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:

Nishmitha K G

1BM23CS216

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 **NISHMITHA K G (1BM23CS216)** during the 5th Semester August 2025-December 2025

.

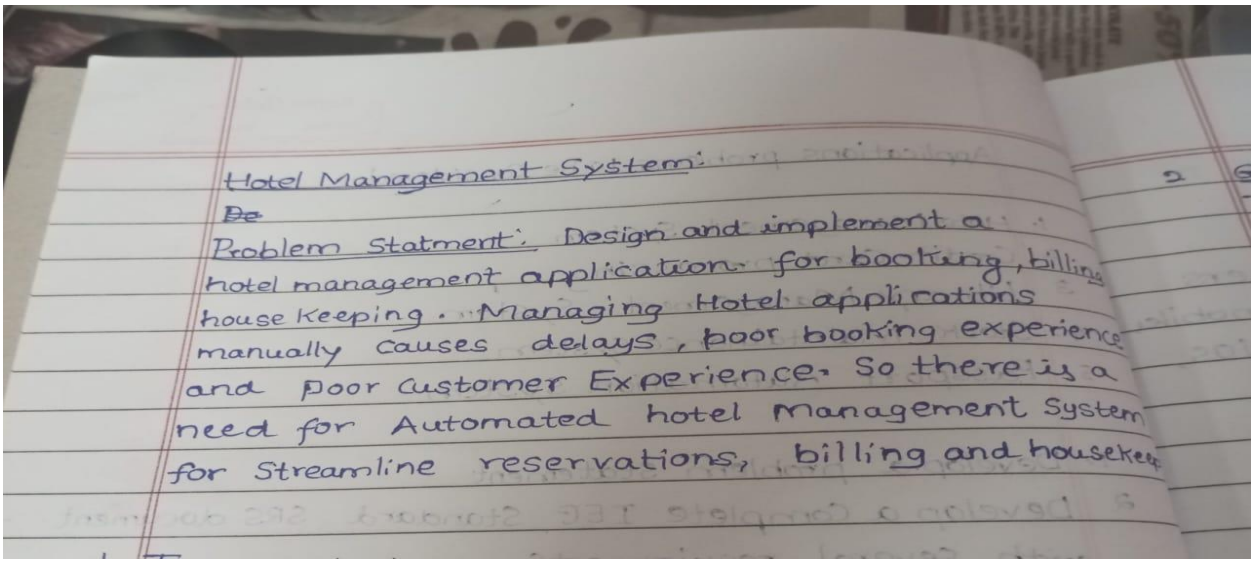
Signature of the Faculty Incharge:

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

Table of Contents

1. Hotel Management System

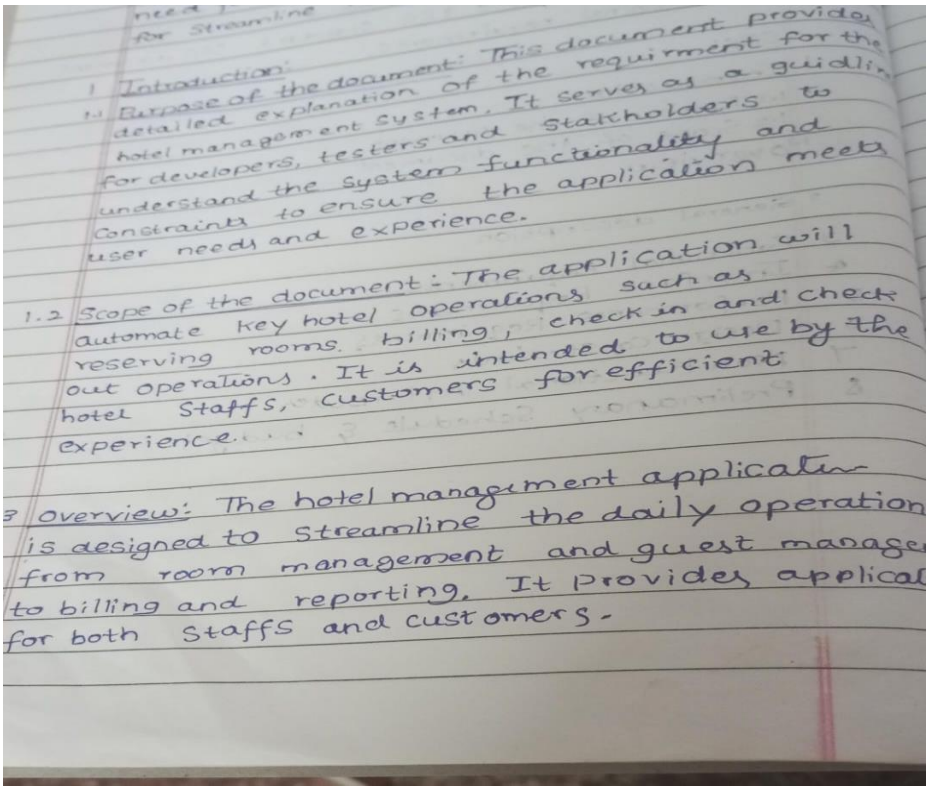
1.1Problem Statement:


Hotel Management System:

Problem Statment: Design and implement a hotel management application for booking, billing house Keeping. Managing Hotel applications manually causes delays, poor booking experience and poor customer Experience. So there is a need for Automated hotel Management System for Streamline reservations, billing and housekee

Fig 1.1

1.2 SRS-Software Requirements Specification:

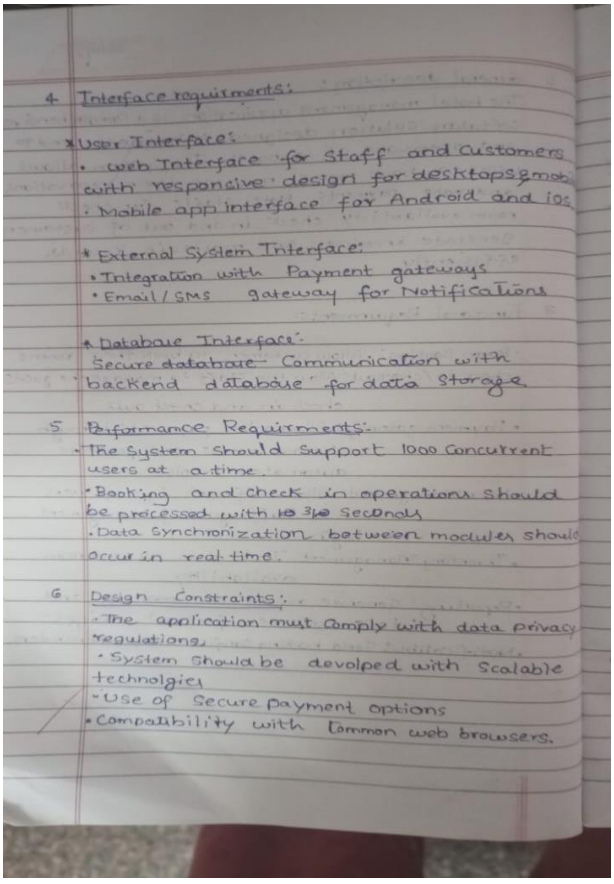

1 Introduction:

1.1 Purpose of the document: This document provides detailed explanation of the requirment for the hotel management system. It serves as a guidline for developers, testers and stakholders to understand the system functionality and Constraints to ensure the application meets user needs and experience.

1.2 Scope of the document: The application will automate key hotel operations such as reserving rooms, billing, check in and check out operations. It is intended to use by the hotel staffs, customers for efficient experience.

3 Overview: The hotel management applicatn is designed to streamline the daily operation from room management and guest manage to billing and reporting. It provides applica for both staffs and customers.

Fig 1.2



4 Interface requirments:

* User Interface:
- web Interface for Staff and Customers with responsive design for desktops & mob
- Mobile app interface for Android and iOS

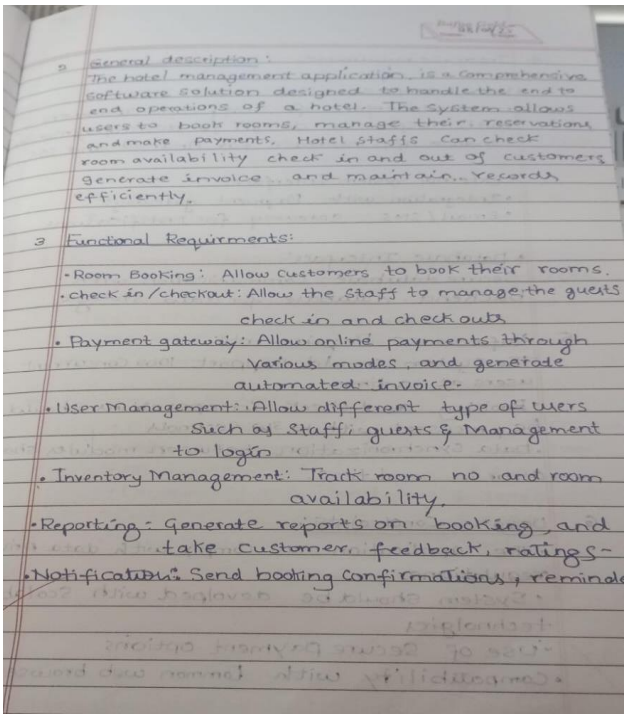
* External System Interface:
- Integration with Payment gateways
- Email / SMS gateway for Notifications

* Database Interface:
Secure database Communication with backend database for data storage.

5 Performance Requirments:
- The System should support 1000 Concurrent users at a time.
- Booking and check in operations should be processed with 10 3/10 seconds
- Data synchronization between modules should occur in real-time.

6 Design Constraints:
- The application must comply with data privacy regulations.
- System should be devolped with scalable technologies
- Use of Secure payment options
- Compatibility with common web browsers.

Fig1.3



3 General description:
The hotel management application is a comprehensive software solution designed to handle the end to end operations of a hotel. The system allows users to book rooms, manage their reservations and make payments. Hotel staffs can check room availability check in and out of customers generate invoice and maintain records efficiently.

3 Functional Requirments:
- Room Booking: Allow customers to book their rooms.
- check in / checkout: Allow the staff to manage the guests check in and check outs
- Payment gateway: Allow online payments through various modes, and genetate automated invoice.
- User Management: Allow different type of users such as staff, guests & Management to login
- Inventory Management: Track room no and room availability.
- Reporting: Generate reports on booking, and take customer, feedback, ratings.
- Notifications: Send booking confirmations, reminde

Fig1.4

1.3 **Advanced class Model**

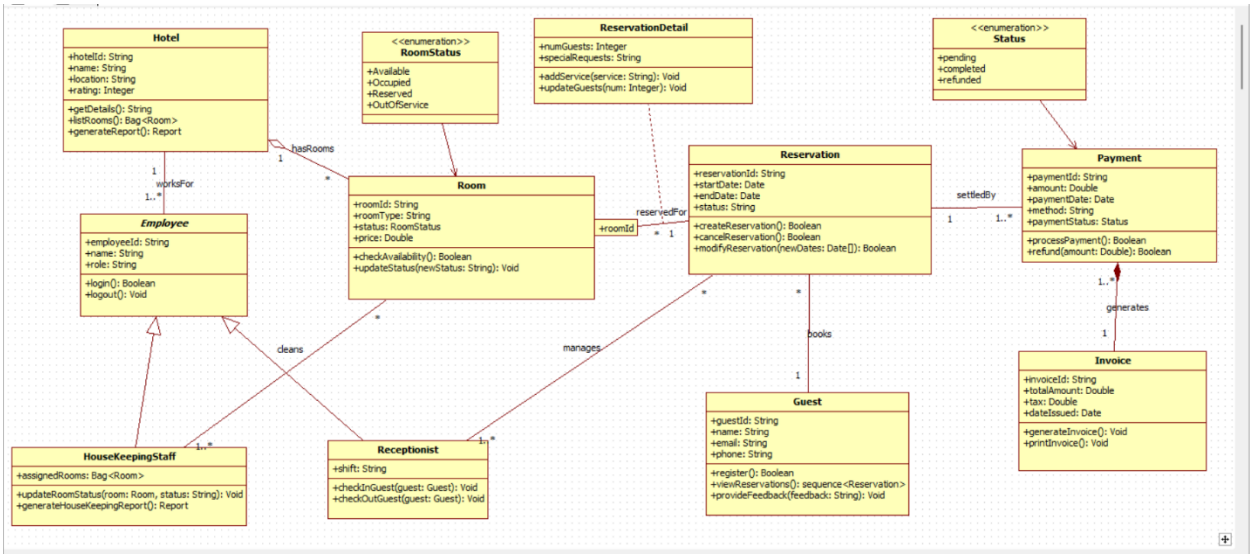


Fig 1.6

This UML diagram represents a Hotel Management System consisting of major components such as Hotel, Room, Guest, Reservation, Payment, and Invoice. The Hotel manages multiple rooms, and employees like Receptionists and Housekeeping Staff work under it. Guests make reservations linked to specific rooms, and payments are processed to generate invoices. Room and payment states are handled through defined status enumerations. Overall, the diagram shows how different system entities interact to manage hotel operations efficiently.

1.4 Simple state Model:

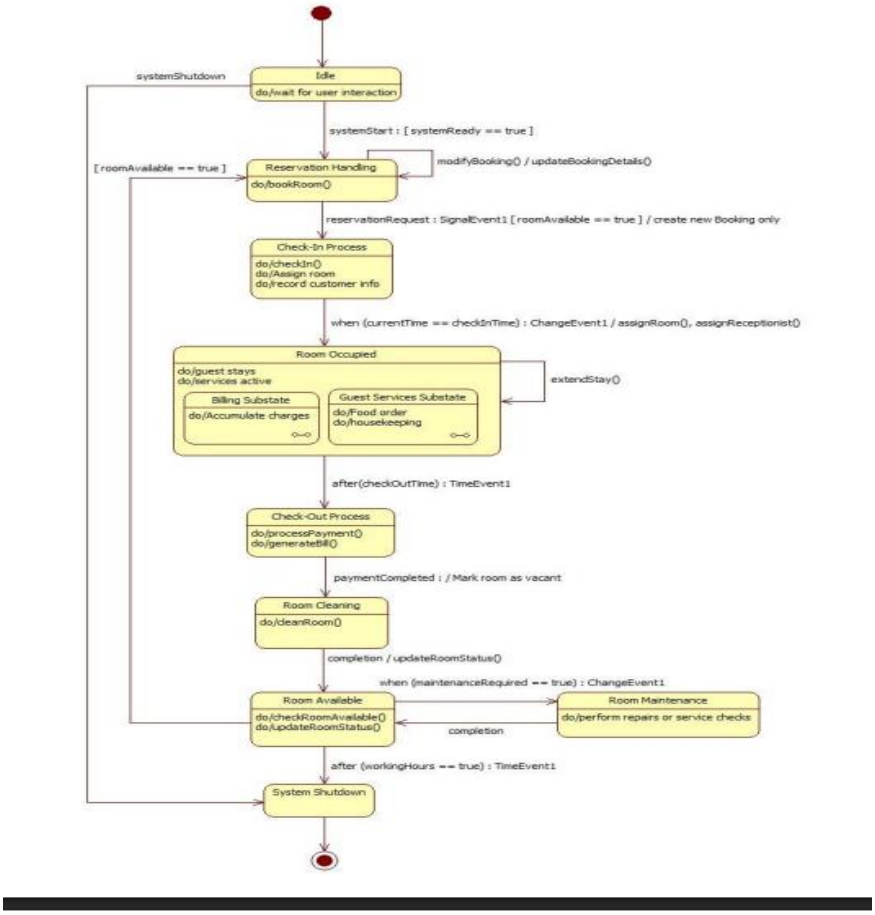


Fig 1.7

This state machine diagram represents the lifecycle of a hotel room, from reservation to checkout and maintenance. The process begins in the Idle state, where the system waits for user interaction, then moves to reservation handling and the check-in process. Once occupied, the room supports billing, guest services, and stay extensions. After checkout, the system handles payment, room cleaning, and room maintenance before marking the room as available again. Finally, the workflow ends with system shutdown when required.

1.5 Advanced State Model:

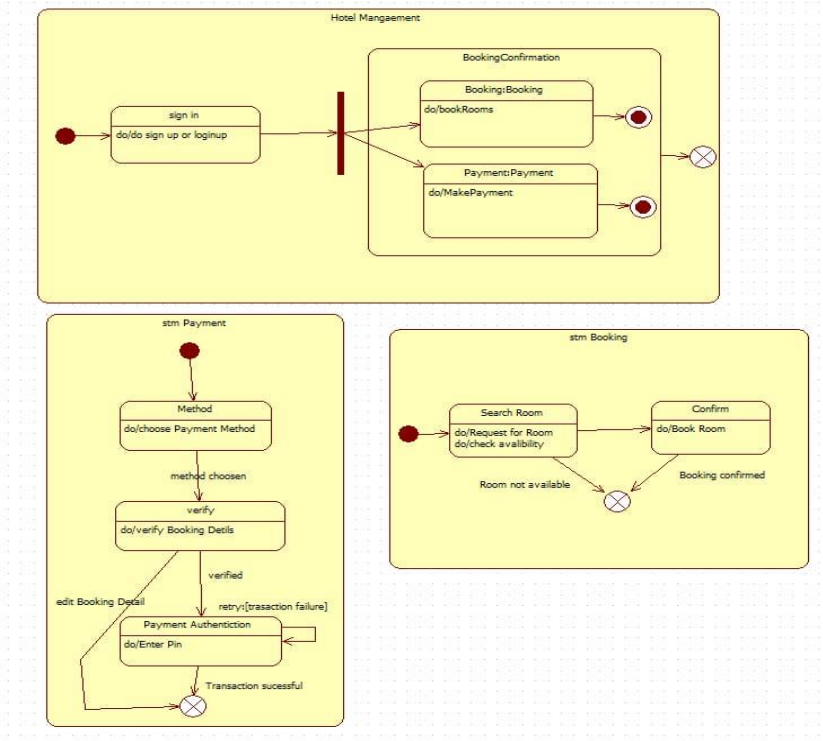


Fig 1.8

This diagram represents the hotel management workflow involving sign-in, room booking, and payment processing. After the user signs in, the system enters a parallel region where booking and payment activities proceed simultaneously. The booking subsystem handles searching for rooms, checking availability, and confirming the booking if a room is available. The payment subsystem manages selecting a payment method, verifying booking details, authenticating the user, and completing the transaction. Together, these coordinated processes ensure a smooth and secure hotel reservation experience.

