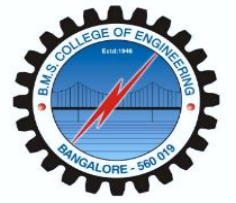
B.M.S. COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



Lab Record

Software Engineering and Object-Oriented Modeling

Submitted in partial fulfillment for the 5th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

SANJANA SURESH
1BM22CS239

Department of Computer Science and Engineering
B.M.S. College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
September 2024-January 2025

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Object-Oriented Analysis and Design(22CS6PCSEO) laboratory has been carried out by Sanjana Suresh(1BM22CS239) during the 5th Semester Oct 24-Jan 2025.

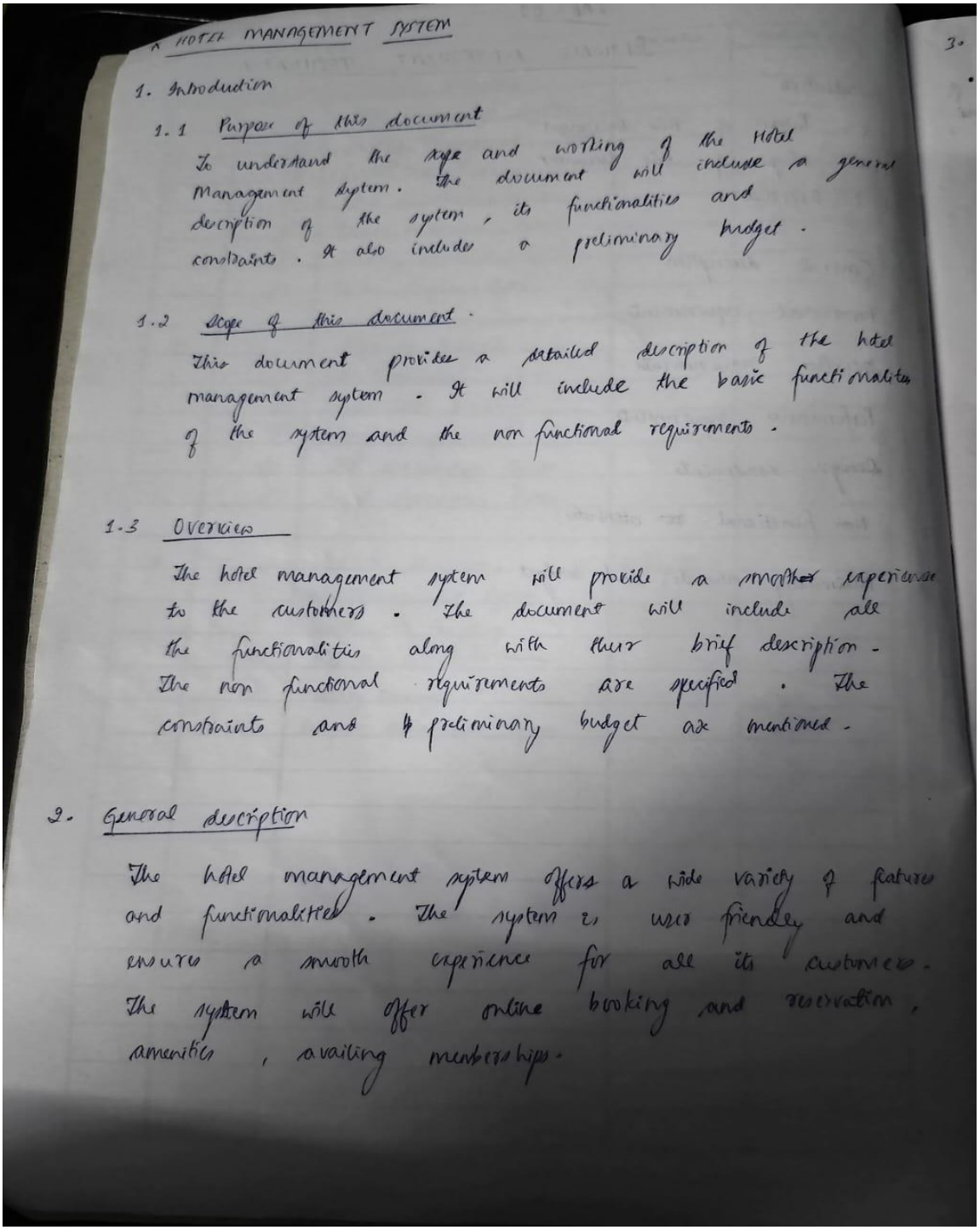
Signature of the Faculty In charge:

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

Table of Contents

1. Hotel Management System

Software Requirement Specification



HOTEL MANAGEMENT SYSTEM

3.

1. Introduction

1.1 Purpose of this document

To understand the type and working of the Hotel Management system. The document will include a general description of the system, its functionalities and constraints. It also include a preliminary budget.

1.2 Scope of this document.

This document provides a detailed description of the hotel management system. It will include the basic functionalities of the system and the non functional requirements.

1.3 Overview

The hotel management system will provide a smoother experience to the customers. The document will include all the functionalities along with their brief description. The non functional requirements are specified. The constraints and a preliminary budget are mentioned.

2. General description

The hotel management system offers a wide variety of features and functionalities. The system is user friendly and ensures a smooth experience for all its customers. The system will offer online booking and reservation, amenities, availing memberships.

3. Functional requirements

- Online booking and reservation

The hotel management system provides facilities for online booking and reservations with memberships and offers. The customer can navigate to the type of suite or room that they require and the time duration of the stay.

- Guest management system

The hotel management system offers the best and smooth navigation to the guests. The guests will be well looked after by the staff.

- Club Membership

The hotel offers memberships and discounts for the in house clubs present in the premises. Depending on the stay of the customer, points will be rewarded.

- Food Management System

The system offers a variety of cuisines to adhere to all the customers. Complimentary beverages and snacks will be provided.

4. Interface requirements

The hotel management system will have a user friendly interface which will ensure a smooth experience to all the customers. The customers can easily navigate between the reservations and services available.

5. Performance requirements

- The online hotel management system can handle a large load of customers at time - around 1000 people can be logged in at a time.
- The hotel management system shall load with 2 seconds of opening and logging in.

6. Design constraints

- The hotel management system will not allow a user to complete its transaction 30 mins after clicking on the payment option.
- The UI-UX should be user friendly to ensure a smooth navigation for customers.

7. Non functional attributes

30/8/24

- Security - The user details after logging in, their reservation and suite numbers shall be stored in a secure database.
- Portability - The online hotel management system can be made available on mobile phones, PC's and laptops.

8. Preliminary budget and schedule

UI - UX design - 2 months

Frontend - 3 months

Backend integration - 4 months.

Class Diagram

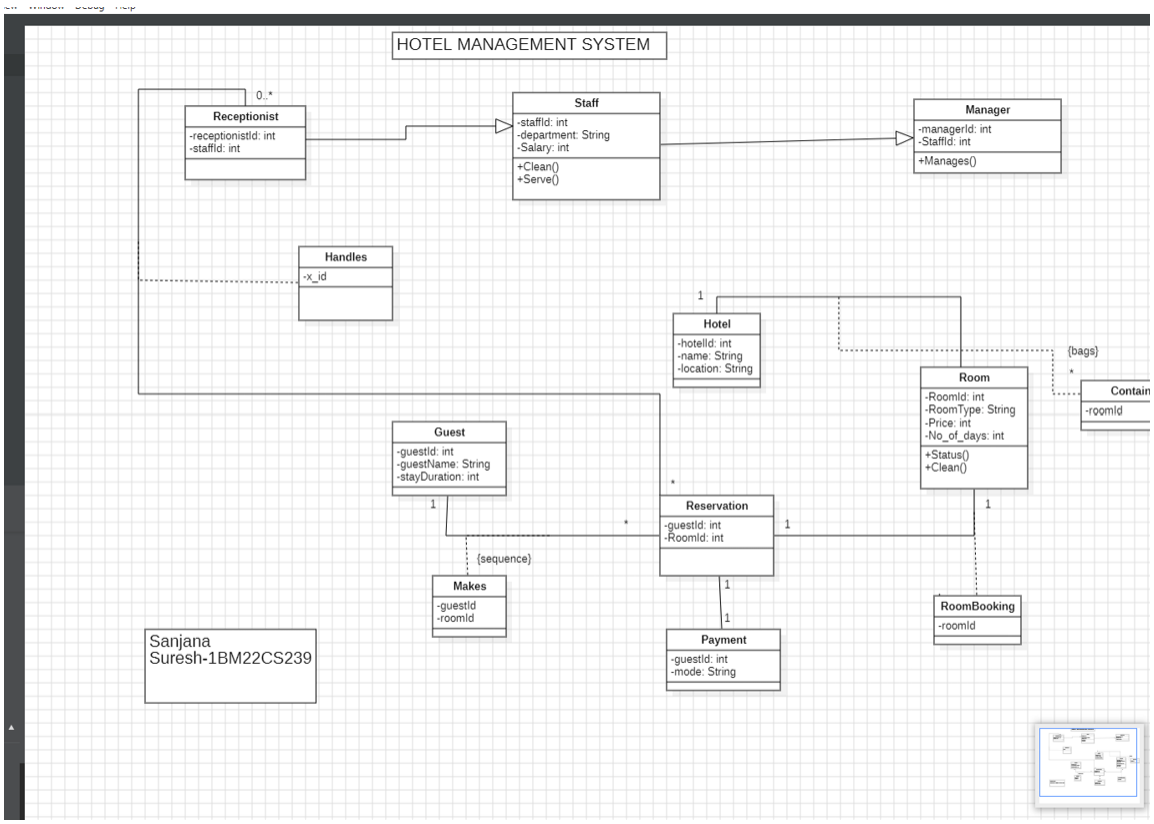


Fig1.1 Class Diagram for Hotel Management System

This class diagram models a Hotel Management System, highlighting entities like Staff (with specialized roles such as Receptionist and Manager), Hotel, Room, Guest, Reservation, and Payment, along with their relationships. A hotel contains multiple rooms, which guests can reserve. Guests make reservations linked to payments, and staff members handle tasks like cleaning and serving, managed by a manager. The system effectively organizes key operations and interactions within a hotel.

State Diagram

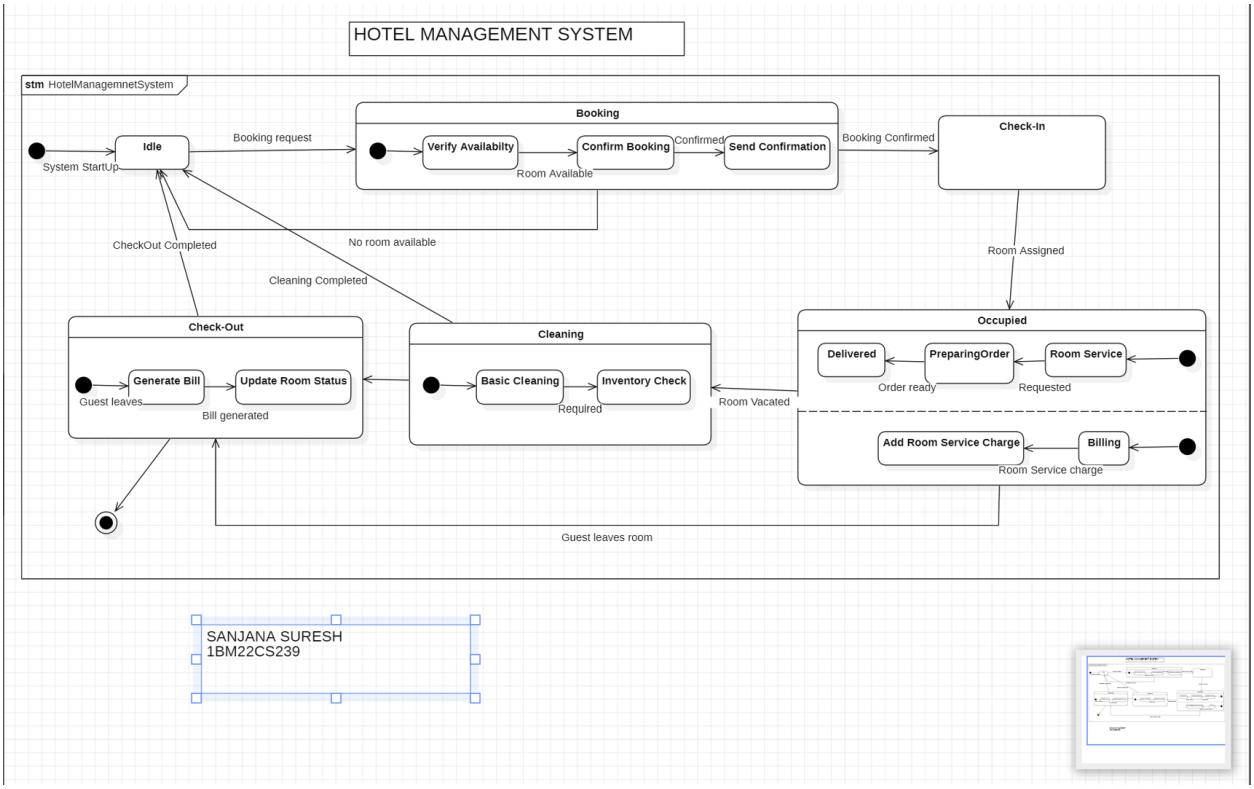


Fig1.2 State Diagram for Hotel Management System

This state diagram illustrates the Hotel Management System's workflow, showing key states and transitions. The system starts in an Idle state, moving to Booking upon a booking request, where availability is verified, booking is confirmed, and a confirmation is sent. Guests then transition to Check-In, with rooms assigned. During the stay (state: Occupied), services like Room Service and Billing can occur. On Check-Out, the system generates bills and updates room statuses. Vacated rooms enter the cleaning process, involving basic cleaning and inventory checks, before returning to the idle state for reuse.

