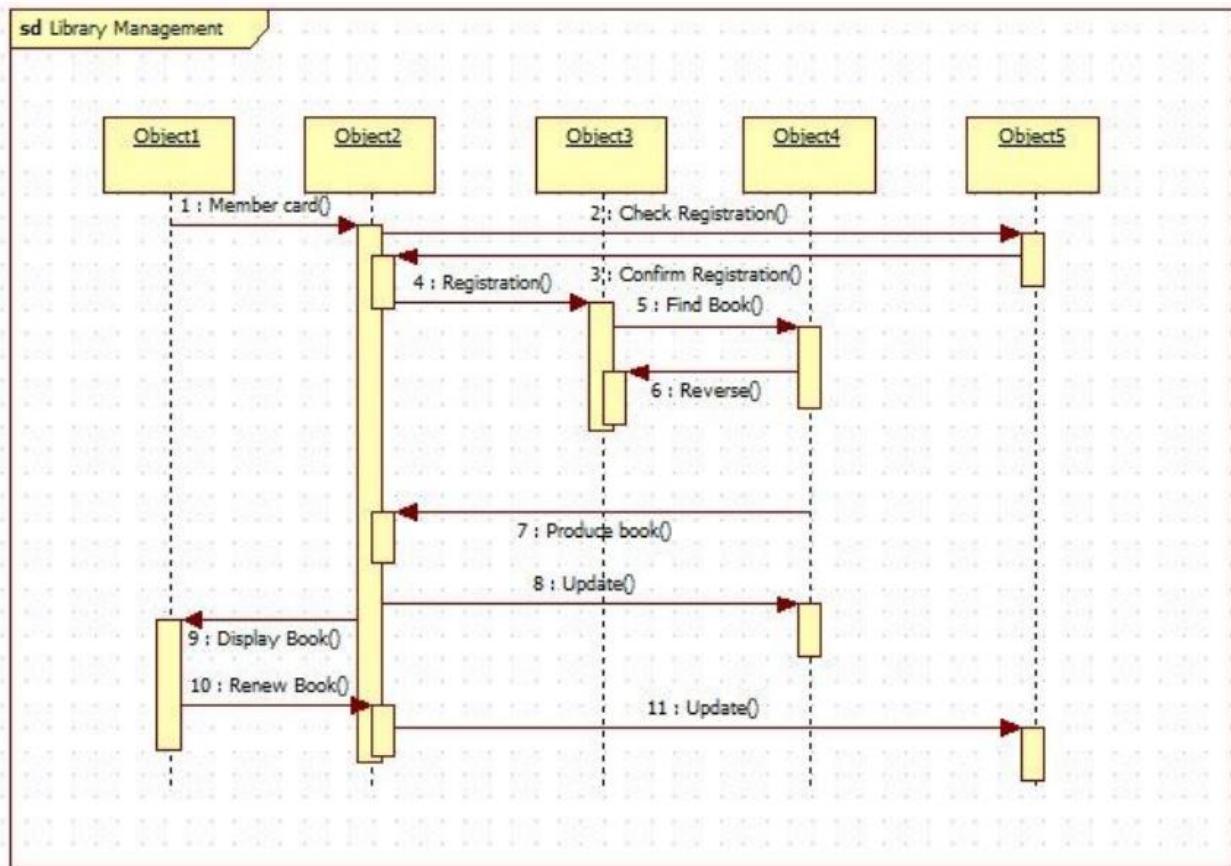


Fig 3.11 LMS advanced sequence diagram

This sequence diagram models typical interactions in library management. It shows how different objects handle member registration, book searching and finding, reservation, book production, updating records, displaying book info, and renewing a book, outlining key function calls and message flows.

Activity Diagrams (Simple and Advanced):

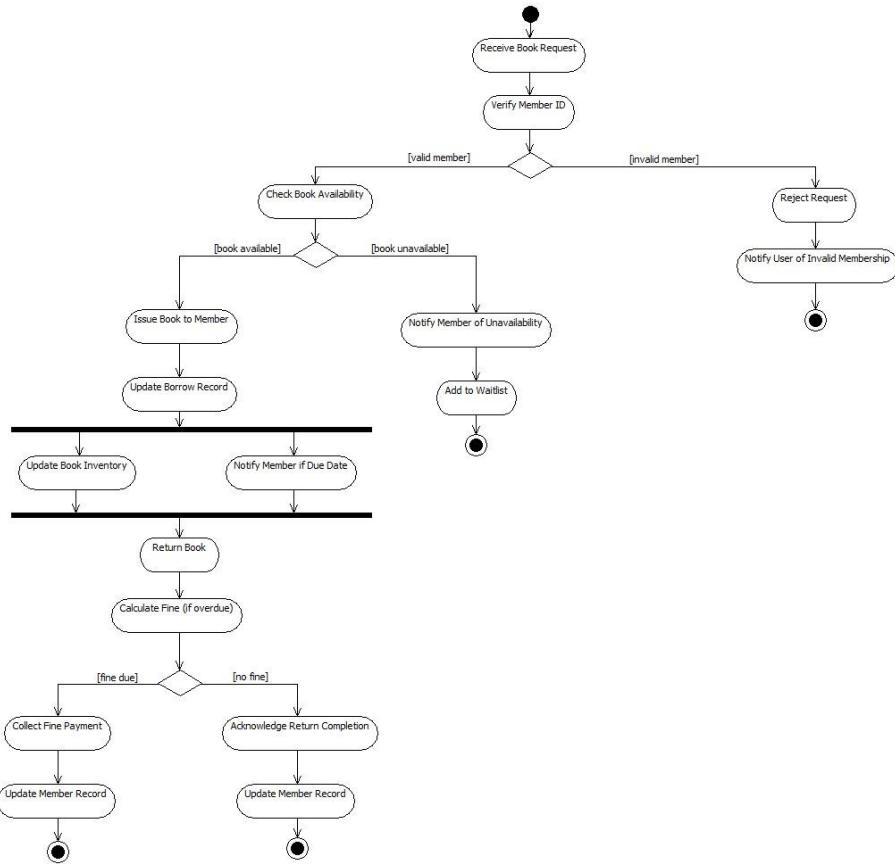


Fig 3.12 LMS simple activity diagram

This activity diagram describes the workflow for issuing and returning books in a library. It starts with verifying a member's ID, checks book availability, handles borrowing and waitlisting, manages returns, and includes steps for overdue fines and updating records depending on different scenarios.

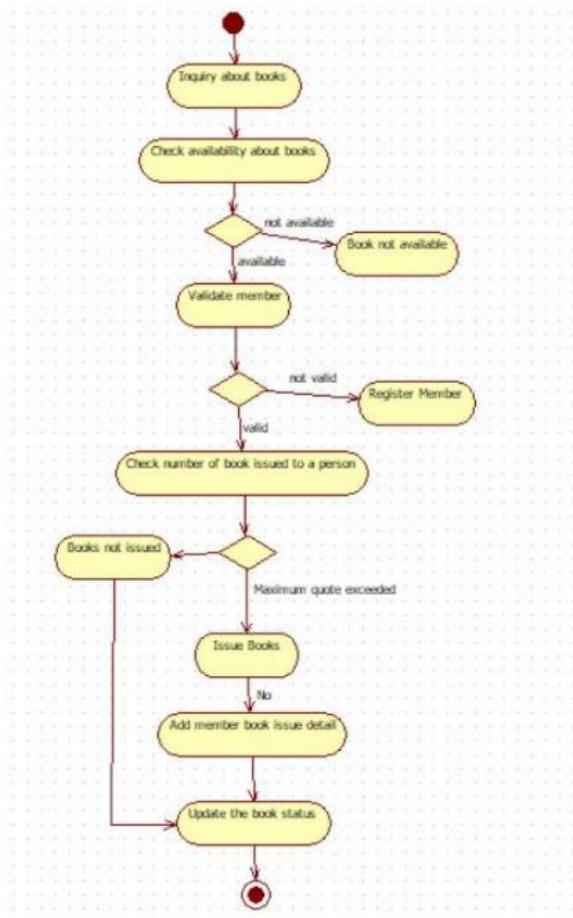


Fig 3.13 LMS advanced activity diagram

This activity diagram illustrates the library's book issuing workflow. It covers steps from book inquiry and availability check, to member validation, enforcing issue limits, registering new members if needed, issuing the book, recording the issue, and updating book status—all while handling exceptions for unavailability and membership issues.