1.6 Simple Sequence:

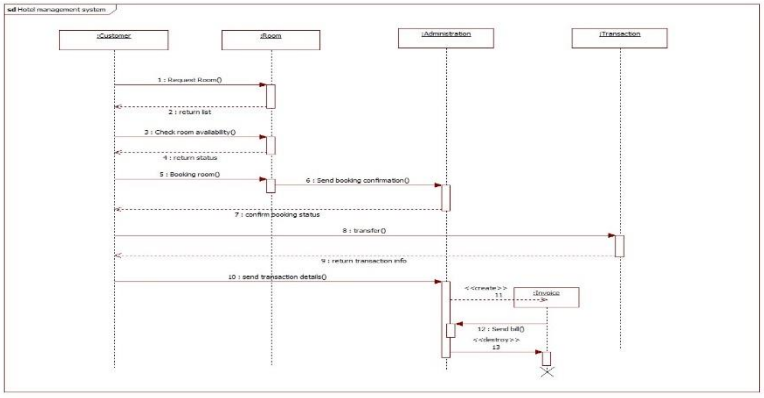


Fig 1.9

This sequence diagram shows the complete flow of booking a hotel room. The guest first sends a booking request, after which the reservation system checks room availability with the hotel system and returns the available options. Once the guest confirms the booking, the reservation system creates a booking record and initiates payment through the payment gateway. If the payment succeeds, the booking is confirmed and a confirmation message is sent to the guest. If payment fails, the system notifies the guest and requests an alternative payment method.

1.7 Advanced Sequence

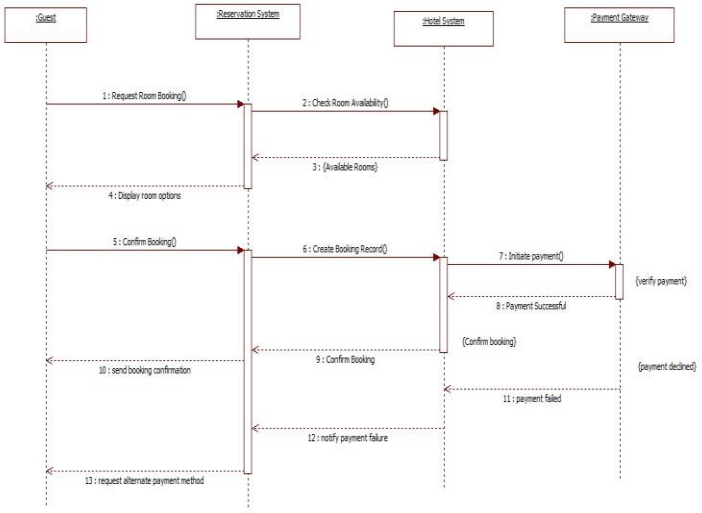


Fig 1.10

The sequence diagram shows the interaction between the Guest, Reservation System, Hotel System, and Payment Gateway during room booking. It covers checking availability, confirming the booking, and processing payment. Based on payment success or failure, the system either sends confirmation or asks for another payment method.

1.8 Simple and Advanced use case:

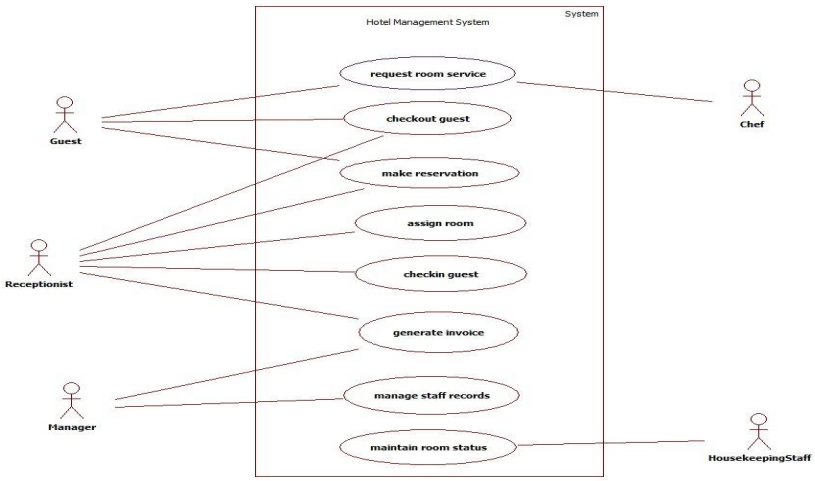


Fig 1.11

This use-case diagram shows how different users interact with the Hotel Management System. Guests, receptionists, managers, chefs, and housekeeping staff perform tasks like reservations, check-in/out, room service, invoicing, and room maintenance. Each actor accesses only the functions relevant to their role.

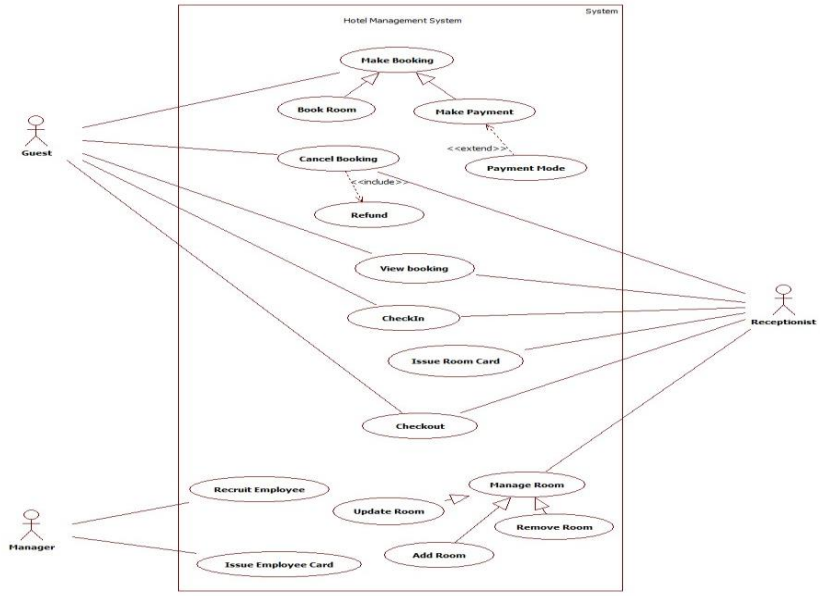


Fig 1.12

This use-case diagram shows how guests, receptionists, and managers interact with the Hotel Management System for booking, payments, check-in/check-out, room management, and employee handling. It includes extended and included use cases such as payment modes .

1.9 Simple Activity :

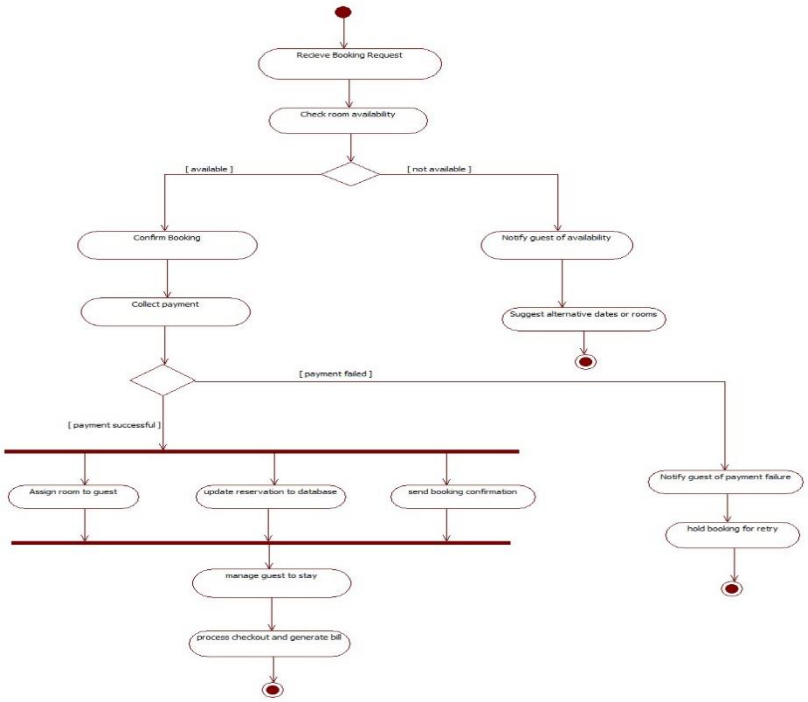


Fig 1.13

This activity diagram shows the flow of hotel room booking from receiving the request to checkout. It includes checking room availability, confirming booking, handling payment success or failure, updating the reservation, assigning rooms, and managing the guest's stay. Both successful and failed payment paths are shown clearly, ending with either booking confirmation or retry options.

1.10 Advanced Activity:

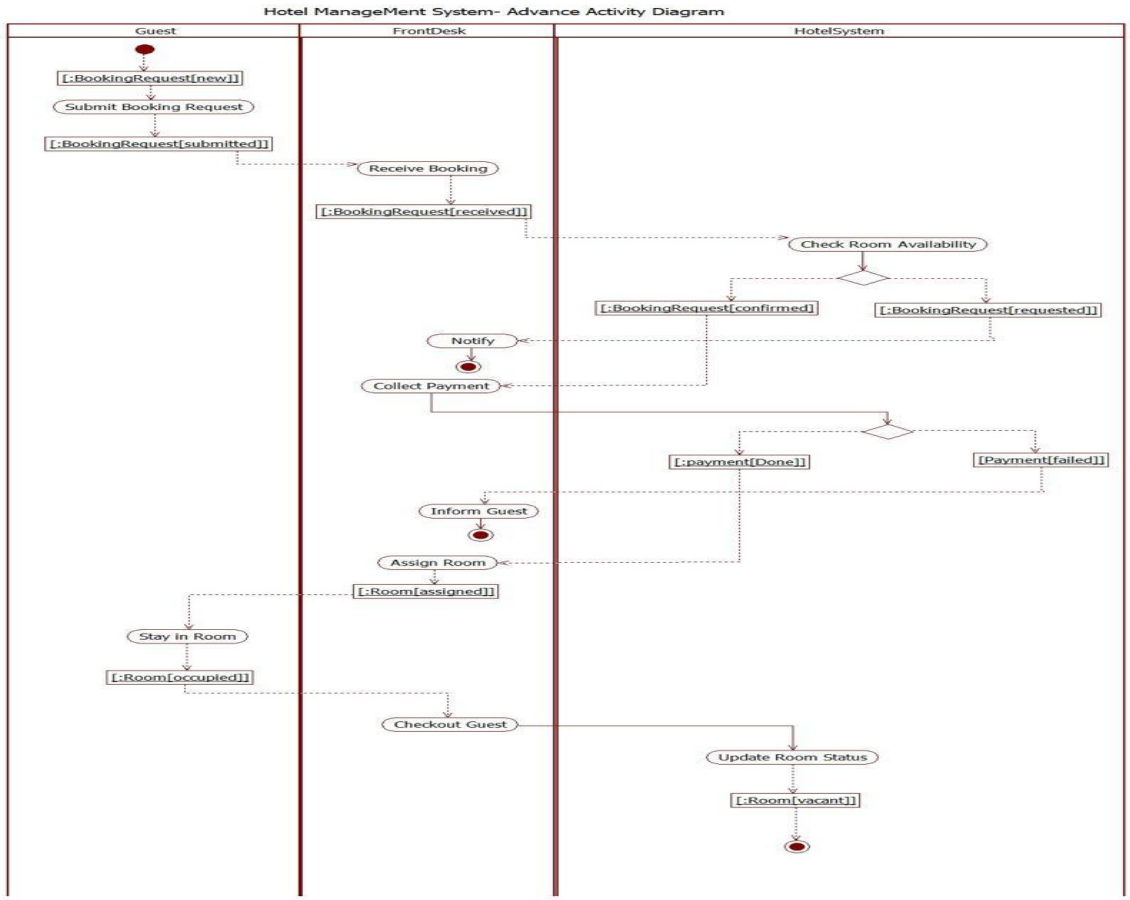


Fig 11.14

This activity diagram shows how a guest submits a booking request, which is received by the front desk and processed by the hotel system to check room availability. If a room is available, payment is collected and the guest is informed, followed by room assignment. The guest then stays in the room and later checks out. Finally, the hotel system updates the room status back to vacant, completing the flow.

2 Credit Card Processing System

2.1 Problem Statement

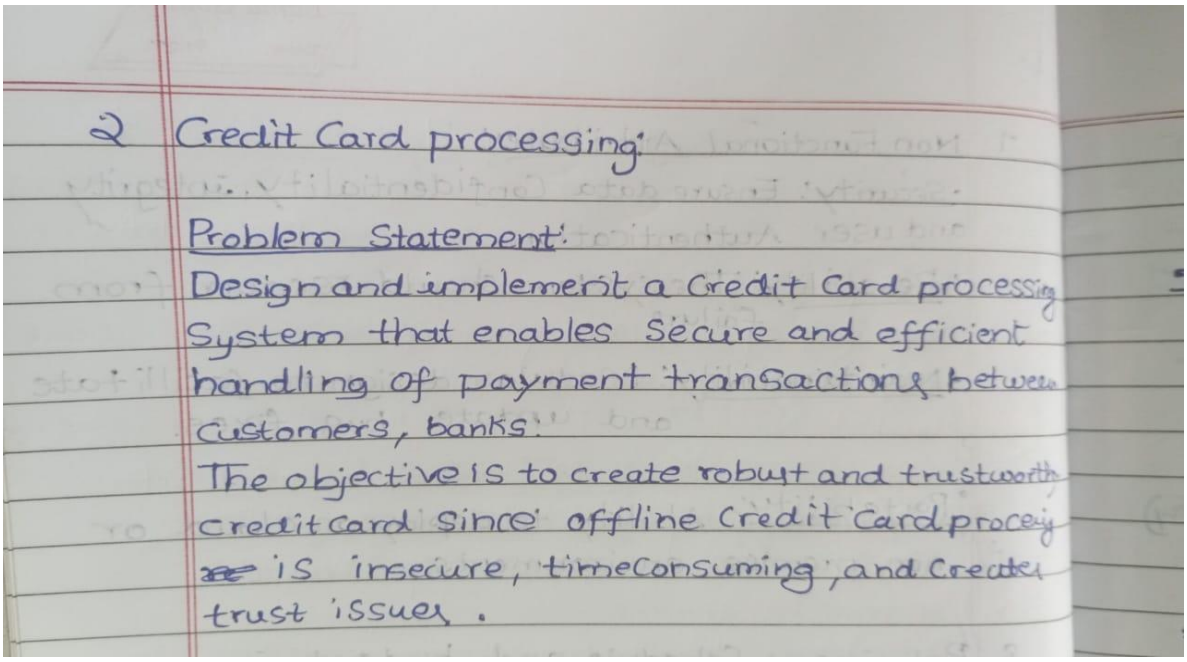
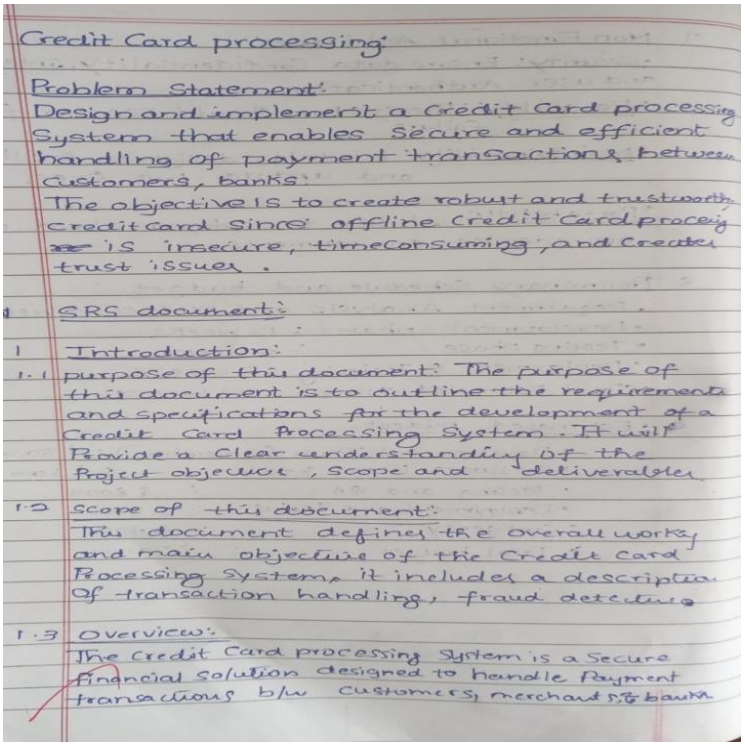
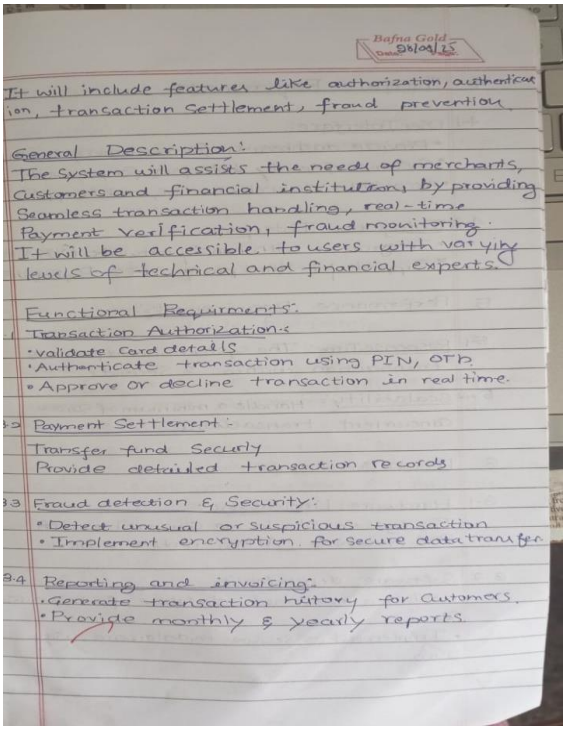


Fig 2.1

2.2 SRS Document





It will include features like authorization, authentication, transaction Settlement, fraud prevention.