Sequence Diagram

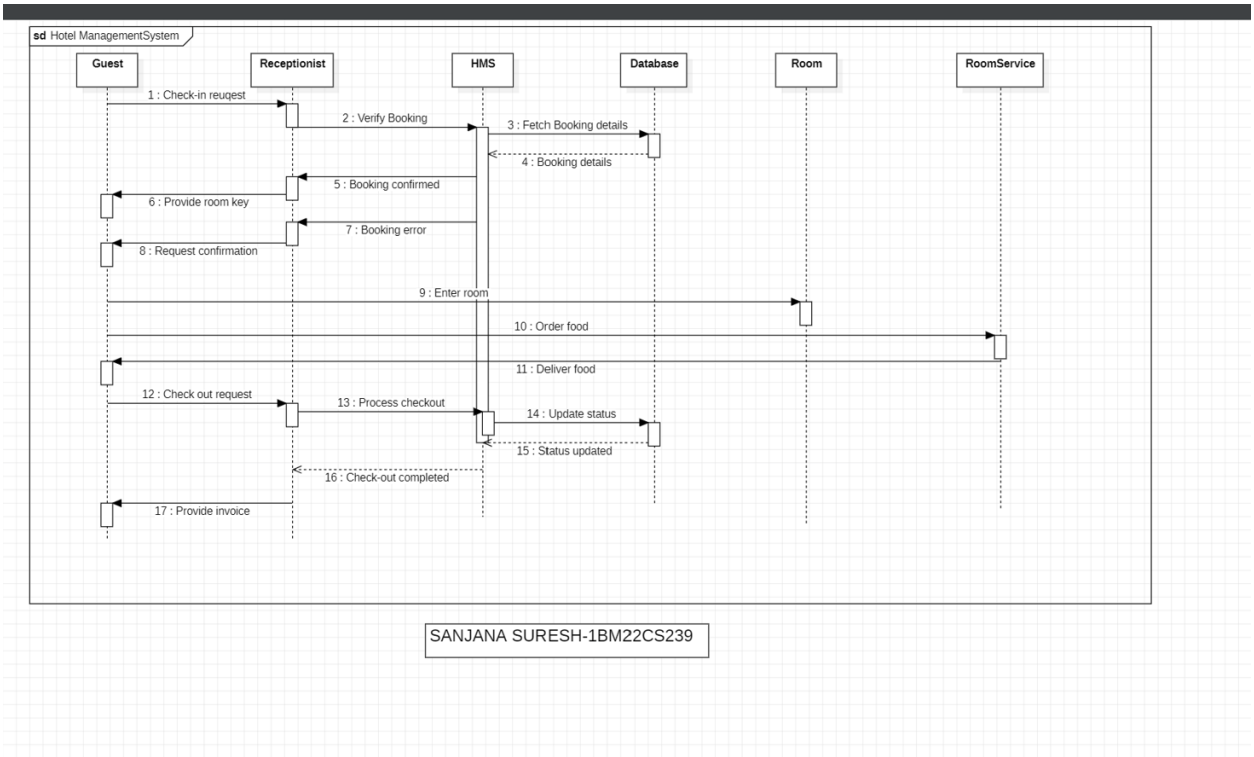


Fig1.3 Sequence Diagram for Hotel Management System

The sequence diagram for a Hotel Management System illustrates the flow of interactions between various entities like Guest, Receptionist, System and Room Service. It begins with the Guest requesting a room, followed by the Receptionist or system verifying room availability. Upon confirmation, the booking is processed, and payment details are handled. Once the guest checks in, interactions for additional services like room service or billing are shown. The diagram concludes with the Check-Out process, where the system generates a bill and updates room status. It emphasizes the chronological order of operations and communication between actors and system components.

Use Case Diagram

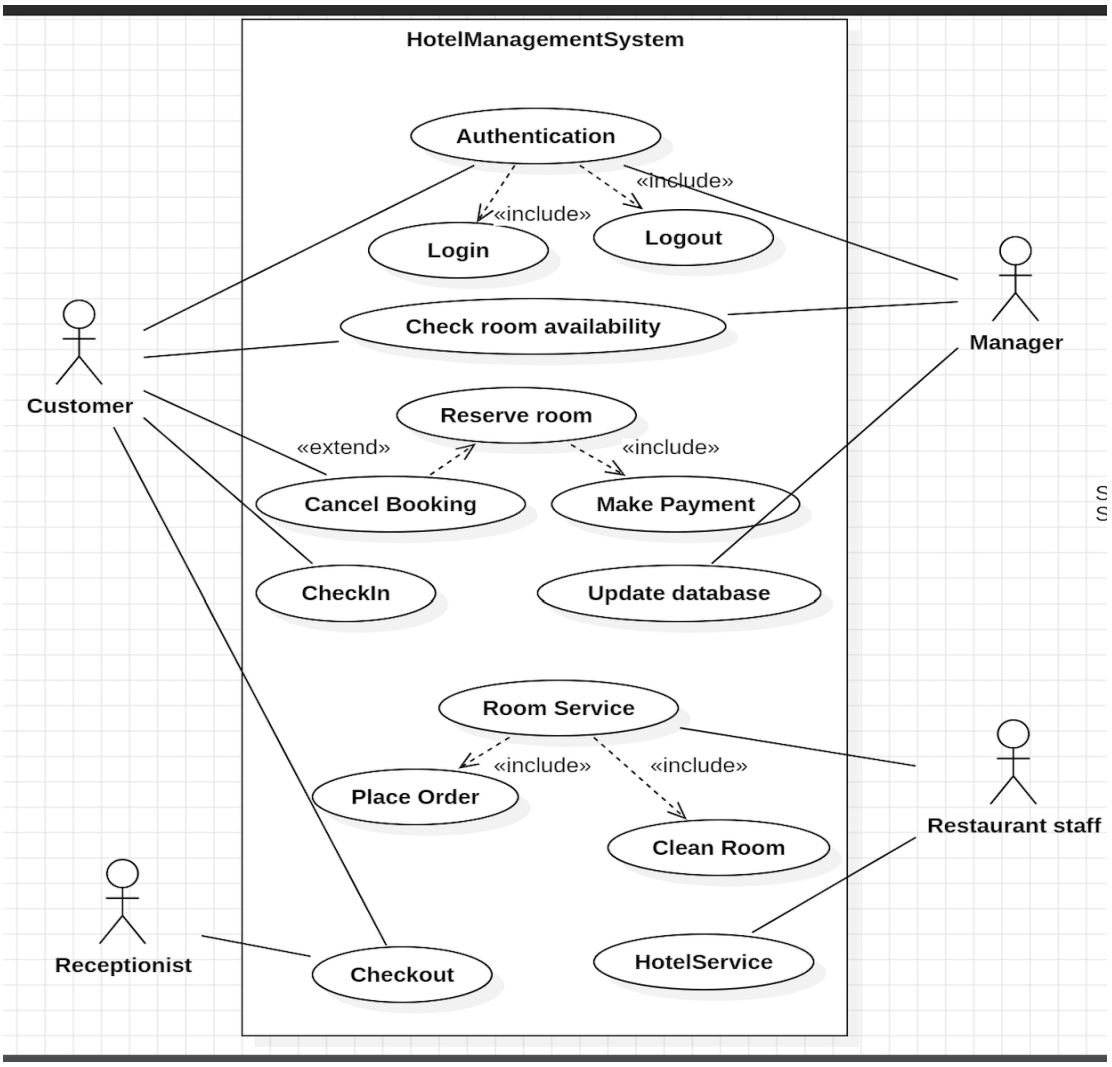


Fig1.4 Use Case Diagram for Hotel Management System

A use case diagram for a Hotel Management System illustrates interactions between actors like Guest, Receptionist, Manager, and Staff with the system's functionalities. Guests can search for rooms, make reservations, request services (e.g., room service), check-in, and check-out. Receptionists manage reservations, assign rooms, handle check-in/check-out, and generate bills. Managers oversee staff operations, monitor bookings, and manage the overall system. Staff handle room cleaning, inventory checks, and service delivery, showcasing the system's comprehensive operational flow.

Activity Diagram

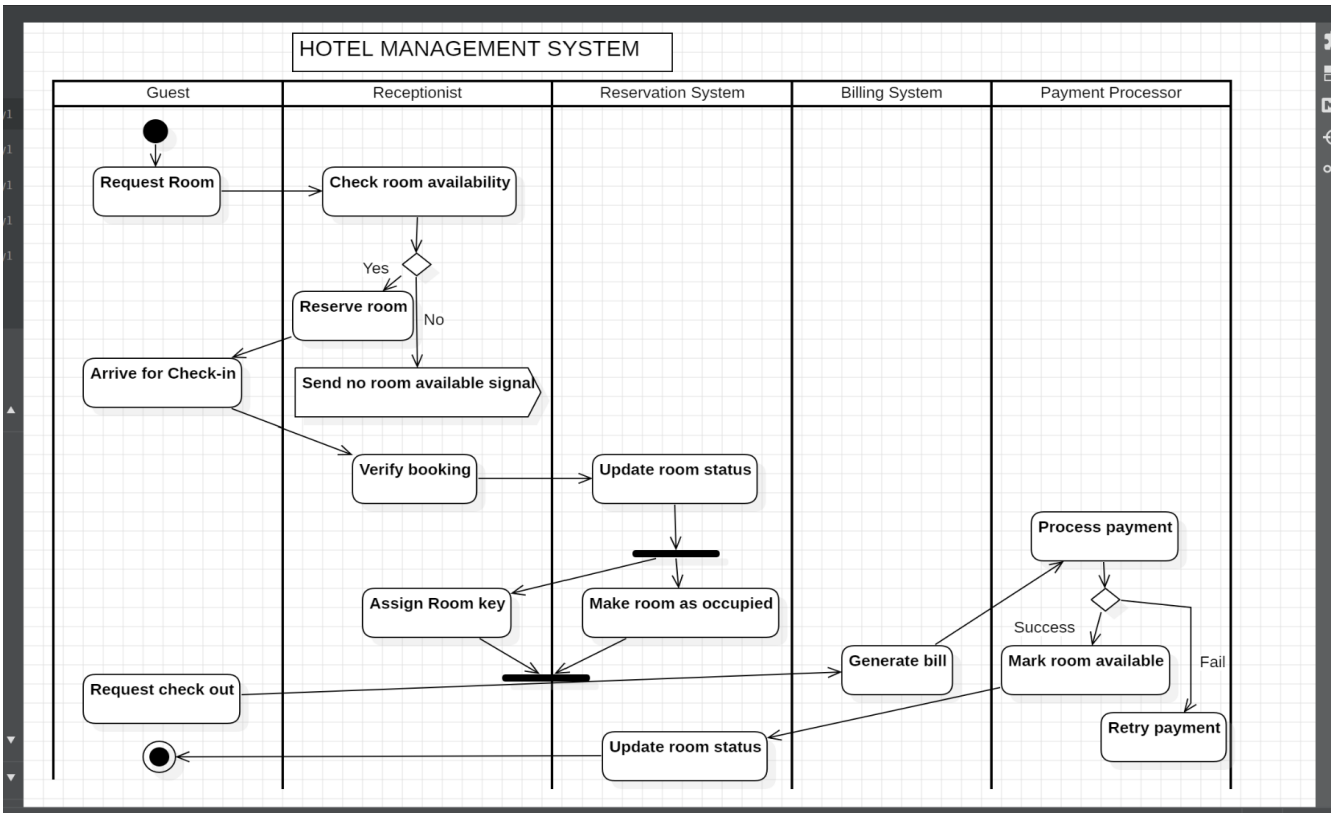
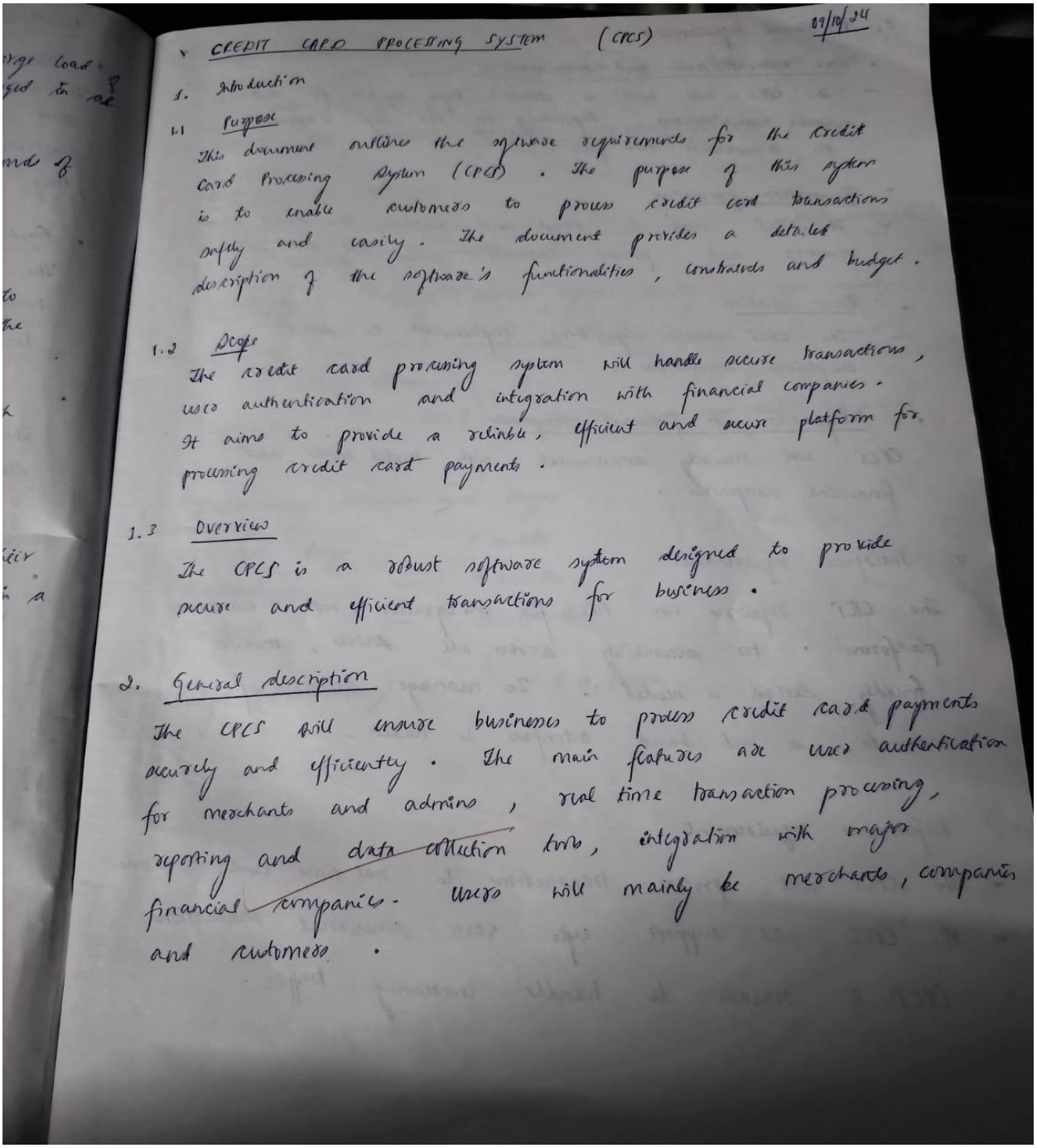


