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:

CHAITHANYA SUDHAN

1BM23CS073

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 Chaithanya Sudhan (1BM23CS073) during the 5th Semester August 2025-December 2025

.

Signature of the Faculty Incharge:

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

TABLE OF CONTENTS

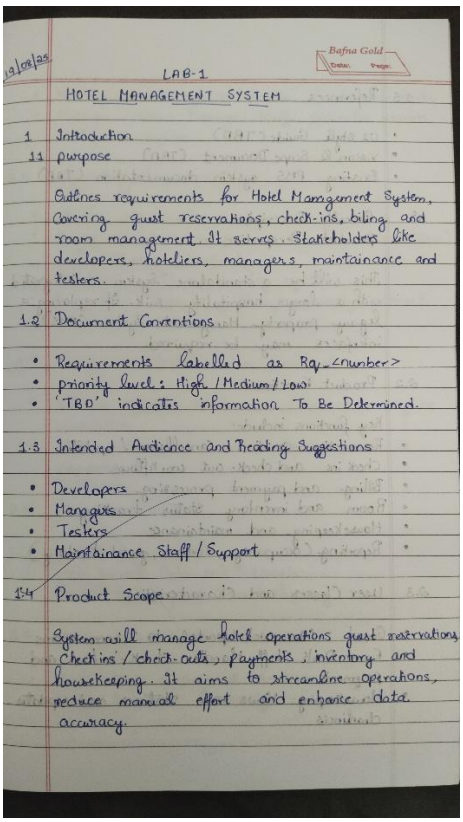
1. Hotel Management System

Problem Statement

Hotels manage many daily tasks like taking reservations, checking in guests, assigning rooms, handling payments, and keeping track of housekeeping. When these tasks are done manually, mistakes can happen easily such as double bookings, delays during check-in, confusion about room availability, or missing records. This makes the work stressful for staff and inconvenient for guests.

To make these processes smoother, a Hotel Management System is needed. This system will help automate the main hotel operations, show real-time room status, manage bookings and payments, support housekeeping updates, and allow managers to view useful reports. The goal is to reduce manual effort, avoid errors, and make the hotel experience easier for both staff and guests.

SRS-Software Requirements Specification



13/08/25

LAB-1

HOTEL MANAGEMENT SYSTEM

1 Introduction

1.1 purpose

Outlines requirements for Hotel Management System, covering guest reservations, check-ins, biling and room management. It serves stakeholders like developers, hoteliers, managers, maintainance and testers.

1.2 Document Conventions

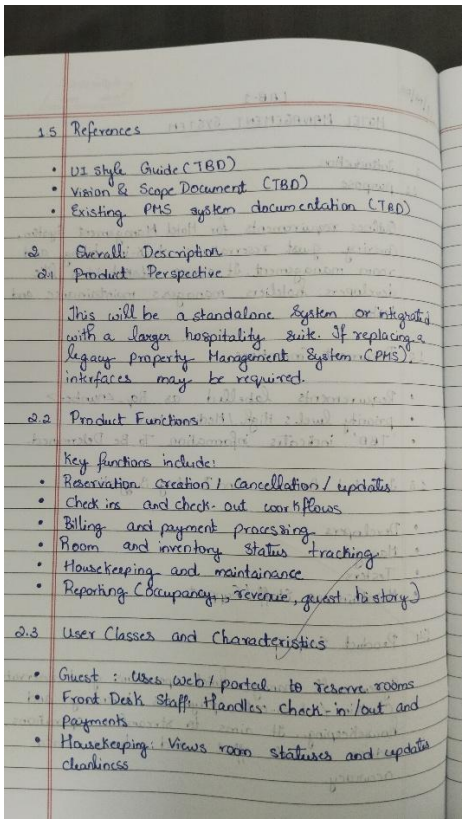
- Requirements labelled as Rq_<number>
- priority level : High / Medium / Low
- 'TBD' indicates information To Be Determined.

1.3 Intended Audience and Reading Suggestions

- Developers
- Managers
- Testers
- Maintainance Staff / Support

1.4 Product Scope

System will manage hotel operations guest reservations Check ins / check-outs, payments, inventory and housekeeping. It aims to streamline operations, reduce manual effort and enhance data accuracy.



1.5 References

- UI style Guide (TBD)
- Vision & Scope Document (TBD)
- Existing PMS system documentation (TBD)

2 Overall Description

2.1 Product Perspective

This will be a standalone System or integrated with a larger hospitality suite. If replacing a legacy property Management System (PMS), interfaces may be required.

2.2 Product Functions

Key functions include:

- Reservation creation / cancellation / updates
- Check ins and check-out workflows
- Billing and payment processing
- Room and inventory status tracking
- Housekeeping and maintainance
- Reporting (Occupancy, revenue, guest history)

2.3 User Classes and Characteristics

- Guest : Uses web / portal to reserve rooms
- Front Desk Staff: Handles check-in / out and payments
- Housekeeping : Views room statuses and updates cleanliness

- Maintainance: Logs and tracks maintainance tasks
- Manage / Admin: Manages staff, generates reports, configures system.

2.4 Operating Environment

- Client: web-based, responsive design
- Server: Cloud or on-premise server (Windows / Linux) with database backend
- Database: SQL-based (eg: MySQL)
- Integrations: Payment gateways possibly external PMS or OTA APIs

2.5 Design and Implementation Constraints

- Use of specific database or framework may be mandated.
- Multi-platform browser support; responsive UI required

2.6 User Documentation

- User Manual (PDF)
- Web-based Help and Tutorials
- Admin Guide and Troubleshooting

2.7 Assumptions and Dependancies

- Availability of payment services
- Stable Internet Connectivity

3 External Interface Requirements

3.1 User Interfaces

- Responsive web-based UI
- Standard layouts with header / footer and navigation bar

3.2 Hardware Interfaces

- Reciept printer for invoices
- POS terminal integration
- optionally, card swipe devices

3.3 Software Interfaces

- payement gateway APIs (PayU)
- Database system (MySQL)
- Potential PMS

3.4 Communication Interfaces

- HTTPS for secure web communication
- Optional email server integration for automated guest notifications.

4 System Features

4.1 ~~System Features~~ : Reservation Management

- priority: high
- 4.1.1 Description and Priority: Enables booking rooms online or in person.
- 4.1.2 Stimulus / Response Sequence: Guest selects date → system shows availability → guest books → confirmation sent

- 4.1.3 Functional Requirement
  - REQ-1: System shows room availability by date and room type.
  - REQ-2: Guest can create / edit / cancel reservation
  - REQ-3: Confirmation emails sent upon booking

4.2 Billing & Payment

- priority: High
- Functional Requirements:
  - ~~REQ-4: Staff can initiate check-in, as~~ Multiple payment methods supported
  - REQ-5: option to split payment across methods.
  - REQ-6: Secure handling of sensitive data

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- system responds to the action within 2 seconds
- Capable of handling 200 concurrent users

5.2 Safety Requirements

- Prevent double booking
- Ensure data integrity during payment processes.

5.3 Security Requirements

- User authentication with role-based access
- Encryption

5.4 Software Quality Attributes

- Usability: Intuitive UI with minimal user training
- Reliability: 99.9% uptime
- Scalability: Support increase in users / bookings.

5.5 Business Rules

- Only front desk staff can override booking details
- Checks before room assignment.
- Refund rules: full refund if cancelling X hours before check in.

6 Other Requirements

- Internationalization: support for multiple currencies
- Database Requirements: Encrypted backups and disaster recovery plan

Appendices

- OTA: Online Travel Agency
- PMS: property Management System.

Class Diagram

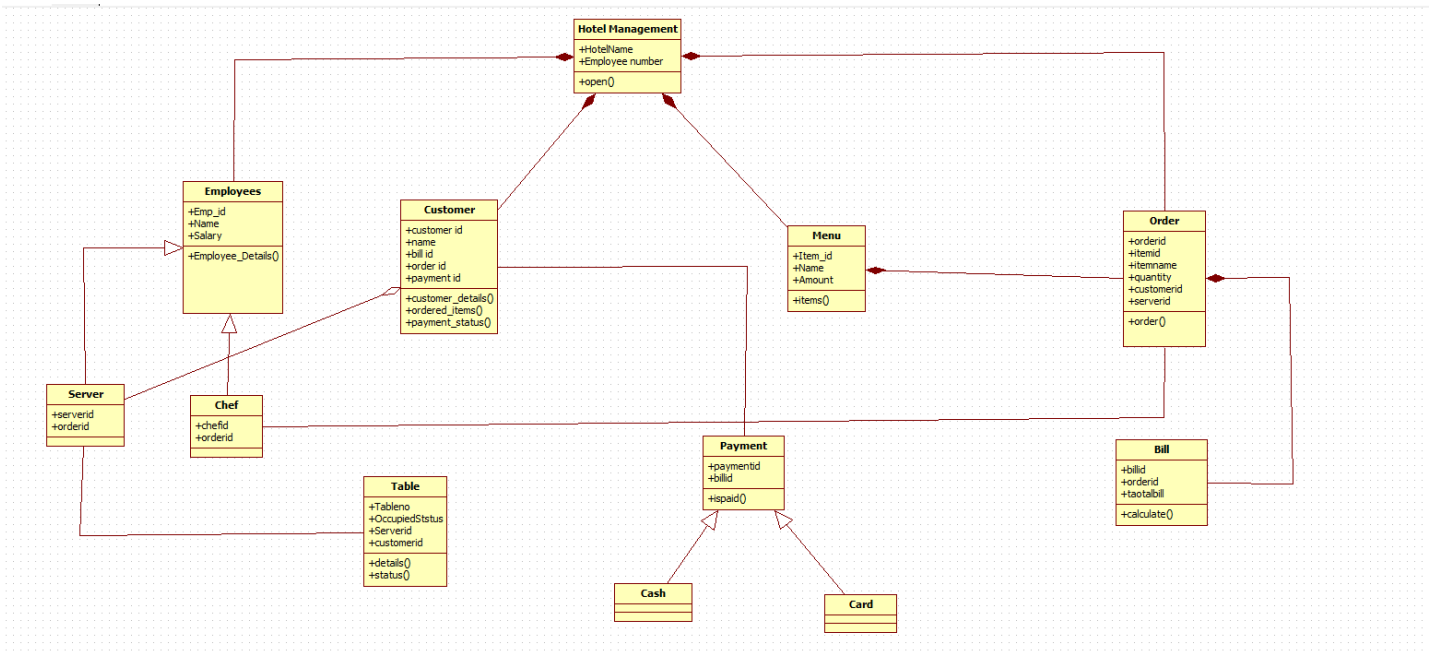


Fig 1.1 Advance Class diagram HMS

This advanced class diagram shows how different entities in the hotel system interact to support smooth hotel operations. At the center, the Hotel Management class connects to key components responsible for handling customers, employees, menu items, table allocation, orders, billing, and payments. Customers can place orders from the menu, and each order includes details such as item name, quantity, and the staff member serving it. Orders generate bills, which are then processed through the Payment class using either cash or card. The diagram also represents how employee roles such as servers and chefs link to specific orders, while tables maintain information about occupancy and assigned customers. Overall, the diagram provides a structured view of how the hotel coordinates staff activities, customer interactions, food ordering, billing, and payment processing in an integrated manner.

State Diagram

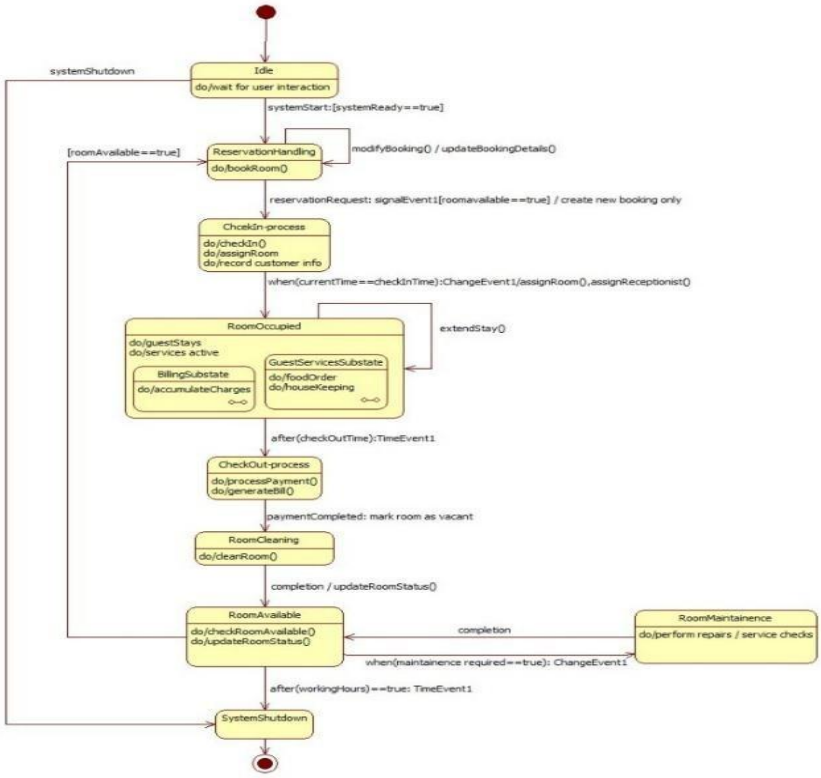


Fig 1.2 Simple State Diagram HMS

The diagram shows the full journey of a hotel room as it moves through different states during a guest's stay. It starts in an idle state, waiting for user interaction, and then moves into reservation handling when a room is available. After booking, the system enters the check-in process, records customer details, and transitions the room into the occupied state where services like food orders, housekeeping, and billing continue. Once the guest checks out, the system processes payment, marks the room as vacant, and sends it for cleaning. After cleaning, the room becomes available again unless maintenance is required, in which case it moves to the maintenance state before returning to availability. Finally, the system can shut down when operations end. The diagram gives a clear, step-by-step view of how a room is managed from booking to checkout in a hotel system.

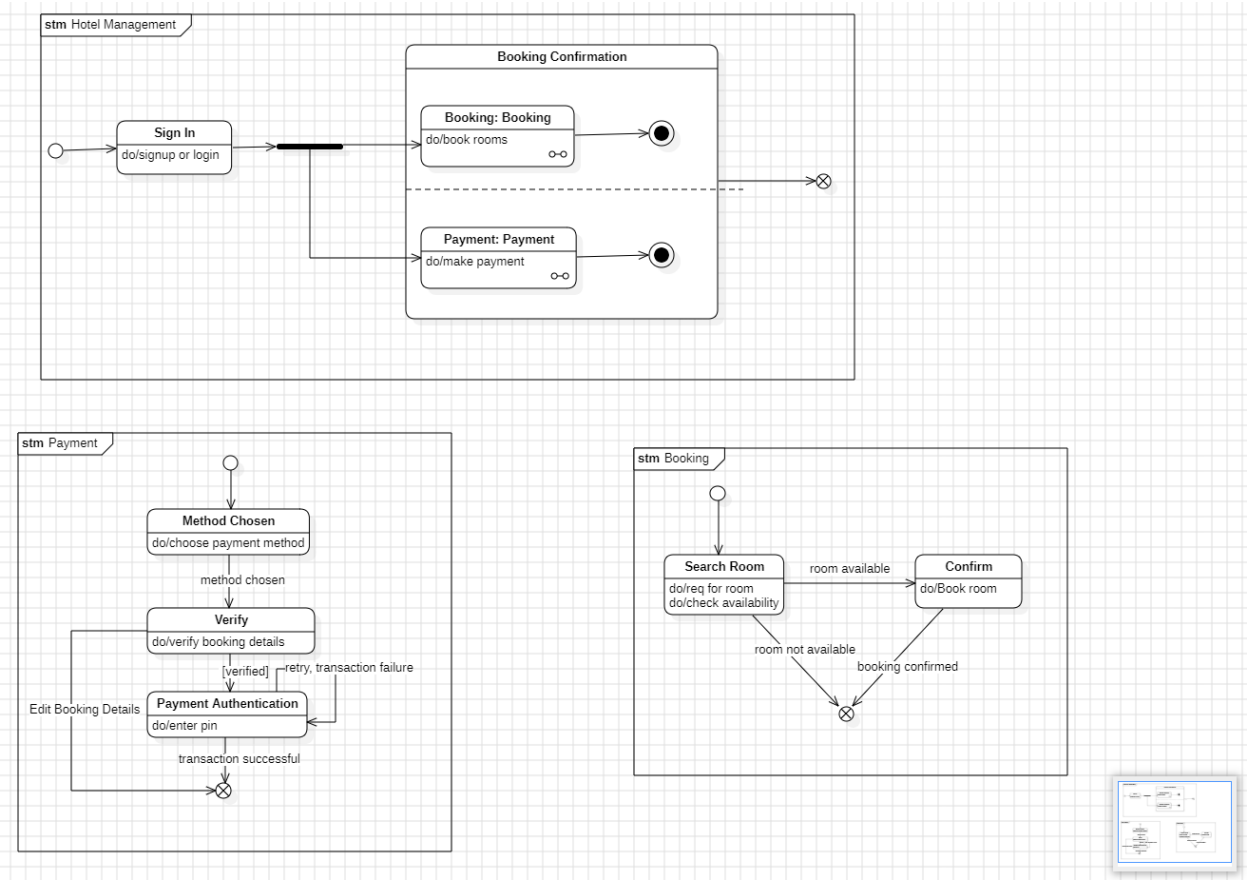


Fig1.3 Advance State diagram HMS

This advanced state diagram represents the complete flow of how a hotel booking is processed, showing multiple activities happening in parallel. After the user signs in, the system enters the booking confirmation phase, where two major processes Booking and Payment can run side by side. In the Booking region, the user searches for available rooms, and the system checks availability before allowing a final confirmation. At the same time, the Payment region handles choosing a payment method, verifying the booking details, and authenticating the transaction to ensure secure payments. Each state includes meaningful actions such as entering a PIN, checking inventory, or verifying user data. The diagram also shows alternate paths like retrying on payment failure or returning to edit booking details. Overall, it provides a detailed, structured view of how different parts of the booking and payment workflow interact simultaneously to complete a hotel reservation smoothly and securely.