General Description:
The System will assists the needs of merchants, customers and financial institutions by providing seamless transaction handling, real-time Payment verification, fraud monitoring.
It will be accessible to users with varying levels of technical and financial experts.

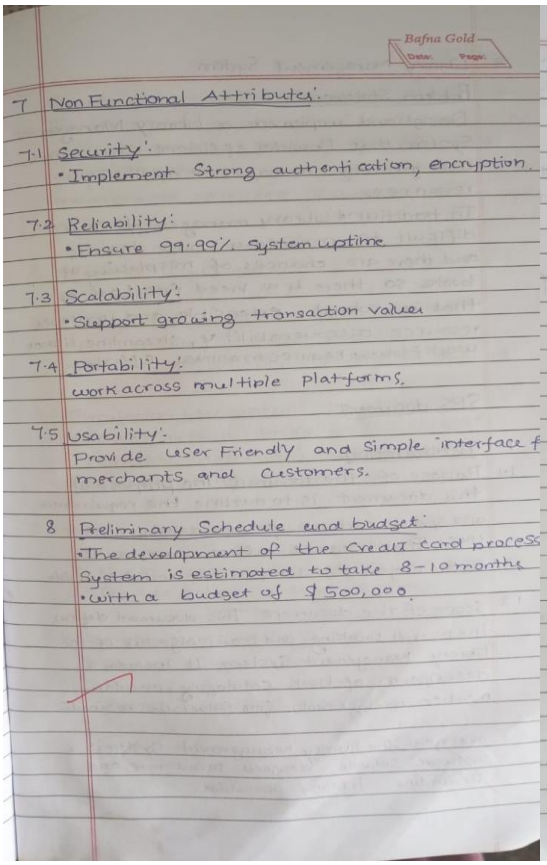
Functional Requirements:
Transaction Authorization:
- Validate card details
- Authenticate transaction using PIN, OTP.
- Approve or decline transaction in real time.

3.2 Payment Settlement:
Transfer fund Securly
Provide detailed transaction records

3.3 Fraud detection & Security:
- Detect unusual or suspicious transaction
- Implement encryption for secure data transfer

3.4 Reporting and invoicing:
- Generate transaction history for customers.
- Provide monthly & yearly reports.

Fig 2.3



7 Non Functional Attributes:

7.1 Security:
- Implement Strong authentication, encryption.

7.2 Reliability:
- Ensure 99.99% System uptime

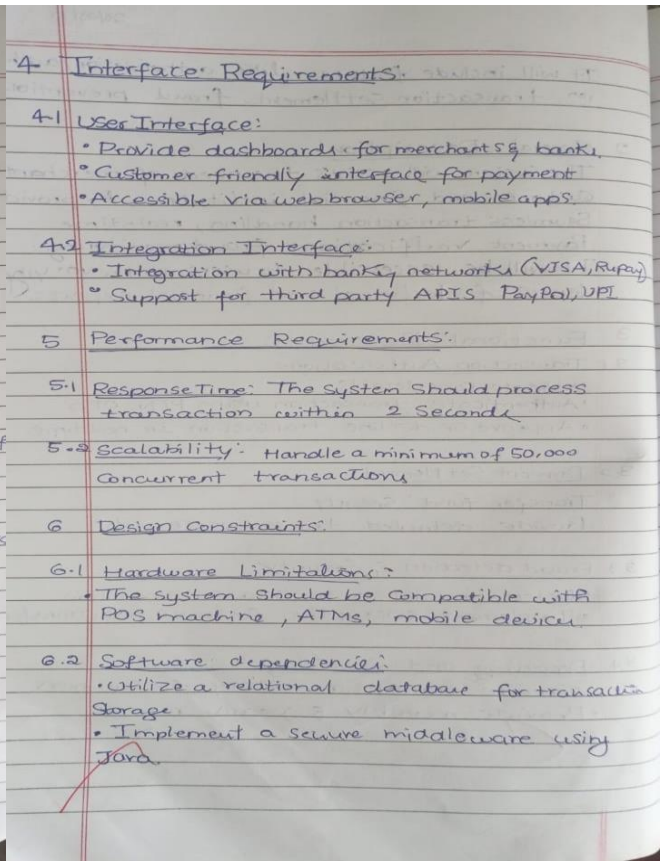
7.3 Scalability:
- Support growing transaction values

7.4 Portability:
work across multiple platforms.

7.5 Usability:
Provide User Friendly and Simple interface for merchants and Customers.

8 Preliminary Schedule and budget:
- The development of the Credit card process System is estimated to take 8-10 months
- with a budget of $500,000.

Fig 2.4



4 Interface Requirements:

4.1 User Interface:
- Provide dashboards for merchants & banks.
- Customer friendly interface for payment
- Accessible via web browser, mobile apps.

4.2 Integration Interface:
- Integration with banking networks (VISA, RuPay)
- Support for third party APIS PayPal, UPI

5 Performance Requirements:

5.1 Response Time: The System Should process transaction within 2 Seconds

5.2 Scalability: Handle a minimum of 50,000 Concurrent transactions

6 Design Constraints:

6.1 Hardware Limitations:
- The System Should be Compatible with POS machine, ATMs, mobile devices.

6.2 Software dependencies:
- Utilize a relational database for transaction storage.
- Implement a secure middleware using Java.

Fig 2.5

2.3 Advanced class Diagram:

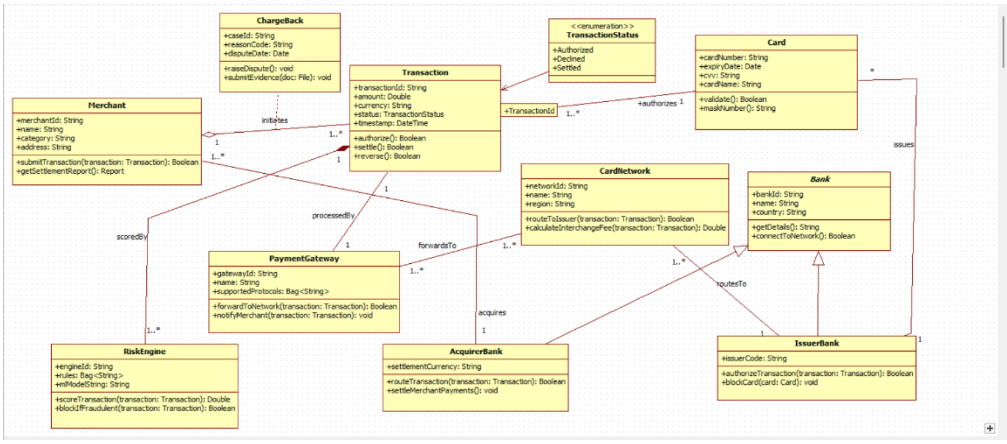


Fig 2.6

The diagram represents a complete card payment processing system involving merchants, payment gateways, banks, and card networks. A Merchant initiates a transaction, which is processed by the Payment Gateway and scored by the Risk Engine for fraud detection. The transaction is then routed through the Card Network to the Issuer Bank for authorization.

2.4 Simple State Diagram

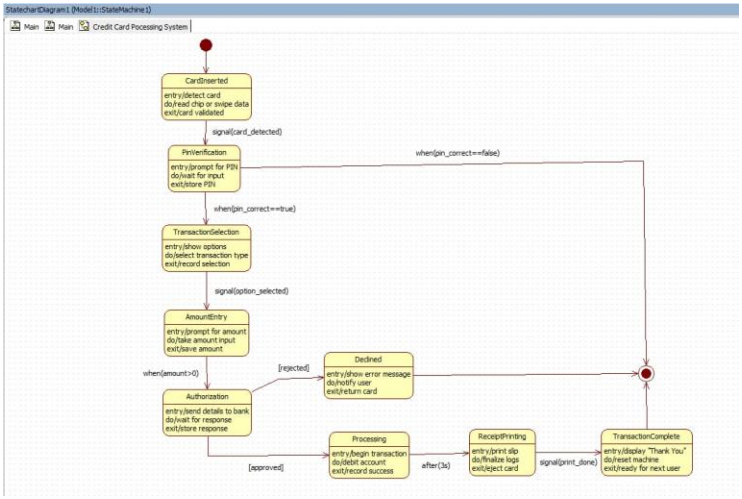


Fig 2.7

This state diagram shows the steps in a credit card processing system, starting from card insertion and PIN verification to transaction selection and authorization. Based on approval or rejection, the system either processes the payment and prints a receipt or displays an error. Finally, it completes the transaction and resets for the next user.

2.5 Advanced State Diagram

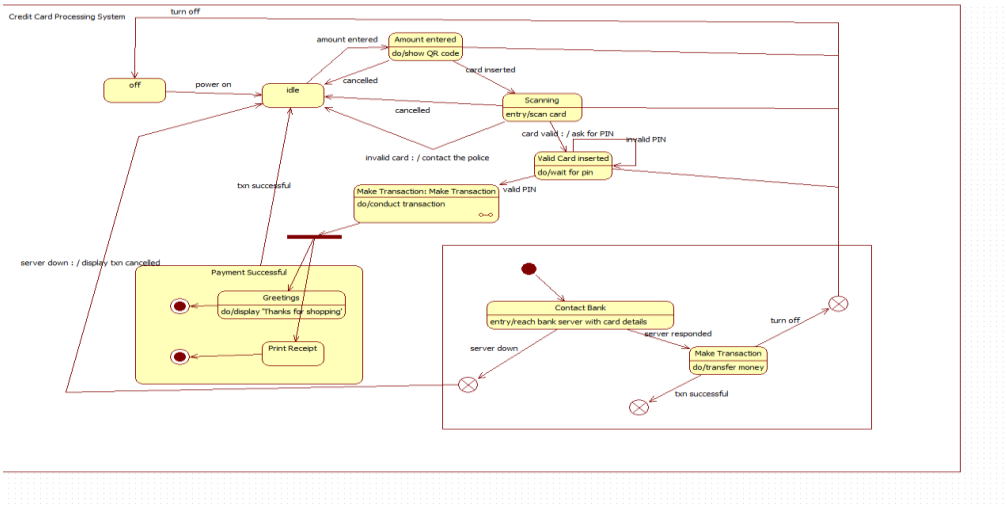


Fig 2.8

This state machine represents the credit card processing workflow, starting from the system being powered on and entering the idle state. When an amount is entered and the card is scanned, the system verifies the card and PIN before proceeding with the transaction. If valid, it contacts the bank, processes the payment, and then moves to the Payment Successful composite state where it displays a thank-you message and prints a receipt.

2.6 Simple Sequence Diagram

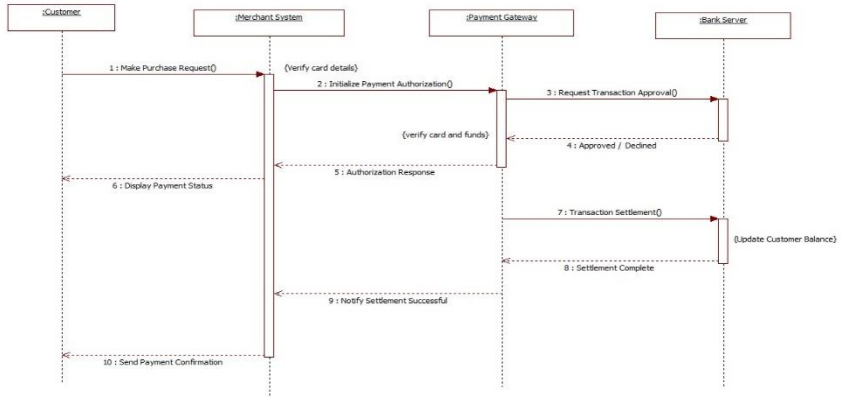


Fig 2.9

This sequence diagram shows how a customer's payment request moves through the merchant system, payment gateway, and bank server for authorization and settlement. Each component verifies details, approves or declines the transaction, and completes settlement. Finally, the customer is notified of the payment status and confirmation**.**

2.7 Advanced Sequence Diagram:

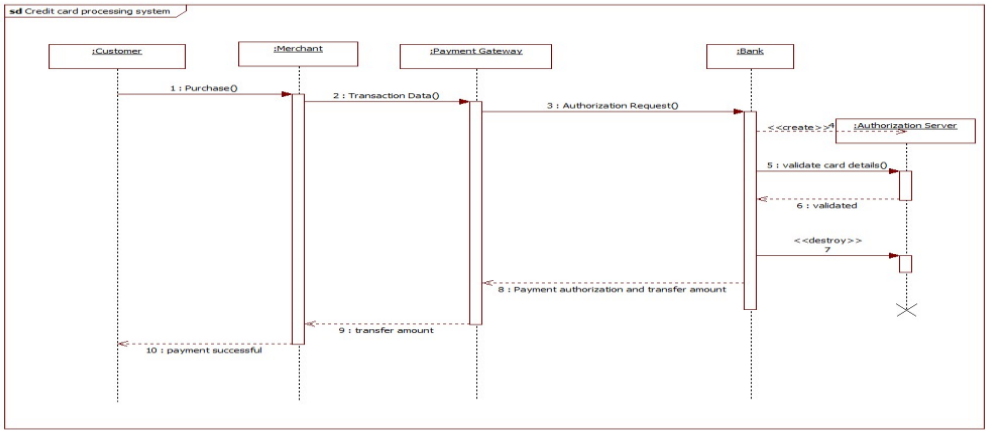


Fig 2.10

This sequence diagram shows how a customer's purchase request is sent to the merchant, forwarded to the payment gateway, and then validated by the bank's authorization server. Once the card details are verified, the bank approves the transaction and transfers the amount. The merchant receives confirmation, and the customer is informed that the payment is successful.

2.8 Simple Use case

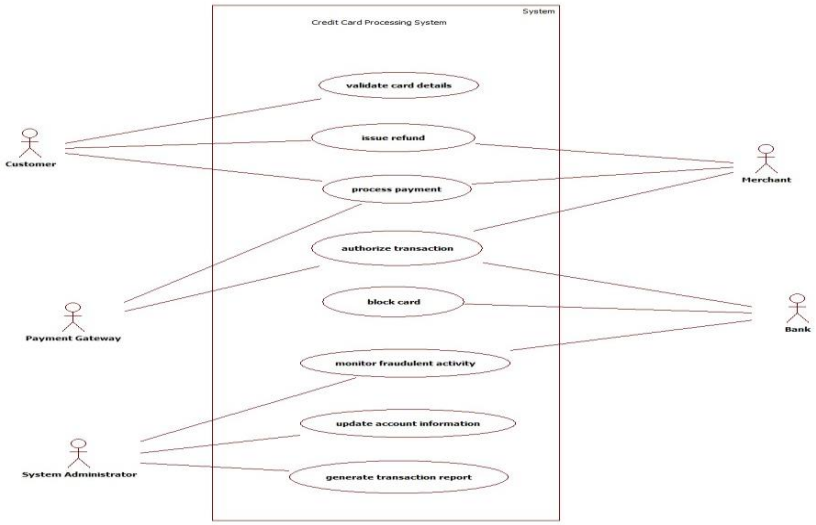


Fig 2.11

This use-case diagram represents a Credit Card Processing System involving multiple actors like Customer, Merchant, Bank, Payment Gateway, and System Administrator. It shows core operations such as validating card details, processing payments, authorizing transactions, issuing refunds, and blocking cards.