Fig1.5 Activity Diagram for Hotel Management System

An activity diagram with swim-lanes for a Hotel Management System visually organizes tasks across actors such as Guest, Receptionist, Staff, and Manager. The Guest initiates actions like searching for rooms, booking, and check-in, while the Receptionist/System handles room availability checks, booking confirmation, check-in, check-out, and billing. The Staff manages room cleaning, preparation, and service delivery, while the Manager/System oversees operations and ensures smooth functionality. The swim lanes clearly delineate responsibilities, highlighting the coordinated flow of activities across the system.

2.Credit Card Processing

Software Requirement Specification



CREDIT CARD PROCESSING SYSTEM (CPCS) 09/10/24

1. Introduction

1.1 Purpose

This document outlines the software requirements for the Credit Card Processing System (CPCS). The purpose of this system is to enable customers to process credit card transactions safely and easily. The document provides a detailed description of the software's functionalities, constraints and budget.

1.2 Scope

The credit card processing system will handle secure transactions, user authentication and integration with financial companies. It aims to provide a reliable, efficient and secure platform for processing credit card payments.

1.3 Overview

The CPCS is a robust software system designed to provide secure and efficient transactions for business.

2. General description

The CPCS will ensure businesses to process credit card payments securely and efficiently. The main features are user authentication for merchants and admins, real time transaction processing, reporting and data collection tools, integration with major financial companies. Users will mainly be merchants, companies and customers.

3. Functional requirements

- User authentication and authorization
  - The CPCS will have a secure login system for customers and administrators. Depending on the login credentials, the transaction take place.

- Transaction processing
  - The CPCS needs to accept and validate the credit card number and process transactions in real time.

- Scam detection
  - The CPCS has algorithms implemented to detect suspicious transactions.

- Integration with financial companies

CPCS will securely communicate with needed banks and financial companies.

4. Interface requirements

The CPCS requires an API for integration with e-commerce platforms. For accessibility across all devices, mobile friendly design is needed. To manage accounts of customers, a web based interface is needed.

5. Performance requirements.

- The CPCS will process transaction in real time under 5 seconds
- CPCS will support upto 5000 concurrent transactions.
- CPCS is scalable to handle increasing traffic.

6. Design Constraints

- The CPCS will cancel a transaction if it takes more than 5 minutes to process data.
- CPCS has approved encryption algorithms for data protection.

7. Non functional attributes

- Security – CPCS has multi factor authentication, data encryption and enable secure transactions.
- Reliability – CPCS ensures system stability and data integrity.
- Portability – CPCS is available on multiple browsers and devices.
- Scalability – CPCS can handle increasing traffic volumes.

8. Preliminary Schedule and Budget

UI - UX design – 3 months

Frontend – 4 months

Backend – 5 months.

Budget (Preliminary) – ₹ 10,00,000.

Class Diagram

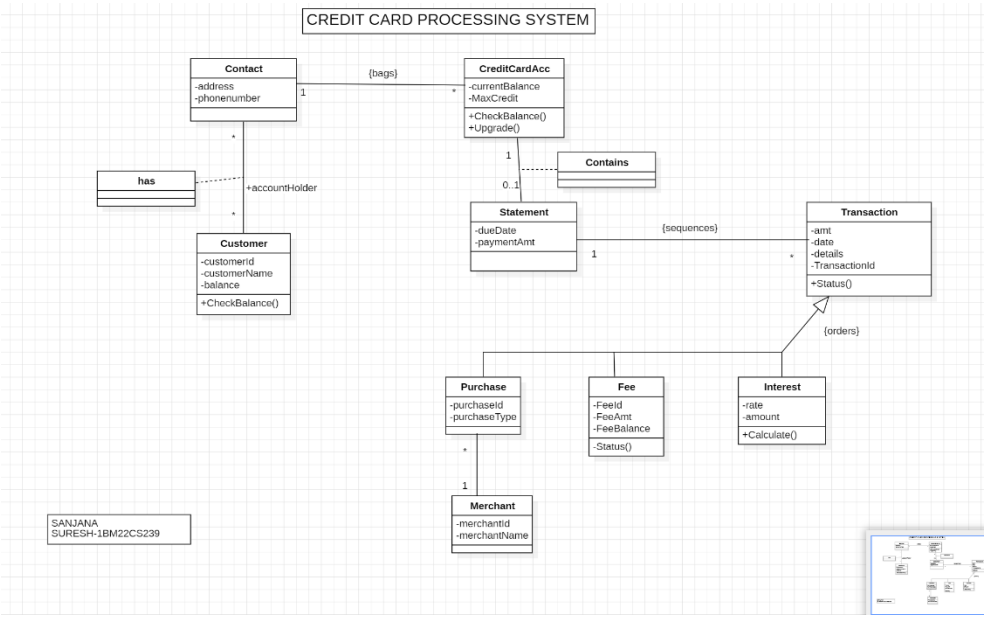


Fig2.1 Class Diagram for Credit Card Processing System

The UML class diagram illustrates a credit card processing system. It outlines the various classes involved, including Customer, CreditCardAcc, Statement, Transaction, Purchase, Fee, Interest, and Merchant. The relationships between these classes are depicted, such as one-to-many associations between CreditCardAcc and Statement, and many-to-many relationships between Transaction and Purchase, Fee, and Interest. The diagram also shows attributes and operations for each class, providing a high-level overview of the system's structure and behaviour.

State Diagram

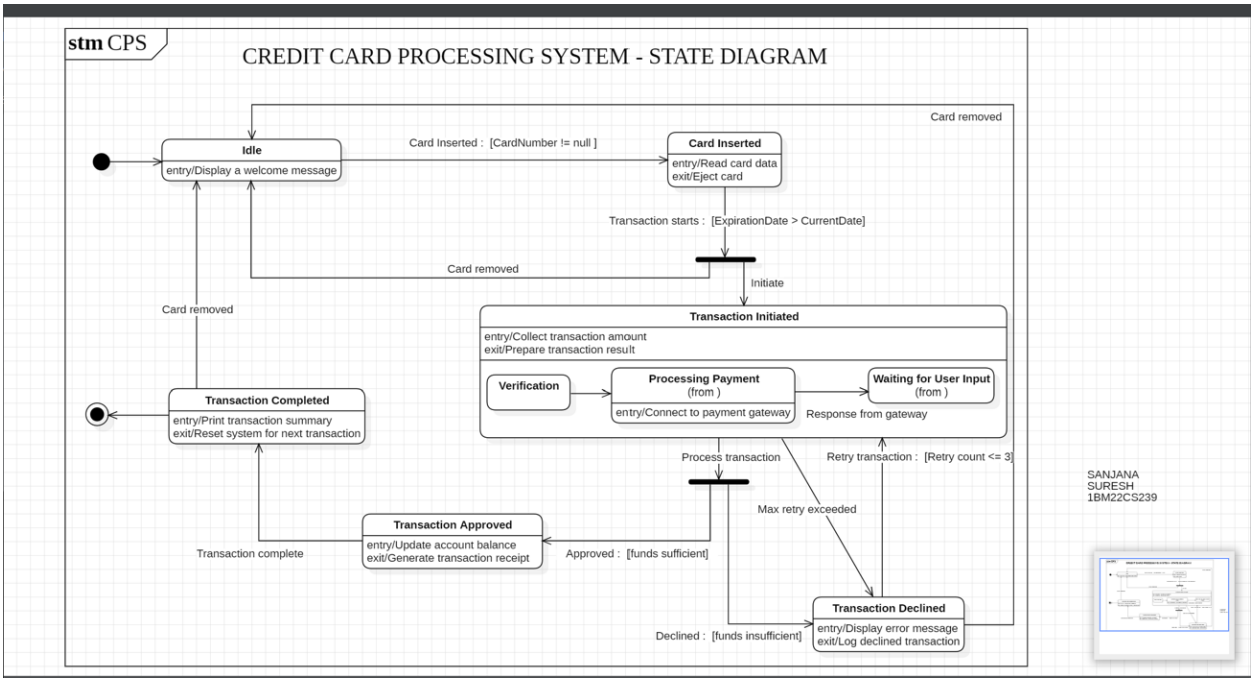


Fig2.2 State Diagram for Credit Card Processing System

The UML state diagram depicts the workflow of a credit card processing system. It begins in the "Idle" state and transitions to "Card Inserted" upon card insertion. The system then verifies the card and initiates the transaction. If successful, it moves to "Processing Payment" and communicates with the payment gateway. Based on the gateway response, it either transitions to "Transaction Approved" or "Transaction Declined". In case of approval, the account is updated, and a receipt is generated. If declined, an error message is displayed, and the transaction is logged. The system returns to the "Idle" state in all scenarios, ready for the next transaction.

Sequence Diagram

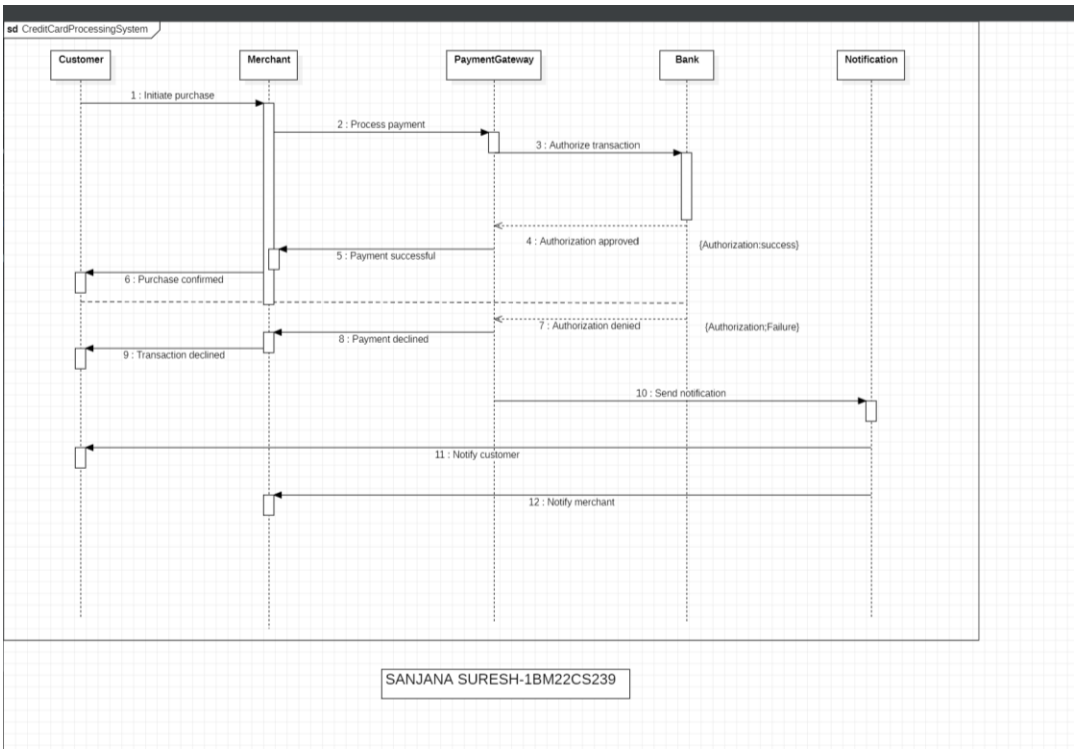


Fig2.3 Sequence Diagram for Credit Card Processing System

The sequence diagram models the credit card transaction process. It starts with the customer initiating a purchase. The merchant then processes the payment and sends a request to the Payment Gateway for authorization. The Gateway forwards this to the Bank for approval. If authorized, the Bank informs the Gateway, and the transaction is confirmed. Both the customer and merchant are notified. However, if the Bank declines the authorization, the transaction is rejected, and notifications are sent accordingly. This diagram provides a step-by-step visualization of the interactions between different entities involved in a credit card transaction.