Use Case Diagram

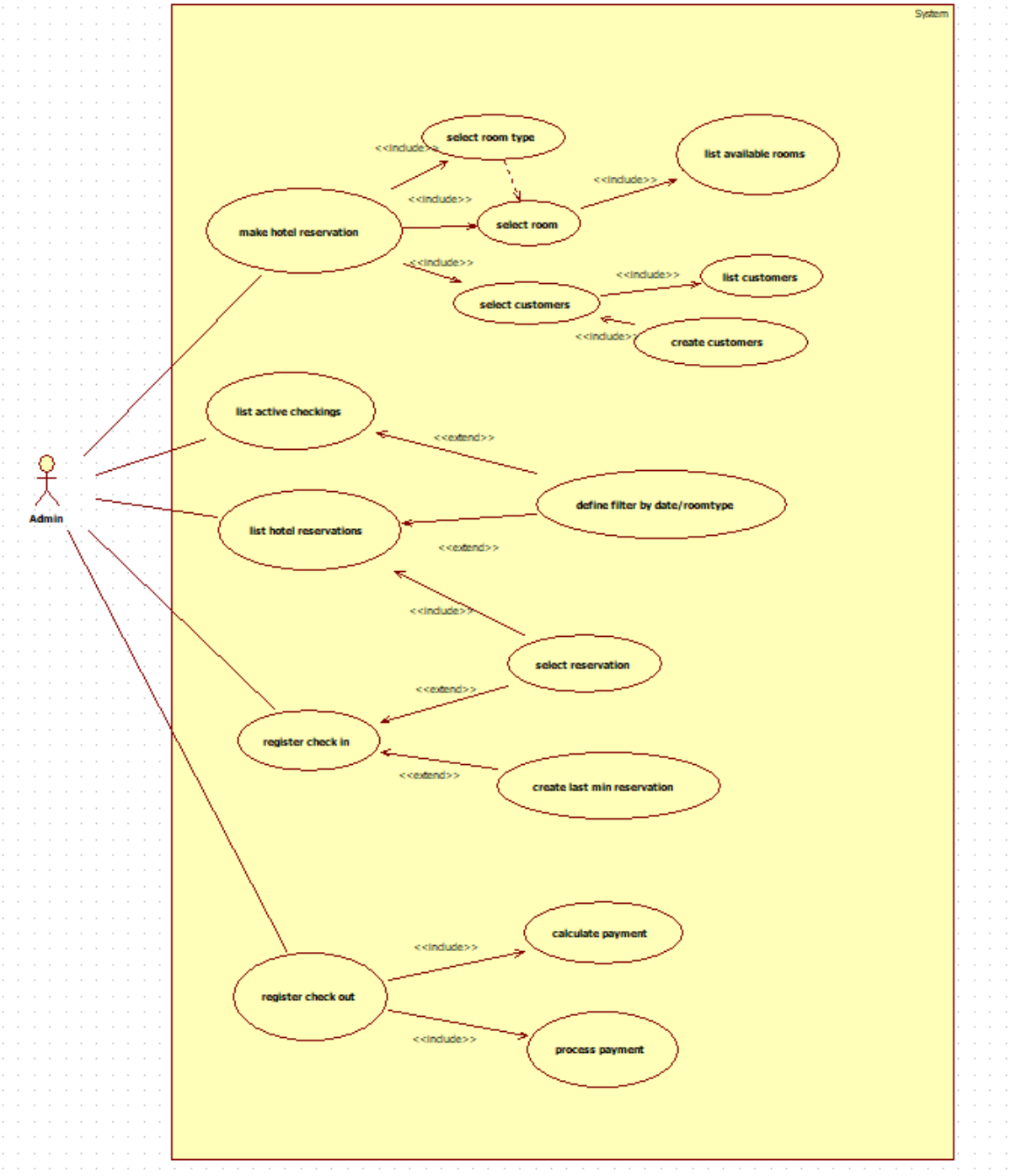


Fig 1.4 Use Case Diagram HMS

This advanced use case diagram shows how the Admin interacts with the hotel reservation system to manage all major operations. The Admin can make hotel reservations, list existing reservations, track active check-ins, and handle both check-ins and check-outs. Each main use case is supported by smaller included actions for example, making a reservation involves selecting the room type, checking available rooms, choosing a room, selecting or creating a customer, and finalizing the reservation details. When listing reservations, the Admin can extend the process by applying filters such as date or room type, or by selecting a specific reservation. During check-in, the system can create last-minute reservations if needed. The check-out process includes calculating the payment and completing the payment transaction. Overall, the diagram provides a complete view of how an Admin controls the reservation lifecycle, ensuring smooth management of bookings, customers, payments, and daily hotel operations.

Sequence Diagram

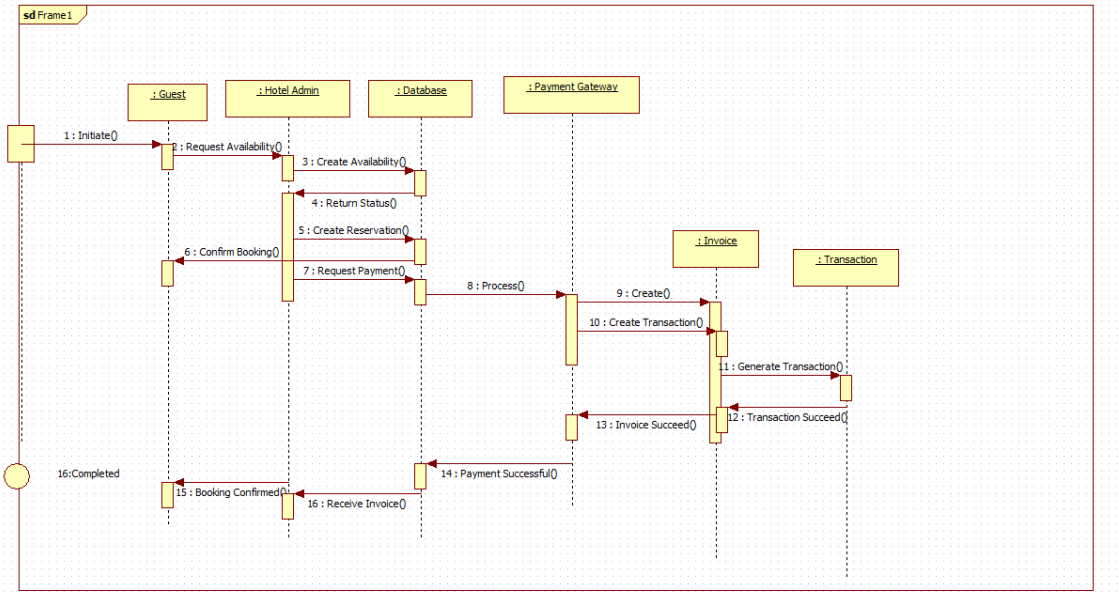


Fig 1.5 Sequence Diagram HMS

This advanced sequence diagram explains the complete flow of how a hotel booking request is handled inside the system. The guest begins by initiating a request for room availability, which the Hotel Admin forwards to the database. After the system returns the availability status, the admin proceeds to create the reservation and then triggers the payment process through the payment gateway. The payment gateway generates the invoice and records the transaction using the Invoice and Transaction components. Once the payment succeeds, the guest receives the invoice and the booking is officially confirmed. The diagram clearly shows how different system components communicate step-by-step to smoothly complete a hotel reservation.

Activity Diagram

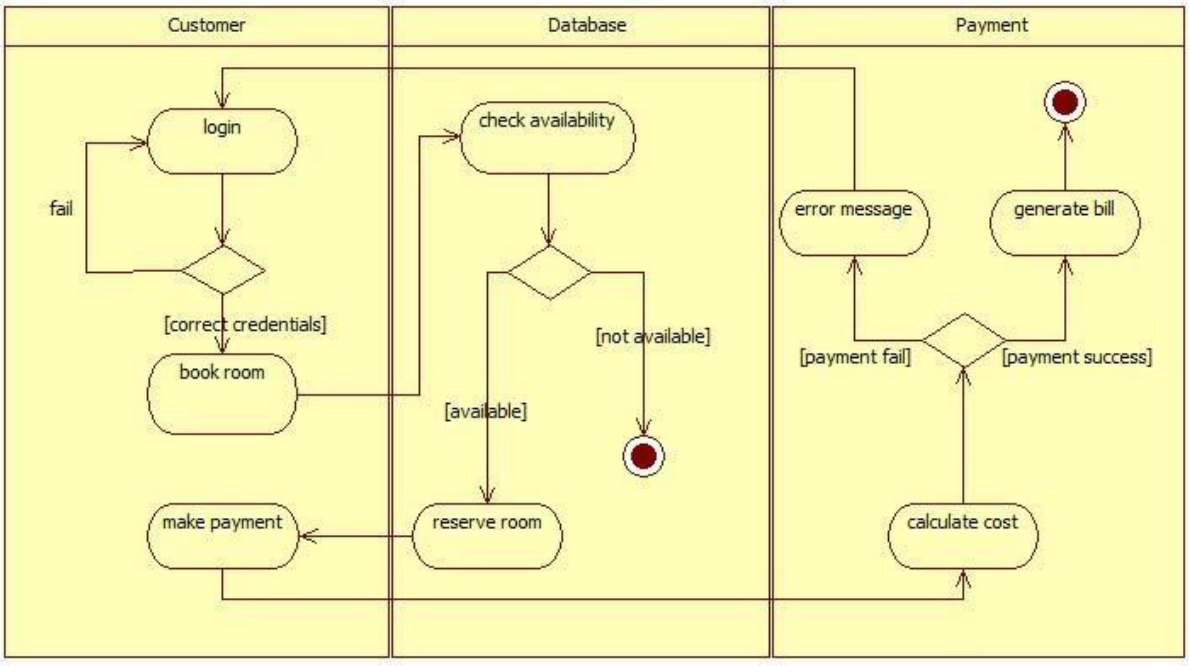


Fig 1.6 Activity Diagram HMS

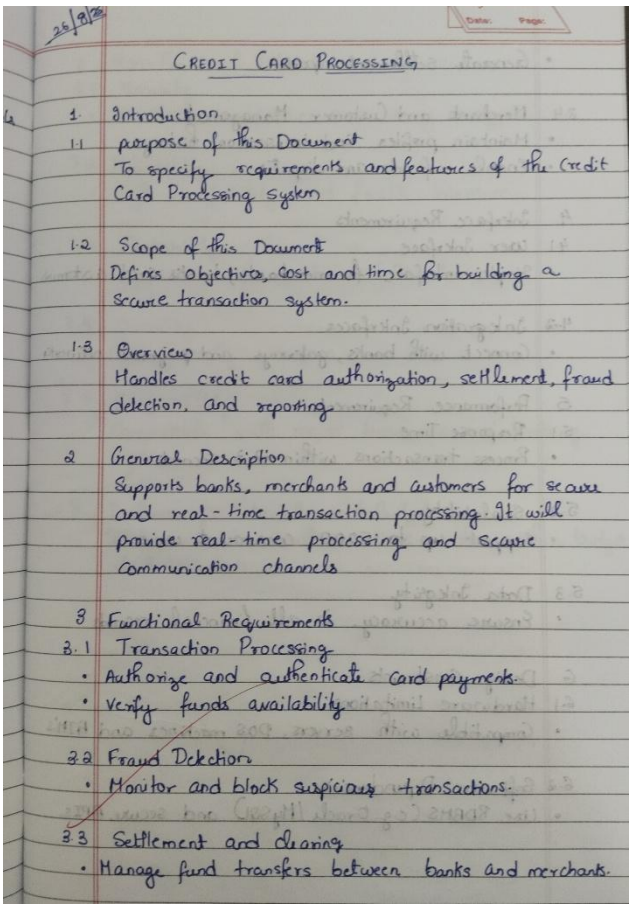
This activity diagram shows how a hotel reservation is handled across the Customer, System, and Admin. The process begins when the customer inquires about a reservation. The system responds by suggesting suitable room details, while the admin inputs the room information into the system. The customer then enters the required room details, which the system uses to verify the customer. If the verification fails, the customer is asked to enter or update their details so a proper customer record can be created. If verification is successful, the system assigns a room to the customer. After the room is assigned, the customer receives a confirmation, marking the completion of the reservation process.

2. Credit Card Processing

Problem Statement

Managing credit card payments involves many steps like verifying card details, checking available funds, preventing fraud, and settling transactions between banks and merchants. When these steps are done manually or through outdated systems, it can lead to delays, errors, and security issues for customers and businesses. To solve this, a reliable Credit Card Processing System is needed to handle payments quickly, securely, and accurately. The system should help banks, merchants, and customers complete transactions in real time while ensuring safety, fraud protection, and smooth communication across all parties.

SRS-Software Requirements Specification



CREDIT CARD PROCESSING

1. Introduction
1.1 purpose of this Document
To specify requirements and features of the Credit Card Processing System

1.2 Scope of this Document
Defines objectives, cost and time for building a secure transaction system.

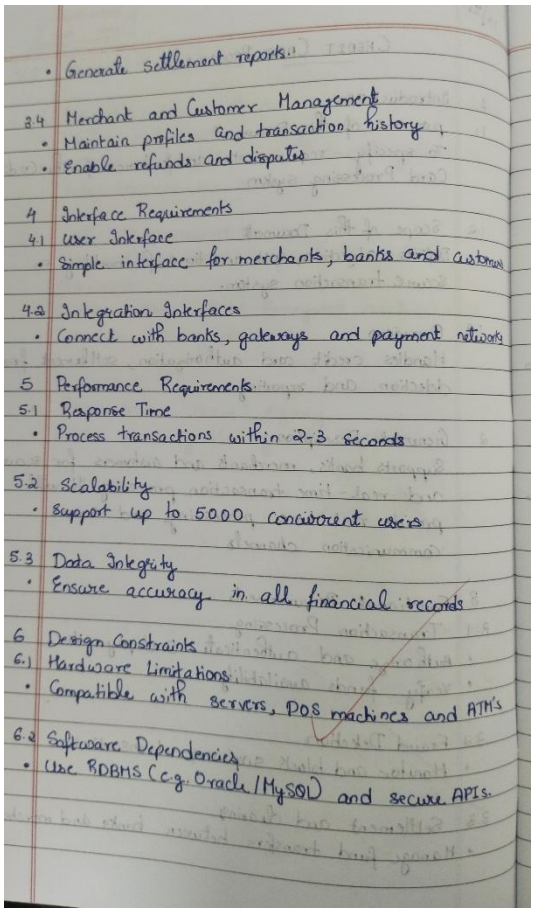
1.3 Overview
Handles credit card authorization, settlement, fraud detection, and reporting

2 General Description
Supports banks, merchants and customers for secure and real-time transaction processing. It will provide real-time processing and secure communication channels

3 Functional Requirements
3.1 Transaction Processing
- Authorize and authenticate card payments.
- Verify funds availability

3.2 Fraud Detection
- Monitor and block suspicious transactions.

3.3 Settlement and clearing
- Manage fund transfers between banks and merchants.



- Generate settlement reports.