2. 9 Advanced Use case

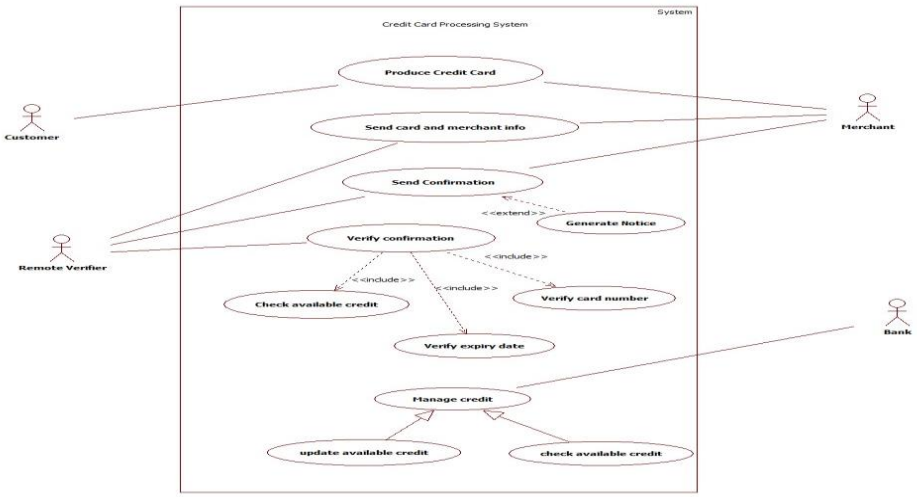


Fig 2.12

The diagram illustrates a Credit Card Processing System involving a Customer, Merchant, Remote Verifier, and Bank. It highlights key use cases such as producing a credit card, sending card and merchant information, and verifying confirmations. The verification process includes checking available credit, validating the card number, and confirming the expiry date. Additionally, the system manages credit by updating and checking the available credit. Optional actions, such as generating notices, further extend the confirmation process.

2.10 Simple Activity Diagram

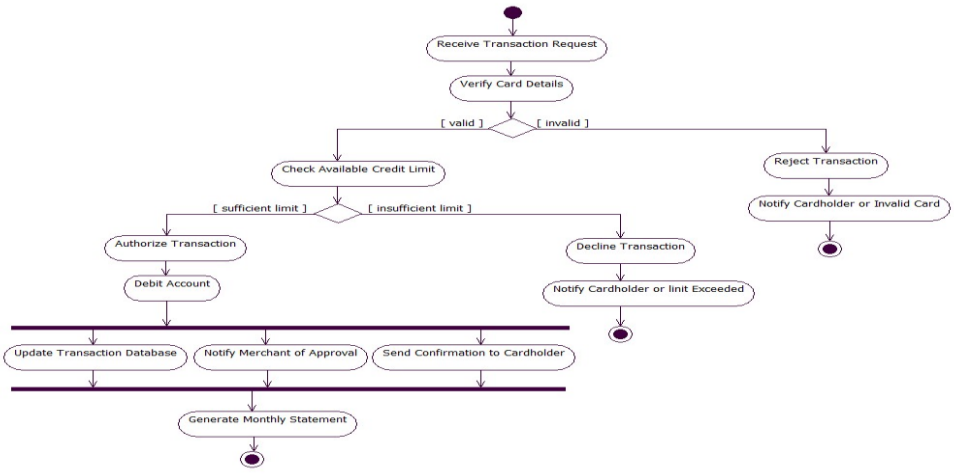


Fig 2.13

This activity diagram shows the credit card transaction processing flow. The system first verifies card details, checks the available credit limit, and then either authorizes or declines the transaction. If authorized, the account is debited, records are updated, and notifications are sent

to the merchant and cardholder. If declined or invalid, the system immediately notifies the cardholder and ends the process.

2.11 Advanced Activity Diagram

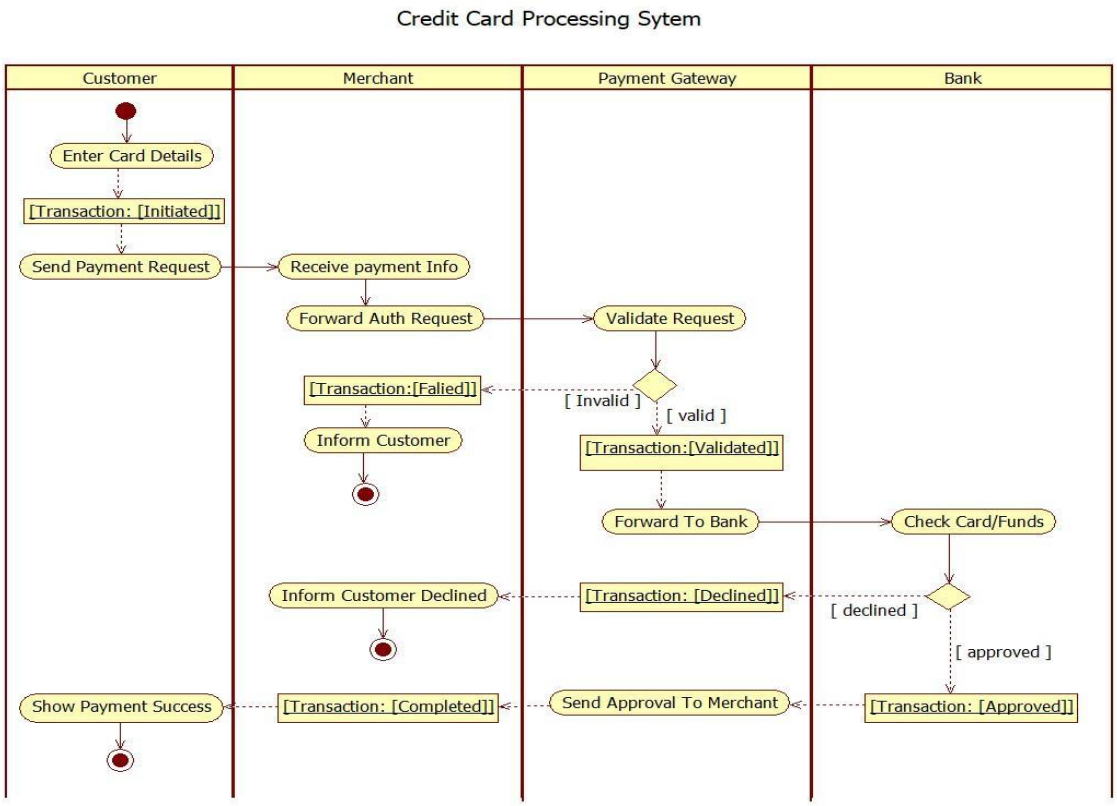
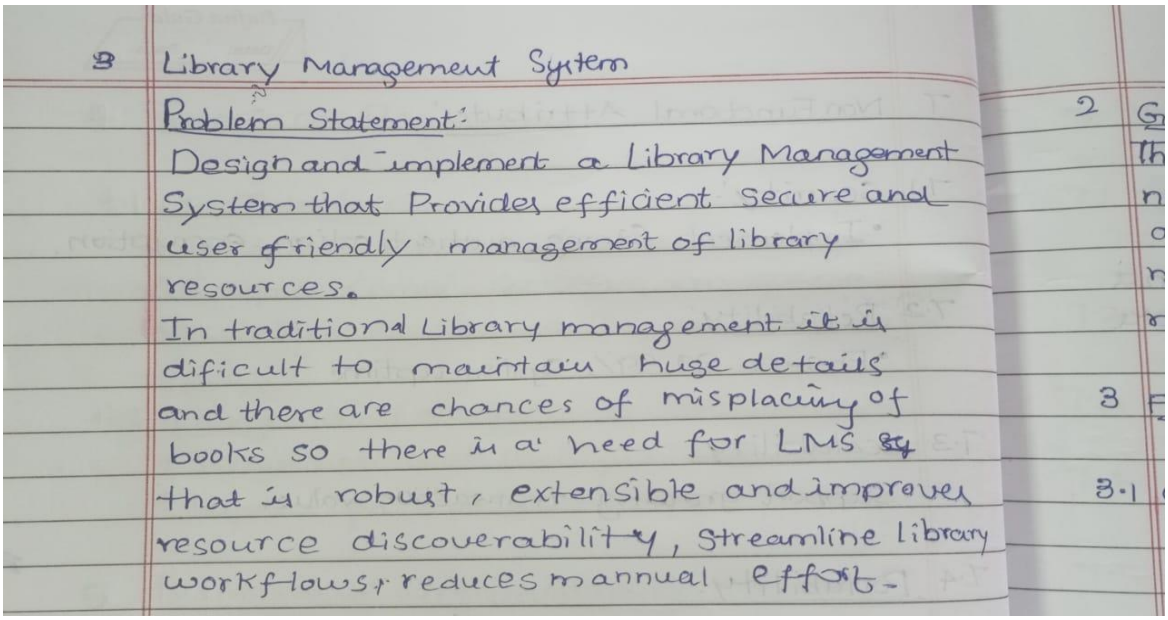


Fig 2.14

The diagram illustrates the complete flow of a credit card payment process involving the Customer, Merchant, Payment Gateway, and Bank. After the customer enters the card details, the merchant receives the payment information and forwards an authorization request to the payment gateway. The gateway then validates the request and forwards it to the bank, where the card details and available funds are checked. Based on whether the transaction is approved or declined, the system sends the appropriate response back to the customer, completing or rejecting the transaction accordingly.

3 Library Management System



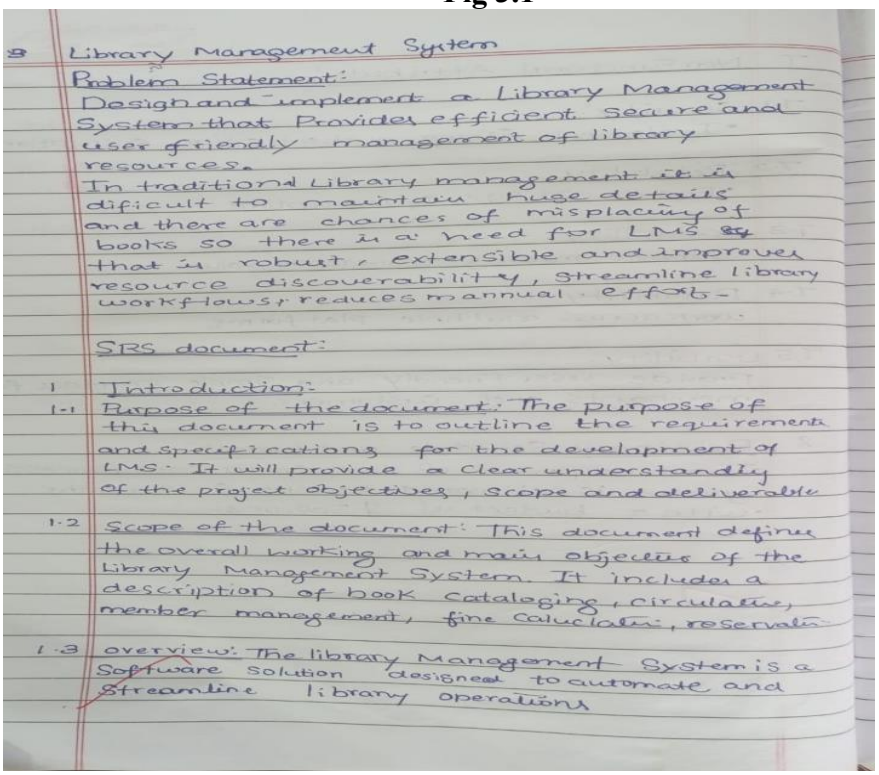
3 Library Management System

Problem Statement:

Design and implement a Library Management System that Provides efficient secure and user friendly management of library resources.

In traditional Library management it is dificult to maintain huge details and there are chances of misplacing of books so there is a need for LMS that is robust, extensible and improves resource discoverability, streamline library workflows, reduces manual effort.

Fig 3.1



3 Library Management System

Problem Statement:

Design and implement a Library Management System that Provides efficient secure and user friendly management of library resources.

In traditional Library management it is dificult to maintain huge details and there are chances of misplacing of books so there is a need for LMS that is robust, extensible and improves resource discoverability, streamline library workflows, reduces manual effort.

SRS document:

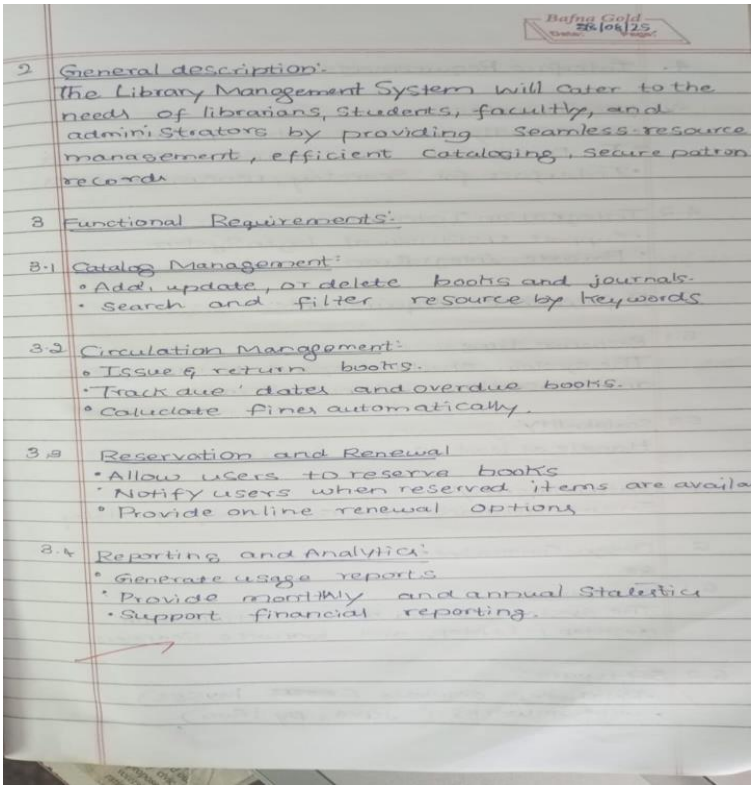
1 Introduction:

1.1 Purpose of the document: The purpose of this document is to outline the requirements and specifications for the development of LMS. It will provide a clear understanding of the project objectives, scope and deliverable

1.2 Scope of the document: This document defines the overall working and main objectives of the Library Management System. It includes a description of book cataloging, circulation, member management, fine calculation, reservation

1.3 overview: The library Management System is a Software solution designed to automate and Streamline library operations

Fig 3.2



2 General description:
The Library Management System will cater to the needs of librarians, students, faculty, and administrators by providing seamless resource management, efficient cataloging, secure patron records.

3 Functional Requirements:

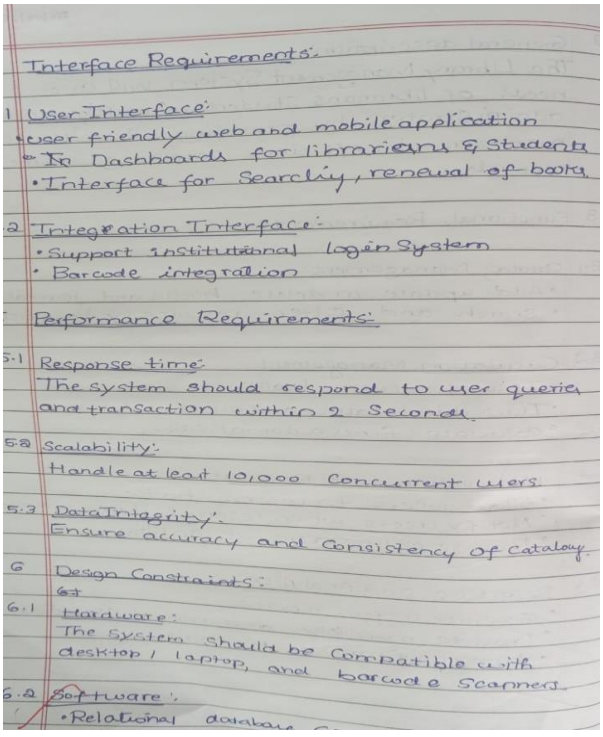
3.1 Catalog Management:
- Add, update, or delete books and journals.
- Search and filter resource by keywords

3.2 Circulation Management:
- Issue & return books.
- Track due dates and overdue books.
- Calculate fines automatically.

3.3 Reservation and Renewal
- Allow users to reserve books
- Notify users when reserved items are availa
- Provide online renewal options

3.4 Reporting and Analytics:
- Generate usage reports
- Provide monthly and annual Statistics
- Support financial reporting.

Fig 3.3



Interface Requirements:

1 User Interface:
- user friendly web and mobile application
- Dashboards for librarians & students
- Interface for Searchy, renewal of books

2 Integration Interface:
- Support institutional login System
- Barcode integration

Performance Requirements:

5.1 Response time:
The system should respond to user queries and transaction within 2 Seconds.

5.2 Scalability:
Handle at least 10,000 Concurrent users

5.3 Data Integrity:
Ensure accuracy and Consistency of cataloy

6 Design Constraints:

6.1 Hardware:
The System should be Compatible with desktop / laptop, and barcode Scanners.

6.2 Software:
- Relational

Fig 3.4

3.2 Advanced Class Model

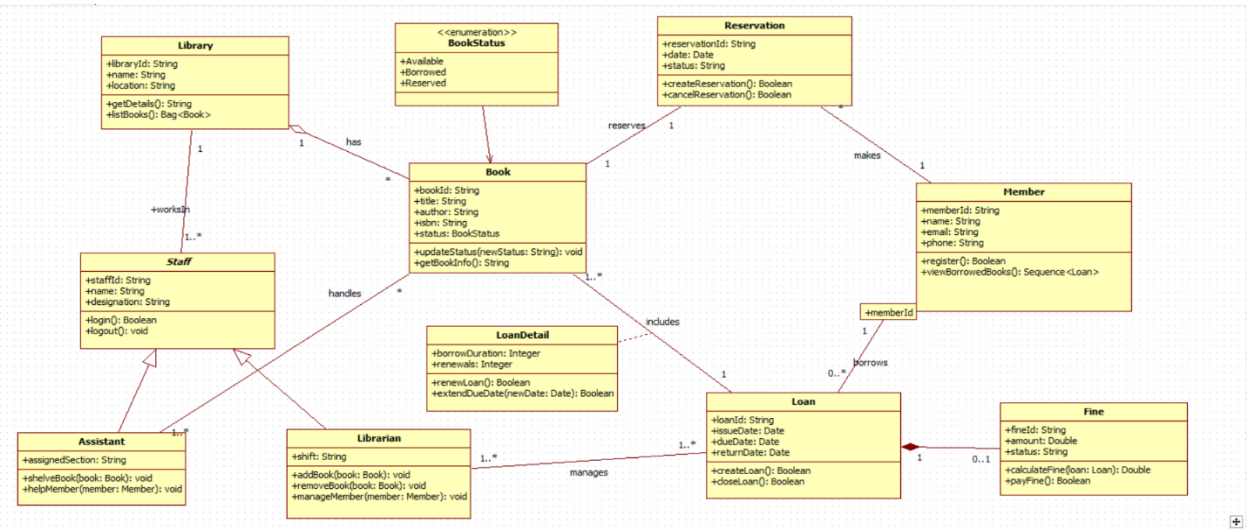


Fig 3.5

This UML diagram shows a library management system with entities such as Library, Book, Member, Staff, Loan, Reservation, and Fine and how they interact.