Use Case Diagram

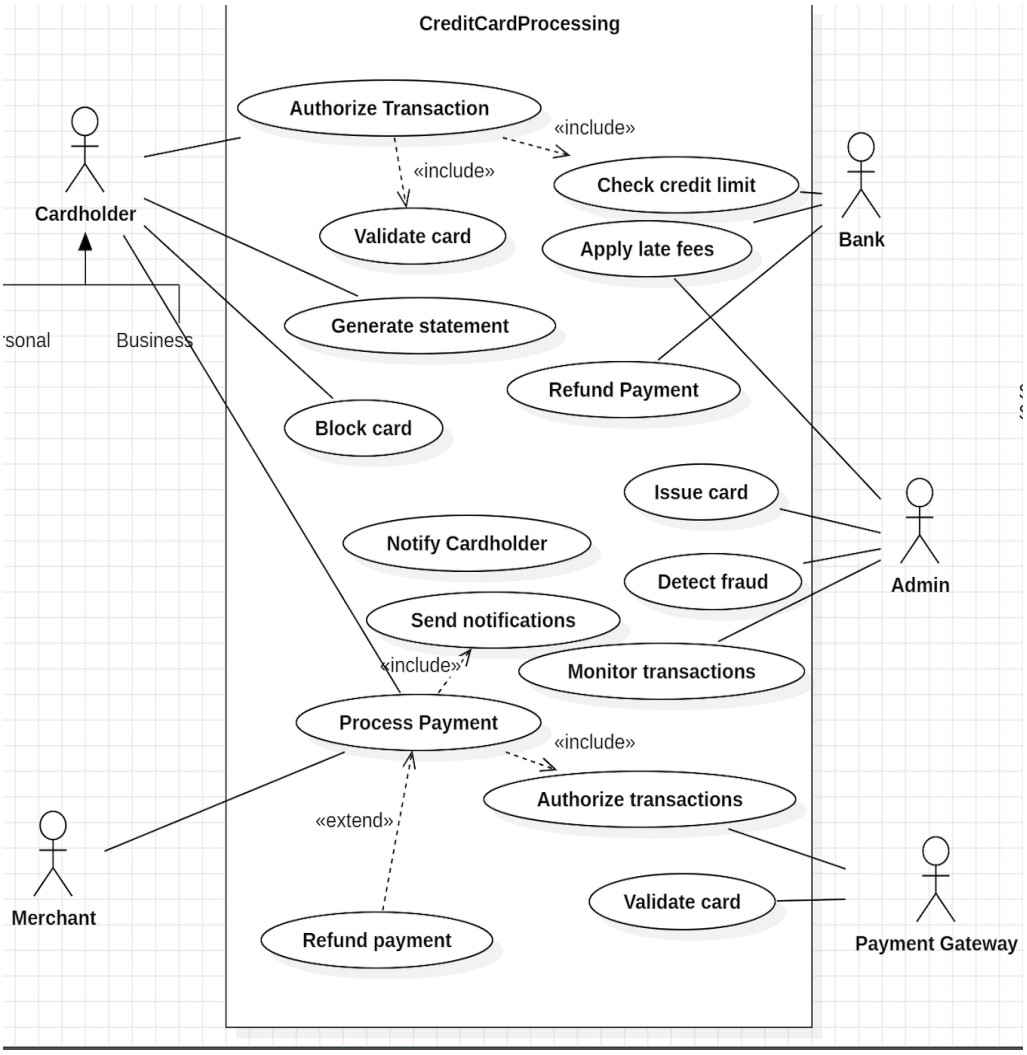


Fig2.4 Use Case Diagram for Credit Card Processing System

The Use Case Diagram illustrates the various functionalities of a credit card processing system. It outlines the interactions between different actors, including Cardholder, Merchant, Bank, Admin, and Payment Gateway. Use cases like Authorize Transaction, Process Payment, Generate Statement, Refund Payment, and Notify Cardholder are defined. Additionally, relationships like "include" and "extend" are used to represent optional or alternative behaviours, such as checking credit limits, applying late fees, and detecting fraud. This diagram provides a high-level overview of the system's capabilities and the interactions involved in credit card management.

Activity Diagram

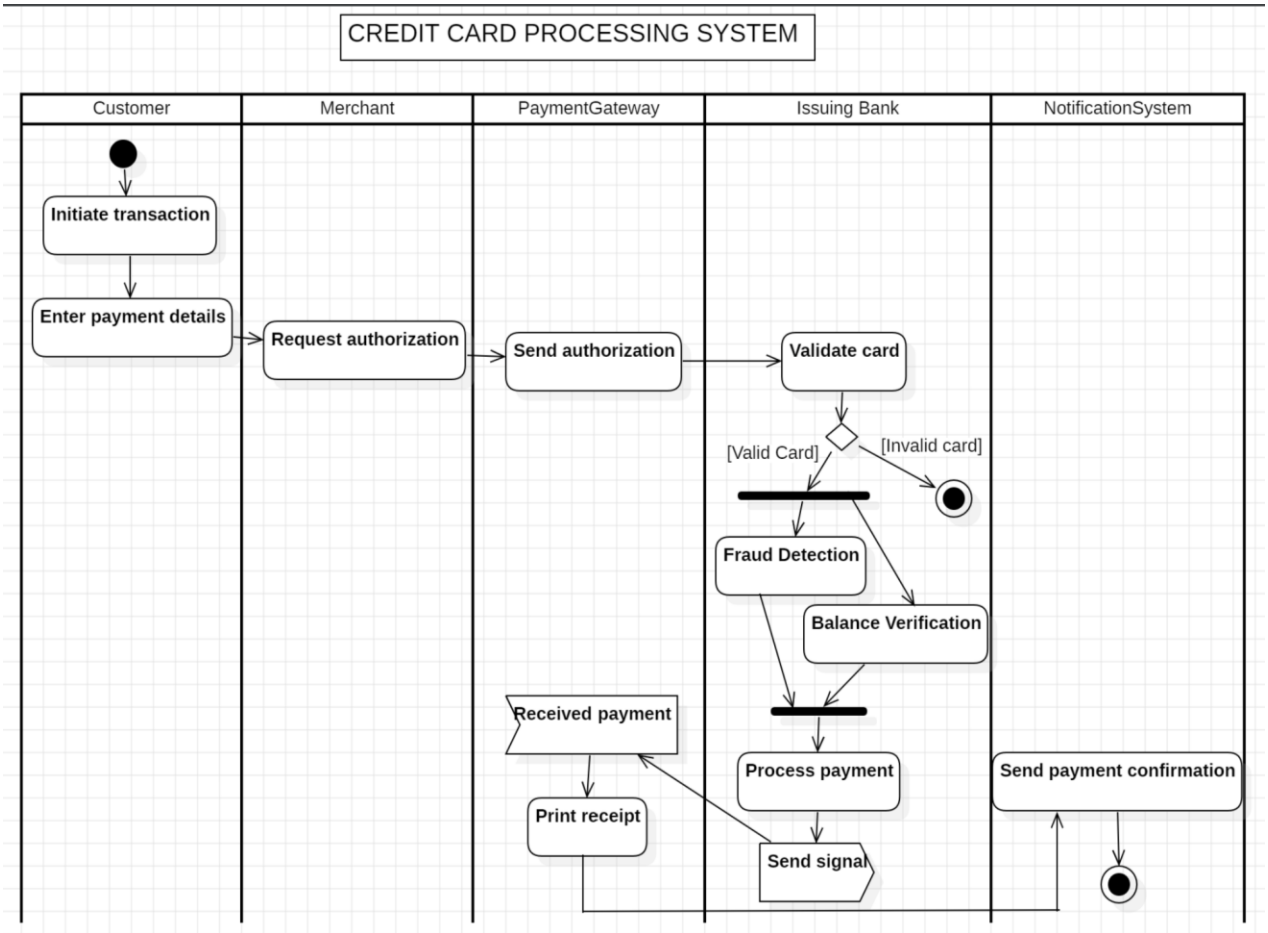
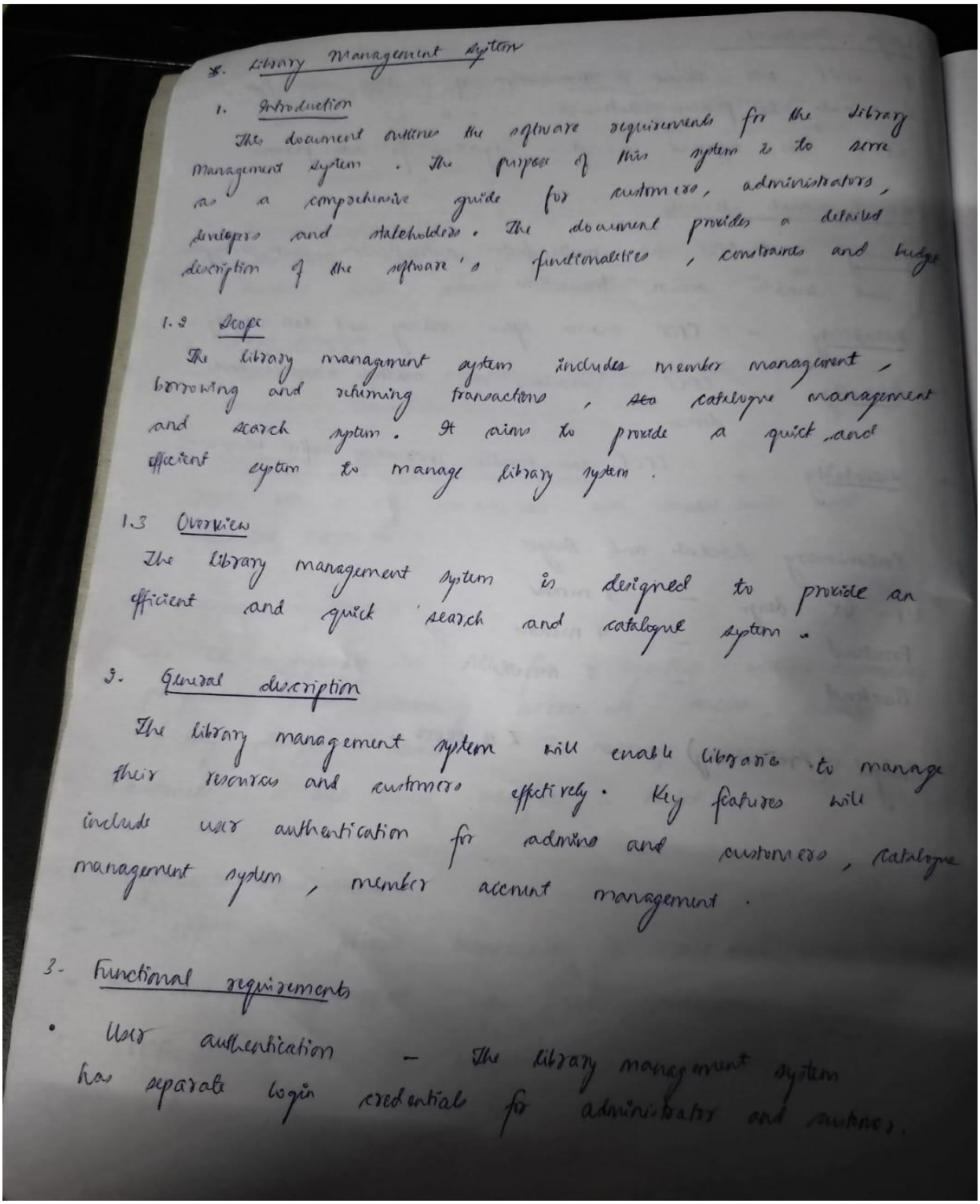


Fig2.5 Activity Diagram for Credit Card Processing System

The Activity Diagram illustrates the process of a credit card transaction. It begins with the Customer initiating the transaction by entering payment details. The Merchant then requests authorization from the Payment Gateway. The Gateway sends the request to the Issuing Bank, which validates the card. If the card is valid, the Bank performs fraud detection and balance verification. Upon successful validation, the Bank approves the transaction, and the Gateway processes the payment. The Merchant then prints a receipt and sends a confirmation to the Customer. Finally, a signal is sent to the Notification System to complete the transaction.