Chapter 4: Stock Maintenance System

Problem statement and SRS:

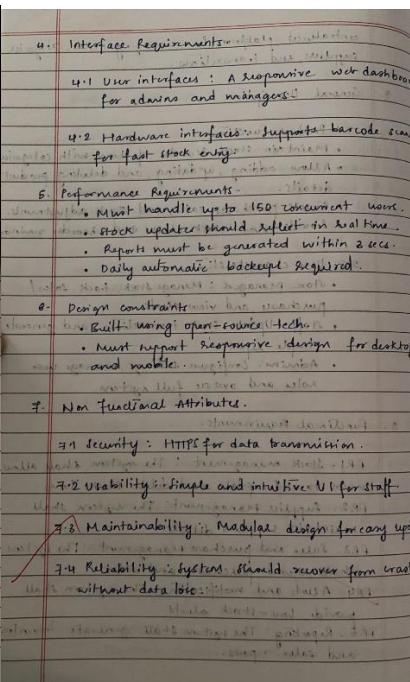
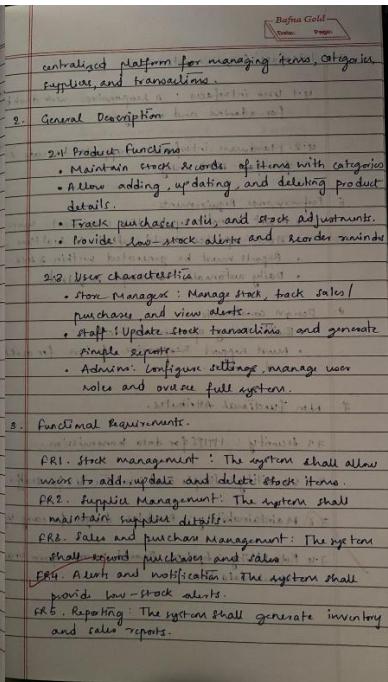
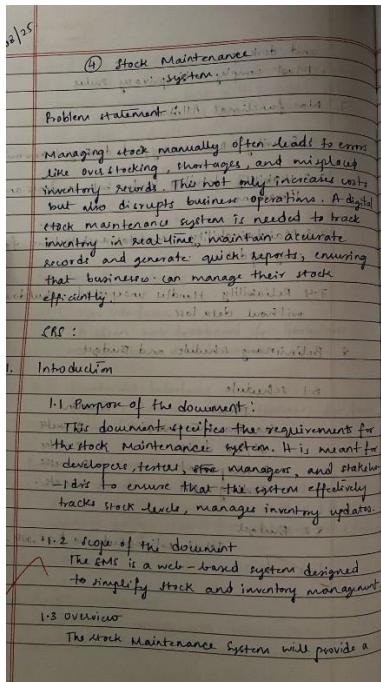


Fig 4.1

Fig 4.2

Fig 4.3

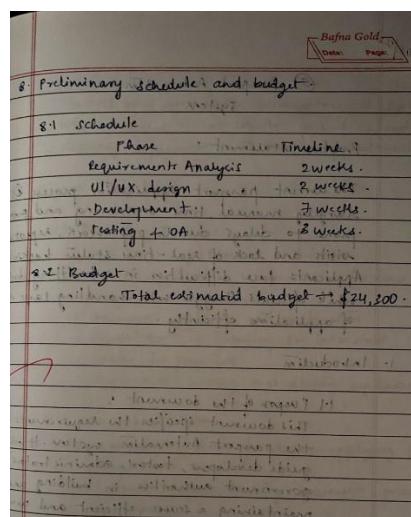


Fig 4.4

Class Diagram:

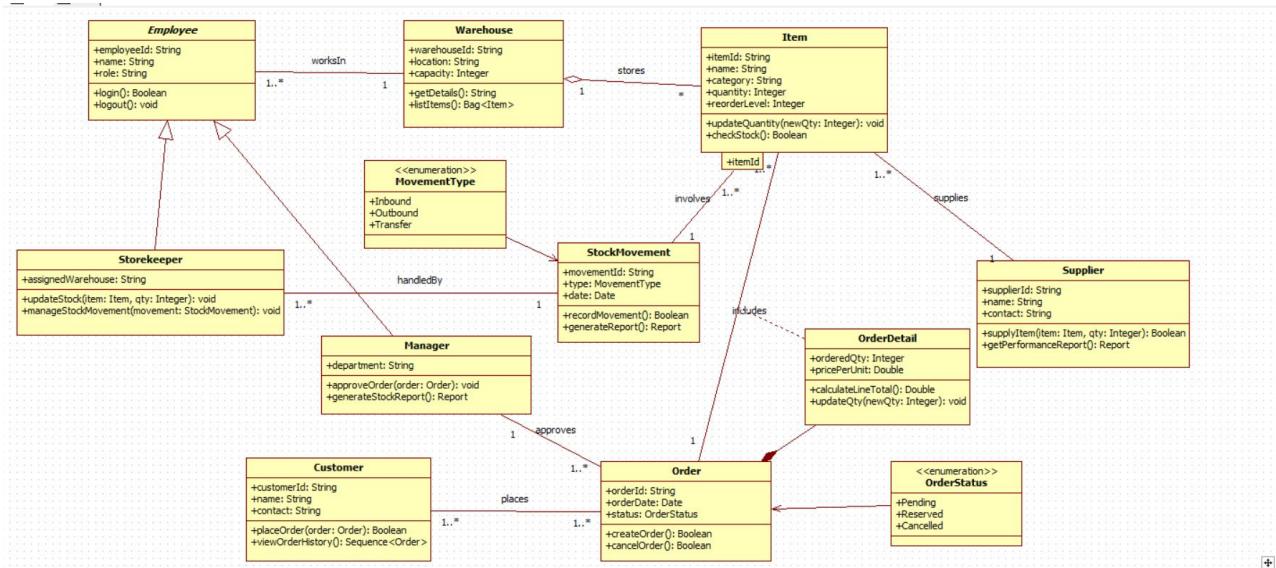


Fig 4.5 SMS class diagram

This UML class diagram represents an inventory and order management system. It models entities like employees, warehouses, items, suppliers, customers, orders, and stock movements, and shows how stock is managed, orders are placed and approved, and inventory changes are tracked through various activities and roles.