It illustrates relationships like members borrowing books, staff managing books and members, and reservations/loans connecting members to books.

Classes include key attributes and methods, capturing how the system tracks book

status, reservations, loans, and fines.

3.3 Simple State

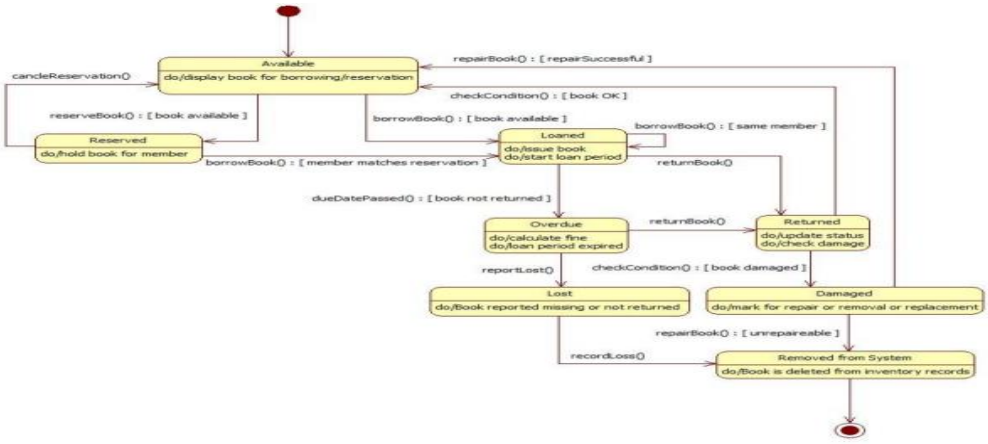


Fig 3.6

his diagram represents the lifecycle of a library book using a UML state machine model. A book transitions between states such as *Available, Reserved, Loaned, Overdue, Returned,* and *Damaged* based on user actions like borrow, return, or reserve.
Each state includes system operations (e.g., calculating fines, updating records, checking conditions) that maintain the book's status accurately.
If the book becomes lost or irreparable, it eventually reaches the Removed from System state, where it is deleted from the inventory.

3.4 Advanced State

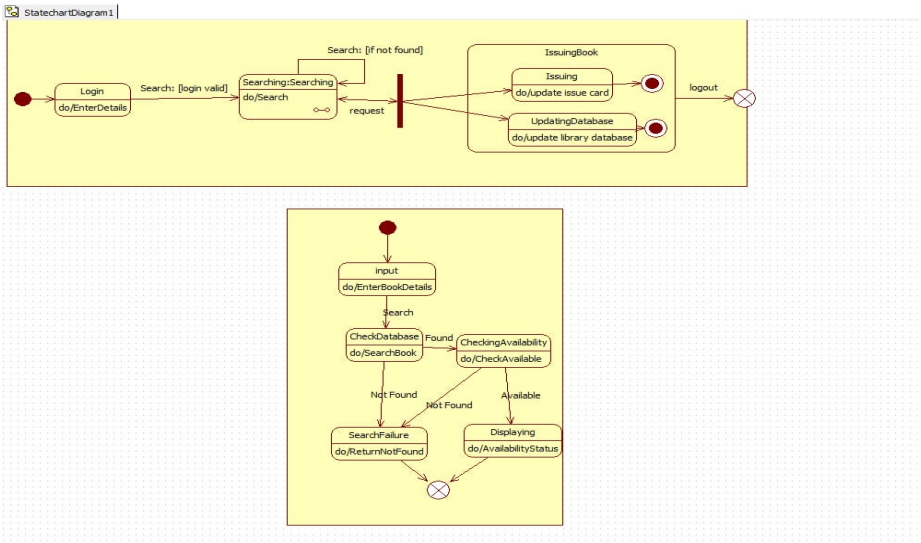


Fig 3.7

This statechart diagram represents the process of searching and issuing a book in a library system. It begins with user login, followed by searching for a book in the database and checking its availability.
If the book is found and available, the system proceeds with issuing the book and updating database records.
If the book is not found or unavailable, the process ends with a failure status and no book issuance.

3.5 Simple Sequence

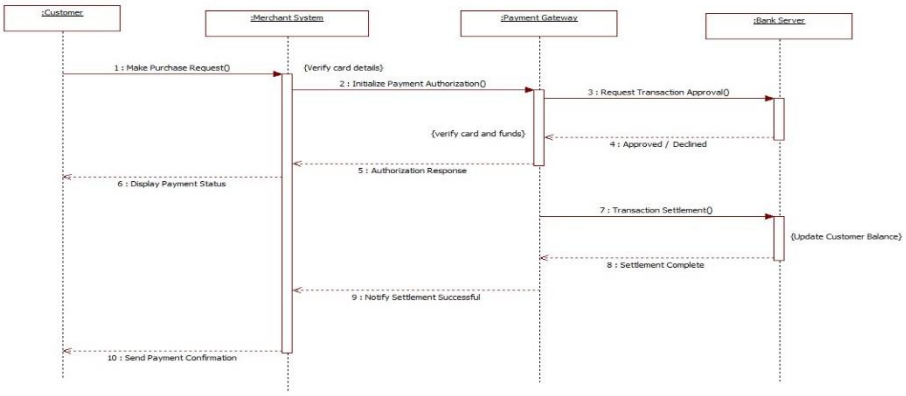


Fig 3.7

This sequence diagram shows the interaction between the customer, merchant system, payment gateway, and bank during an online payment process.
After the customer initiates a purchase, the merchant system requests authorization through the payment gateway, which verifies the transaction with the bank.
Once approved and settled, the system updates records and sends payment confirmation back to the customer.

3.6 Advanced Sequence

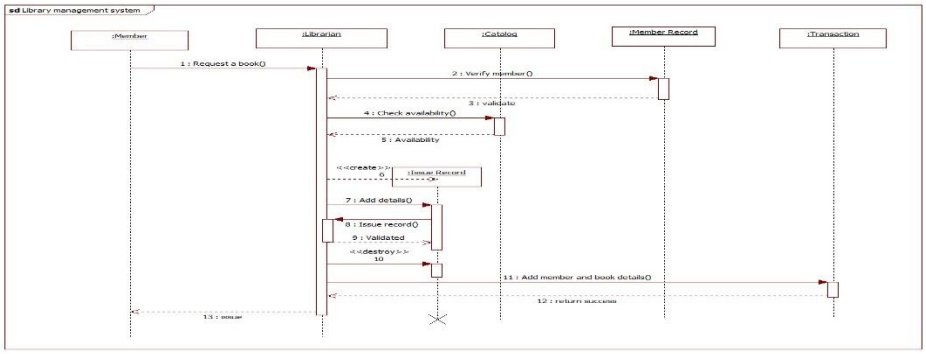


Fig 3.8

This sequence diagram explains how a book is issued in a library system when a member makes a request. The librarian verifies the member, checks book availability, creates an issue record, and updates member and transaction details. Finally, the system confirms the successful issue of the book back to the member.

3.7 Simple Use case

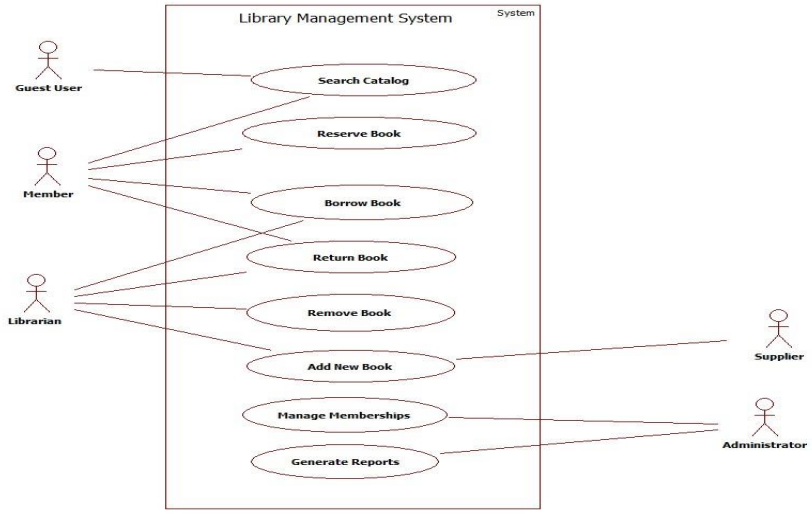


Fig 3.8

This use case diagram represents different users interacting with a Library Management System. Guest users can only search the catalog, while members can reserve, borrow, and return books. Librarians have additional permissions such as removing and adding books, managing memberships, and generating reports. The administrator and supplier are also involved in supporting system maintenance and resource updates.

3.8 Advanced Use case

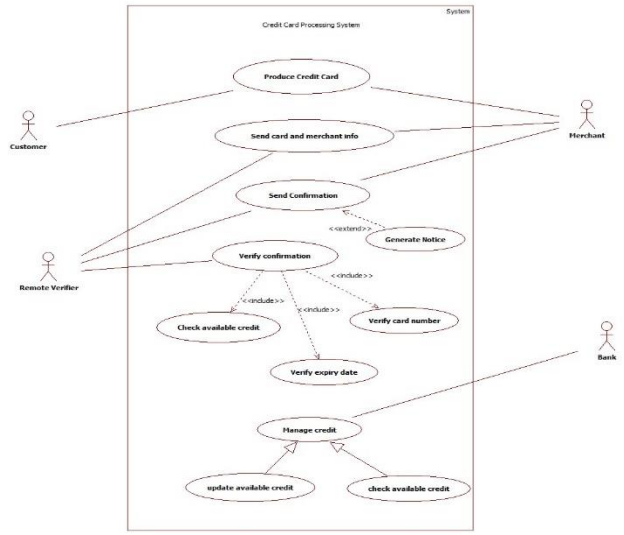


Fig 3.9

This use case diagram represents how different users interact with a Credit Card Processing System. The customer and merchant initiate transactions by sending credit card and merchant

information, which triggers verification steps like checking card number, expiry date, and available credit. The remote verifier and bank assist in validating and updating credit details, ensuring secure transaction processing. After successful verification, the system manages credit and sends confirmation or notices when required.

3.9 Simple Activity:

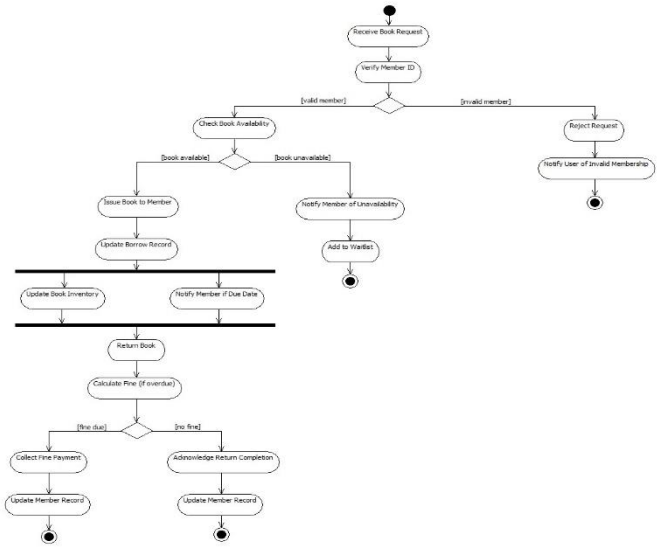
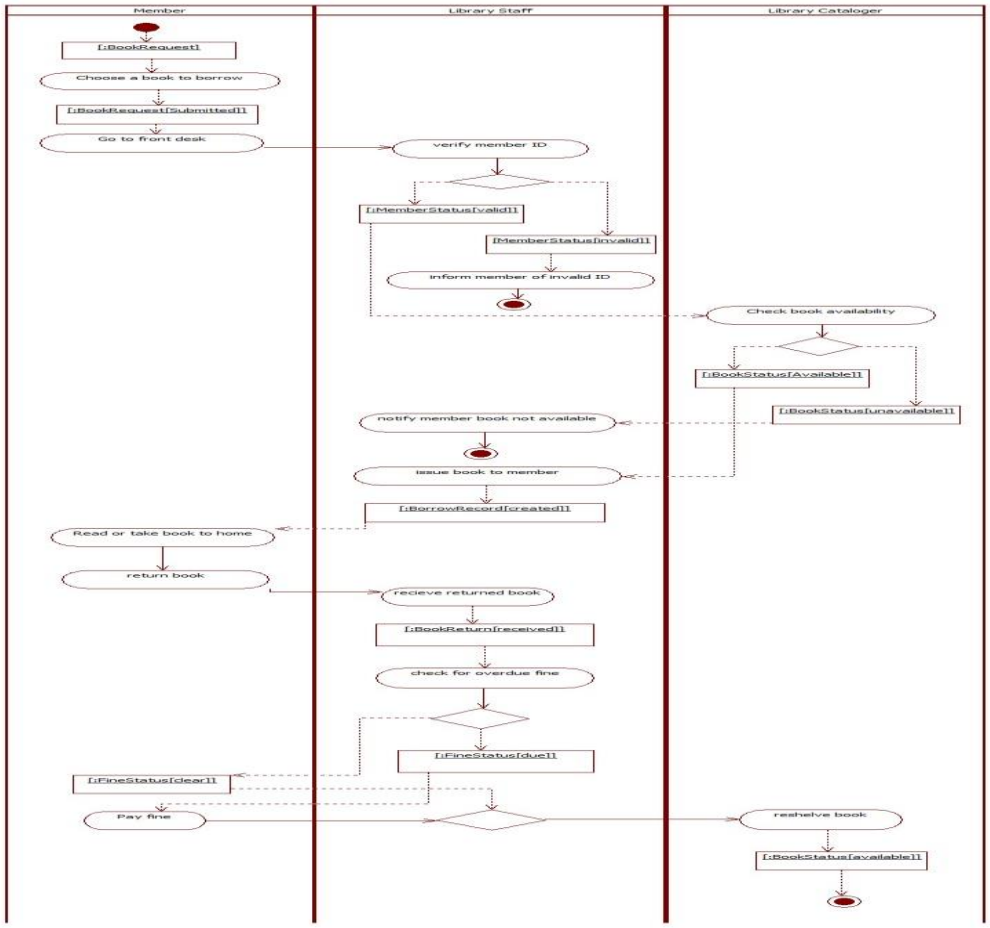


Fig 3.10

This activity diagram shows the workflow of issuing and returning a book in a library system. The process begins with verifying the member and checking book availability before issuing or placing the user on a waitlist. When the book is returned, the system calculates overdue fines if applicable and updates both the inventory and member record. Finally, the process ends after confirmation of return and any required fine payment**.**

3.10 Advanced Activity



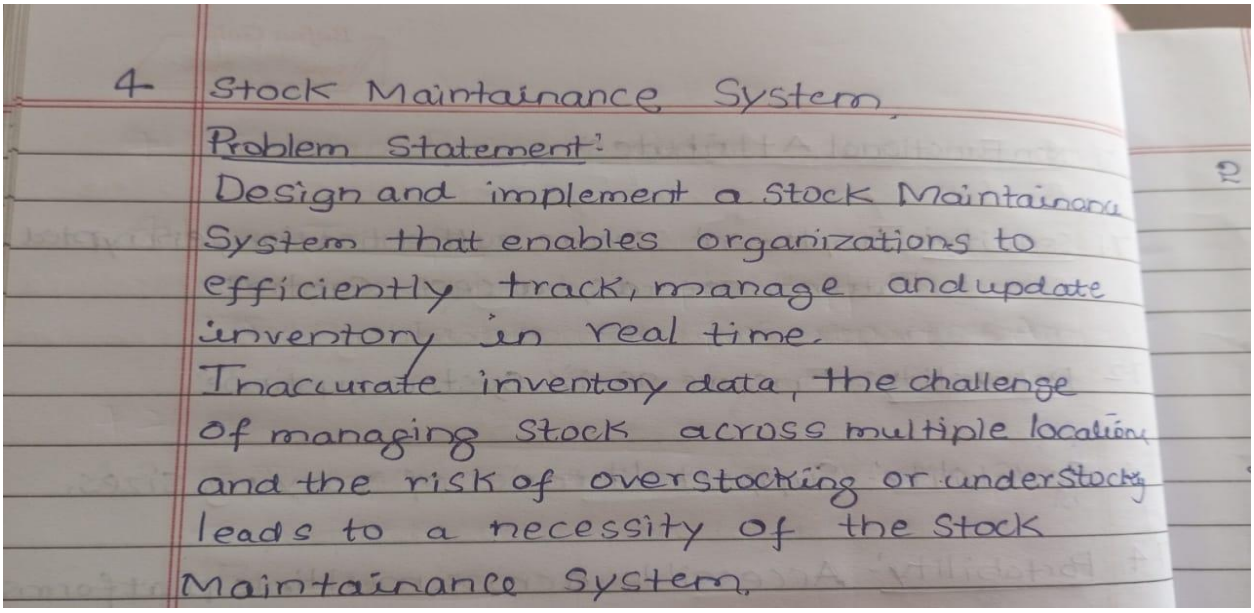
+65

Fig 3.11

This activity diagram represents the borrow and return process in a library system, involving the member, library staff, and cataloger. It includes validating the member ID, checking book availability, issuing the book, and recording the borrowing details. When the book is returned, the system checks for overdue fines, processes payments if required, and updates the inventory before reshelving the book.