3.Library Management

Software Requirement Specification



*. Library Management System

1. Introduction

The document outlines the software requirements for the library Management system. The purpose of this system is to serve as a comprehensive guide for customers, administrators, developers and stakeholders. The document provides a detailed description of the software's functionalities, constraints and budget

1.2 Scope

The library management system includes member management, borrowing and returning transactions, catalogue management and search system. It aims to provide a quick and efficient system to manage library system.

1.3 Overview

The library management system is designed to provide an efficient and quick search and catalogue system.

2. General description

The library management system will enable libraries to manage their resources and customers effectively. Key features will include user authentication for admins and customers, catalogue management system, member account management.

3. Functional requirements

- User authentication – The library management system has separate login credential for administrator and customer.

- Catalogue management – The books are arranged in order of their genre and ISBN code.
- Borrowing and returning – The system will ~~autom~~ handle the process of borrowing and returning by scanning code.
- Member Account management – The system will ensure easy account creation and deletion for customers.

4. Interface requirements

The library management system requires barcode scanners for easy borrowing and returning books. It carries an API for easy sorting and searching system. For accessibility across all devices, a mobile friendly device is needed.

5. Performance requirements

- The library management system will search for books in under 3 seconds.
- The returning / borrowing process will be completed in about 5 seconds.
- The system can handle upto 5000 concurrent customers.

6. Design constraints

- The system will cancel a transaction if it takes more than 5 minutes.

7. Non functional attributes

- Security – The users information, along with their credential will be stored in a database encrypted with secure algorithms.
- Portability – The library management system is available on multiple websites and browsers.

- Reliability – The system ensures smooth transactions and data integrity.
- Scalability – The system can handle increasing traffic volume.

* 8. Preliminary schedule and budget

UI- UX design – 2 months
Frontend – 3 months
Backend – 4 months

Budget – 12,00,000/-

Class Diagram

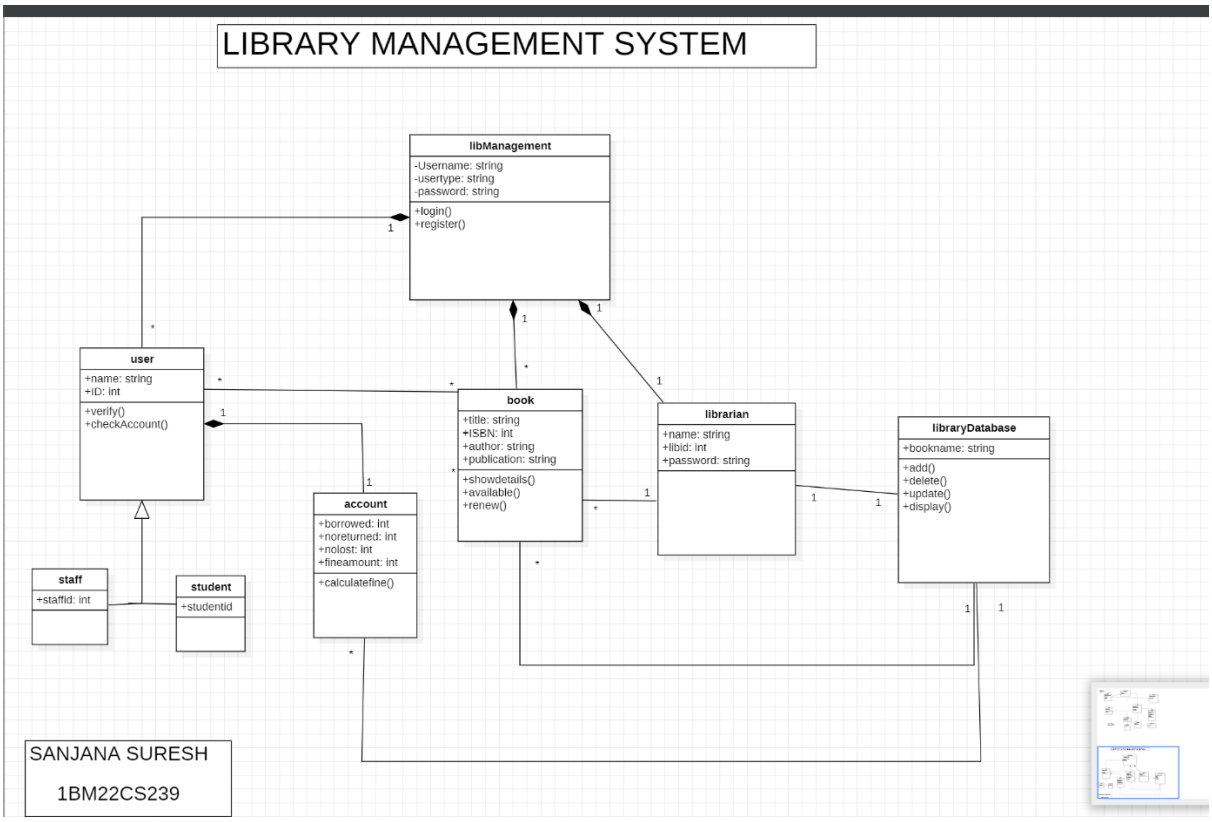


Fig3.1 Class Diagram for Library Management System

The UML class diagram illustrates a Hotel Management System. It shows classes like Receptionist, Staff, Manager, Hotel, Room, Guest, Reservation, Payment, and RoomBooking with their respective attributes and operations. Relationships are depicted, such as inheritance between Receptionist and Staff, and Manager and Staff. Associations are shown between classes like Hotel and Room, Guest and Reservation, and Reservation and Payment. The diagram provides a visual representation of the system's structure and how different entities interact within the hotel management context.

State Diagram

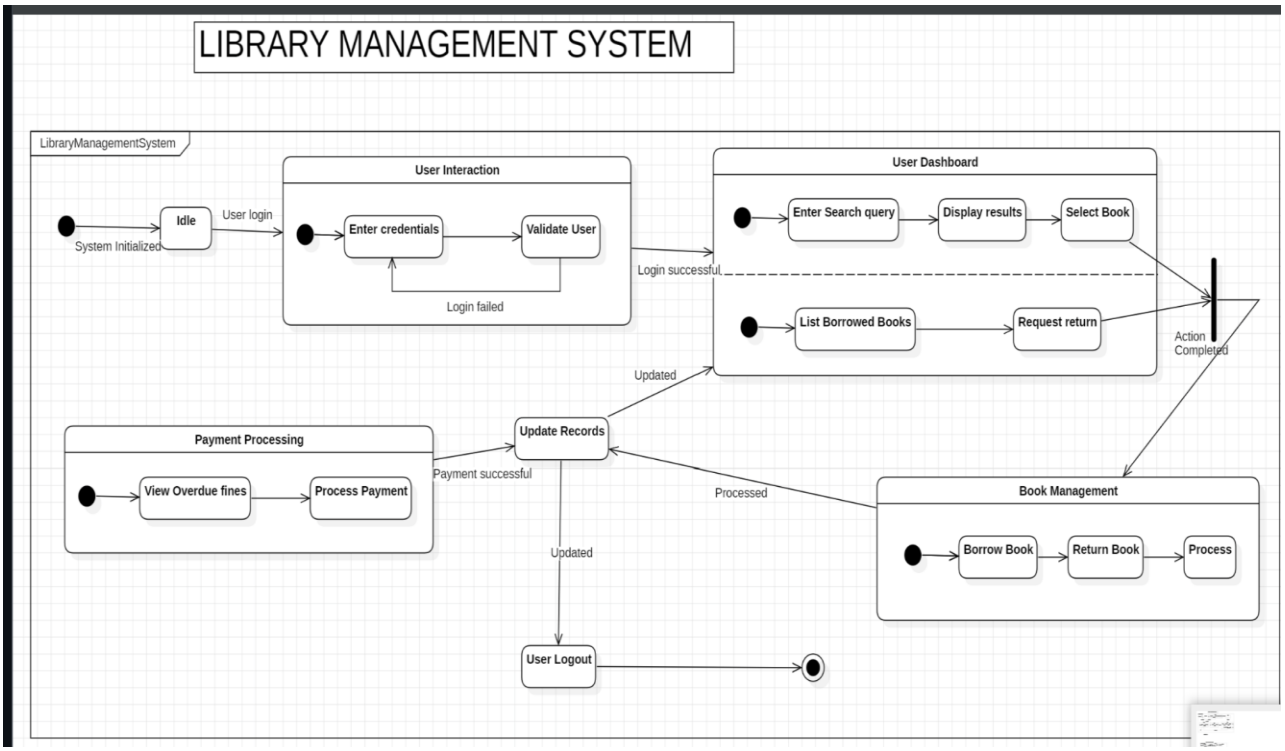


Fig3.2 State Diagram for Library Management System

The diagram illustrates the workflow of a Library Management System. It begins with system initialization and user login. Upon successful login, the user enters the User Dashboard where they can search for books, view borrowed books, and request returns. The system also handles payment processing for overdue fines and updates records accordingly. The Book Management module allows for borrowing and returning books. The system can be accessed by users and updated as needed. Finally, the user can log out of the system.

Sequence Diagram

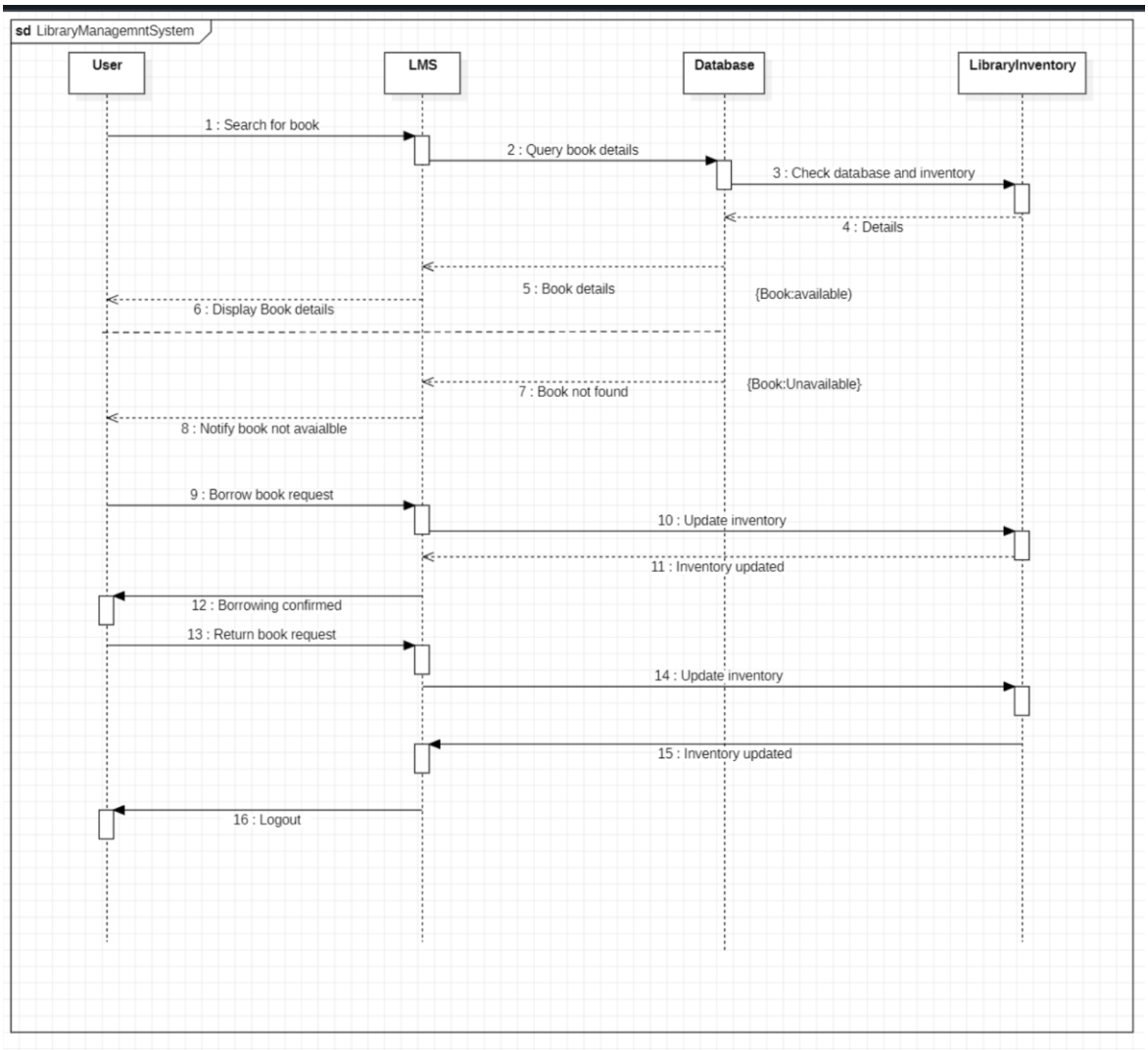


Fig3.3 Sequence Diagram for Library Management System

The sequence diagram illustrates the interaction between a User, the Library Management System (LMS), Database, and Library Inventory during a book search and borrowing process. The User initiates by searching for a book. The LMS queries the Database for book details, which then checks the Library Inventory for availability. Details are sent back to the User, who can then request to borrow the book. The LMS updates the Inventory accordingly and confirms the borrowing. Finally, the User can request to return the book, triggering another inventory update. This diagram provides a step-by-step visualization of the interactions involved in a book borrowing transaction within the Library Management System.