State Diagrams (Simple and Advanced):

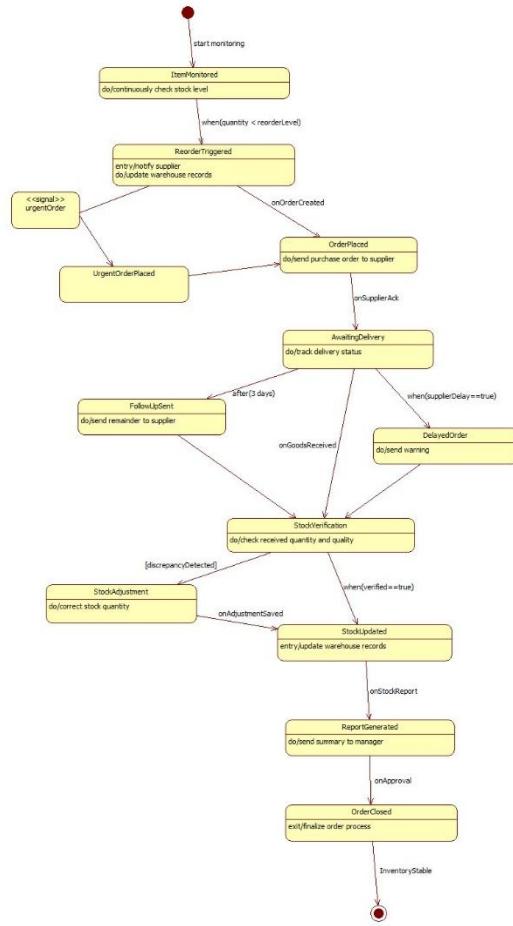


Fig 4.6 SMS simple state diagram

This state diagram illustrates the workflow of inventory monitoring and restocking in a warehouse system. It includes continuous stock level checks, order creation and delivery tracking, urgent order response, stock verification and adjustment, supplier coordination, and final inventory stabilization after discrepancies are handled or goods are approved.

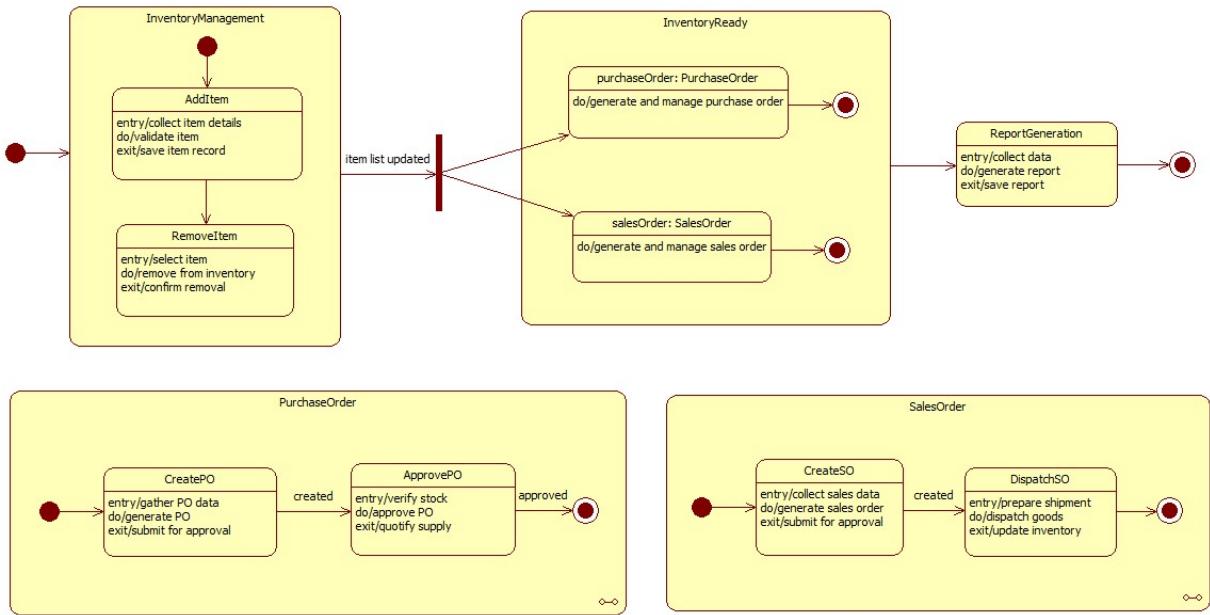


Fig 4.7 SMS advanced state diagram

This state diagram models inventory and order processing in a supply chain system. It shows how items are added or removed from inventory, sales and purchase orders are created and approved, and inventory levels are updated. The diagram also includes the generation of reports, reflecting the end-to-end flow from inventory changes to transactions and reporting.

Use Case Diagrams (Simple and Advanced):

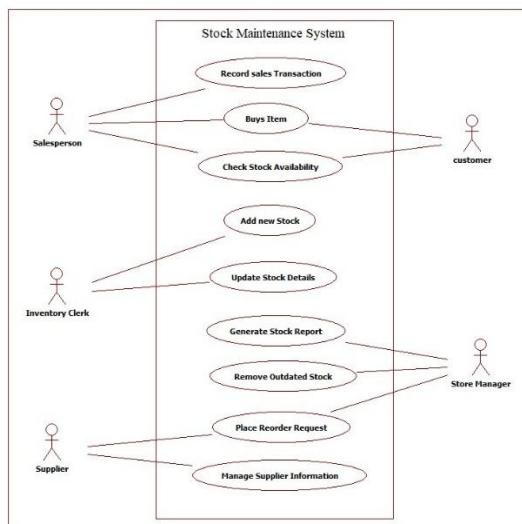


Fig 4.8 SMS simple use case

This use case diagram shows the main operations of a stock maintenance system. It illustrates how different actors—such as the salesperson, customer, inventory clerk, supplier, and store manager—interact with system functions like buying items, adding or updating stock, checking stock availability, managing supplier information, placing reorder requests, and generating stock reports.

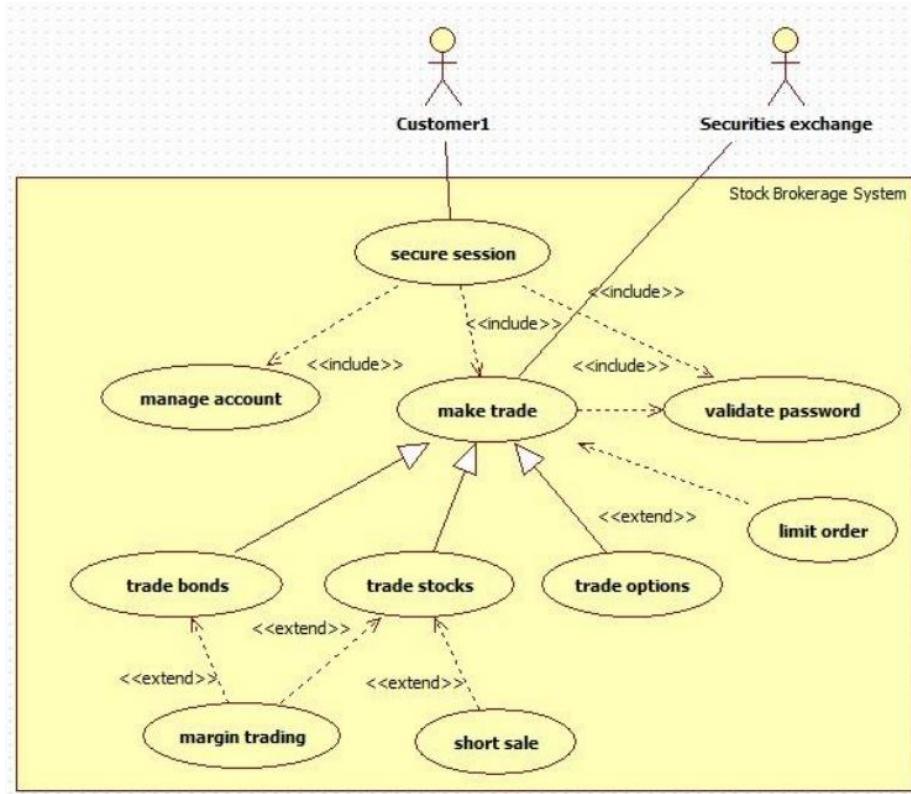


Fig 4.9 advanced use case

This use case diagram models a stock brokerage system, focusing on trading functionalities for a customer. It shows how the customer manages accounts, makes trades (stocks, bonds, options), secures sessions, and validates passwords. The diagram also includes advanced actions such as limit orders, margin trading, and short sales, with system interactions involving the securities exchange.