4 Stock Maintenance System

4.1 Problem Statement


4 Stock Maintainance System.

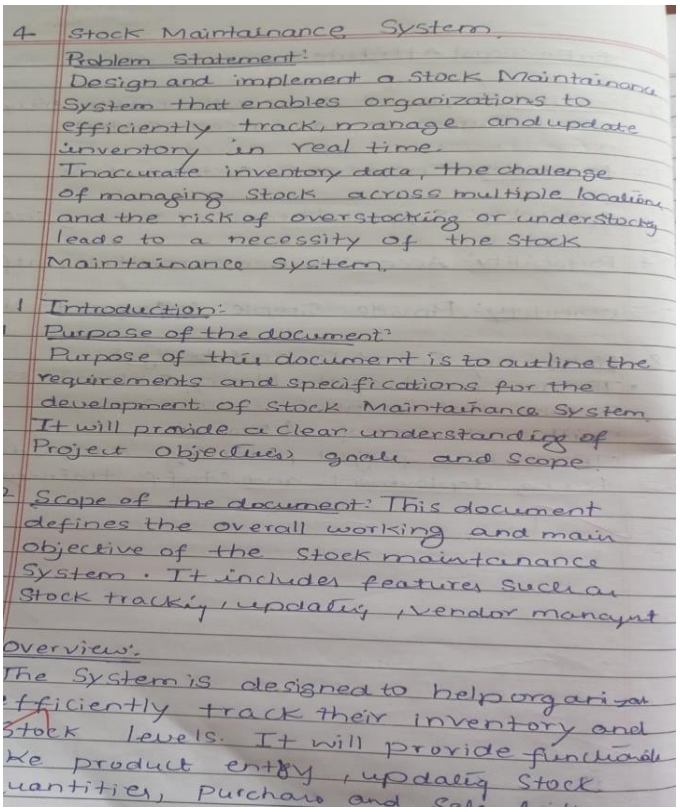
Problem Statement:

Design and implement a Stock Maintainance System that enables organizations to efficiently track, manage and update inventory in real time.

Inaccurate inventory data, the challenge of managing stock across multiple locations and the risk of overstocking or understocking leads to a necessity of the Stock Maintainance System.

Fig 4.1

4.2 SRS Document


4 Stock Maintainance System.

Problem Statement:

Design and implement a Stock Maintainance System that enables organizations to efficiently track, manage and update inventory in real time.

Inaccurate inventory data, the challenge of managing stock across multiple locations and the risk of overstocking or understocking leads to a necessity of the Stock Maintainance System.

1 Introduction:

Purpose of the document:

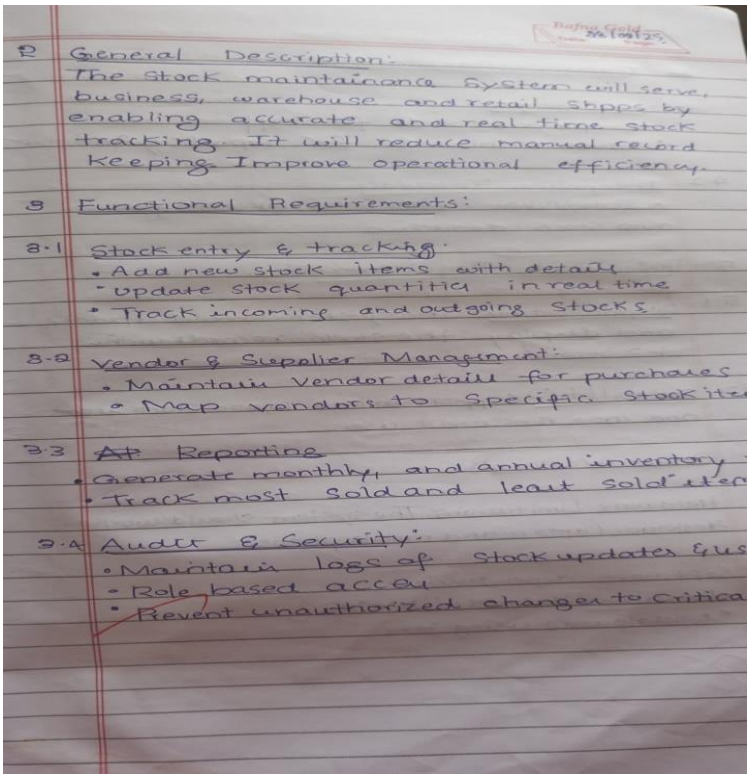
Purpose of this document is to outline the requirements and specifications for the development of Stock Maintainance System. It will provide a clear understanding of Project objectives, goals and scope.

2 Scope of the document: This document defines the overall working and main objective of the Stock maintenance System. It includes features such as Stock tracking, updating, Vendor managment

Overview:

The System is designed to help organizations efficiently track their inventory and stock levels. It will provide functionalities like product entry, updating stock quantities, Purchase and

Fig 4.2



2 General Description:
The stock maintaiance system will serve, business, warehouse and retail shops by enabling accurate and real time stock tracking. It will reduce manual record keeping. Improve operational efficiency.

3 Functional Requirements:

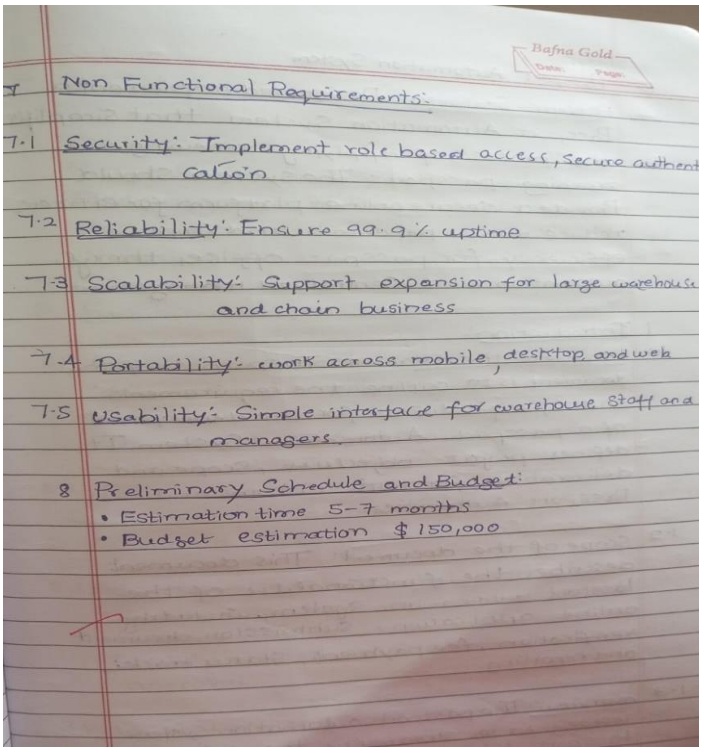
3.1 Stock entry & tracking:
- Add new stock items with details
- Update stock quantities in real time
- Track incoming and outgoing stocks

3.2 Vendor & Supplier Management:
- Maintain vendor details for purchases
- Map vendors to Specific stock ite

3.3 ~~At~~ Reporting
- Generate monthly, and annual inventory
- Track most sold and least sold item

3.4 Audit & Security:
- Maintain logs of stock updates & us
- Role based acces
- Prevent unauthorized changes to critica

Fig 4.3



7 Non Functional Requirements:

7.1 Security: Implement role based access, secure authentication

7.2 Reliability: Ensure 99.9% uptime

7.3 Scalability: Support expansion for large warehouse and chain business

7.4 Portability: work across mobile, desktop and web

7.5 Usability: Simple interface for warehouse staff and managers.

8 Preliminary Schedule and Budget:
- Estimation time 5-7 months
- Budget estimation $150,000

Fig 4.4

4.3 Advanced Class Model

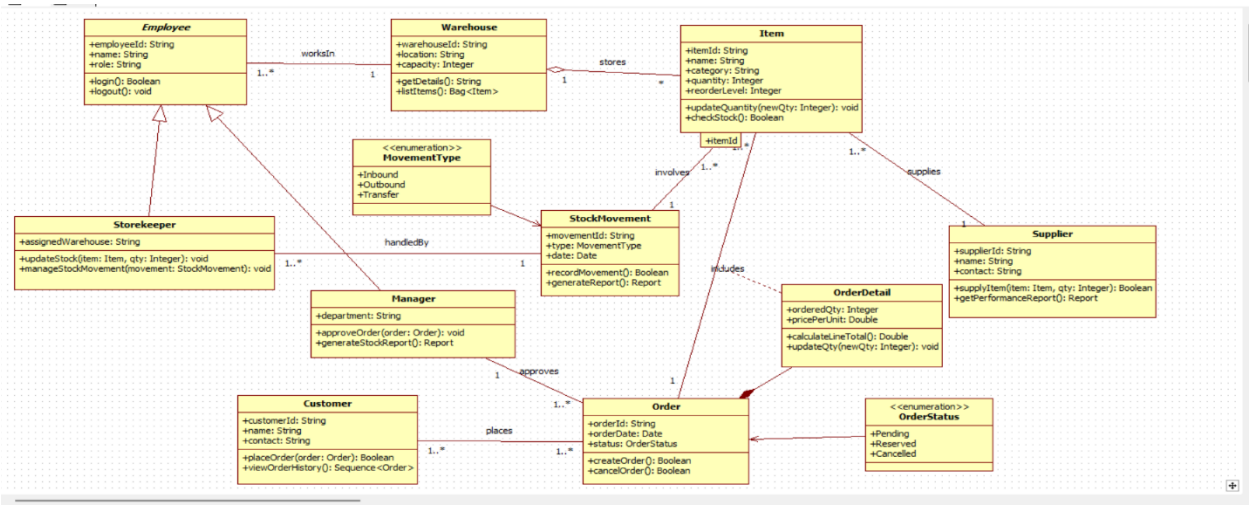


Fig 4.5

It shows relationships between core entities like Employee, Warehouse, Item, Supplier, and Customer. Managers approve orders and generate reports, while Storekeepers update stock and manage movements. Orders and stock movements track the lifecycle of items and transactions in the system.

4.4 Simple State model

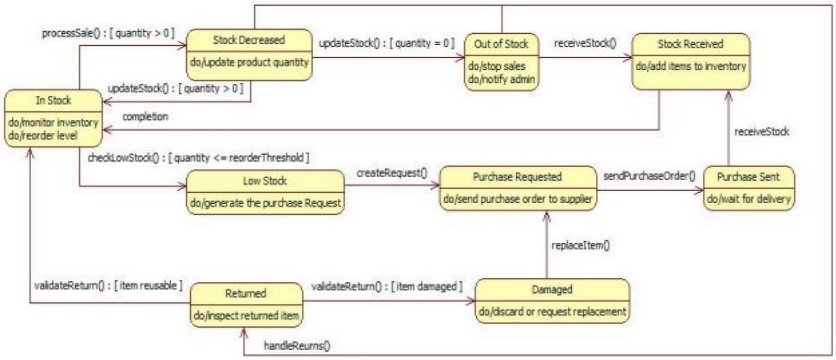


Fig 4.5

The cycle starts with items being in stock, and transitions occur based on sales, low quantity, returns, or receiving new stock. Inventory may move to states such as stock decreased, out of stock, low stock, purchase requested, or damaged. Actions like generating purchase requests and handling returns ensure inventory control and replenishment.

4.5 Advanced State Model

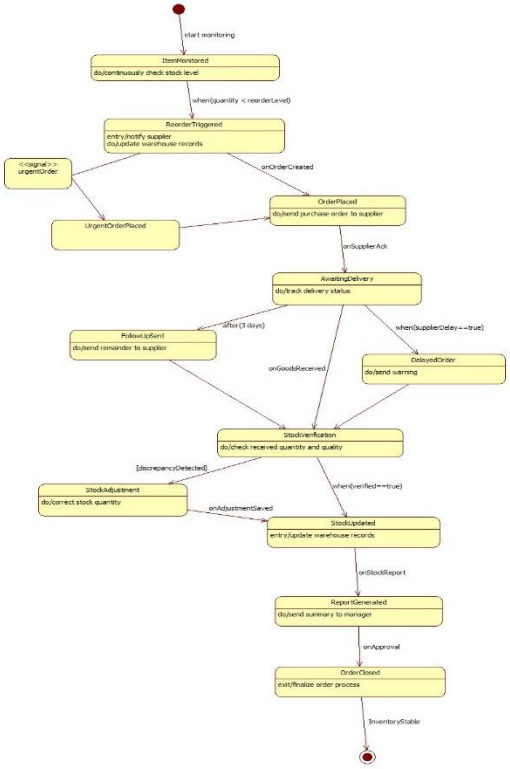


Fig 4.6

Inventory items are continuously monitored, and a reorder is triggered when quantity falls below a set level. Orders can be placed, tracked, followed up, or escalated if delivery is delayed. Verification and adjustment of received stock ensures accuracy before updating warehouse records. The order process closes after a manager reviews a summary report, leading to inventory stability.

4.6 Simple Sequence

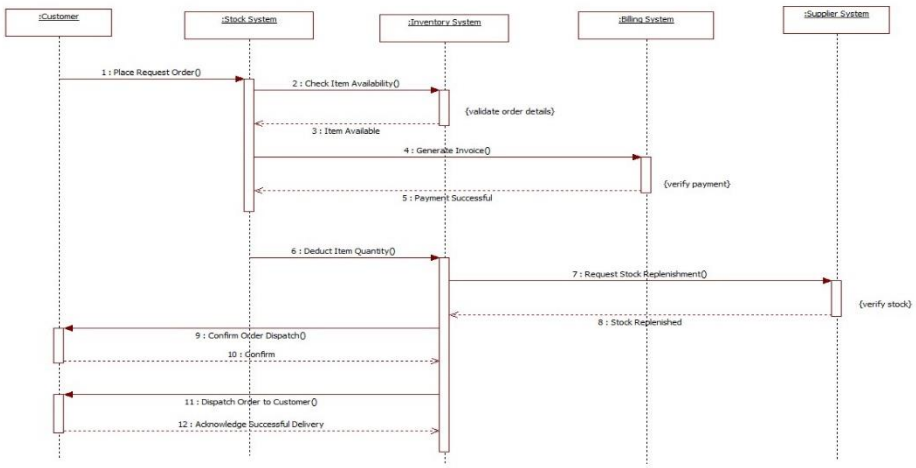


Fig 4.7

This sequence diagram illustrates an e-commerce order fulfillment process involving five systems: Customer, Stock System, Inventory System, Billing System, and Supplier System. The workflow begins when a customer places an order, which triggers inventory validation, invoice generation, and payment verification. Upon successful payment, the system deducts item quantity, requests stock replenishment from the supplier, and confirms order dispatch. The process concludes with order delivery to the customer and acknowledgment of successful delivery.