3.4 Merchant and Customer Management
- Maintain profiles and transaction history
- Enable refunds and disputes

4 Interface Requirements
4.1 User Interface
- Simple interface for merchants, banks and customers

4.2 Integration Interfaces
- Connect with banks, gateways and payment networks

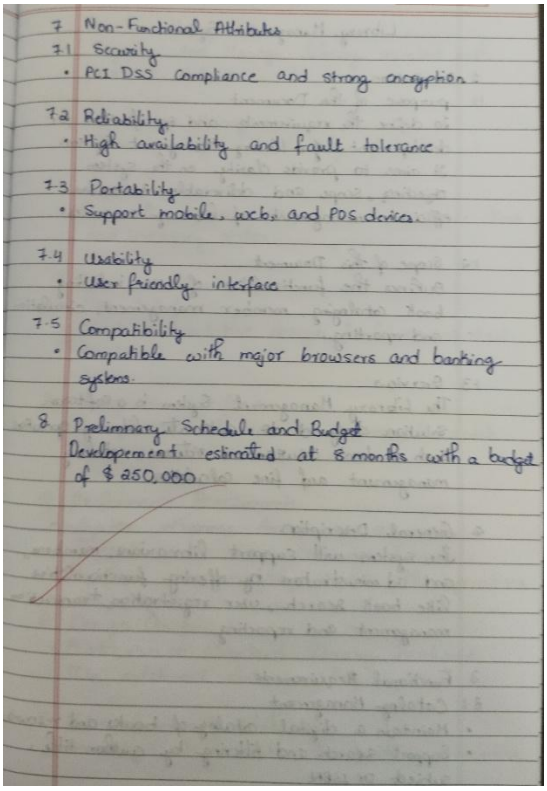
5 Performance Requirements
5.1 Response Time
- Process transactions within 2-3 seconds

5.2 Scalability
- Support up to 5000 concurrent users

5.3 Data Integrity
- Ensure accuracy in all financial records

6 Design Constraints
6.1 Hardware Limitations
- Compatible with servers, POS machines and ATM's

6.2 Software Dependencies
- Use RDBMS (e.g. Oracle/MySQL) and secure APIs.


7 Non-Functional Attributes

7.1 Security
- PCI DSS compliance and strong encryption.

7.2 Reliability
- High availability and fault tolerance.

7.3 Portability
- Support mobile, web, and POS devices.

7.4 Usability
- User friendly interface.

7.5 Compatibility
- Compatible with major browsers and banking systems.

8 Preliminary Schedule and Budget
Development estimated at 8 months with a budget of $ 250,000.

Class Diagram

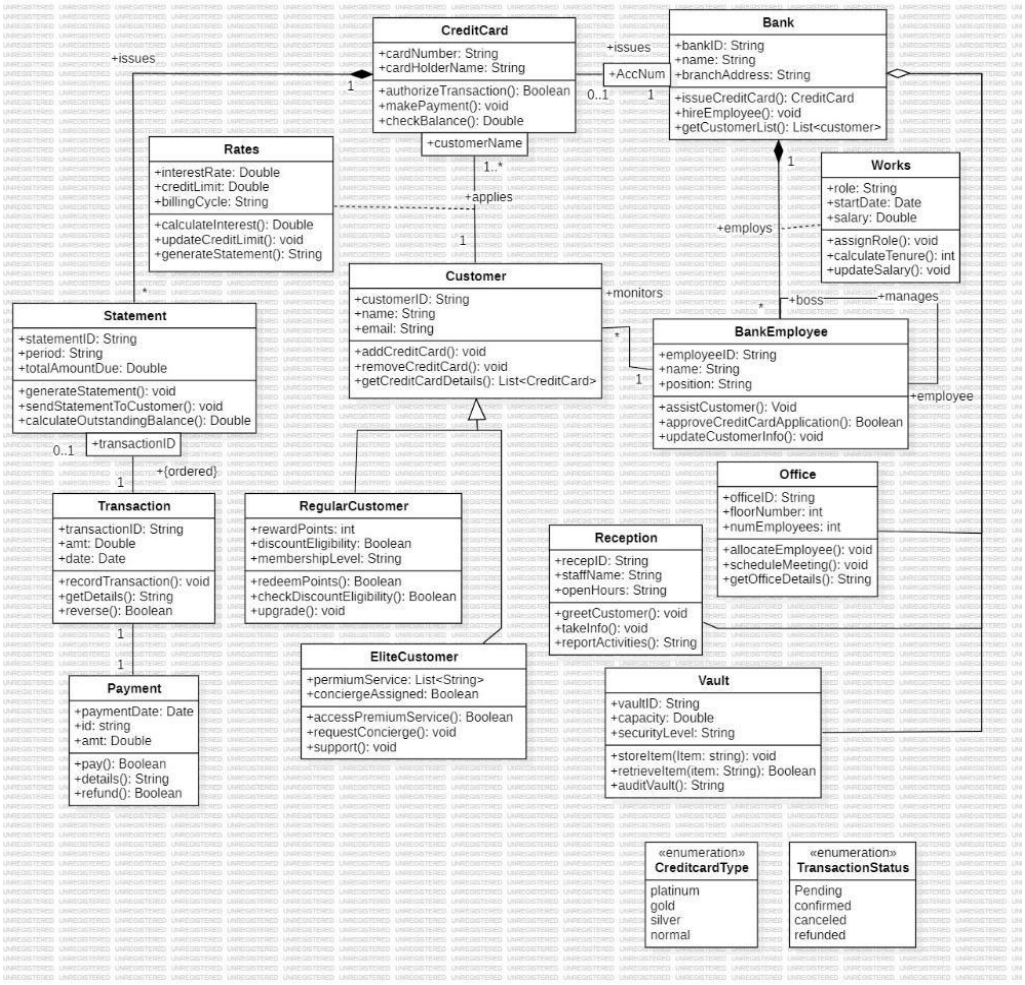


Fig 2.1 Class Diagram CCPS

This class diagram shows how a credit card processing system manages customers, banks, employees, and financial transactions. The Customer class connects to different types of customers Regular and Elite each with their own benefits and services. A customer can own multiple CreditCards, which store details like card number, billing cycle, and credit limits, and handle actions such as authorizing transactions and checking balances. Each card is issued by a Bank, which employs BankEmployees who review applications, assist customers, and manage office operations. Transactions, Payments, and Statements record all financial activities, including amounts, dates, refunds, and outstanding balances. Supporting classes like Rates calculate interest and billing cycles, while Vault and Office handle internal bank operations. Enumerations such as CreditCardType and TransactionStatus define card categories and transaction outcomes. Overall, the diagram models the complete workflow of issuing cards, managing customer accounts, processing payments, and maintaining secure bank operations.

State Diagram

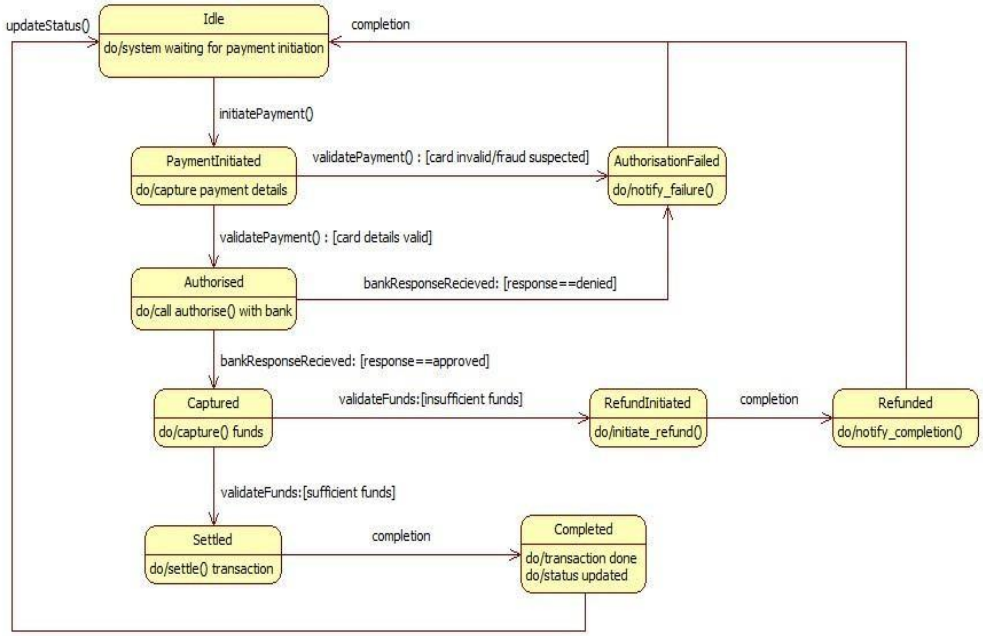


Fig 2.2 Simple State Diagram CCPS

This state diagram shows the journey of a payment as it moves through a transaction system—from waiting to begin, to validating card details, getting bank approval, capturing funds, settling the transaction, and finally marking it as completed. If any issues occur (like invalid card, fraud suspicion, denial by the bank, or insufficient funds), the flow diverts into failure or refund paths. Overall, it represents how the system carefully checks, authorizes, processes, and completes or refunds a payment to ensure everything happens securely and correctly.

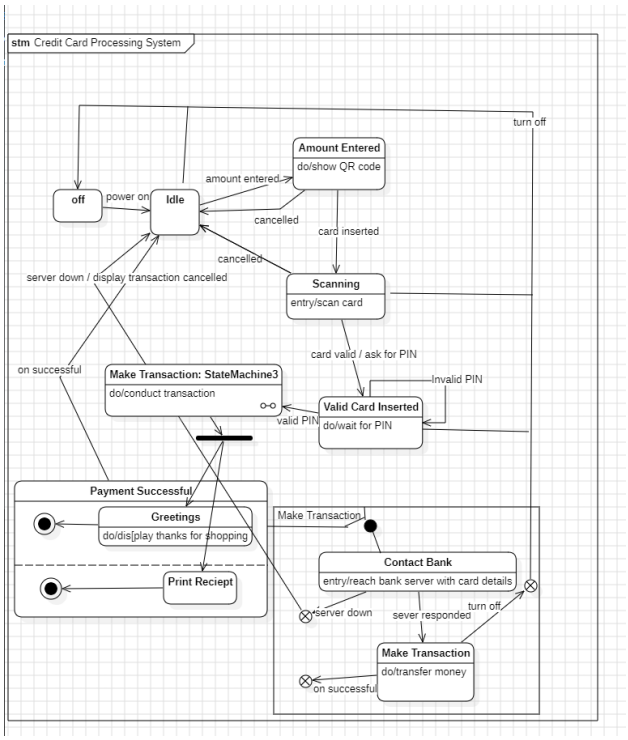


Fig 2.3 Advance State Diagram

This state diagram shows how a credit card processing system guides a transaction from start to finish. It begins from an idle state, waits for an amount to be entered, scans the card, verifies it, checks the PIN, and then contacts the bank to process the payment. If everything goes well, the system completes the transaction, displays a thank-you message, and prints the receipt. If the card is invalid, the PIN is wrong, the server is down, or the user cancels, the flow safely returns to idle or stops the process. Overall, it illustrates how the system handles each step to ensure a smooth and secure payment.

Use Case Diagram

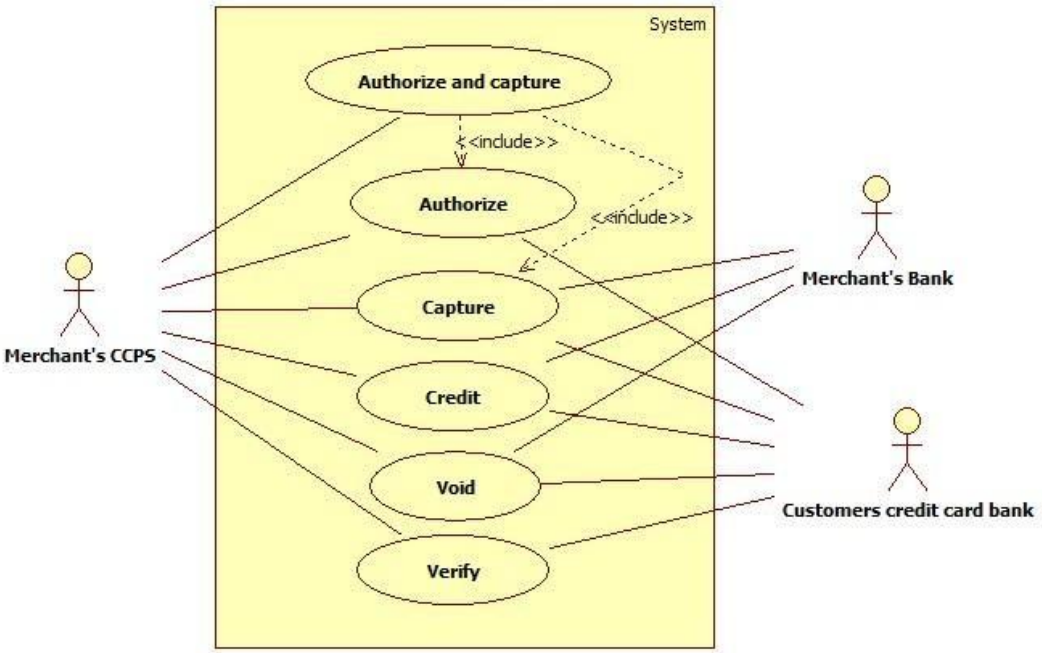


Fig 2.4 Use Case Diagram CCPS

This use case diagram shows how the credit card processing system works between the merchant, the merchant's bank, and the customer's credit card bank. The merchant's system can perform actions like authorizing a payment, capturing funds, issuing credit (refund), voiding a transaction, and verifying card details. Some actions depend on others — for example, "Authorize and Capture" includes both authorizing the card and then capturing the funds. Each use case interacts with the banks to confirm card validity, approve payments, or process refunds. Overall, it illustrates how different parties coordinate to securely approve, charge, and manage credit card transactions.