Sequence Diagrams (Simple and Advanced):

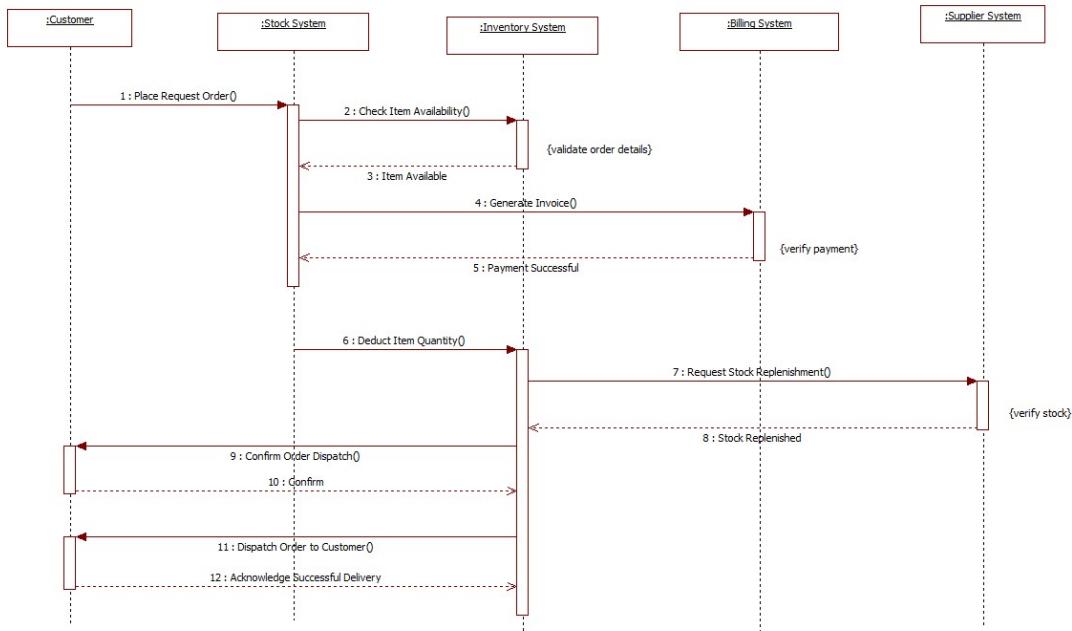


Fig 4.10 SMS simple sequence diagram

This sequence diagram illustrates the order processing workflow in an inventory system. It shows how a customer places an order, how stock and item availability are checked, payments processed, dispatch and delivery are managed, and any necessary replenishment requests are sent to the supplier to restock items.

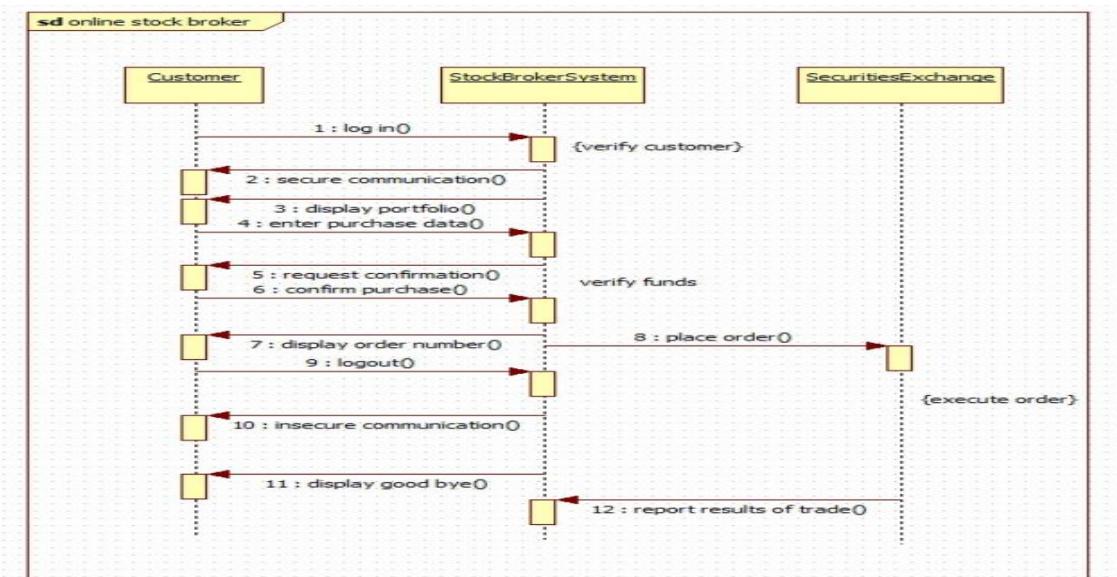


Fig 4.11 SMS advanced sequence diagram

This sequence diagram outlines how an online stock brokering system operates. It shows a customer logging in, securely communicating with the stockbroker system, performing trades, and confirming purchases. The system interacts with the securities exchange to execute orders, and finally reports trade results and manages session termination and logout.

Activity Diagrams (Simple and Advanced):

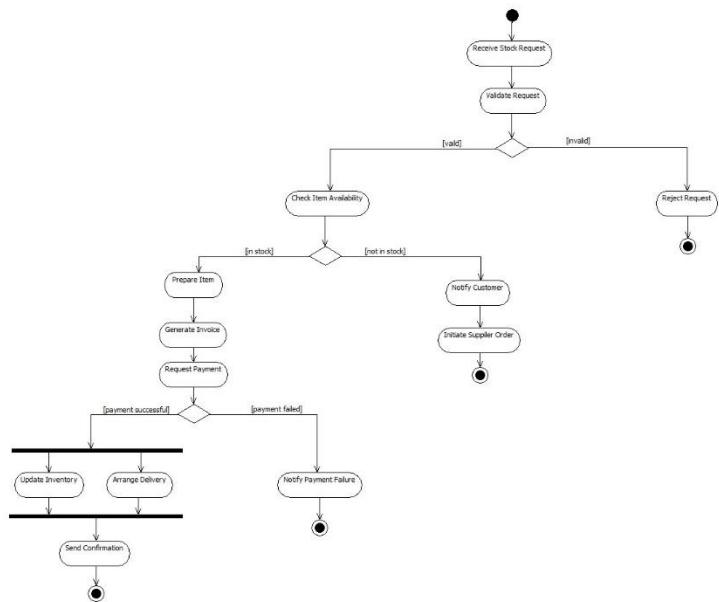


Fig 4.12 SMS simple activity diagram

This activity diagram shows the workflow for handling a stock request in an inventory system. It covers request validation, item availability checks, item preparation, invoicing, payment handling, updating inventory, arranging delivery, failure notifications, and, if necessary, supplier ordering, making the process clear from request to fulfillment or exception.