4.7 Advanced Use case:

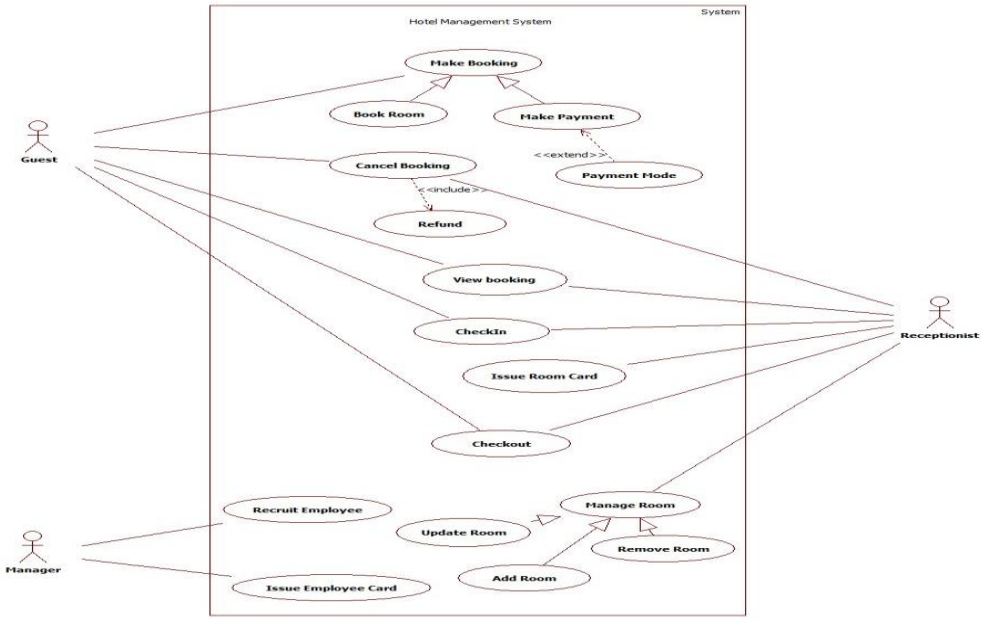


Fig 4.8

This use case diagram depicts a Hotel Management System with three primary actors: Guest, Receptionist, and Manager. Guests can make bookings with room selection and payment options, cancel bookings with refunds, view their reservations, and perform check-in/checkout operations with room card issuance. Receptionists handle guest check-in/checkout processes, issue room cards, view bookings, and manage room operations including adding, updating, and removing rooms. Managers oversee administrative functions such as recruiting employees, issuing employee cards, and supervising room management activities within the hotel system**.**

4.8 Simple Activity:

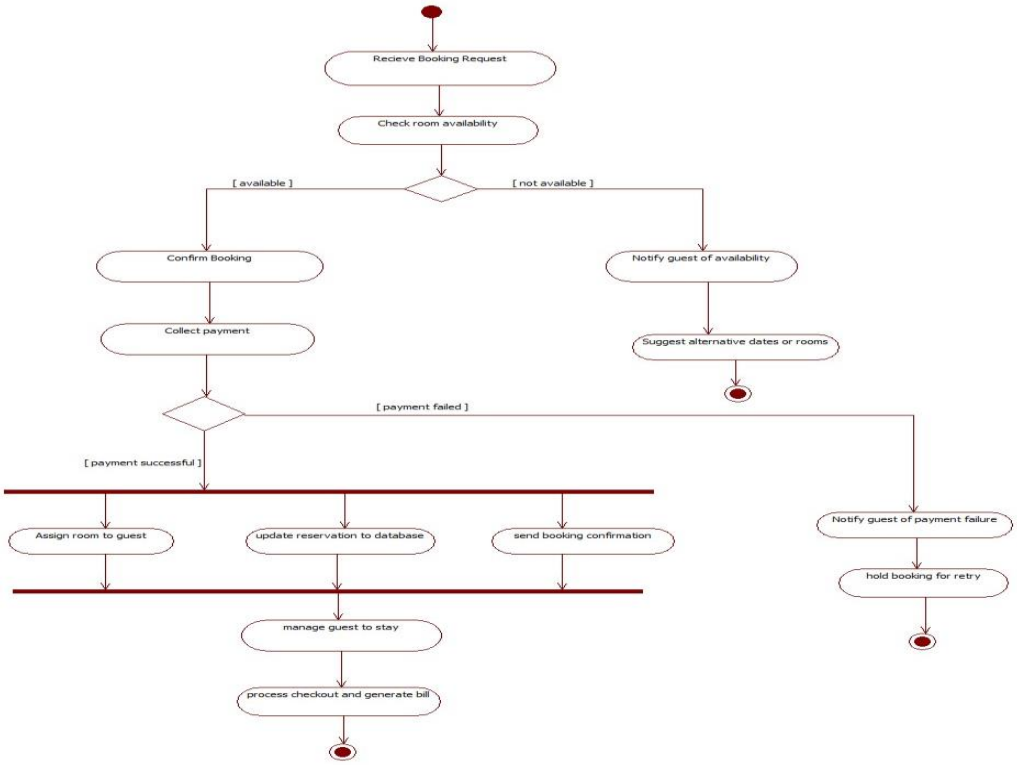


Fig 4.9

This activity diagram illustrates a hotel room booking workflow that begins when the system receives a booking request and checks room availability. If rooms are available, the system confirms the booking and collects payment; if successful, it executes three parallel activities: assigning the room to the guest, updating the reservation database, and sending booking confirmation, followed by managing the guest's stay and processing checkout with bill generation. If rooms are not available, the system notifies the guest and suggests alternative dates or rooms before terminating. In case of payment failure, the system notifies the guest of the failure, holds the booking for retry, and then ends the process.

4.9 Advanced Activity:

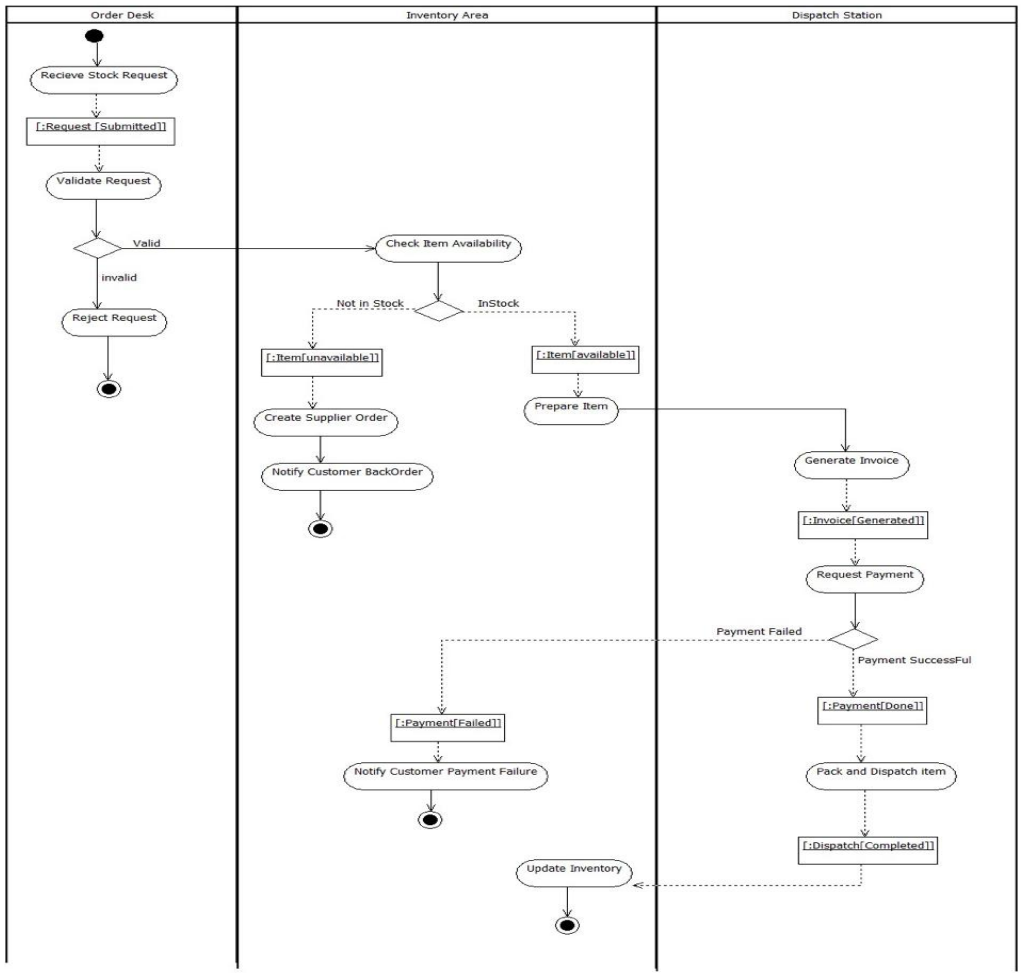
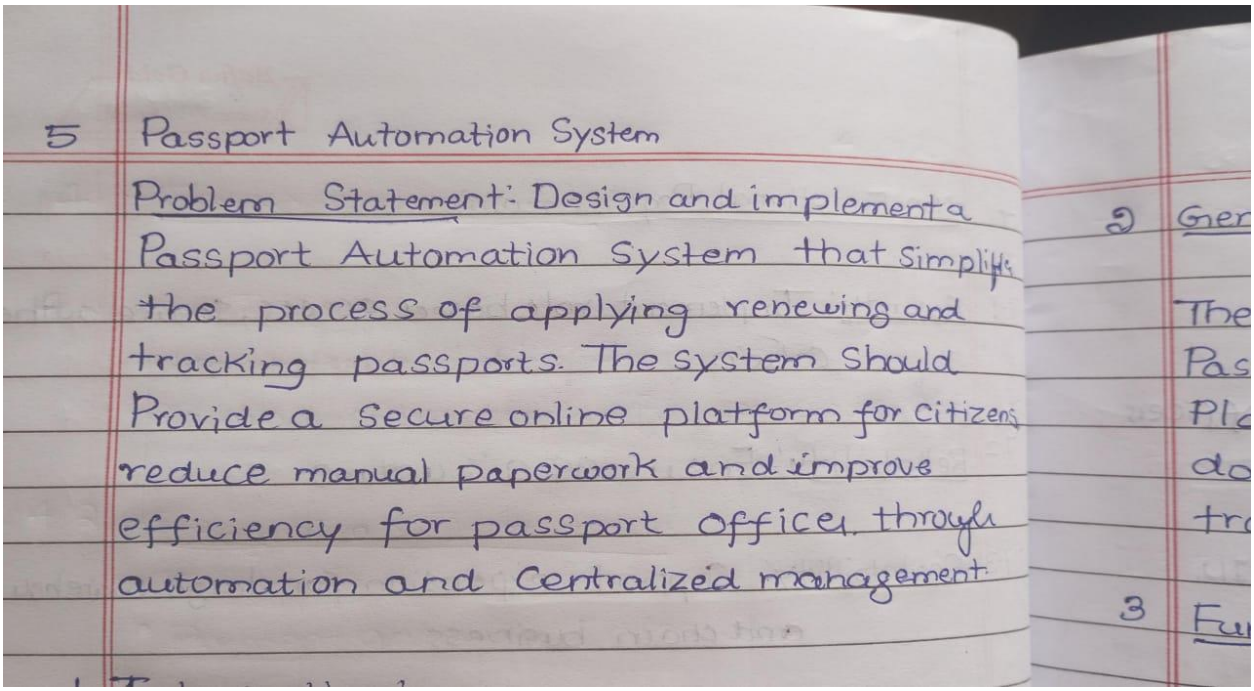


Fig 4.9

This activity diagram represents an inventory management system divided into three swimlanes: Order Desk, Inventory Area, and Dispatch Station. The process begins at the Order Desk where a stock request is received, validated, and either approved or rejected; valid requests proceed to the Inventory Area to check item availability, where items in stock are prepared for dispatch while unavailable items trigger supplier orders and customer backorder notifications. If items are available, the Dispatch Station generates an invoice and requests payment; successful payments lead to packing and dispatching the item with inventory updates, while payment failures result in customer notification and process termination.

5 Passport Automation System

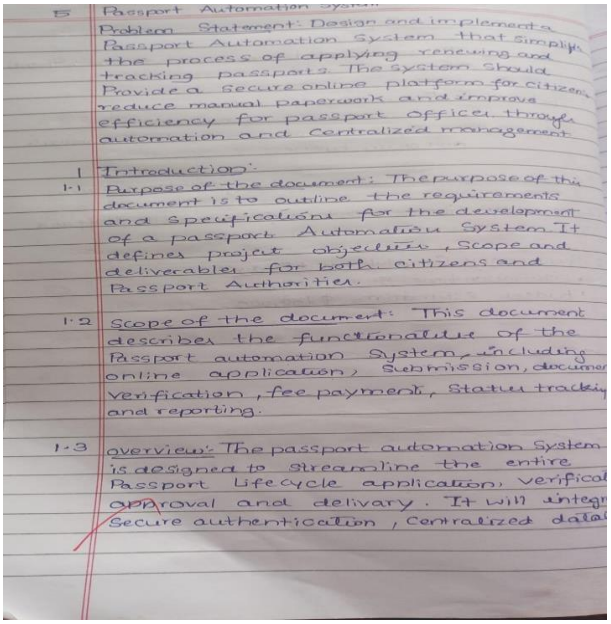
5.1 Problem Statement


5 Passport Automation System

Problem Statement: Design and implement a Passport Automation System that simplifies the process of applying renewing and tracking passports. The system should Provide a secure online platform for citizens reduce manual paperwork and improve efficiency for passport offices. through automation and Centralized management.

Fig 5.1

5.2 SRS Document


5 Passport Automation System

Problem Statement: Design and implement a Passport Automation System that simplifies the process of applying renewing and tracking passports. The system should Provide a secure online platform for citizens reduce manual paperwork and improve efficiency for passport offices. through automation and Centralized management.

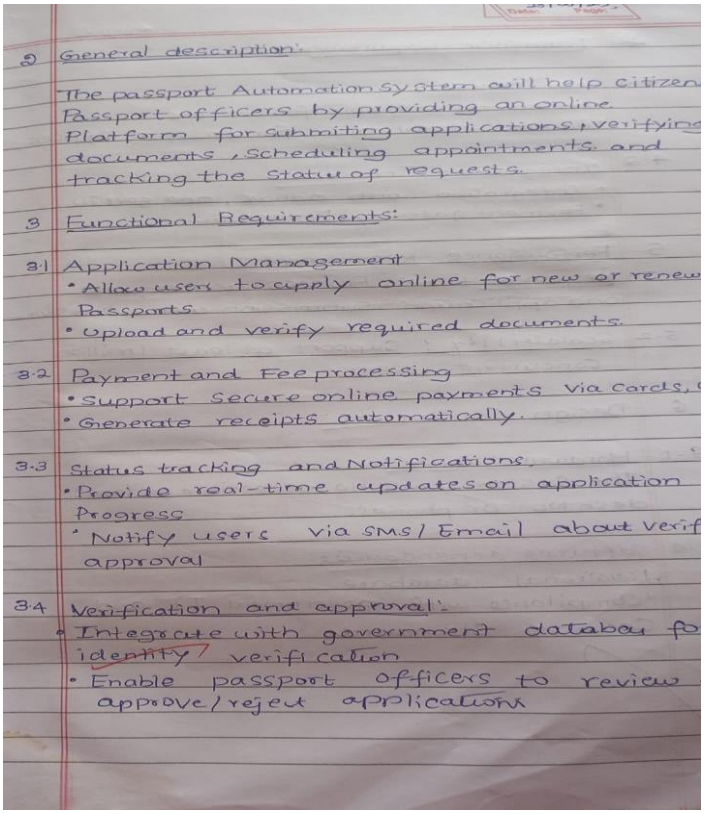
1 Introduction:

1.1 Purpose of the document: The purpose of this document is to outline the requirements and specifications for the development of a passport Automation System. It defines project objectives, Scope and deliverables for both citizens and Passport Authorities.

1.2 Scope of the document: This document describes the functionalities of the Passport automation System, including online application, Submission, documen verification, fee payment, Status tracki and reporting.

1.3 overview: The passport automation System is designed to streamline the entire Passport lifecycle application, verifica approval and delivery. It will integr Secure authentication, Centralized data

Fig 5.2



2 General description:

The passport Automation System will help citizen Passport officers by providing an online Platform for submiting applications, verifying documents, scheduling appointments and tracking the status of requests.

3 Functional Requirements:

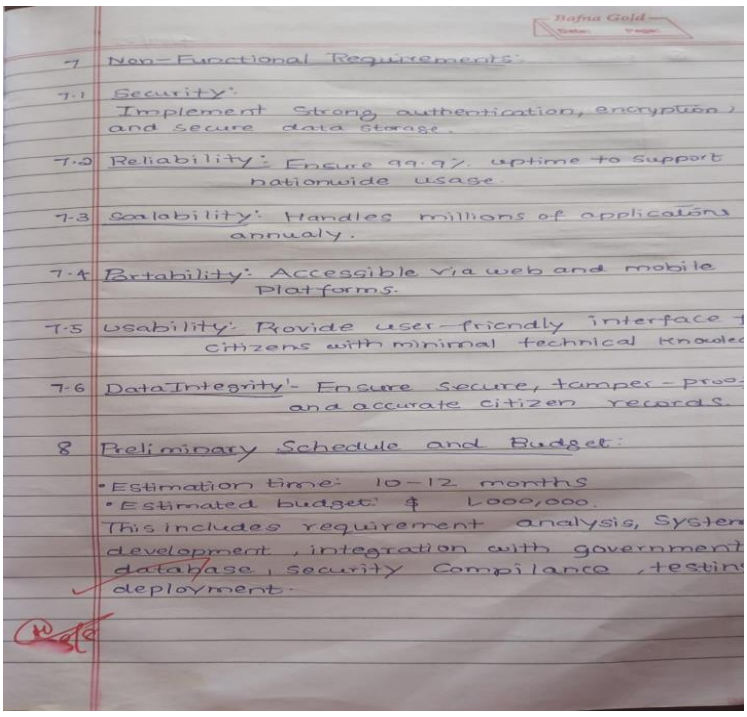
3.1 Application Management
- Allow users to apply online for new or renew Passports
- Upload and verify required documents

3.2 Payment and Fee processing
- Support secure online payments via cards,
- Generate receipts automatically.

3.3 Status tracking and Notifications.
- Provide real-time updates on application Progress
- Notify users via SMS/Email about Verif approval

3.4 Verification and approval:
- Integrate with government databas fo identity verification
- Enable passport officers to review approve/reject applications

Fig 5.3



7 Non-Functional Requirements:

7.1 Security:
Implement Strong authentication, encryption, and secure data storage.

7.2 Reliability: Ensure 99.9% uptime to support nationwide usage.

7.3 Scalability: Handles millions of applications annualy.

7.4 Portability: Accessible via web and mobile Platforms.

7.5 Usability: Provide user-friendly interface citizens with minimal technical knowle

7.6 Data Integrity: Ensure secure, tamper-proo and accurate citizen records.

8 Preliminary Schedule and Budget:
- Estimation time: 10-12 months
- Estimated budget: $ 1,000,000.

This includes requirement analysis, System development, integration with government database, security compilance, testin deployment.