Sequence Diagram

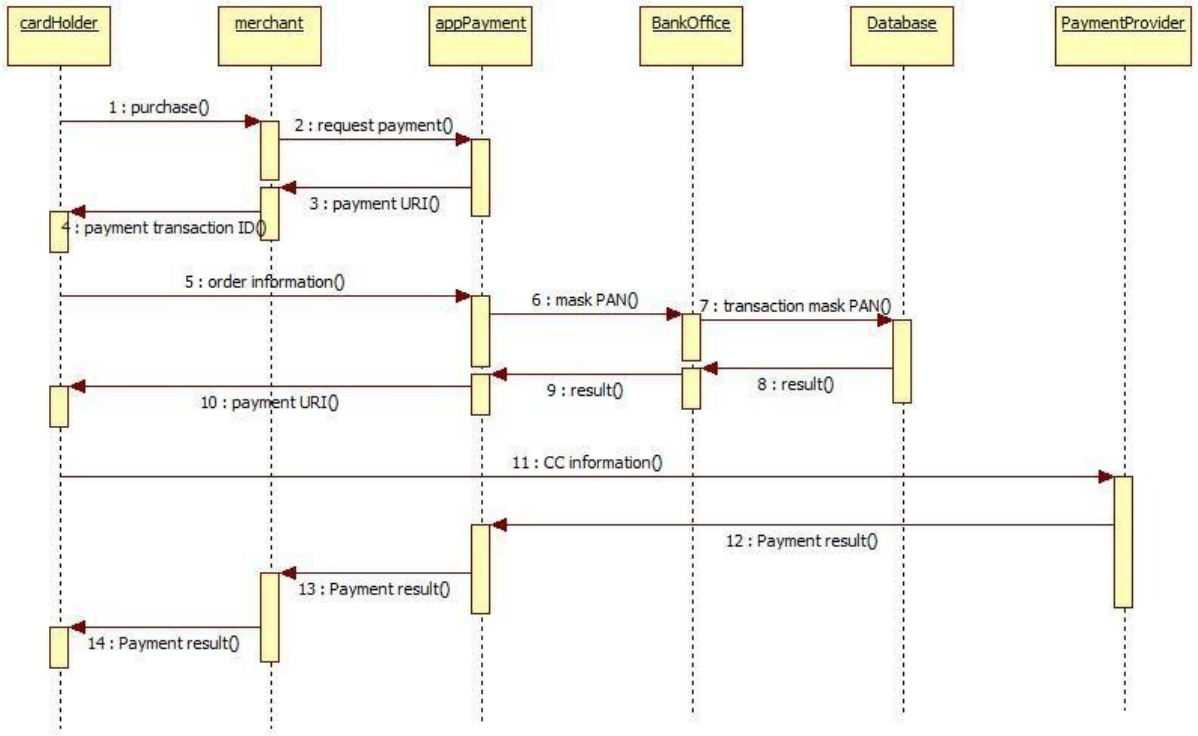


Fig 2.5 Sequence Diagram CCPS

This sequence diagram shows the full communication flow during an online payment. It begins when the cardholder makes a purchase and the merchant requests payment details from the payment application. The system generates a payment link or URI and returns it to the merchant, who sends it back to the customer. The app sends order information for processing, and the bank masks sensitive card data before passing it to the database. Each component returns results back through the chain. Once card details are verified and processed, the payment provider delivers the final payment result, which flows back to the bank office, the payment app, the merchant, and finally reaches the cardholder. Overall, it captures how different systems coordinate to securely validate, mask, process, and confirm a payment transaction.

Activity Diagram

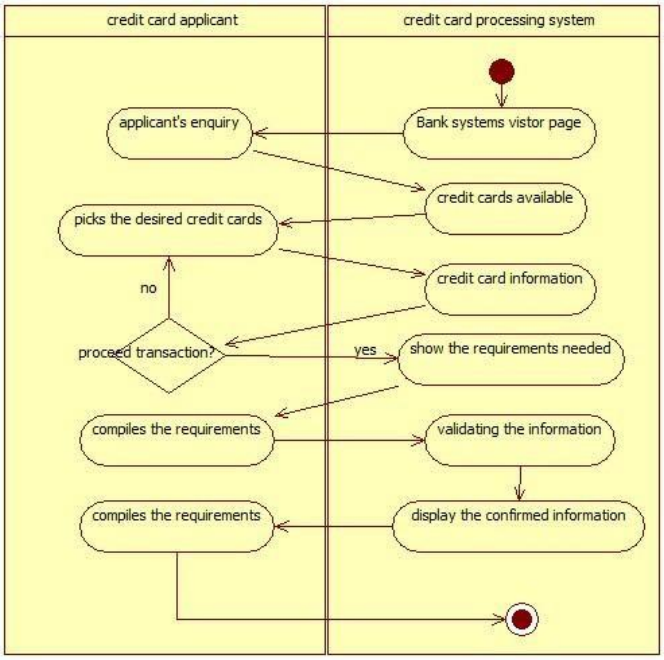


Fig 2.6 Activity Diagram CCPS

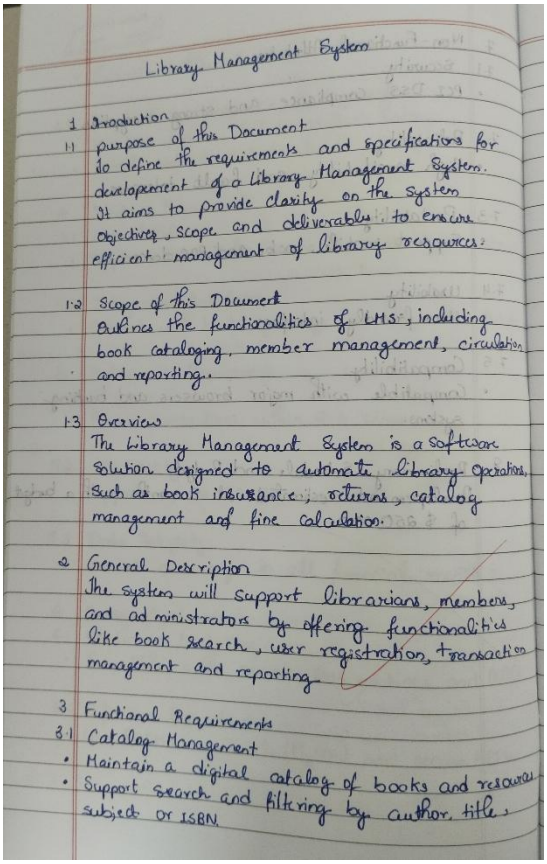
This activity diagram shows how a credit card applicant interacts with the credit card processing system to apply for a card. The process begins when the applicant makes an enquiry, and the system displays its visitor page with the available credit card options. The applicant selects a card and the system provides detailed card information. If the applicant chooses to proceed, the system shows the requirements needed, and the applicant compiles the required documents. The system then validates the submitted information and displays the confirmed details back to the applicant. Once everything is compiled and verified, the process reaches completion. The diagram clearly illustrates the back-and-forth flow between the applicant and the system during a credit card application.

3 Library Management

Problem Statement

A library handles many daily tasks like tracking books, managing members, issuing and returning books, maintaining records, and handling fines. When these tasks are done manually, it becomes difficult to keep accurate records, avoid misplaced books, check availability, or manage member information efficiently. This often leads to delays, errors, and extra workload for librarians and inconvenience for users. To overcome these issues, a Library Management System is needed to automate book management, streamline borrowing and returning, maintain accurate records, and provide quick access to information. The goal is to make library operations faster, more organized, and easier for both librarians and members.

SRS-Software Requirements Specification



Library Management System

1 Introduction

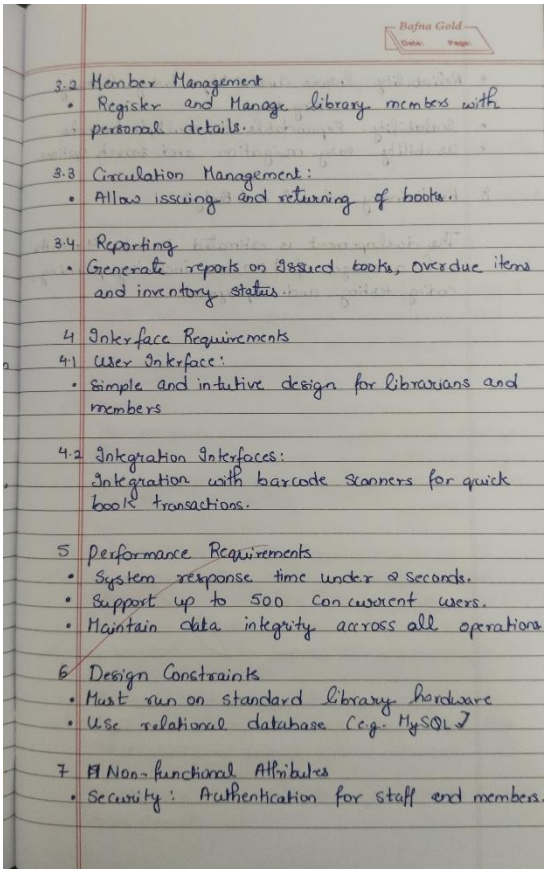
1.1 purpose of this Document
To define the requirements and specifications for development of a Library Management System. It aims to provide clarity on the system objectives, scope and deliverables to ensure efficient management of library resources.

1.2 Scope of this Document
Outlines the functionalities of LMS, including book cataloging, member management, circulation and reporting.

1.3 Overview
The Library Management System is a software solution designed to automate library operations such as book issuance, returns, catalog management and fine calculation.

2 General Description
The system will support librarians, members, and administrators by offering functionalities like book search, user registration, transaction management and reporting.

3 Functional Requirements
3.1 Catalog Management
- Maintain a digital catalog of books and resources
- Support search and filtering by author, title, subject or ISBN



Bafna Gold

3.2 Member Management
- Register and Manage library members with personal details.

3.3 Circulation Management:
- Allow issuing and returning of books.

3.4 Reporting
- Generate reports on issued books, overdue items and inventory status.

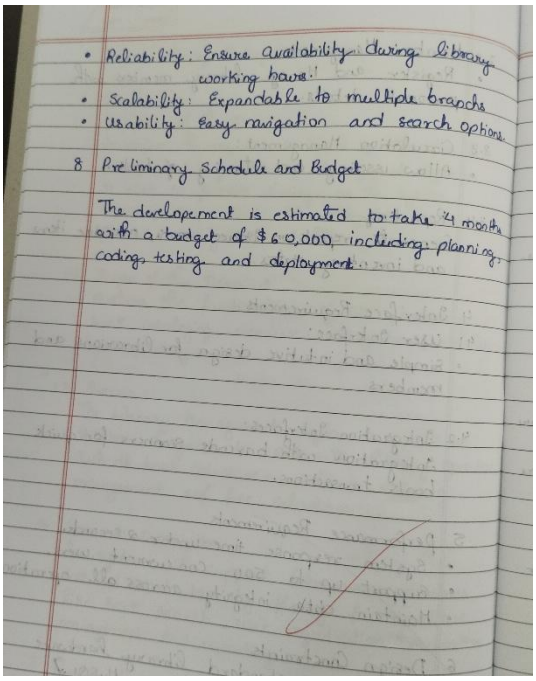
4 Interface Requirements
4.1 User Interface:
- Simple and intuitive design for librarians and members

4.2 Integration Interfaces:
Integration with barcode scanners for quick book transactions.

5 Performance Requirements
- System response time under 2 seconds.
- Support up to 500 concurrent users.
- Maintain data integrity across all operations

6 Design Constraints
- Must run on standard library hardware
- Use relational database e.g. MySQL

7 Non-functional Attributes
- Security: Authentication for staff and members.



Class Diagram

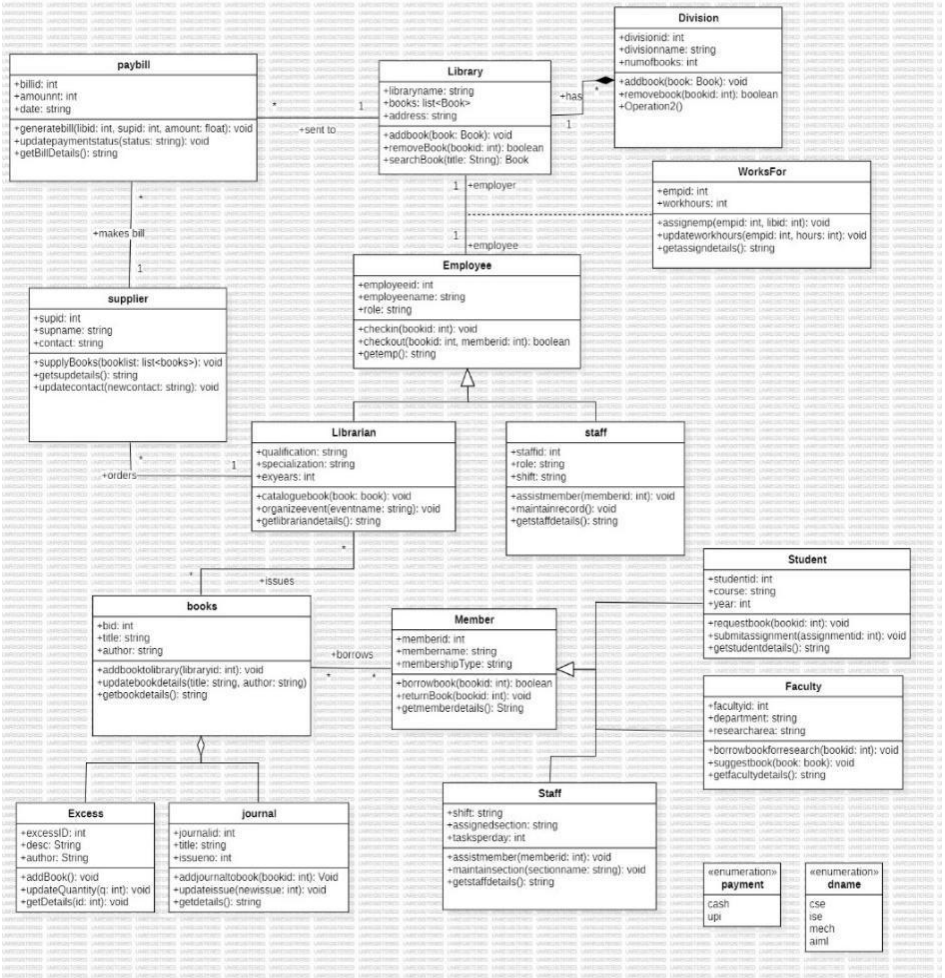


Fig 3.1 Class Diagram LMS

This class diagram shows how a library manages its books, members, staff, and daily operations through different interconnected components. The Library class keeps track of books, journals, and excess items, while Librarians handle cataloging, issuing books, organizing events, and managing records. Members—whether students or faculty—can borrow and return books, with their details and borrowing history stored in the Member class. Staff assist with book maintenance, shelving, and member services, while Employees are linked to divisions and work schedules through the Division and WorksFor classes. Suppliers provide books and are linked to paybills for payments. The system also tracks payments, fines, and book details through dedicated classes like Paybill, Books, Journal, and Excess. Altogether, the diagram represents how a library coordinates its inventory, users, employees, suppliers, and financial processes to run smoothly and efficiently.

State Diagram

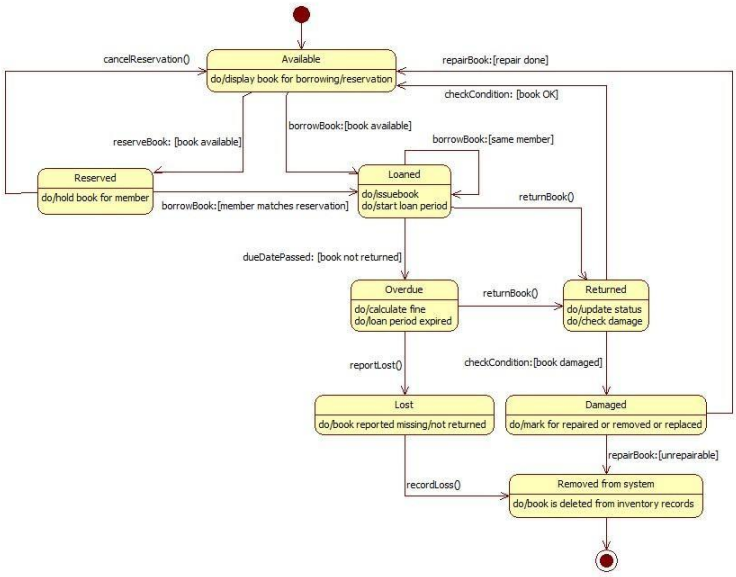


Fig 3.2 Simple Sate Diagram LMS

This state machine diagram shows how a library book moves through different states from the moment it is available until it is returned, lost, damaged, or removed from the system. A book starts in the Available state, where it can be borrowed or reserved by a member. If reserved, it is held for that member; if borrowed, it enters the Loaned state. When the loan period ends and the book is not returned, it becomes Overdue, and fines may be applied. If the book is reported missing, it moves to the Lost state. If the book is returned, it enters the Returned state, where its condition is checked—if fine, it goes back to Available; if damaged, it moves to the Damaged state for repair or replacement. If the book is beyond repair, it reaches the Removed from System state and is deleted from the library's records. The diagram clearly maps the full lifecycle of a book in a library system.