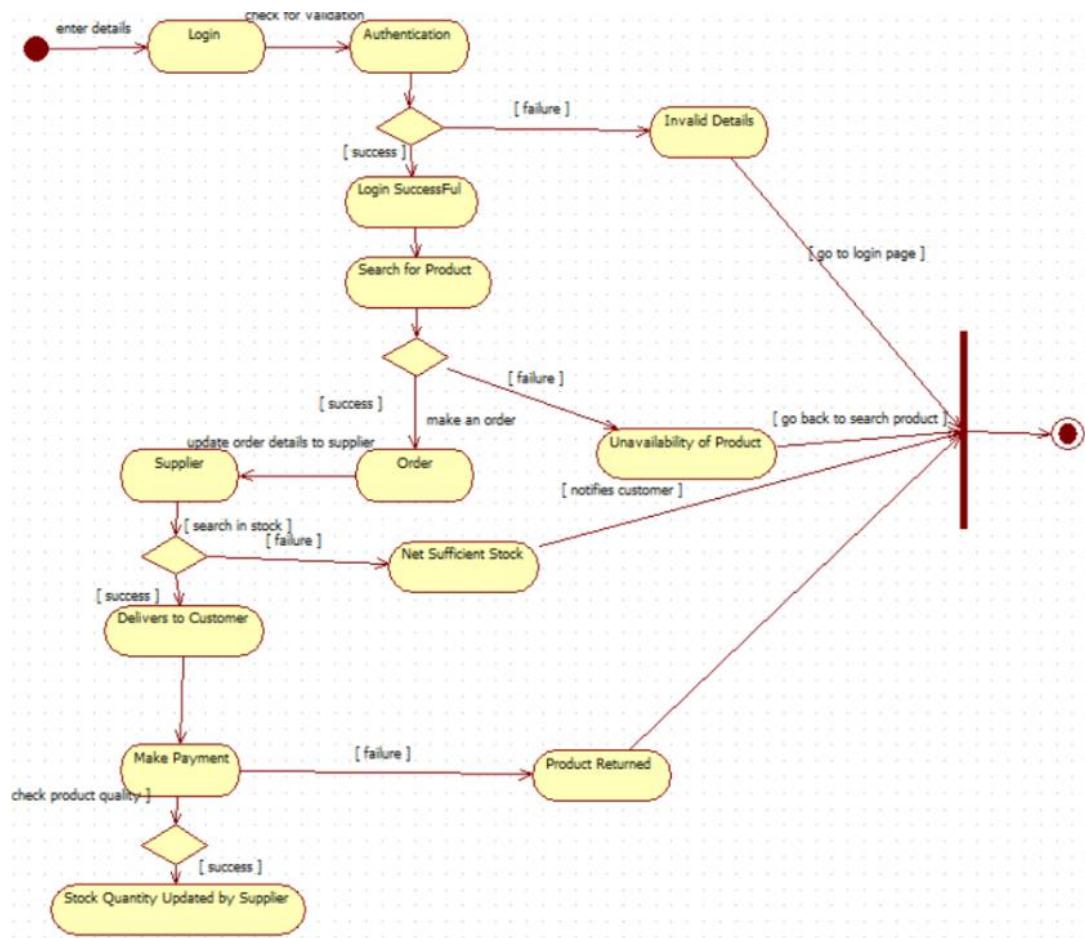


Fig 4.13 SMS advanced use case diagram

This activity diagram shows the flow for a customer placing an order in an inventory or sales system. It starts from login and authentication, proceeds through product search, stock checking, order placement, supplier coordination, delivery, payment, and stock update, and also handles failures or product returns.

Chapter 5: Passport Automation System

Problem statement and SRS:

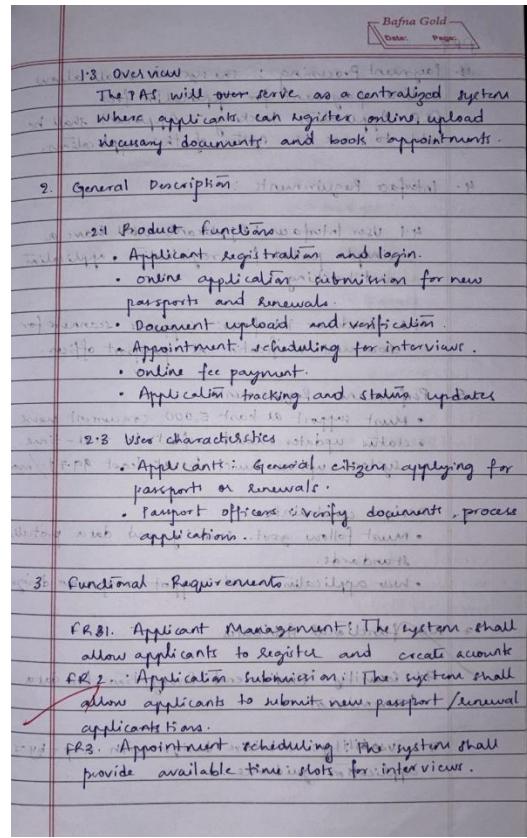
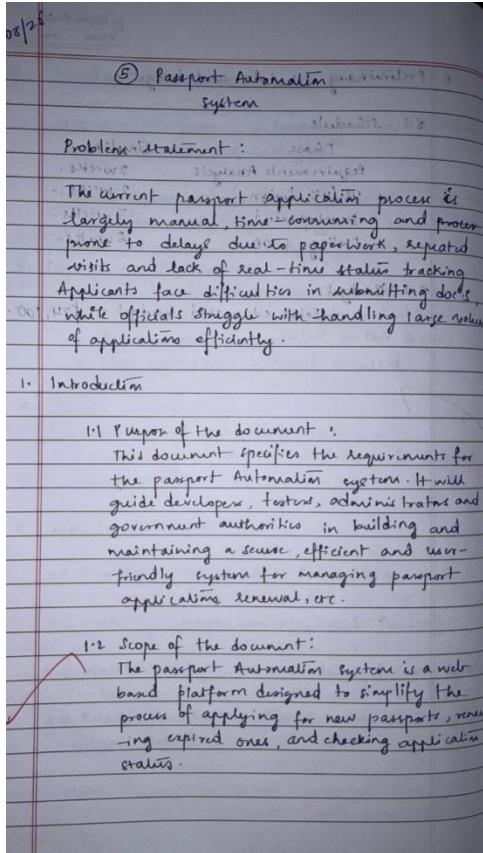


Fig 5.1

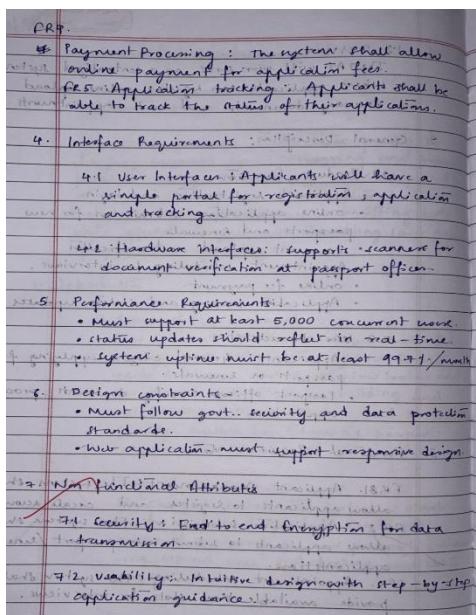


Fig 5.3

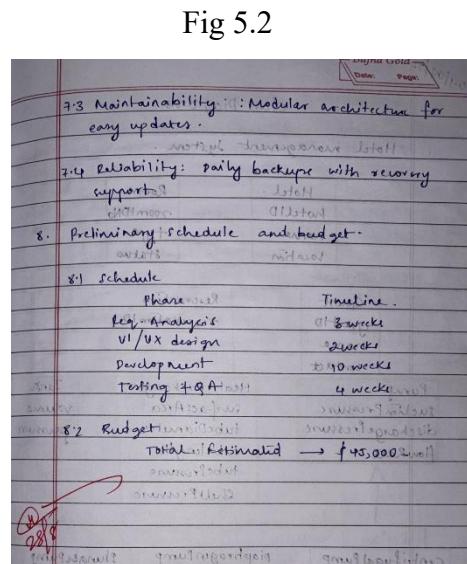


Fig 5.4

Class Diagrams:

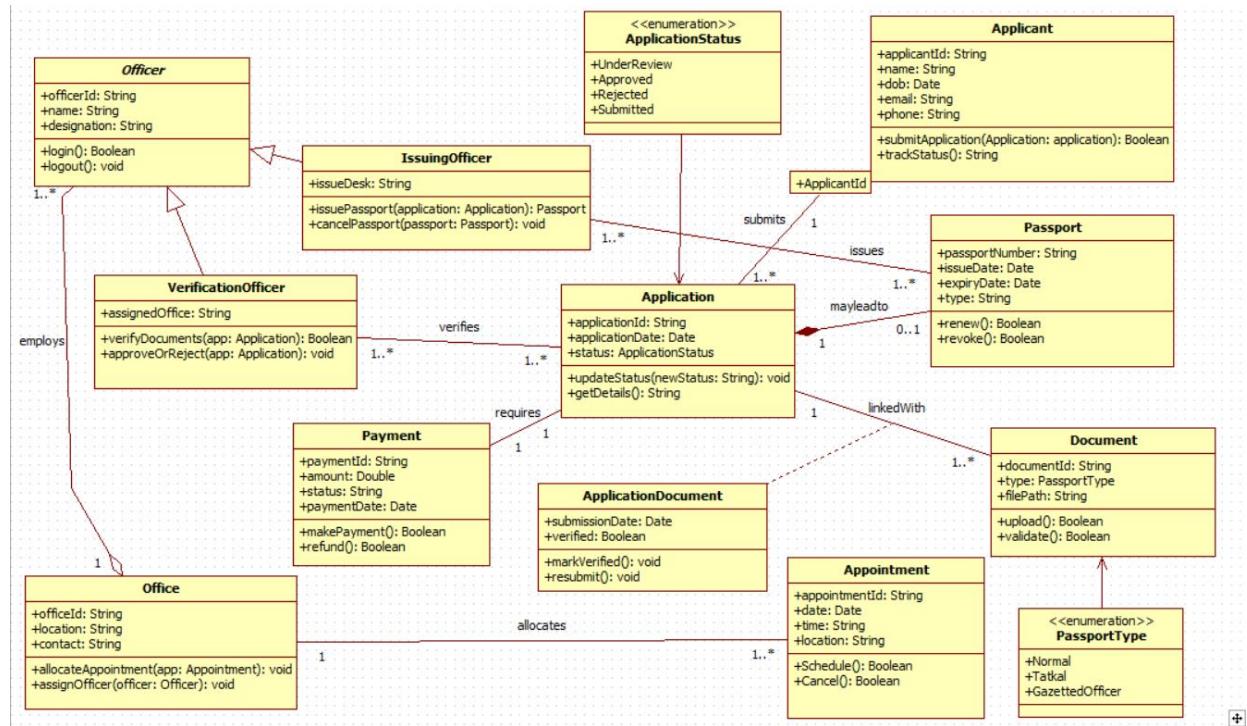


Fig 5.5 PAS class diagram

This UML class diagram represents a passport application management system. It defines entities such as applicant, officer, application, passport, appointment, document, payment, and the relationships between them. The diagram shows how applications are submitted, verified, approved, and issued, including supporting processes like payments, document management, scheduling, and status tracking.

State Diagrams (Simple and advanced):

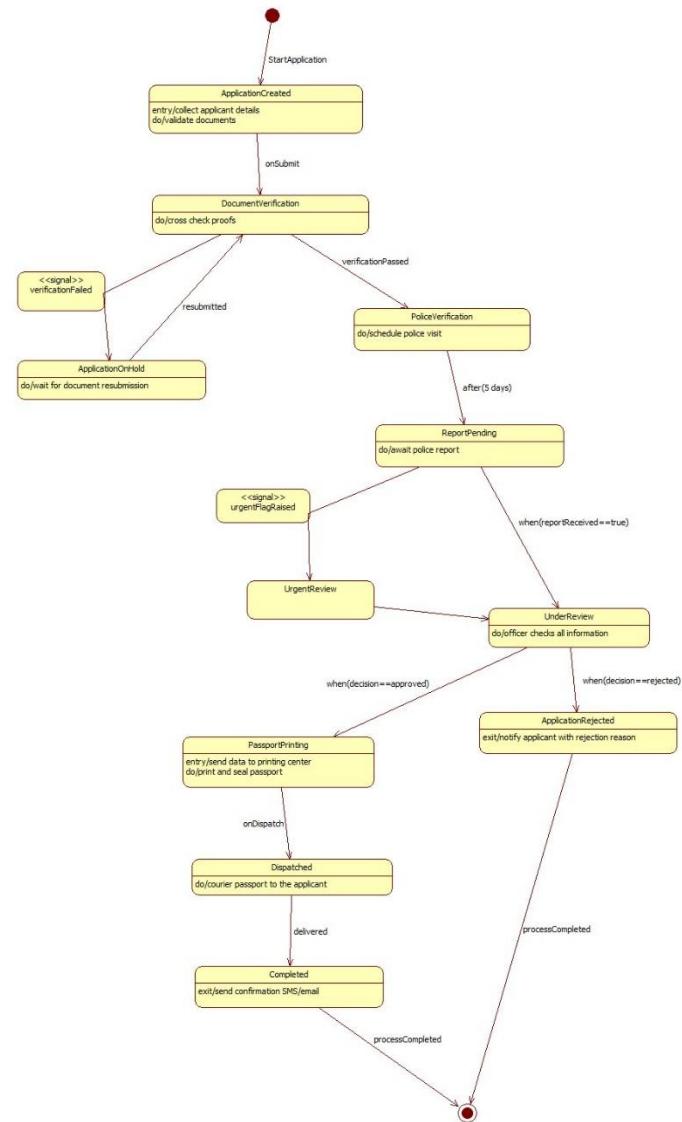


Fig 5.6 PAS simple state diagram

This state diagram represents the passport application process. It follows stages from application creation and document verification to police checks, urgent handling, review, approval or rejection, passport printing, dispatch, and final completion, including paths for resubmission if verification fails.

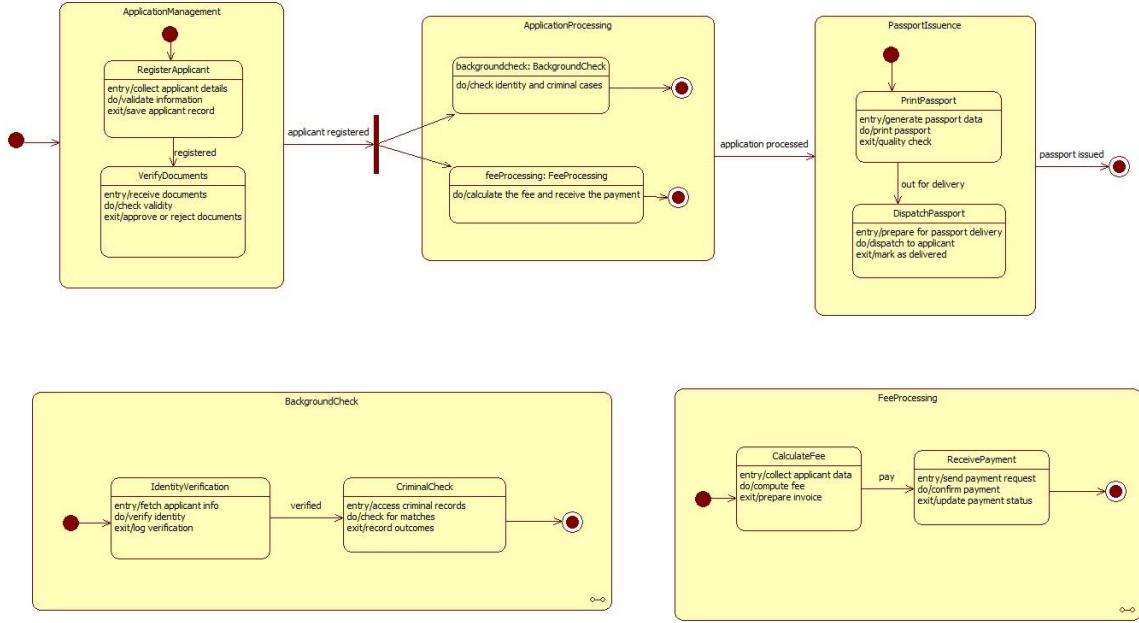


Fig 5.7 PAS advanced state diagram

This diagram breaks down the passport processing system into clear stages: application management, verification (including background and fee checks), app processing, and final passport issuance. Steps include applicant registration, identity and criminal checks, payment, printing the passport, and dispatching it for delivery.