Fig 5.4

5.3 Class Diagram

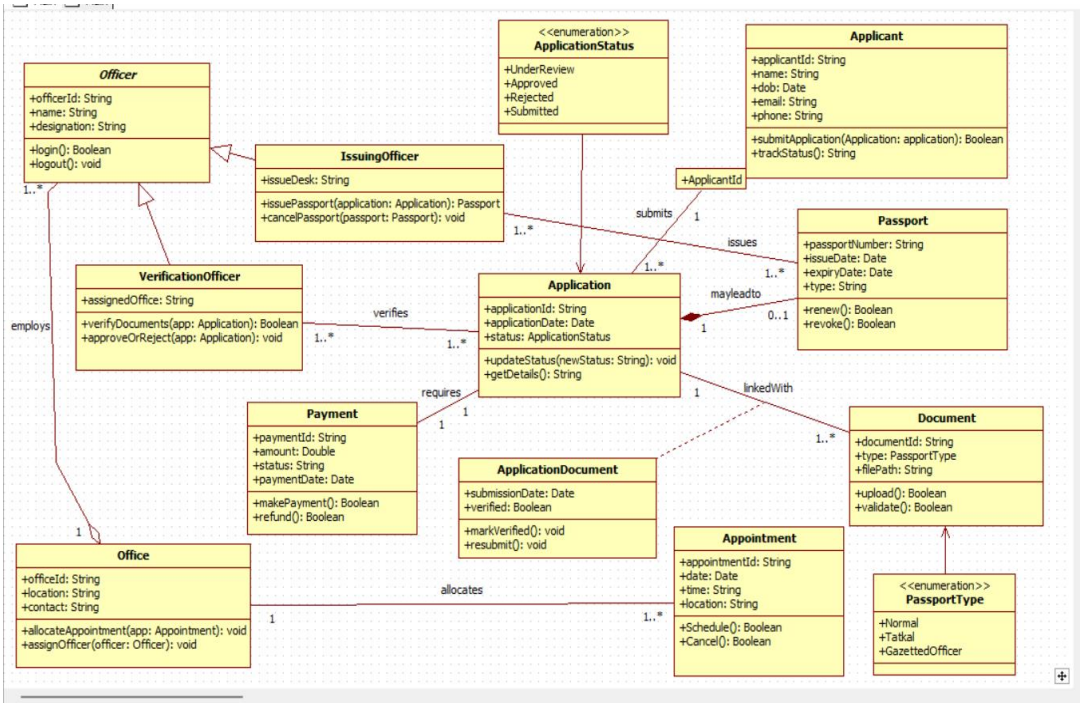


Fig 5.5

This diagram models a Passport Automation System, centered on the Application entity, which is *submitted by* an Applicant and is processed through various states (ApplicationStatus).

The application requires Documents and a Payment, and the workflow involves different Officer roles (VerificationOfficer and IssuingOfficer) to verify the application, which may eventually *lead to* a Passport.

The system uses supporting entities like Office to *allocate* Appointments and manages the lifecycle of the application, payment, and passport issuance via defined attributes and operations.

5.4 State Diagram

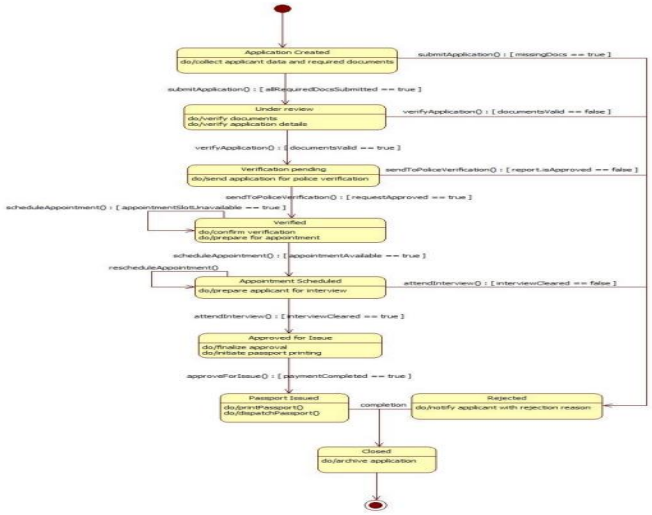


Fig 5.5

This state machine diagram illustrates the complete lifecycle of a passport application, showing how an application transitions between different states based on specific events and conditions.

The process begins in the Application Created state, and the application moves to the Under Review state once all required documents are submitted. If documents are missing or verification fails, the application loops back to a prior state or transitions to Rejected. Once verification passes, the application moves to Verification Pending, triggering the police verification process.

5.5 Simple sequence Model

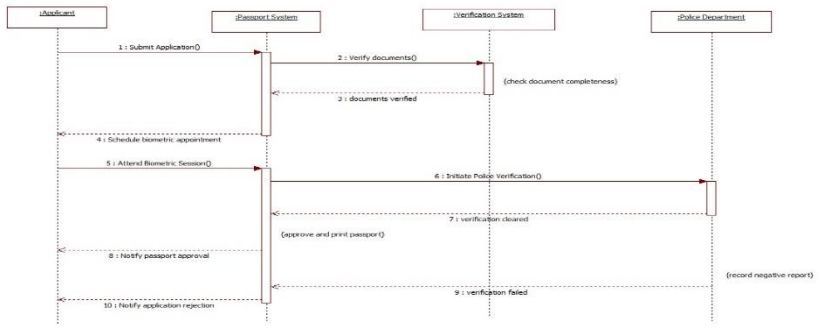


Fig 5.6

This sequence diagram illustrates the chronological interaction among four actors: Applicant, Passport System, Verification System, and Police Department, for a passport application.

The Applicant initiates the process by submitting the application and attending a biometric session; the Passport System then directs the Verification System to verify documents, and, following biometrics, initiates the police verification with the Police Department

5.6 Advanced Sequence

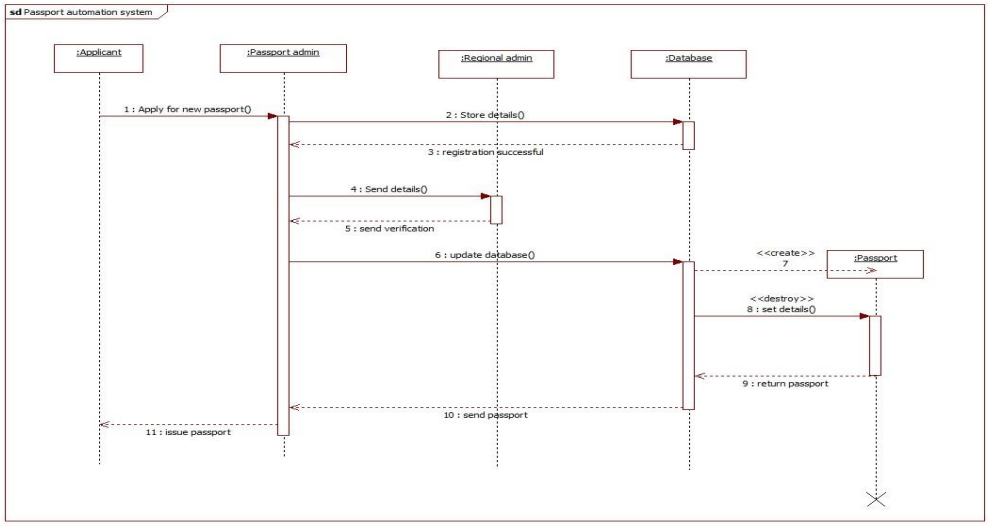


Fig 5.7

This sequence diagram illustrates the chronological flow of interactions between four main participants: the Applicant, the Passport Admin, the Regional Admin, and the Database, during the passport issuance process.

The process is initiated when the Applicant sends an Apply for new passport() request to the Passport Admin, which in turn instructs the Database to Store details() and confirm registration success. The Passport Admin then sends applicant details to the Regional Admin for verification, and uses the update database() message to reflect status changes.

5.7 Simple Use case

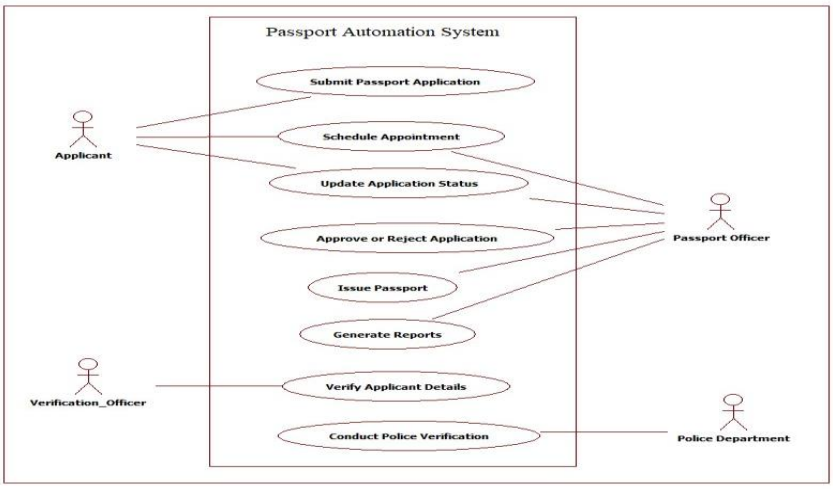


Fig 5.7

This use case diagram illustrates the functional requirements and external actors of a Passport Automation System, showing how different users interact with the system.

The main external actor is the Applicant, who can Submit Passport Application, Schedule Appointment, and Update Application Status. The Passport Officer is responsible for system administration and approval, interacting with use cases like Schedule Appointment, Update Application Status, Approve or Reject Application, Issue Passport, and Generate Reports.

5.8 Advanced Use Case

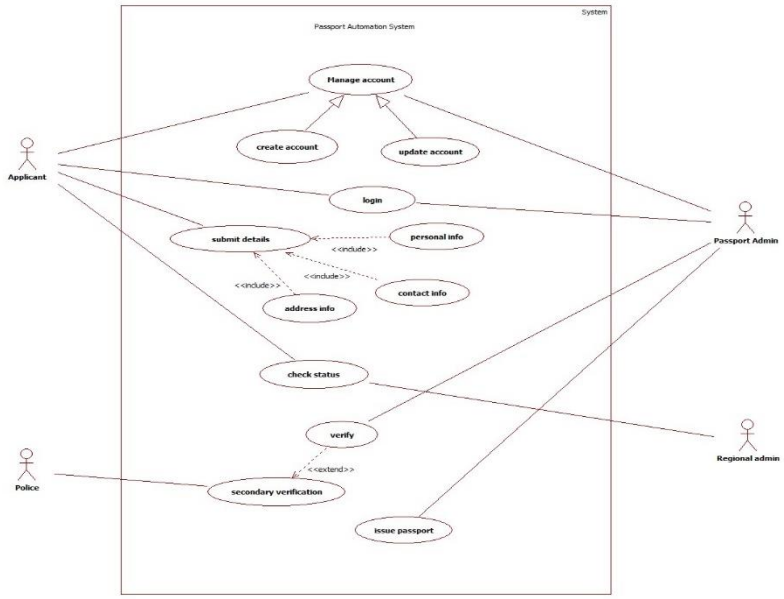


Fig 5.8

This use case diagram models the advanced functionalities and actors within the Passport Automation System, focusing on account management, detailed submission, and multi-layered verification.

1. The Applicant is responsible for system entry (login) and managing their account, which involves the specialized use cases of create account and update account (part of Manage account).

2. The application submission is broken down into core included components: submit details always requires personal info, contact info, and address info, after which the Applicant can check status.

3. Verification is a multi-step process, beginning with verify (handled by the Regional Admin), which can be *extended* by secondary verification involving the Police, before the Passport Admin performs the final issue passport action.

5.8 Simple Activity

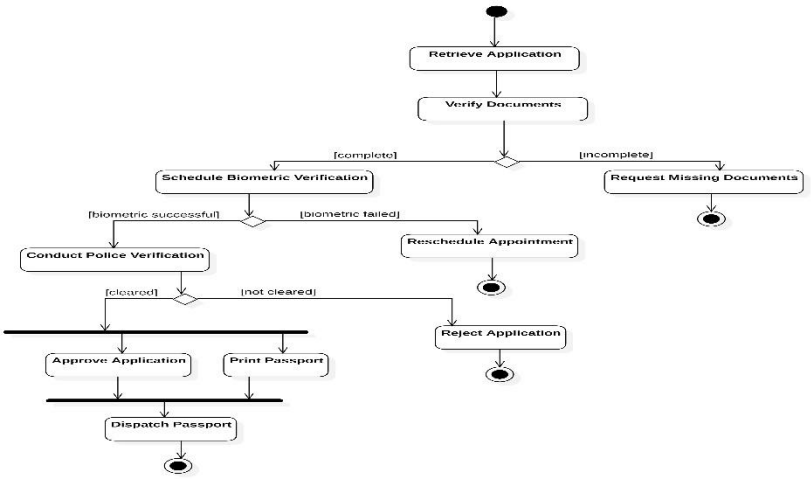


Fig 5.9

This activity diagram illustrates the sequential flow for processing a passport application, beginning with Retrieving and Verifying Documents. Based on verification completeness, the process either requests missing documents or proceeds to Schedule Biometric Verification.

If biometrics are successful, the system conducts a Police Verification; failure at either the biometric or police verification stages leads directly to Reject Application or prompts a Reschedule Appointment.

Once both verifications are cleared, the workflow proceeds in parallel to Approve Application and Print Passport, culminating in the final step of Dispatch Passport.

5.9Advanced Activity Diagram

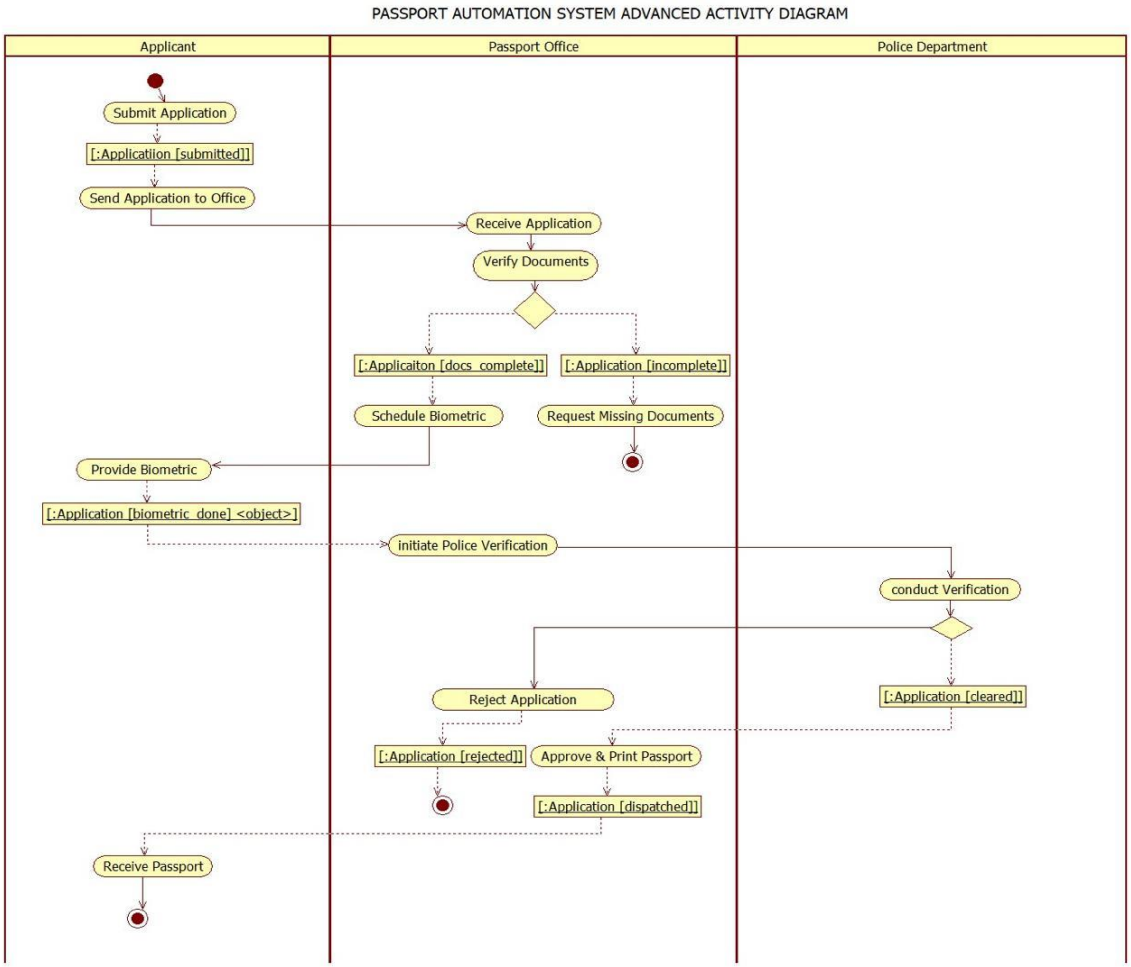


Fig 5.10

This activity diagram maps the end-to-end flow of a passport application across three swimlanes: Applicant, Passport Office, and Police Department.

The Passport Office receives the application, verifies documents (branching on completeness), schedules biometrics, and initiates the Police Department's verification.

The final decision is based on police verification: if cleared, the office Approve & Print Passport; if not, the application is Rejected, with the Applicant concluding the process by Receive Passport.