Use Case Diagram

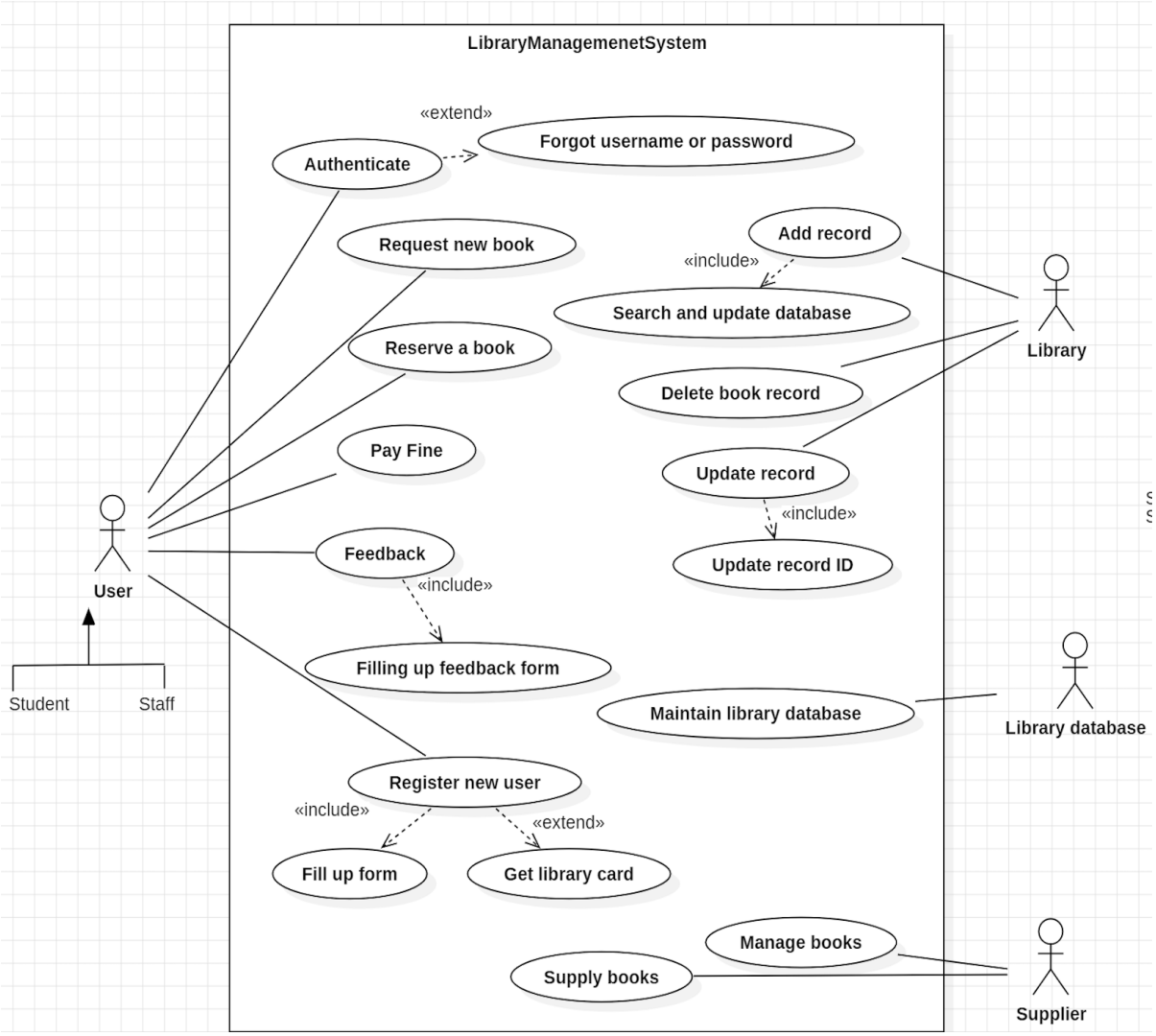


Fig3.4 Use Case Diagram for Library Management System

The Use Case Diagram illustrates the functionalities of a Library Management System. It shows various use cases like Authenticate, request new book, reserve a book, Pay Fine, Feedback, register new user, and Manage books. The diagram also includes relationships like “include” and “extend” to represent optional or alternative behaviours. For example, “Forgot username or password” extends the “Authenticate” use case. The system interacts with different actors such as User (including Student and Staff), Library, Library database, and Supplier. This diagram provides a high-level overview of the system’s capabilities and the interactions between different entities in the library management context.

Activity Diagram

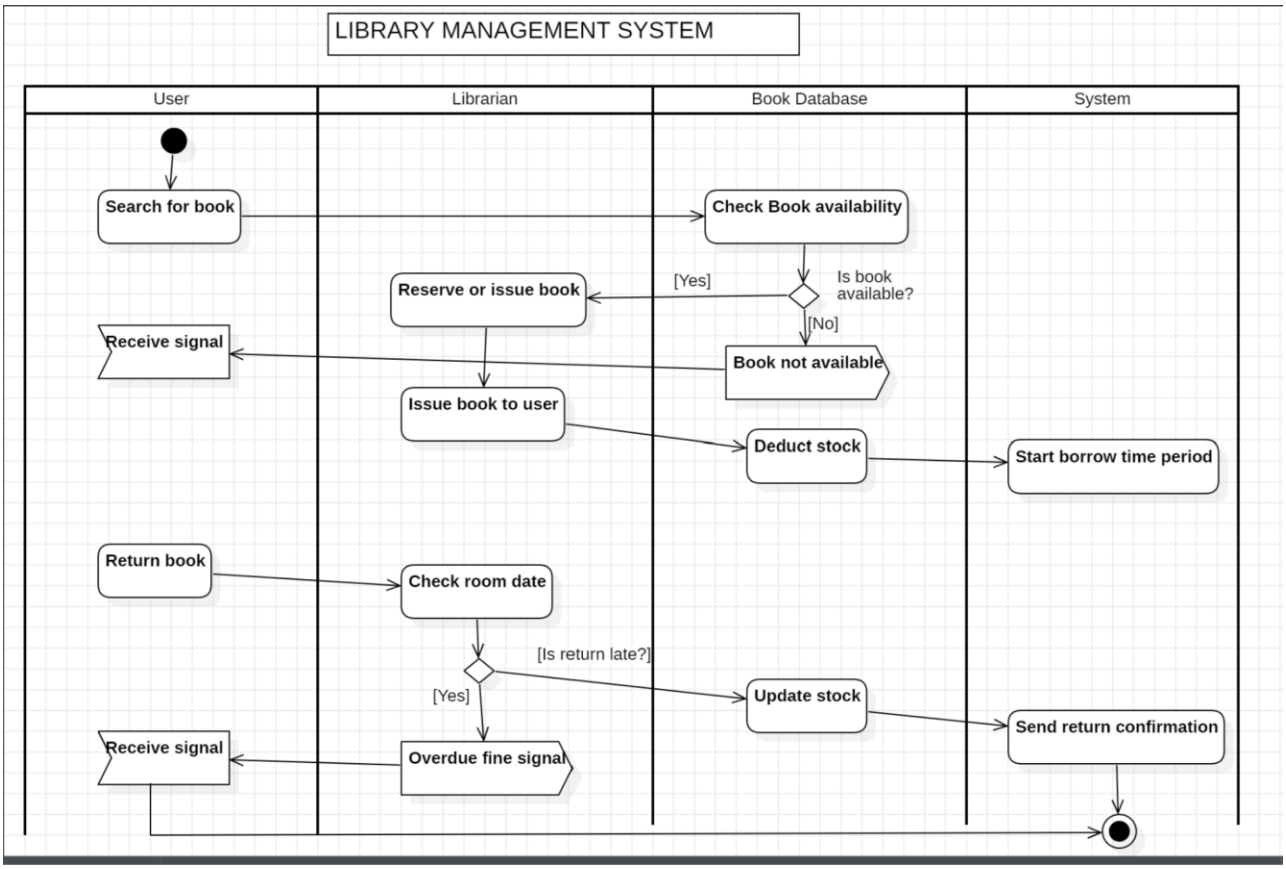
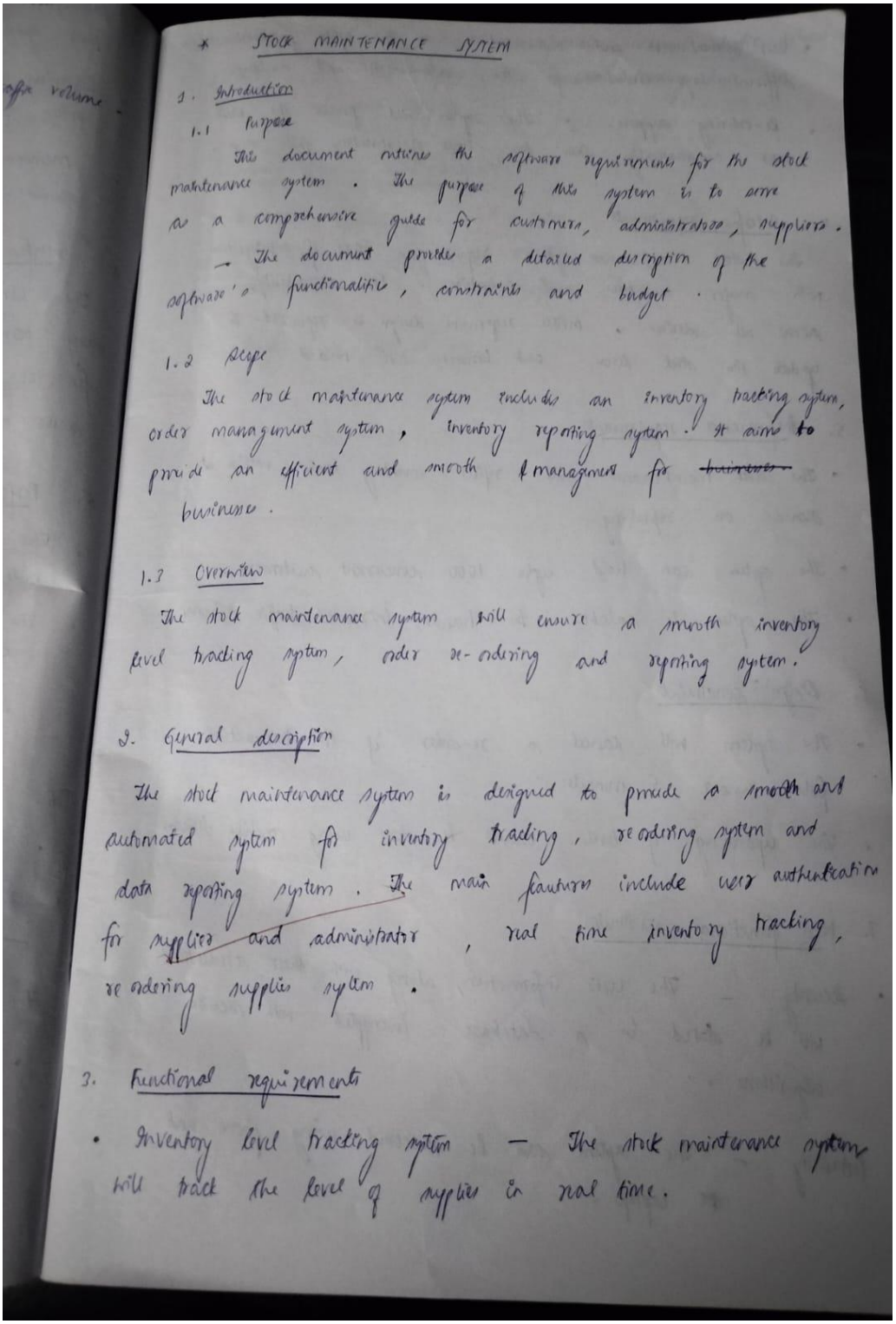


Fig3.5 Activity Diagram for Library Management System

The Activity Diagram illustrates the workflow of a Library Management System. It starts with the User searching for a book. The Librarian checks the Book Database for availability. If available, the book is issued to the User, and the stock is deducted. A borrow time period is initiated. When the User returns the book, the Librarian checks the return date. If overdue, a fine is imposed. The stock is updated, and a return confirmation is sent. This diagram visually represents the key steps and decisions involved in book borrowing and returning within the library system.

4. Stock Maintenance System

Software Requirement Specification



* STOCK MAINTENANCE SYSTEM

1. Introduction

1.1 Purpose

This document outlines the software requirements for the stock maintenance system. The purpose of this system is to serve as a comprehensive guide for customers, administrators, suppliers.

→ The document provides a detailed description of the software's functionalities, constraints and budget.