Use Case Diagram (Simple and Advanced):

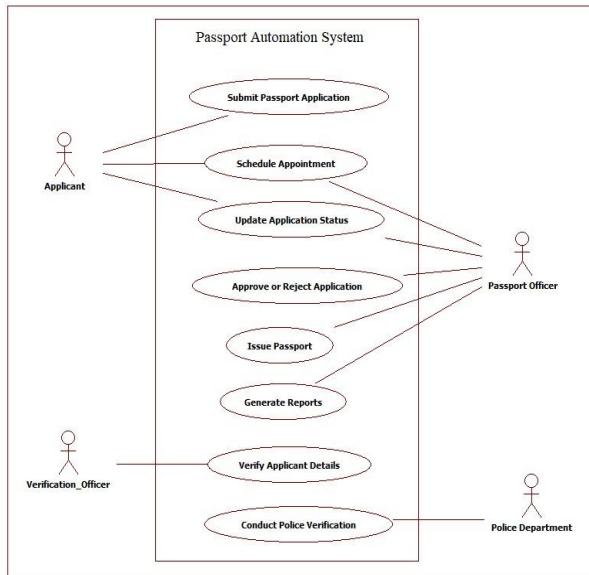


Fig 5.8 PAS simple use case

This use case diagram models the passport application system, showing how applicants submit applications, schedule appointments, upload documents, and interact with police and passport officers. The system administrator manages user and officer accounts, databases, and backups, while the passport officer and printing authority handle verification, issuing, and reporting processes.

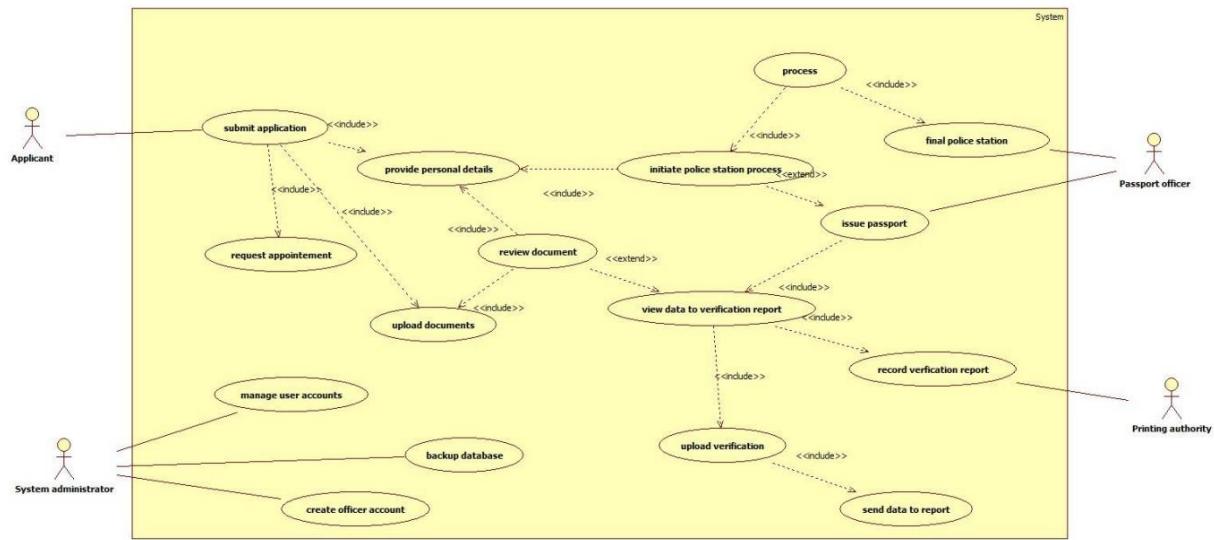


Fig 5.8 PAS advanced use case diagram

This use case diagram models the passport application system, showing how applicants submit applications, schedule appointments, upload documents, and interact with police and passport officers. The system administrator manages user and officer accounts, databases, and backups, while the passport officer and printing authority handle verification, issuing, and reporting processes.

Sequence Diagram (Simple and advanced):

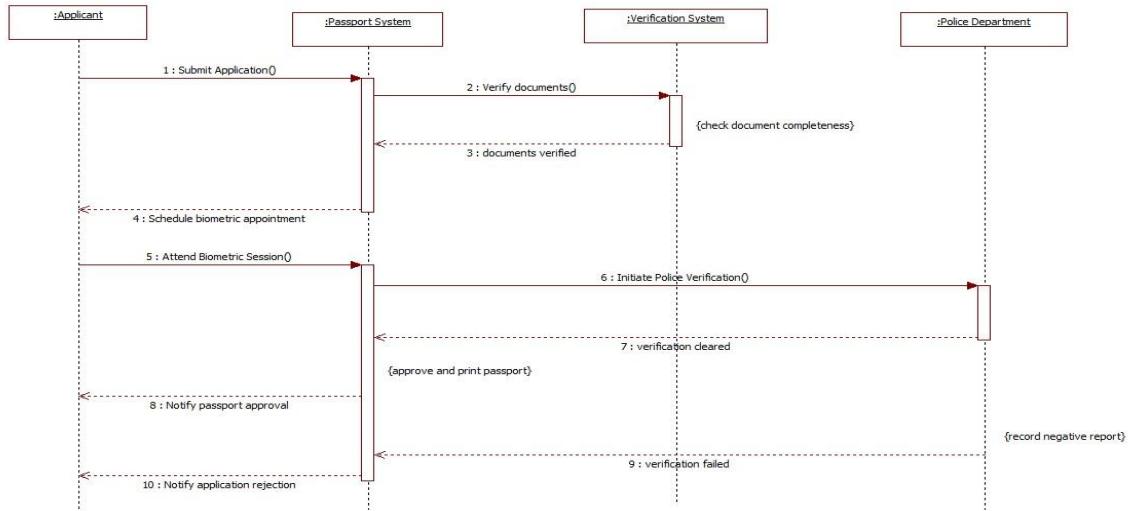


Fig 5.9 PAS simple sequence diagram

This sequence diagram shows the major steps in a passport application process. The applicant submits their application and attends a biometric session, while the passport system coordinates document and police verification, and communicates approval or rejection back to the applicant based on verification results.

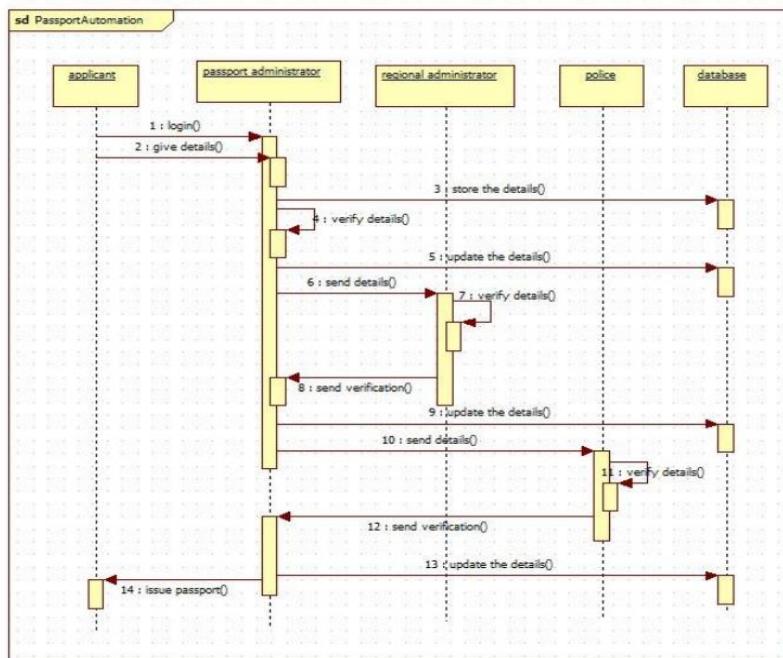


Fig 5.10 PAS advanced sequence diagram

Activity Diagrams (Simple and advanced):