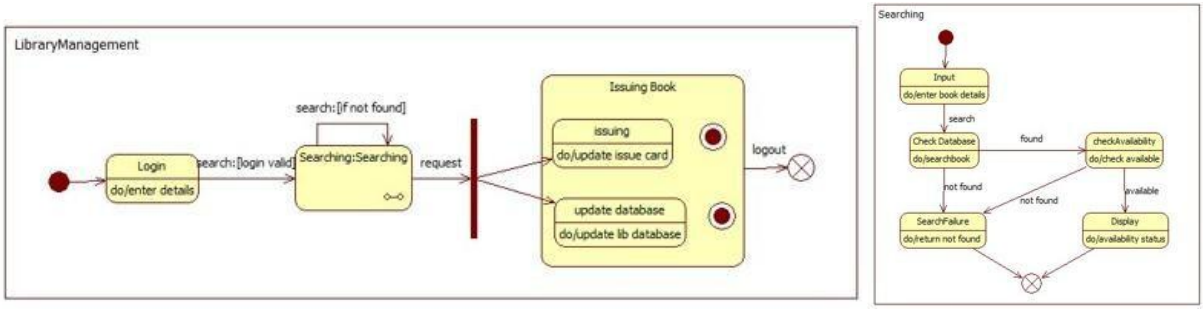


Fig 3.3 Advanced State Diagram LMS

The Library Management diagram shows how a user logs into the system, enters their details, and begins searching for a book. Once the search is valid, the system transitions to the Issuing Book process, where the book is issued, the issue card is updated, and the library database is refreshed. After completing the transaction, the user can log out, ending the session. The separate Searching diagram explains what happens during the book search: the user enters book details, the system checks the database, and if the book is found, it checks availability and displays the current status; if not found, the system returns a search-failure message. Together, these diagrams describe how a user searches for a book and how the library system processes the issuing of that book in a smooth, step-by-step workflow.

Use Case Diagram

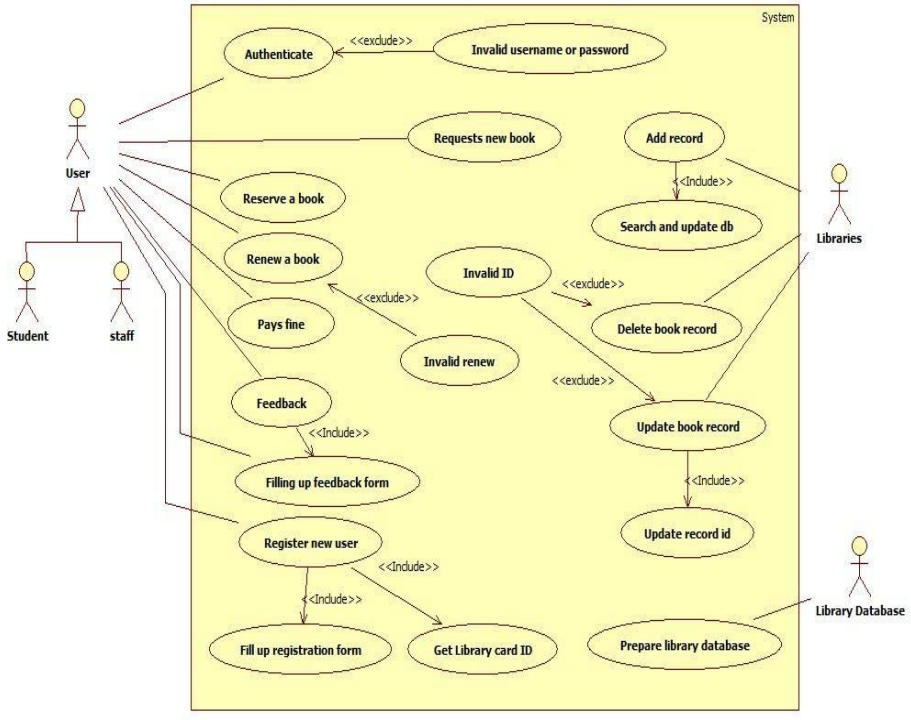


Fig 3.4 Use Case Diagram LMS

This use case diagram shows how different users—students and staff—interact with the library management system to perform everyday tasks. A user can log in, reserve or renew books, request new books, pay fines, give feedback, and register as a new member. The system checks for invalid logins, wrong IDs, or invalid renewals and guides the user through forms like registration and feedback. Behind the scenes, the system works with librarians and the library database to add new records, update or delete book information, search the database, and prepare or update library IDs. Overall, the diagram illustrates a complete picture of how users and librarians work together with the library system to manage books, user accounts, and database updates smoothly.

Sequence Diagram

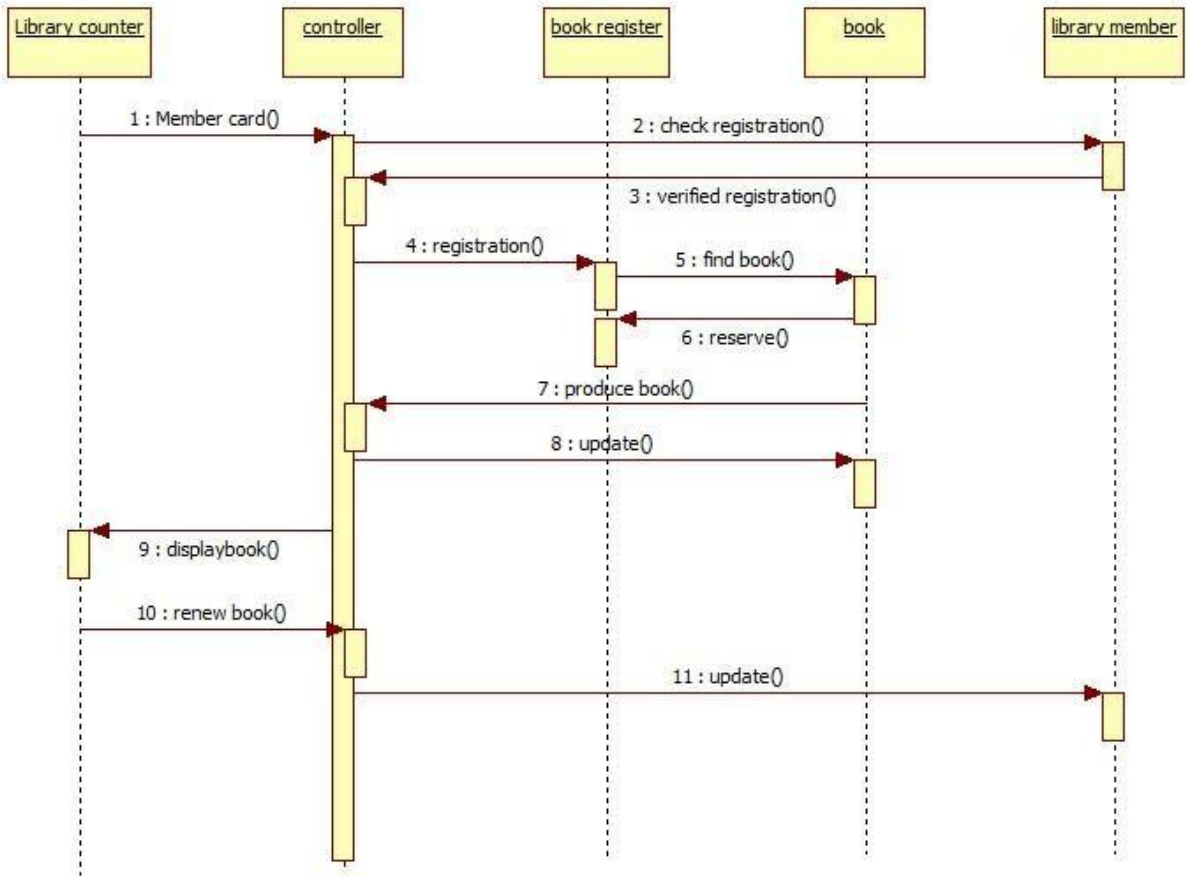


Fig 3.5 Sequence Diagram LMS

This sequence diagram shows the step-by-step process of how a library member interacts with the system to register and borrow or renew a book. The flow begins when the member presents their card at the library counter, and the controller checks their registration with the book register. After the registration is verified, the controller requests the book register to locate the required book, and the book is then reserved for the member. Once the book is found and prepared, the system updates the records. Later, the member can view the book details at the counter and request a renewal,

which again triggers an update in the system. Overall, the diagram clearly illustrates how the controller coordinates between the library counter, book register, book entity, and the library member to handle registration, reservation, issuing, and renewal activities.

Activity Diagram

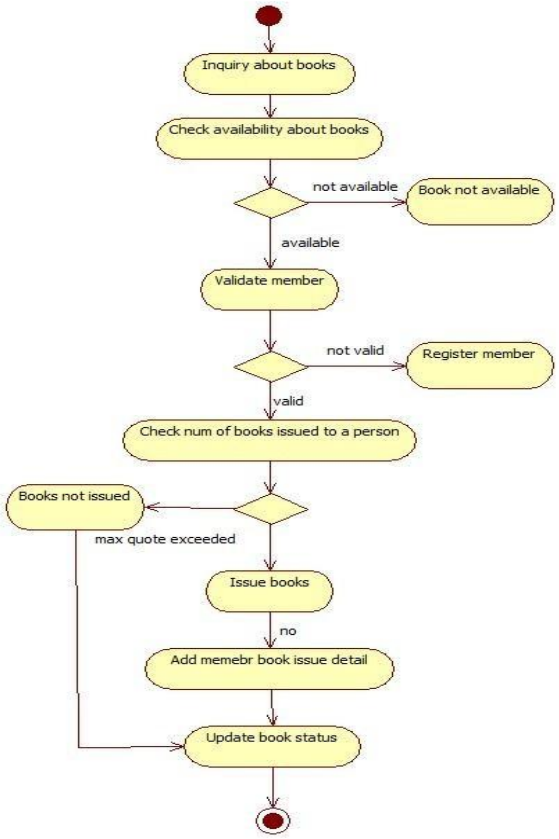


Fig 3.6 Activity Diagram LMS

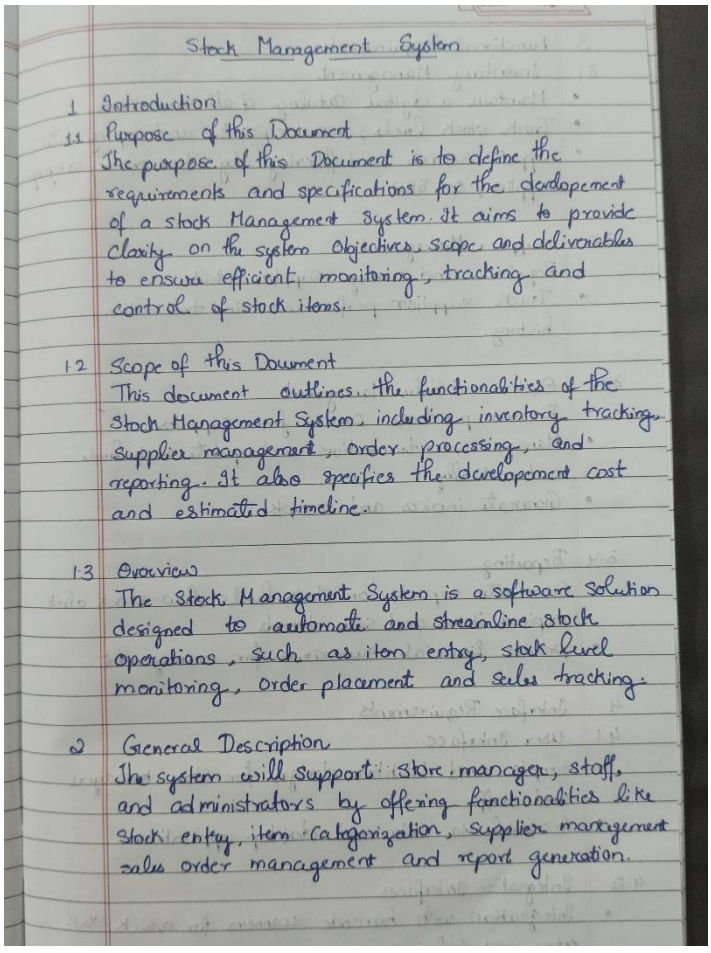
This activity diagram shows the step-by-step process a library follows when a user wants to issue a book. The process begins with an inquiry about a book, after which the system checks if the book is available. If it is not available, the process ends there; if available, the system verifies whether the person is a valid library member. Non-members are directed to register first, while valid members proceed to the next step, where the system checks how many books the member has already borrowed. If they have exceeded the limit, no books are issued; otherwise, the system issues the requested books, updates the member's issue details, and finally updates the book's status in the database. This ensures proper tracking of book availability and member borrowing limits.

4 Stock Management System

Problem Statement

A business needs to keep track of its inventory so it always knows what items are available, what needs to be reordered, and how stock is moving in and out. When this is done manually, it often leads to mistakes like over-stocking, running out of essential items, misplaced products, or incorrect stock counts. These issues can slow down operations, increase costs, and affect customer satisfaction. To solve this, a Stock Management System is needed to automatically monitor inventory levels, update stock in real time, record purchases and sales, and alert staff when items need restocking. The goal is to make inventory handling faster, more accurate, and easier for the organization.

SRS-Software Requirements Specification



Stock Mangement System

1 Introduction

1.1 Purpose of this Document

The purpose of this Document is to define the requirements and specifications for the developement of a stock Management System. It aims to provide clarity on the system objectives, scope and deliverables to ensure efficient monitoring, tracking and control of stock items.

1.2 Scope of this Doument

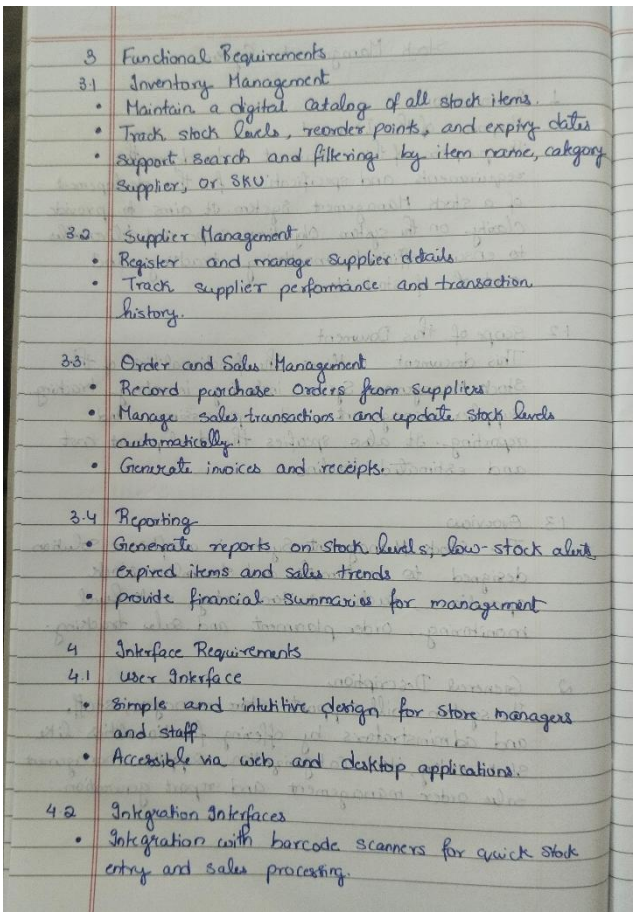
This document outlines the functionalities of the Stock Management System, including inventory tracking, supplier management, order processing, and reporting. It also specifies the developement cost and estimated timeline.

1.3 Overview

The Stock Management System is a software solution designed to automate and streamline stock operations, such as item entry, stock level monitoring, order placement and sales tracking.

2 General Description

The system will support store manager, staff, and administrators by offering functionalities like stock entry, item categorization, supplier management sales order management and report generation.