1.2 Scope

The stock maintenance system includes an inventory tracking system, order management system, inventory reporting system. It aims ~~to~~ provide an efficient and smooth ~~f~~ management for ~~business~~ businesses.

1.3 Overview

The stock maintenance system will ensure a smooth inventory level tracking system, order re-ordering and reporting system.

2. General description

The stock maintenance system is designed to provide a smooth and automated system for inventory tracking, reordering system and data reporting system. The main features include user authentication for suppliers and administrator, real time inventory tracking, reordering supplies system.

3. Functional requirements

- Inventory level tracking system — The stock maintenance system will track the level of supplies in real time.

- User authentication and authorization - The system provides different login credentials for the administrator and customer.
- Re-ordering system - The system will process the new order automatically when the levels of inventory falls down.

4. Interface requirements

The stock maintenance system requires an API for integration with major hardware and industries. For accessibility, across all devices, mobile responsive design is required. To update the stock levels, web browsers are needed.

5. Performance requirements

- The stock maintenance will reflect inventory levels within 2 seconds on restocking.
- The system can hold upto 1000 concurrent customers.
- The system is scalable to handle increased traffic volume.

6. Design constraints

- The system will cancel a re-order if the transaction fails under 5 minutes.
- The updating of levels cannot be done using mobile devices.

7. Non functional attributes

- Security - The user information, along with their credentials will be stored in a database encrypted with secure algorithms.
- Portability - The system can be accessed using phone and pc laptop.

- Scalability - The system can handle increased traffic volume.
- Reliability - The system ensures smooth trading hours and data integrity.

8. Preliminary schedule and budget

UI - UX design - 2 months
Frontend - 3 months
Backend - 4 months.

Budget - ₹20,00,000

Class Diagram

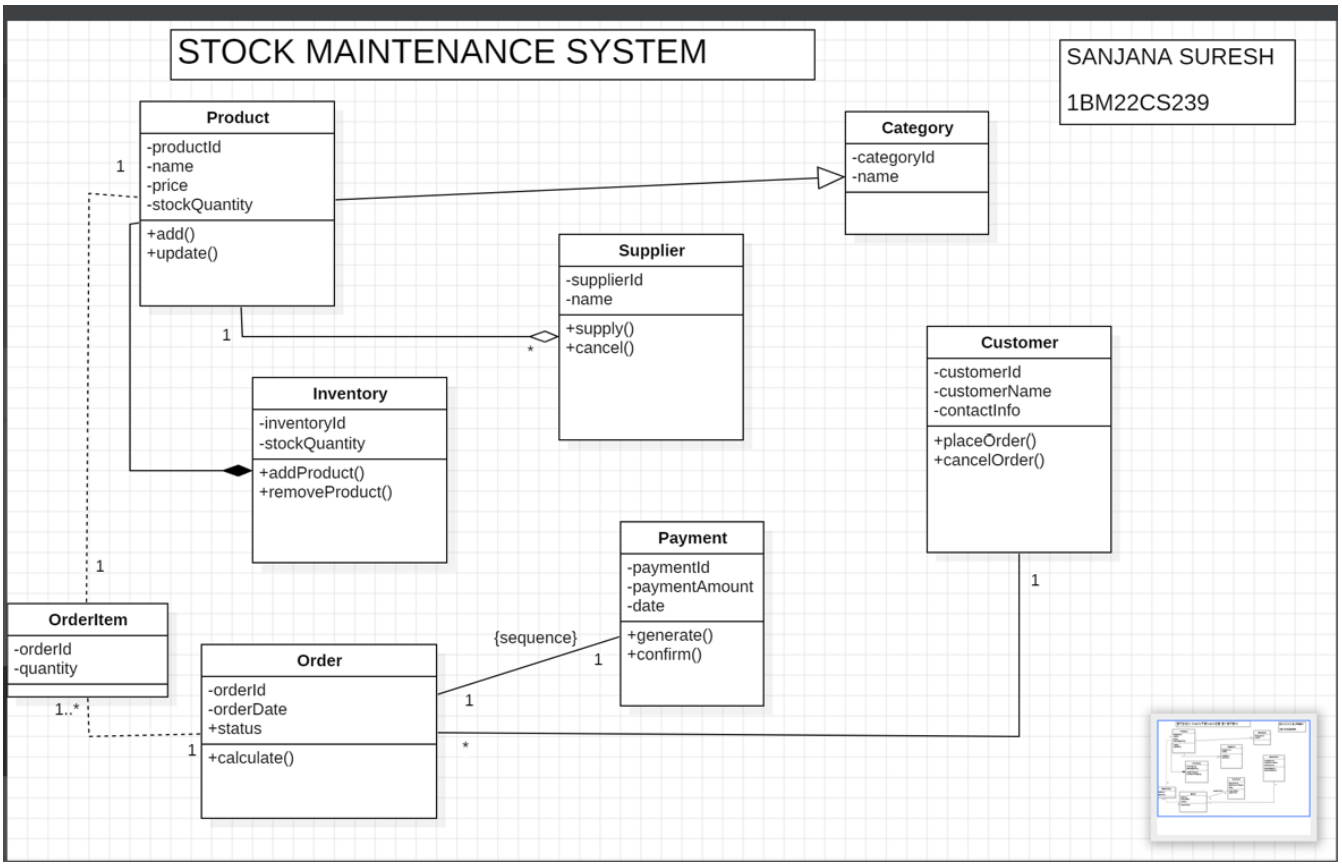


Fig4.1 Class Diagram for Stock Maintenance System

The UML class diagram illustrates a Stock Maintenance System. It shows classes like Product, Category, Supplier, Customer, Inventory, Order, OrderItem, and Payment with their respective attributes and operations. Relationships are depicted, such as one-to-many associations between Product and Inventory, Supplier and Product, Customer and Order, and Order and OrderItem. The diagram provides a visual representation of the system's structure and how different entities interact within the stock management context.

State Diagram

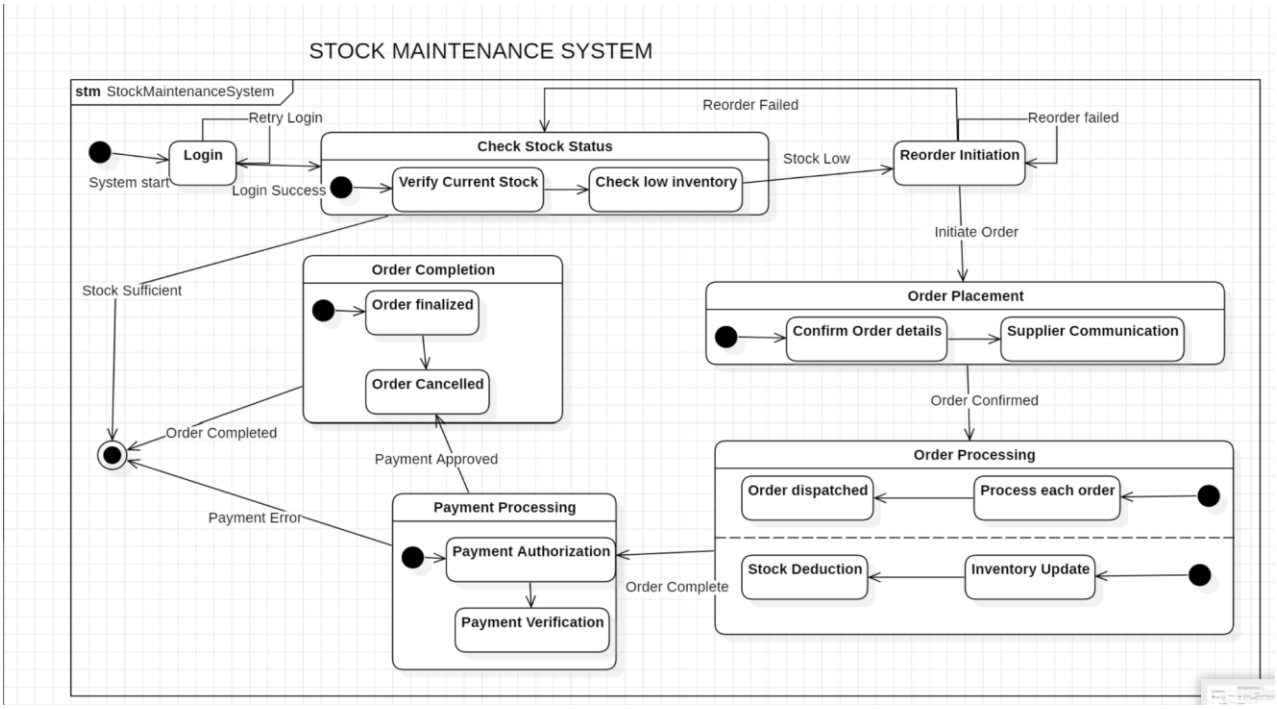


Fig4.2 State Diagram for Stock Maintenance System

The diagram illustrates the workflow of a Stock Maintenance System. It starts with system initialization and user login. The system then checks the stock status and initiates a reorder if stock is low. The order placement process involves confirming order details and communicating with the supplier. Once the order is confirmed, it proceeds to order processing, which includes order dispatch, stock deduction, and inventory update. Payment processing involves authorization and verification. The system can handle order completion, cancellation, or payment errors. This diagram provides a visual representation of the key steps involved in maintaining stock levels within the system.

Sequence Diagram

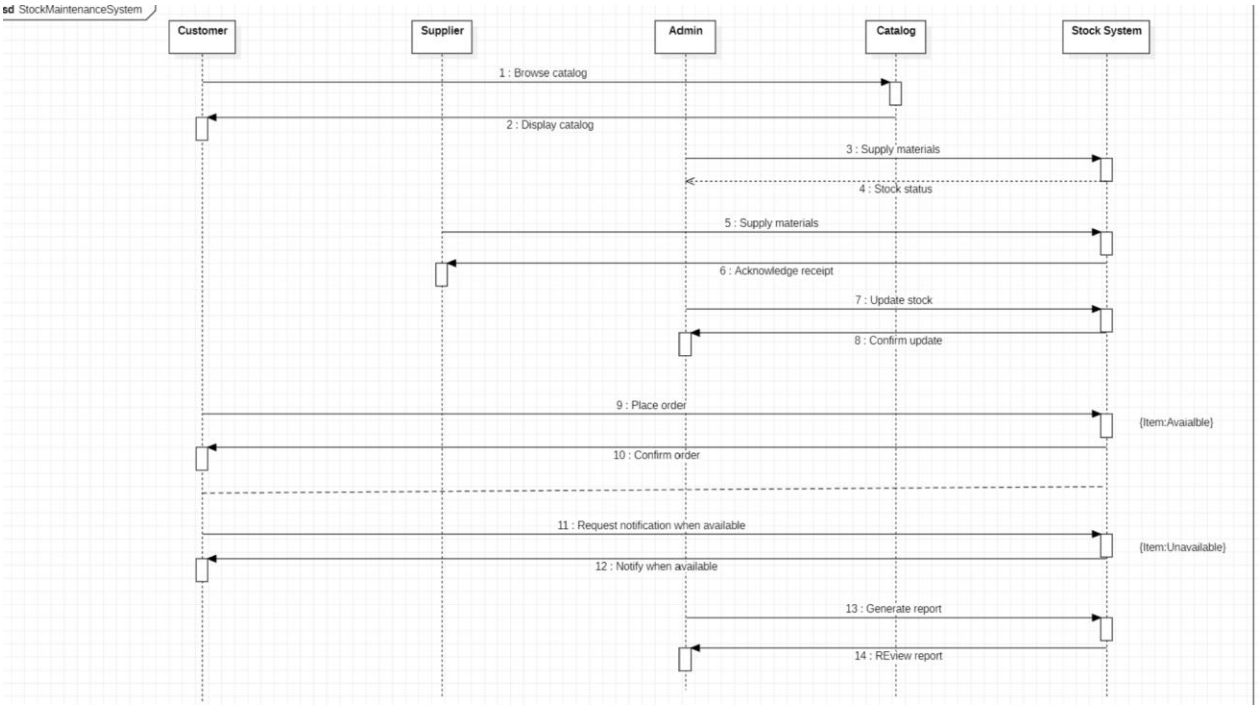


Fig4.3 Sequence Diagram for Stock Maintenance System

The sequence diagram illustrates the interactions between different entities in a stock management system. It begins with the Customer browsing the Catalog. The Supplier then supplies materials, and the Stock System updates its status. The Customer can place orders, and the Stock System confirms the order. The Customer can also request notifications when an item becomes available. Finally, the Admin can generate and review reports on stock levels. This diagram provides a step-by-step visualization of the interactions involved in managing stock within the system.