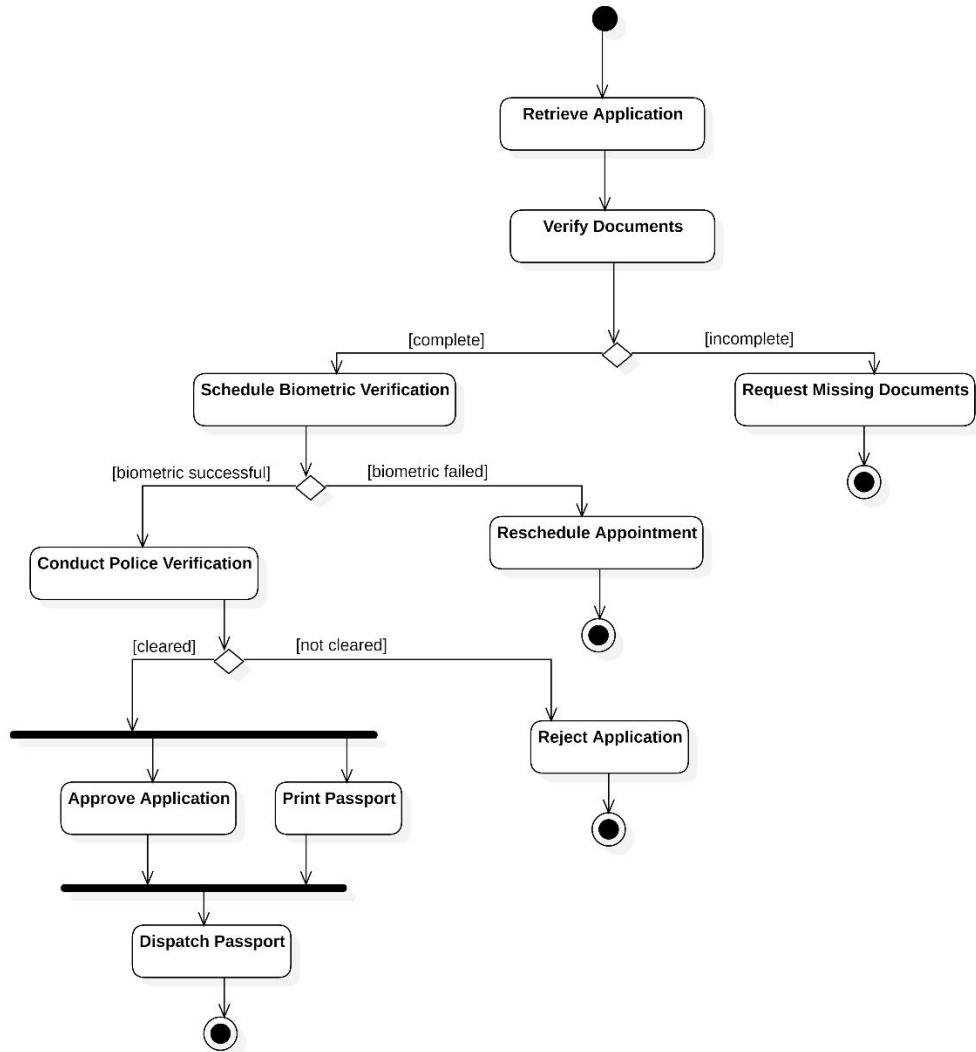


Fig 5.11 PAS simple activity diagram

This activity diagram outlines the key steps of a typical passport application process. It shows document verification, scheduling biometrics, police verification, approval or rejection, passport printing, and dispatch, including alternative paths if documents are missing or verification fails.

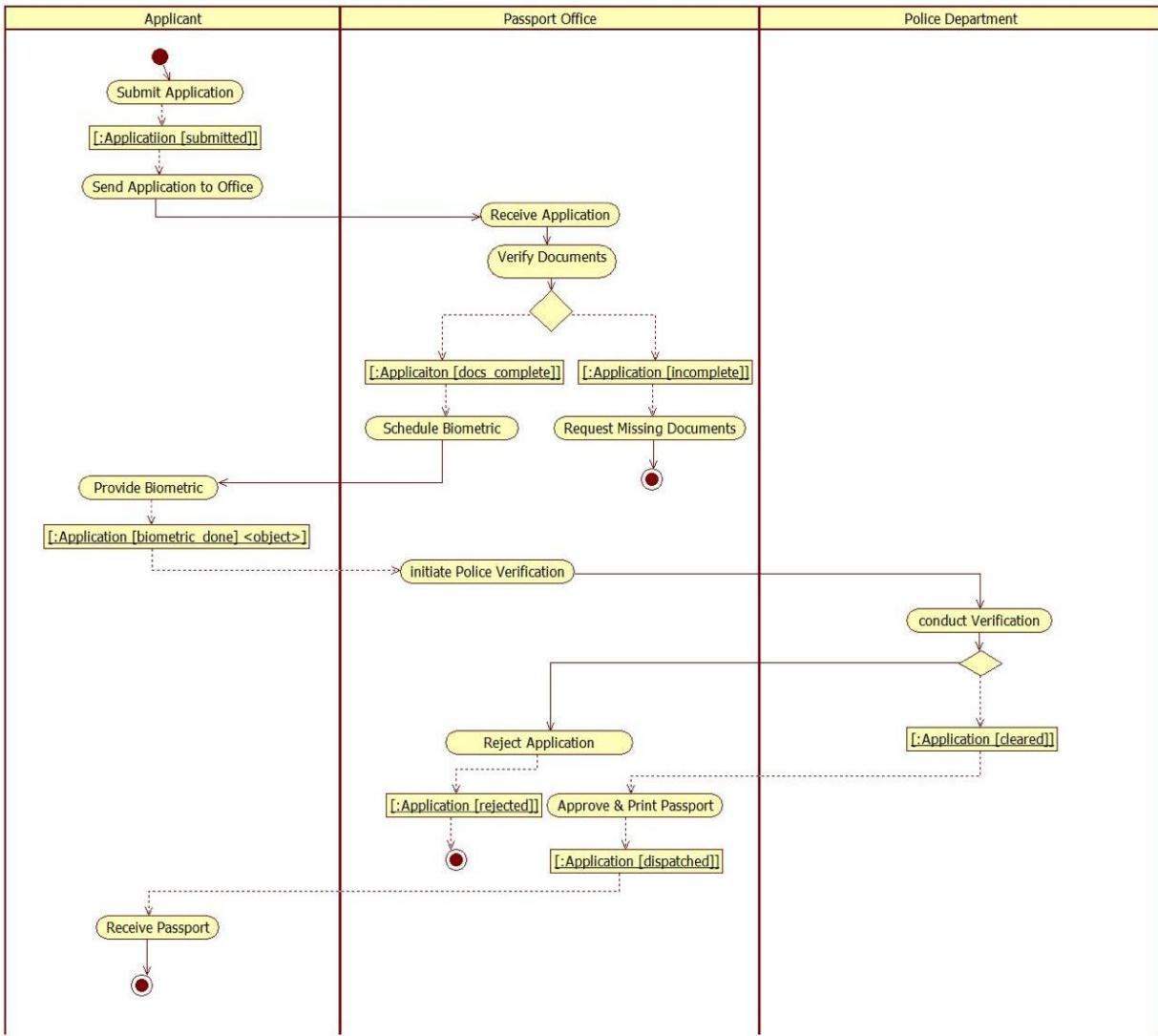


Fig 5.12 PAS advanced Activity diagram