3 Functional Requirements

3.1 Inventory Management

- Maintain a digital catalog of all stock items.
- Track stock levels, reorder points, and expiry dates
- Support search and filtering by item name, category supplier, or SKU

3.2 Supplier Management

- Register and manage supplier details
- Track supplier performance and transaction history.

3.3 Order and Sales Management

- Record purchase Orders from suppliers
- Manage sales transactions and update stock levels automatically.
- Generate invoices and receipts.

3.4 Reporting

- Generate reports on stock levels, low-stock alert expired items and sales trends
- provide financial summaries for management

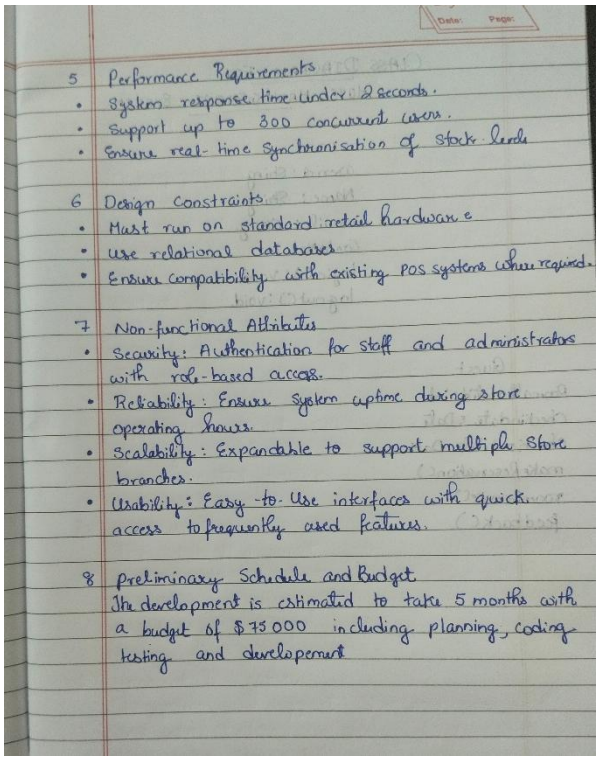
4 Interface Requirements

4.1 User Interface

- Simple and intuitive design for store managers and staff
- Accessible via web and desktop applications.

4.2 Integration Interfaces

- Integration with barcode scanners for quick stock entry and sales processing.



5 Performance Requirements
- System response time under 2 seconds.
- Support up to 300 concurrent users.
- Ensure real-time synchronisation of stock levels

6 Design Constraints
- Must run on standard retail hardware
- Use relational database
- Ensure compatibility with existing POS systems where required.

7 Non-functional Attributes
- Security: Authentication for staff and administrators with role-based access.
- Reliability: Ensure system uptime during store operating hours.
- Scalability: Expandable to support multiple store branches.
- Usability: Easy-to-use interface with quick access to frequently used features.

8 Preliminary Schedule and Budget
The development is estimated to take 5 months with a budget of $75 000 including planning, coding, testing and development

Class Diagram

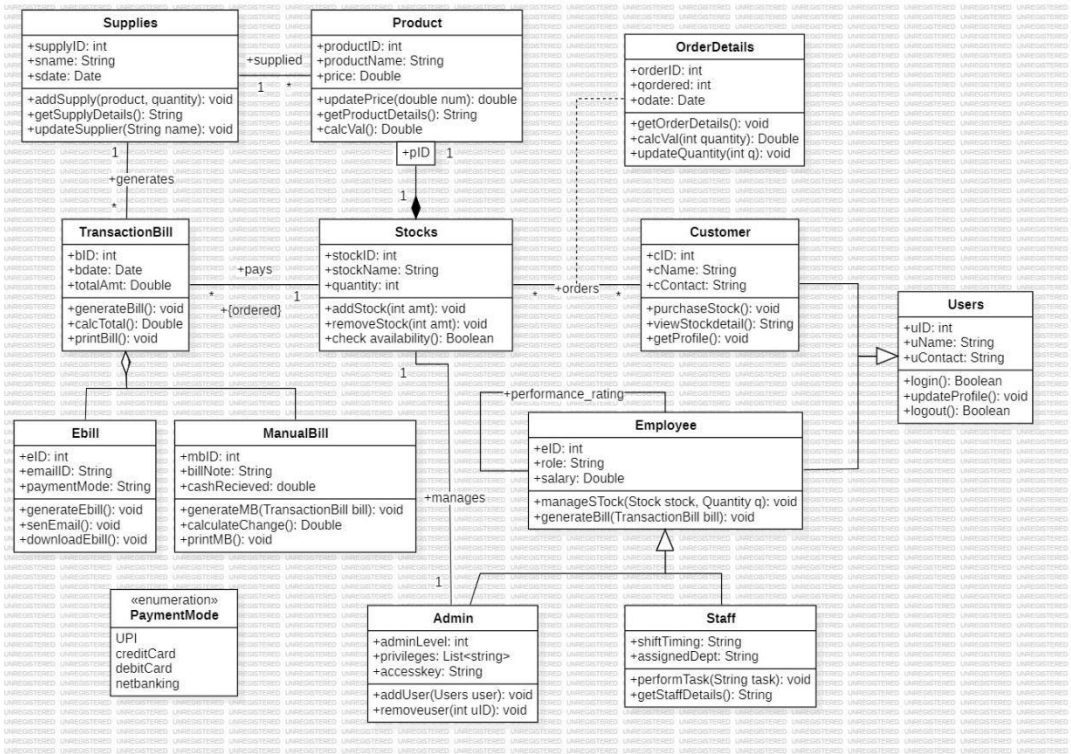


Fig 4.1 Class Diagram SMS

This class diagram represents how a stock management system tracks products, suppliers, inventory, customers, bills, and employees. Supplies deliver products, which are stored as Stock with information like name, quantity, and ID. Customers place orders for different products, and Order Details keep track of quantities and dates. Employees manage stock levels, update product quantities, and generate bills, while admins oversee users and system privileges. The system supports both electronic bills and manual bills, each storing payment details, totals, and methods like UPI, credit card, or net banking. Users such as staff and customers can log in, update profiles, and perform tasks based on their roles. Overall, the diagram shows how stock is added, removed, purchased, billed, and tracked across different parts of the business, ensuring smooth and organized inventory handling.

State Diagram

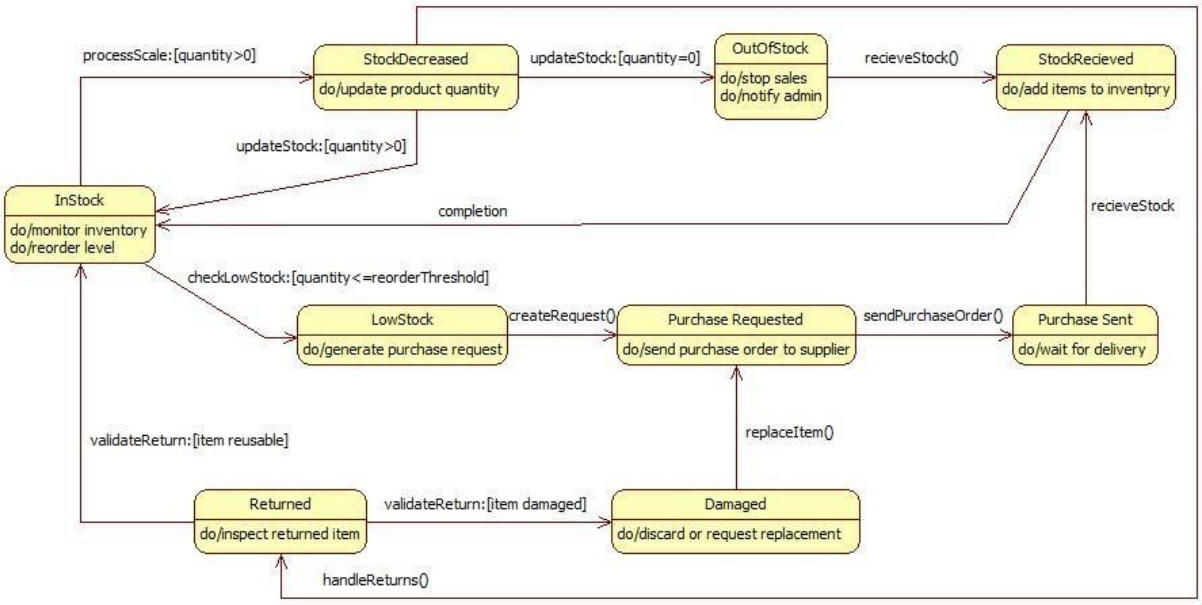


Fig 4.2 Simple State Diagram SMS

This state machine diagram shows how stock items move through different stages in an inventory system. Items start in the InStock state, where their levels are monitored. When stock decreases, the system updates quantities, and if the item runs out, it moves to the OutOfStock state, notifying the admin and stopping sales. If an item is returned, it is inspected and either accepted back as reusable or marked as damaged and discarded or replaced. When stock becomes low, the system creates a purchase request, sends an order to the supplier, and waits for the delivery. Once the new items arrive, they move to the StockReceived state and are added back to inventory, returning the system to the InStock state. The diagram clearly shows how stock is tracked, replenished, reused, or discarded throughout its entire lifecycle.

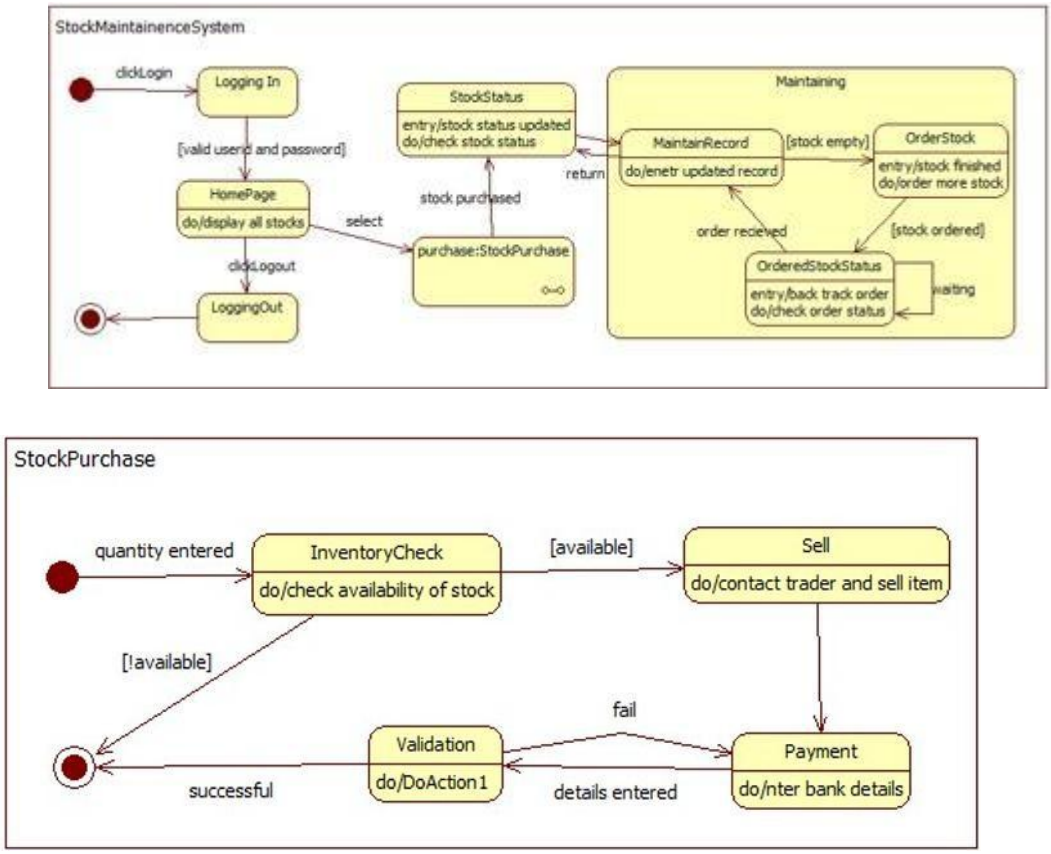


Fig 4.3 Advanced State Diagram SMS

The Stock Maintenance System diagram shows how a user logs into the system, checks the current stock status, and either purchases stock or updates records based on what is available. From the homepage, the user can view all items and navigate to stock status, where they can see if inventory is low or empty. If stock is empty, the system moves into the maintenance area, where records are updated and new stock orders are placed. Once an order is made, the system tracks the order status until the stock arrives. The Stock Purchase diagram describes how the system checks the availability of stock when a purchase quantity is entered. If the item is available, it proceeds to selling and then to payment; if unavailable, the process ends. Payments are validated, and once successful, the purchase is completed. Together, both diagrams show the entire flow of how stock is checked, updated, ordered, purchased, and maintained within an inventory system.

Use Case Diagram

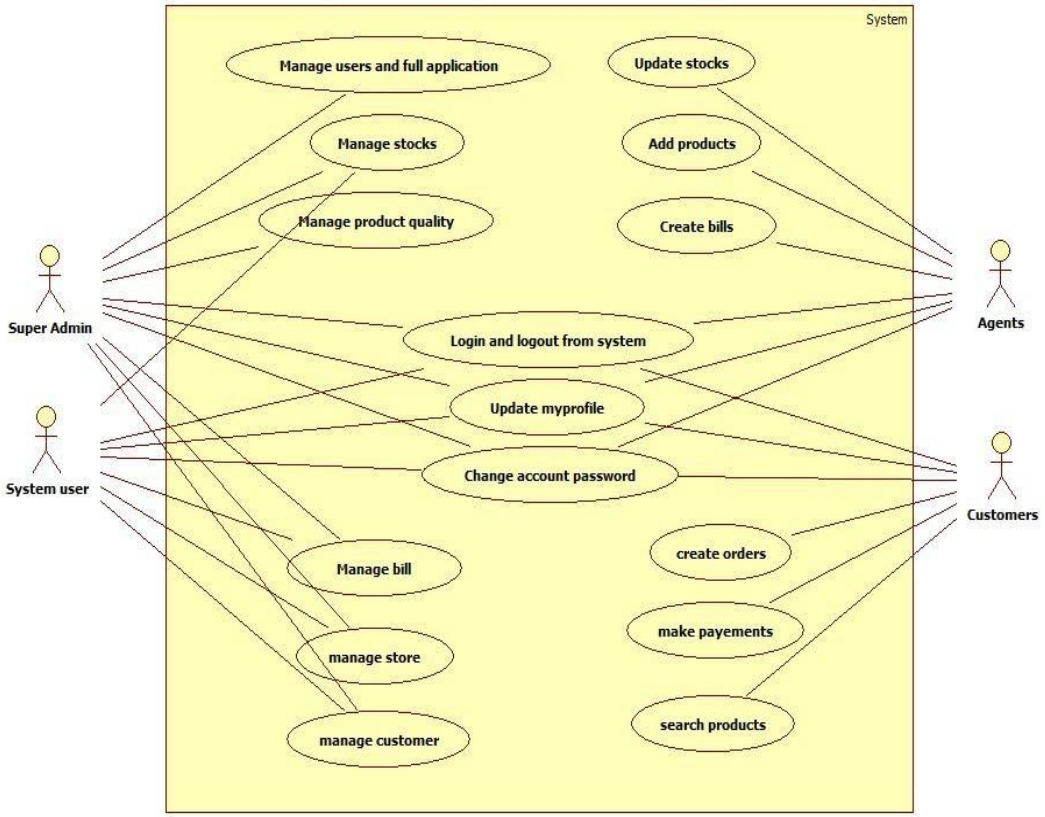


Fig 4.4 Use Case Diagram SMS

This use case diagram shows how different users interact with the stock management system to perform their specific tasks. Super Admins have the highest control, allowing them to manage users, monitor the entire application, update stocks, manage product quality, and handle billing, stores, and customers. System Users perform daily operations like updating their profiles, managing bills, managing stores, handling customers, and ensuring smooth stock updates. Agents mainly handle product-related tasks such as updating stocks, adding products, and creating bills. Customers can search for products, create orders, make payments, and manage their own account details like profile updates and password changes. Altogether, the diagram clearly shows how each type of user works with the system to maintain inventory, process orders, and ensure smooth stock management.

Sequence Diagram

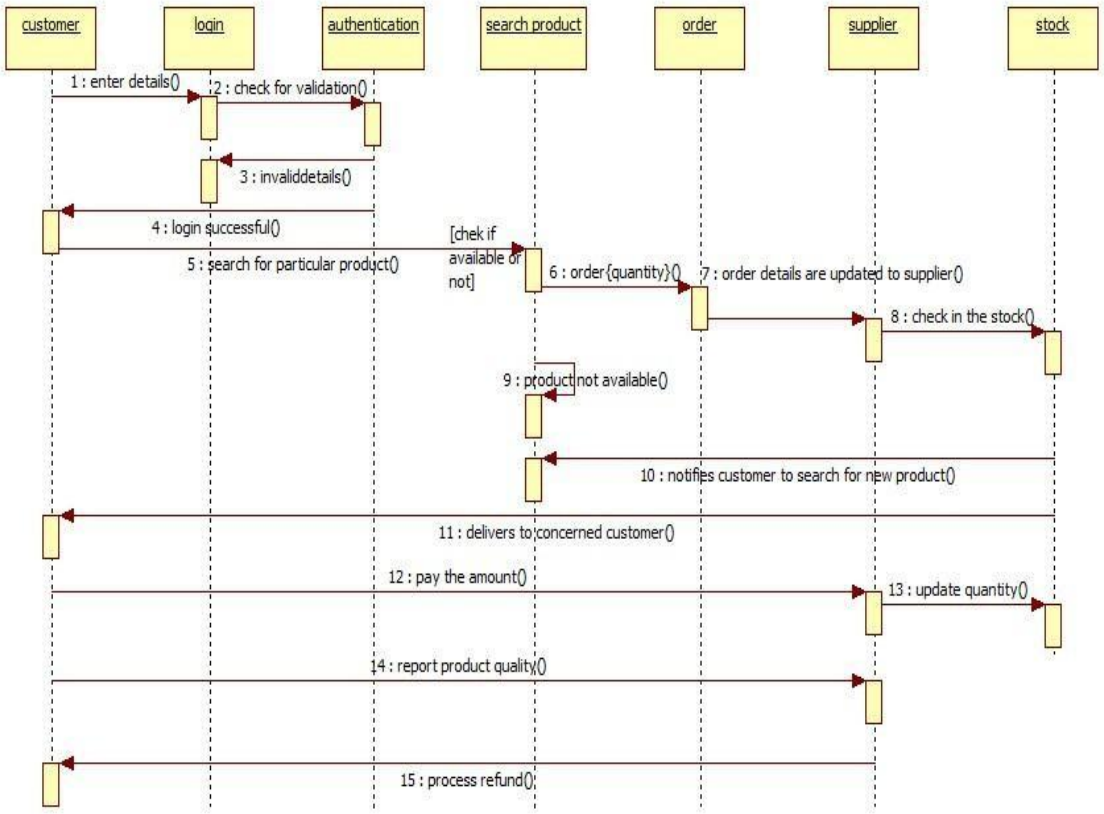


Fig 4.5 Sequence Diagram SMS

This sequence diagram shows how a customer searches for and orders a product through the stock management system. The process begins when the customer enters login details, which are checked for validation. Once logged in, the customer searches for a product, and the system checks its availability in stock. If available, an order is placed, and the order information is sent to the supplier, who then checks stock and delivers the product to the customer. After receiving the product, the customer pays, reports any quality issues if needed, and the system processes refunds when required. If the product is not available, the system notifies the customer to search for an alternative product. Overall, the diagram explains how login, product search, ordering, delivery, payment, and refund requests flow smoothly between the customer, system, supplier, and stock.

Activity Diagram

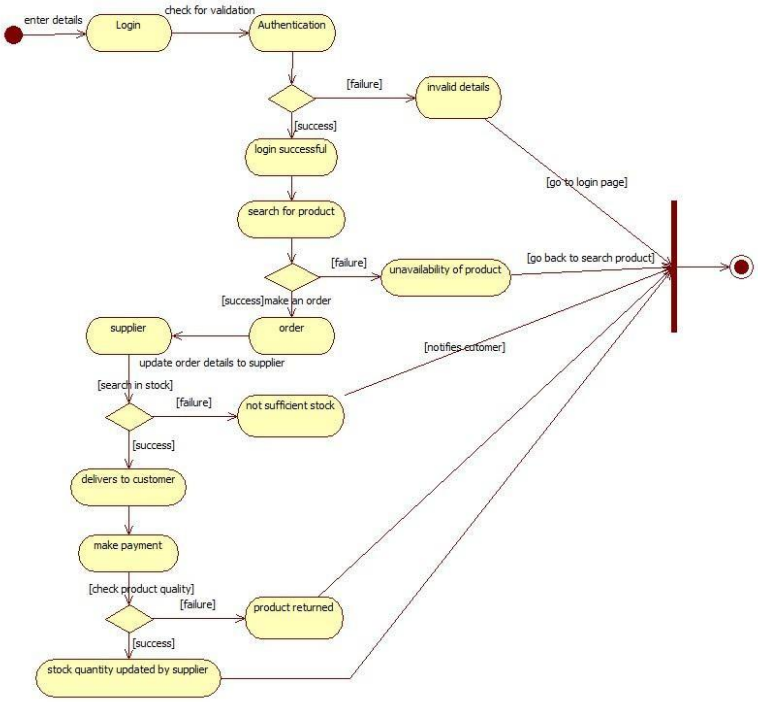


Fig 4.6 Activity Diagram SMS

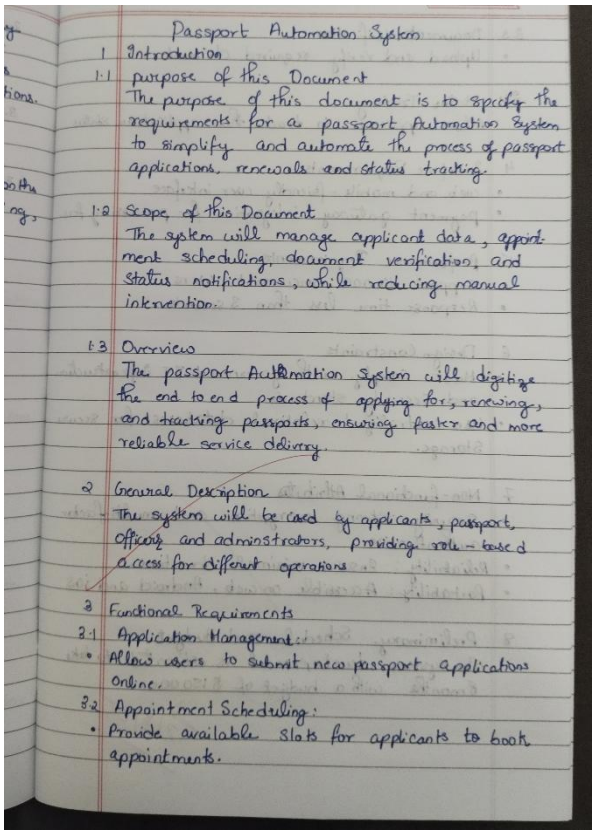
This activity diagram shows the full flow of how a customer logs into a stock management system, searches for a product, and places an order. The process begins when the customer enters their details; if authentication fails, the system shows invalid details and directs them back to login. Once logged in, the customer searches for a product—if it is unavailable, the system notifies them to try again. If available, the customer places an order, and the supplier checks stock levels. When stock is sufficient, the supplier confirms and delivers the product to the customer. After receiving it, the customer makes a payment and the system checks product quality; if the product is defective, it is returned, and the supplier updates the stock accordingly. The diagram clearly shows how login, search, order placement, supplier processing, delivery, payment, and returns are all connected in the stock management workflow.

5 Passport Automation System

Problem Statement

Applying for a passport often involves long queues, manual paperwork, repeated visits, and delays caused by slow verification and processing. These manual steps increase the chances of errors, lost documents, and long waiting times for applicants. To overcome these problems, a Passport Automation System is needed to streamline the entire process—from submitting applications and uploading documents to scheduling appointments, verifying information, and tracking application status. The goal is to make passport services faster, more accurate, and more convenient for citizens while reducing workload and errors for officials**.**

SRS-Software Requirements Specification



Passport Automation System

1 Introduction

1.1 purpose of this Document

The purpose of this document is to specify the requirements for a passport Automation System to simplify and automate the process of passport applications, renewals and status tracking.

1.2 Scope of this Document

The system will manage applicant data, appointment scheduling, document verification, and status notifications, while reducing manual intervention.

1.3 Overview

The passport Automation System will digitize the end to end process of applying for, renewing, and tracking passports, ensuring faster and more reliable service delivery.

2 General Description

The system will be used by applicants, passport officers and administrators, providing role-based access for different operations.

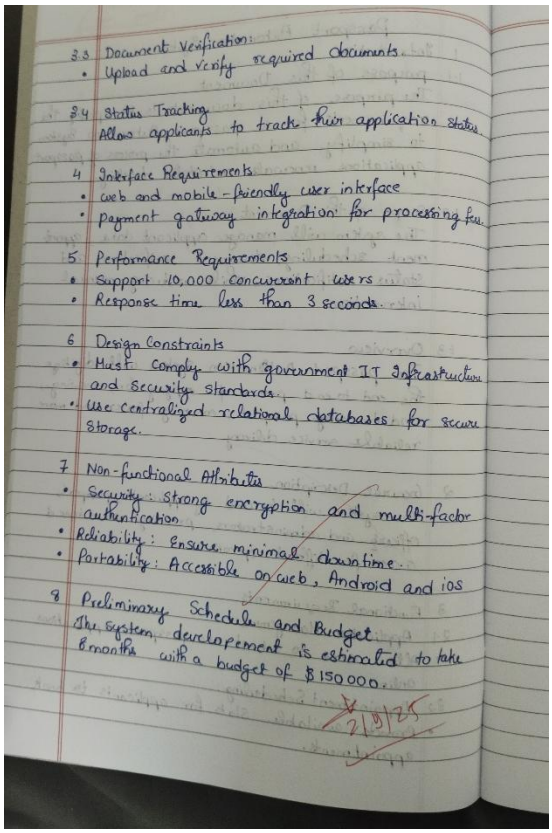
3 Functional Requirements

3.1 Application Management:

- Allow users to submit new passport applications online.

3.2 Appointment Scheduling:

- Provide available slots for applicants to book appointments.



3.3 Document Verification:

- Upload and verify required documents.

3.4 Status Tracking

Allow applicants to track their application status

4 Interface Requirements

- web and mobile-friendly user interface
- payment gateway integration for processing fee

5 Performance Requirements

- Support 10,000 concurrent users
- Response time less than 3 seconds.

6 Design Constraints

- Must comply with government IT infrastructure and security standards.
- Use centralized relational databases for secure storage.

7 Non-functional Attributes

- Security: Strong encryption and multi-factor authentication
- Reliability: Ensure minimal downtime.
- Portability: Accessible on web, Android and iOS

8 Preliminary Schedule and Budget

The system development is estimated to take 8 months with a budget of $150000.

21/9/25

Class Diagram

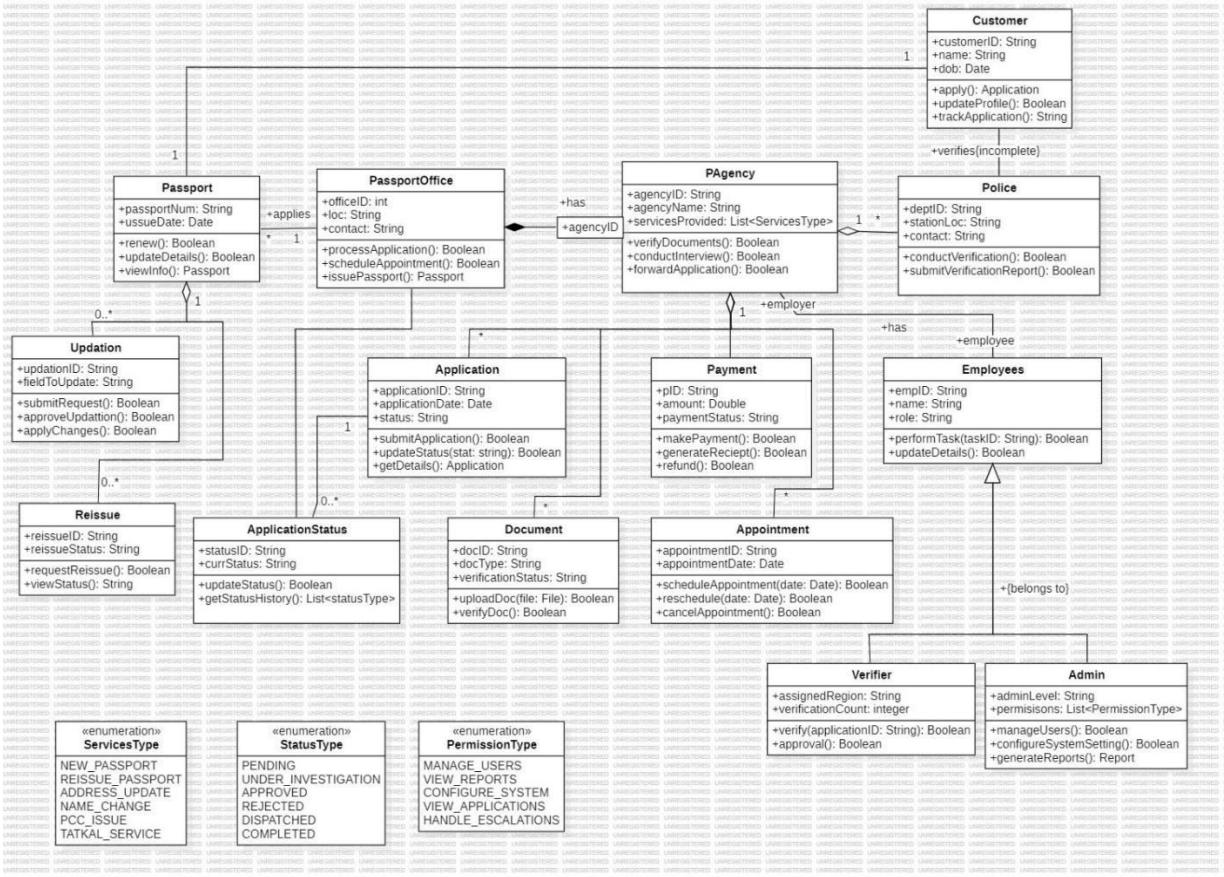


Fig 5.1 Class Diagram PAS

The Passport Automation System streamlines how a person applies for, updates, or renews a passport by connecting all the departments involved—customers, passport offices, police verification teams, payment units, and administrative staff—into one organised digital workflow. A customer submits an application, books appointments, uploads documents, and makes payments, while the system routes the request to the appropriate agency for document checks and to the police for background verification. Each application moves through stages like submission, review, verification, approval, or rejection, with employees and verifiers handling specific tasks. The system keeps track of passport status, reissue requests, updates, payments, and appointment schedules, ensuring the entire process is transparent, accurate, and faster than manual procedures.

State Diagram

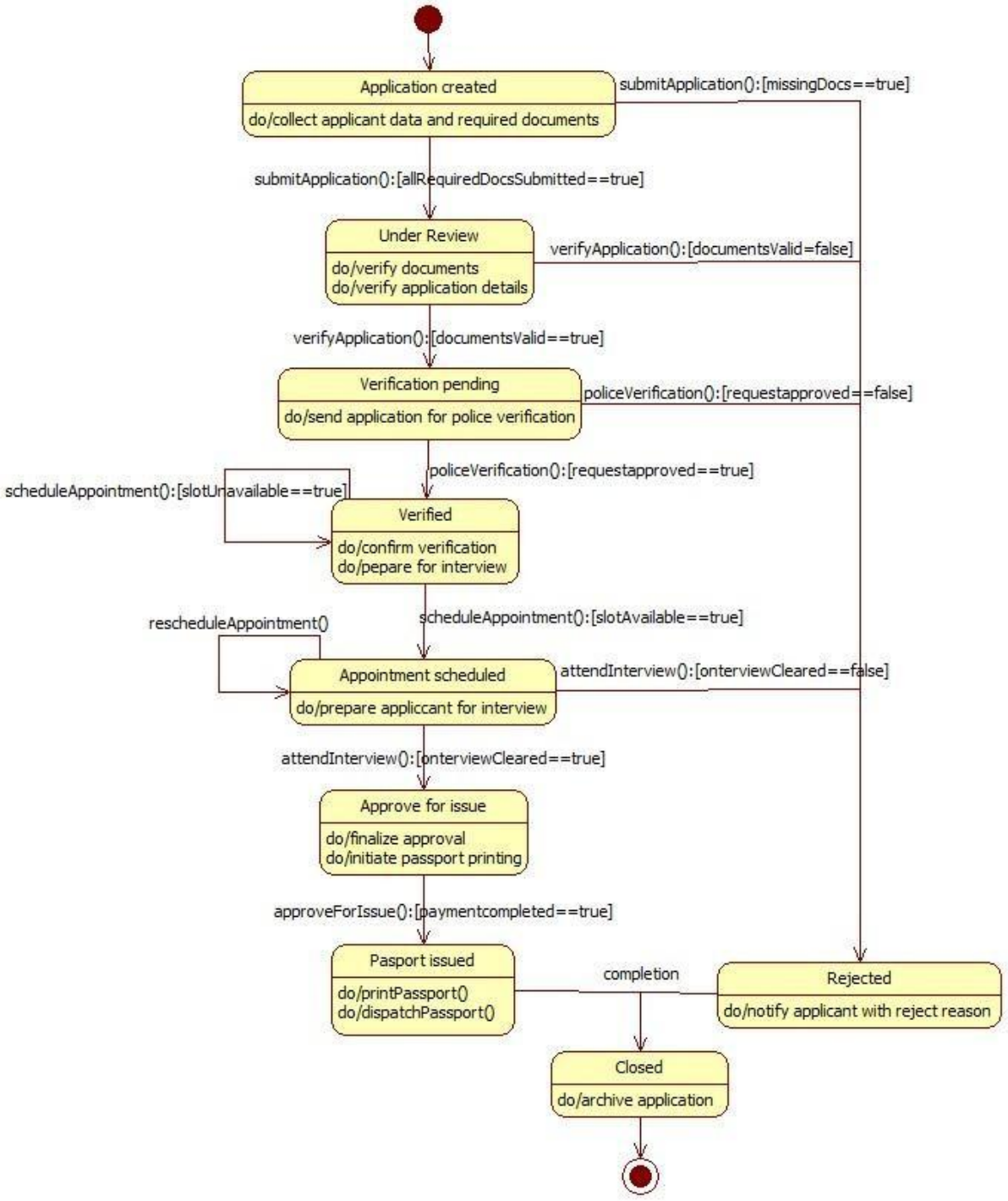


Fig 5.2 Simple State Diagram PAS

This state diagram shows the journey of a passport application from the moment it is created until it is either approved or rejected. The process begins with collecting the applicant's details and documents, after which the application goes under review for verification. If the documents are correct, the request is sent for police verification; otherwise, the application moves to

rejection. Once police approval comes through, the applicant is scheduled for an interview, with options to reschedule if slots aren't available. After successfully clearing the interview and completing payment, the system approves the application, prints the passport, and dispatches it to the applicant. If any stage fails—whether documents, verification, or interview—the application is shifted to the "Rejected" state. Finally, once the passport is issued or rejected, the system closes and archives the application.

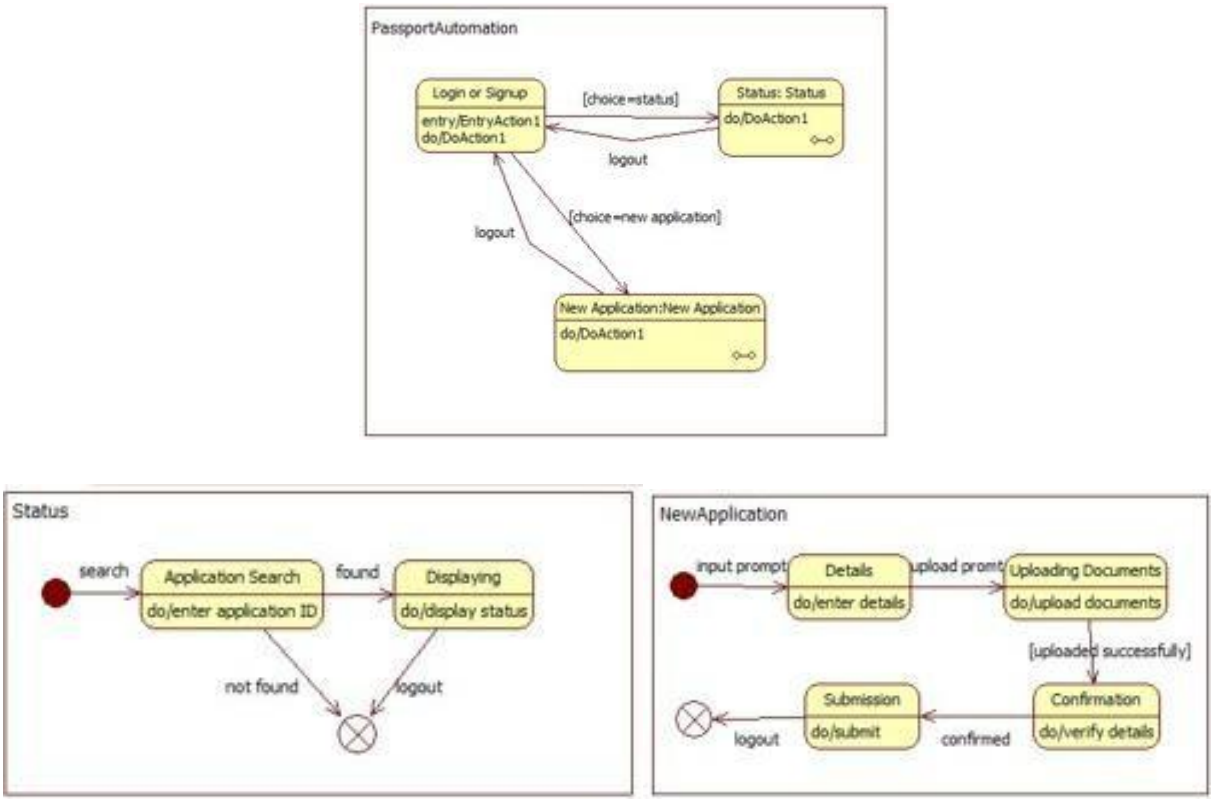


Fig 5.3 Advanced State Diagram PAS

This diagram explains how a user moves through different stages of the passport automation system, starting from logging in or signing up. After entering the system, the user can either check the status of an existing passport application or begin a new one. If they choose to check the status, they search for their application ID, and the system either displays the status or shows that no record was found. If they choose to submit a new application, the system guides them through entering personal details, uploading documents, verifying the information, and finally submitting the application. At any point, users can log out, and the system safely exits. Overall, the diagram shows how the system manages both new applications and status checks in a smooth, organised way.

Use Case Diagram

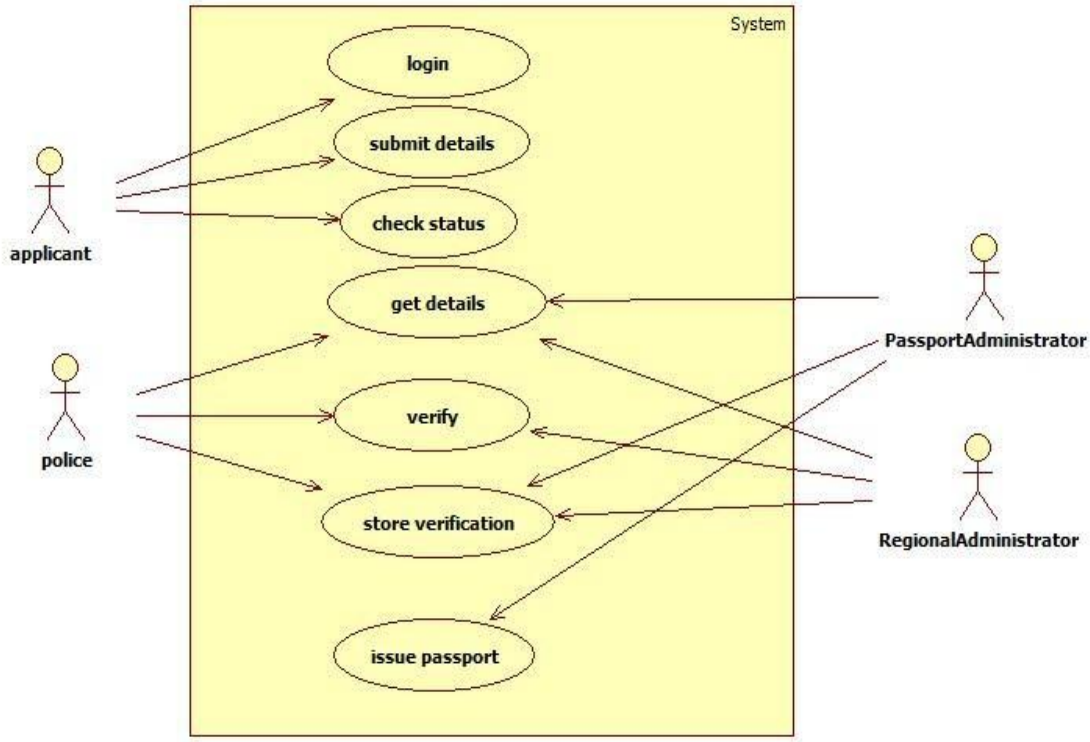


Fig 5.4 Use Case Diagram PAS

This use-case diagram shows how different people interact with the Passport Automation System during the passport application process. The applicant logs into the system, submits personal details, checks the status of their application, and retrieves their information whenever needed. The police department plays a crucial role by accessing applicant details, conducting background verification, and sending verification results back to the system. Passport Administrators and Regional Administrators handle internal tasks like reviewing applicant details, confirming verification, storing verification records, and finally approving and issuing the passport. Overall, the diagram shows how applicants, police, and administrators work together through the system to ensure a secure and smooth passport-issuing process.

Sequence Diagram

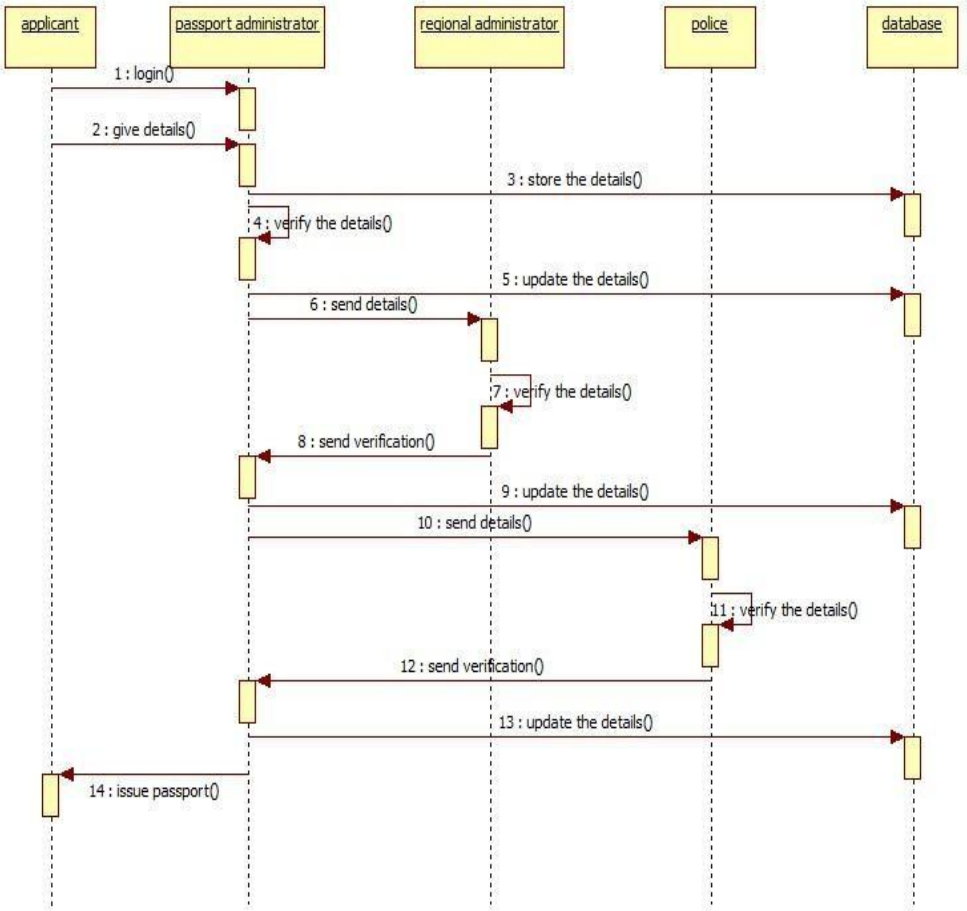


Fig 5.5 Sequence Diagram PAS

This sequence diagram shows how a passport application moves through different authorities before the passport is finally issued. The process begins when the applicant logs in and submits their personal details to the Passport Administrator, who verifies the information and forwards it to the Regional Administrator. The Regional Administrator stores and updates these details in the database, then sends them to the police for background verification. The police review the applicant's information, verify it, and send their verification report back to the Regional Administrator, who updates the database again and forwards the confirmed details to the Passport Administrator. Once all checks are cleared and verification is received, the Passport Administrator instructs the system to issue the passport. Overall, the diagram captures how multiple stakeholders work together step-by-step to securely process a passport application

Activity Diagram

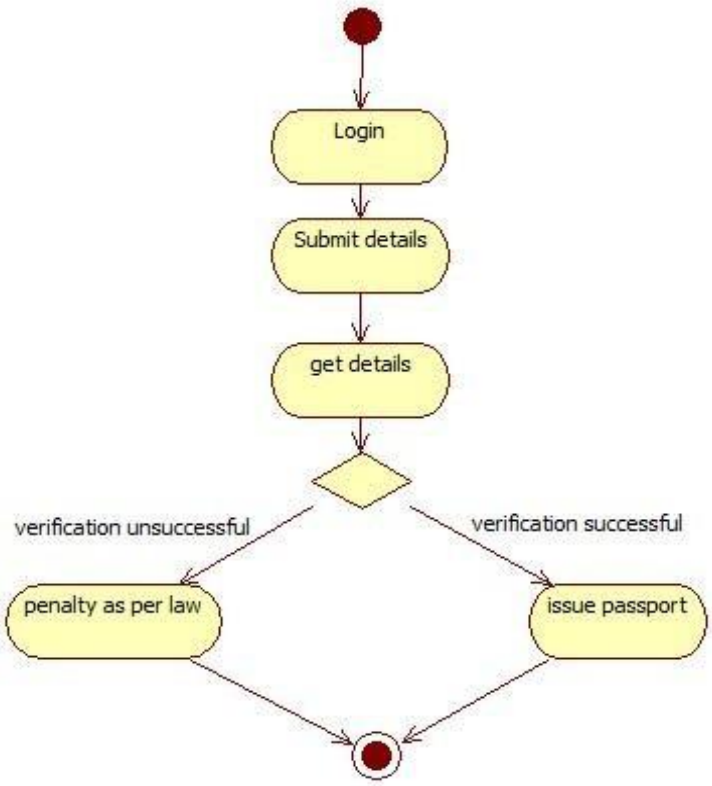


Fig 5.6 Activity Diagram PAS

This diagram shows the basic flow of how a passport application is processed. The applicant begins by logging into the system and submitting their personal details. The system then retrieves and reviews these details, leading to a verification step. If the verification is successful, the passport is approved and issued to the applicant. However, if the verification fails—due to incorrect details, missing information, or legal issues—the process leads to a penalty as per the law. The diagram neatly captures this clear decision point and the two possible outcomes, making the entire passport workflow easy to understand.