Use Case Diagram

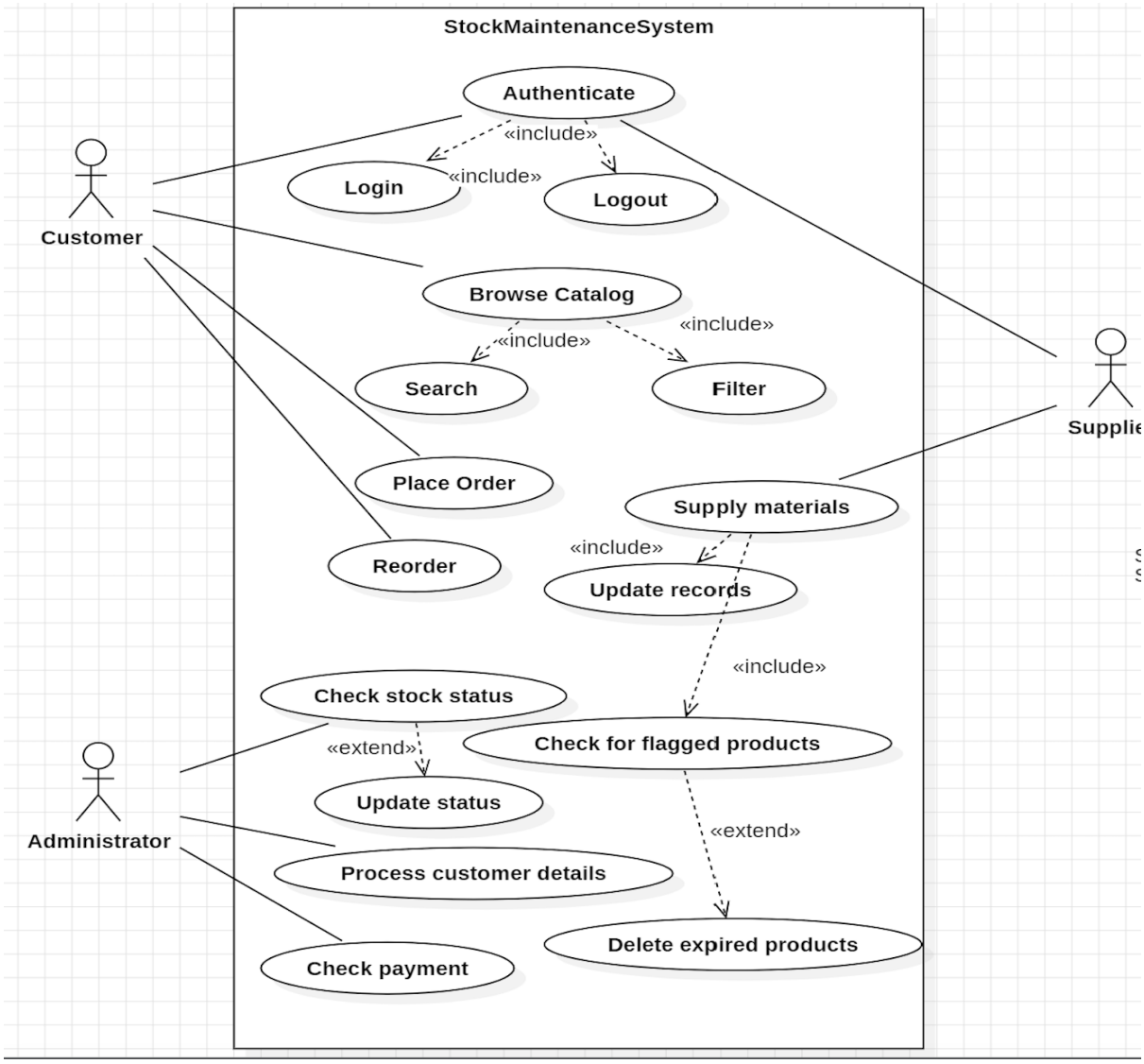


Fig4.4 Use Case Diagram for Stock Maintenance System

The Use Case Diagram illustrates the functionalities of a Stock Maintenance System. It shows various use cases like Authenticate, Browse Catalog, Place Order, Supply materials, Update records, and Check stock status. The diagram also includes relationships like "include" and "extend" to represent optional or alternative behaviours. For example, "Search" and "Filter" are included in "Browse Catalog". The system interacts with different actors such as Customer, Supplier, and Administrator. This diagram provides a high-level overview of the system's capabilities and the interactions between different entities in the stock management context.

Activity Diagram

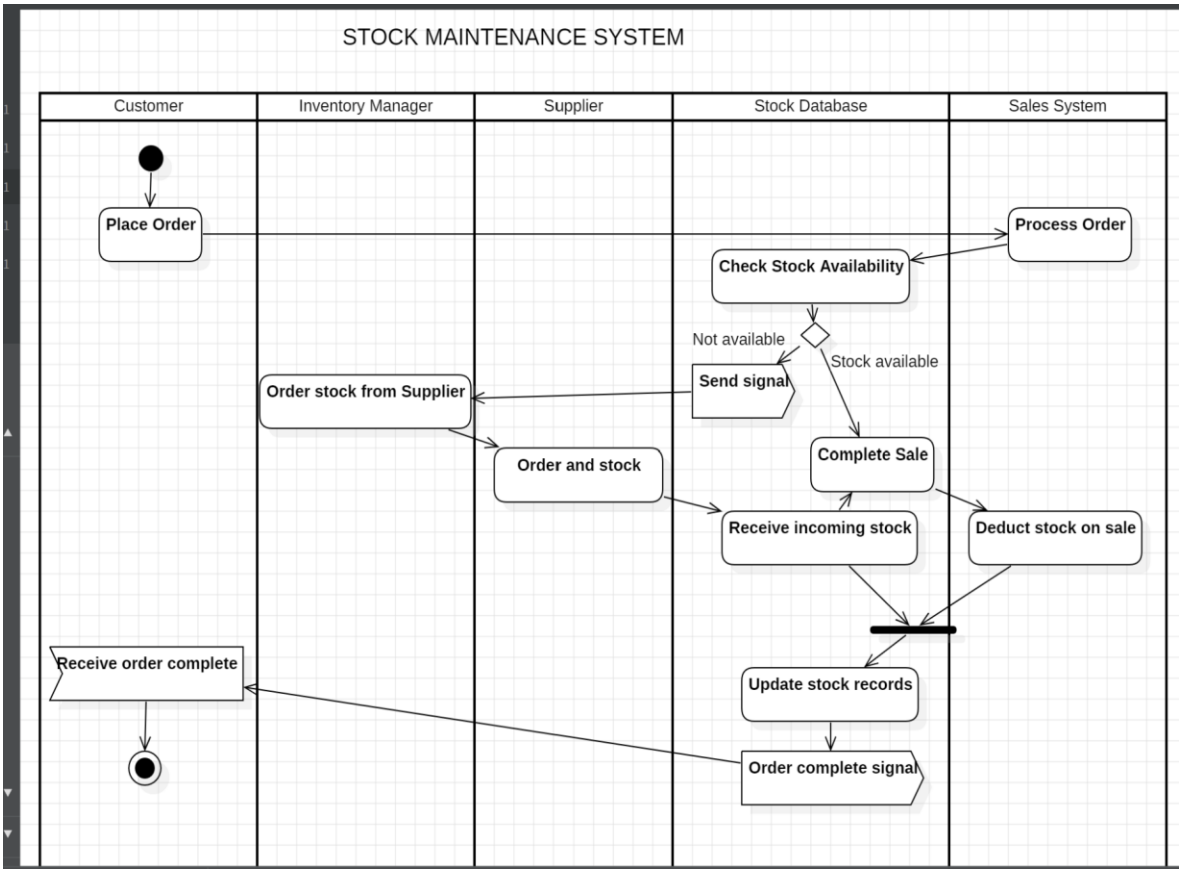
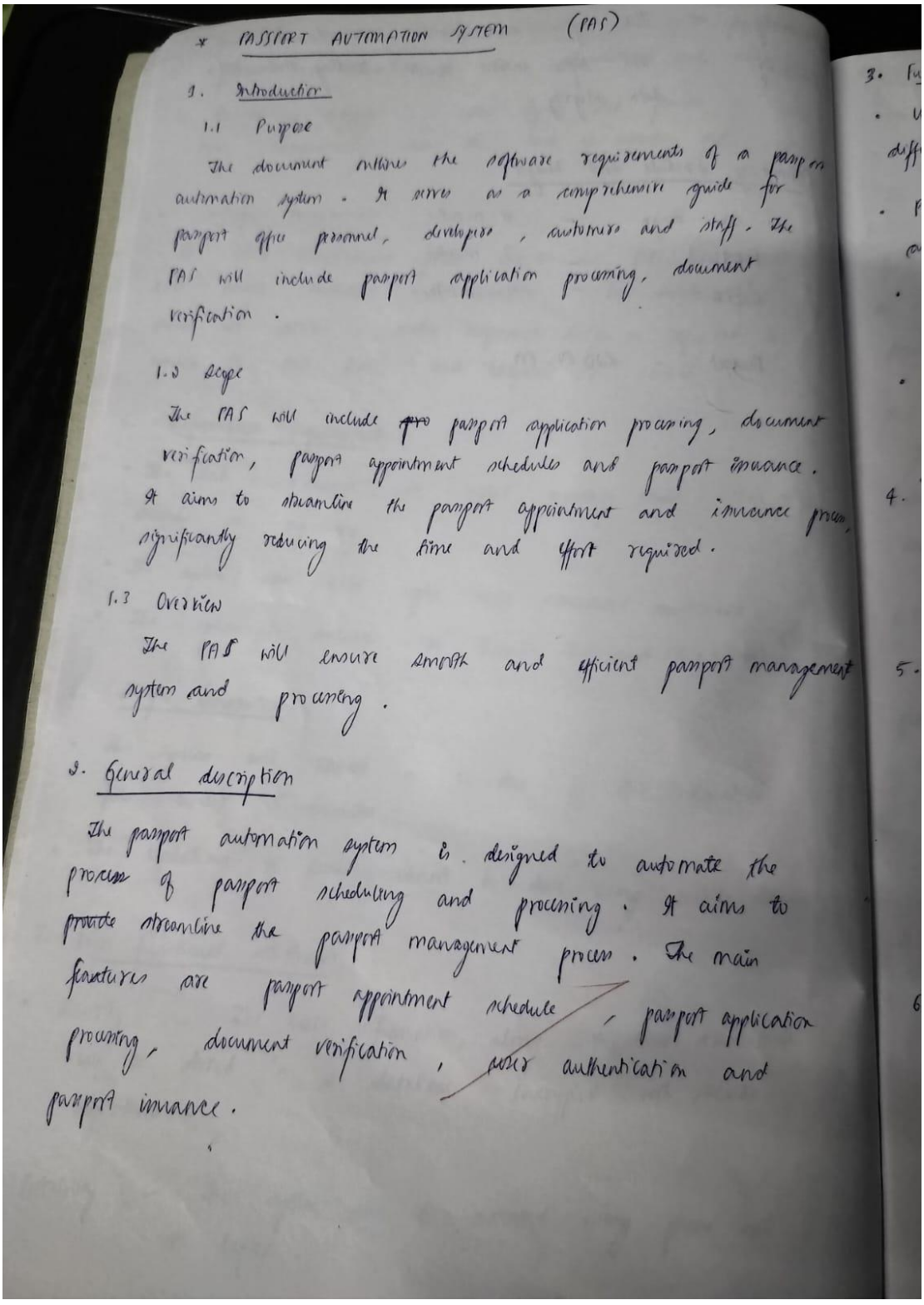


Fig4.5 Activity Diagram for Stock Maintenance System

The Activity Diagram illustrates the workflow of a Stock Maintenance System. It starts with the Customer placing an order. The Inventory Manager checks the Stock Database for availability. If the stock is not available, an order is placed with the Supplier. Once the Supplier sends the order and stock, the Inventory Manager receives it and deducts the stock on sale. The Sales System processes the order and completes the sale. Finally, the Inventory Manager updates the stock records and sends an order complete signal. This diagram visually represents the key steps and interactions involved in managing stock and fulfilling orders within the system.

5.Passport Automation System

Software Requirement Specification



* PASSPORT AUTOMATION SYSTEM (PAS)

1. Introduction

1.1 Purpose

The document outline the software requirements of a passport automation system. It serve as a comprehensive guide for passport office personnel, developers, customers and staff. The PAS will include passport application processing, document verification.

1.2 Scope

The PAS will include passport application processing, document verification, passport appointment schedules and passport issuance. It aims to streamline the passport appointment and issuance process, significantly reducing the time and effort required.

1.3 Overview

The PAS will ensure smooth and efficient passport management system and processing.

2. General description

The passport automation system is designed to automate the process of passport scheduling and processing. It aims to provide streamline the passport management process. The main features are passport appointment schedule, passport application processing, document verification, user authentication and passport issuance.

3. Functional attributes

- User authentication and authorization – The PAS system offers different login credentials for applicants and administrators.
- Passport appointment system – The PAS app has an easy and accessible appointment process.
- Passport application processing – The PAS displays and shows real time processing of applicant's application.
- Document verification system – The PAS has an efficient verification system.

4. Interface requirements

Web based interface for applicants and staff. For accessibility on various devices, mobile responsive design.

5. Performance requirements

- PAS can handle upto 10,000 concurrent users.
- Process new applications within 4 minutes.
- Data backup every 6 hours.
- PAS can generate reports within 30 seconds.

6. Non functional requirements

- Security – PAS implements a robust data encryption, secure authentication.
- Reliability – PAS will ensure system stability, data integrity
- Scalability – PAS has the ability to handle increasing application volume.

7. Preliminary Schedule and Budget

- Development Timeline - 9 months.

  UI - UX design - 2 months
  Frontend - 3 months
  Backend - 4 months

- Budget - Rs 25,00,000.

Class Diagram

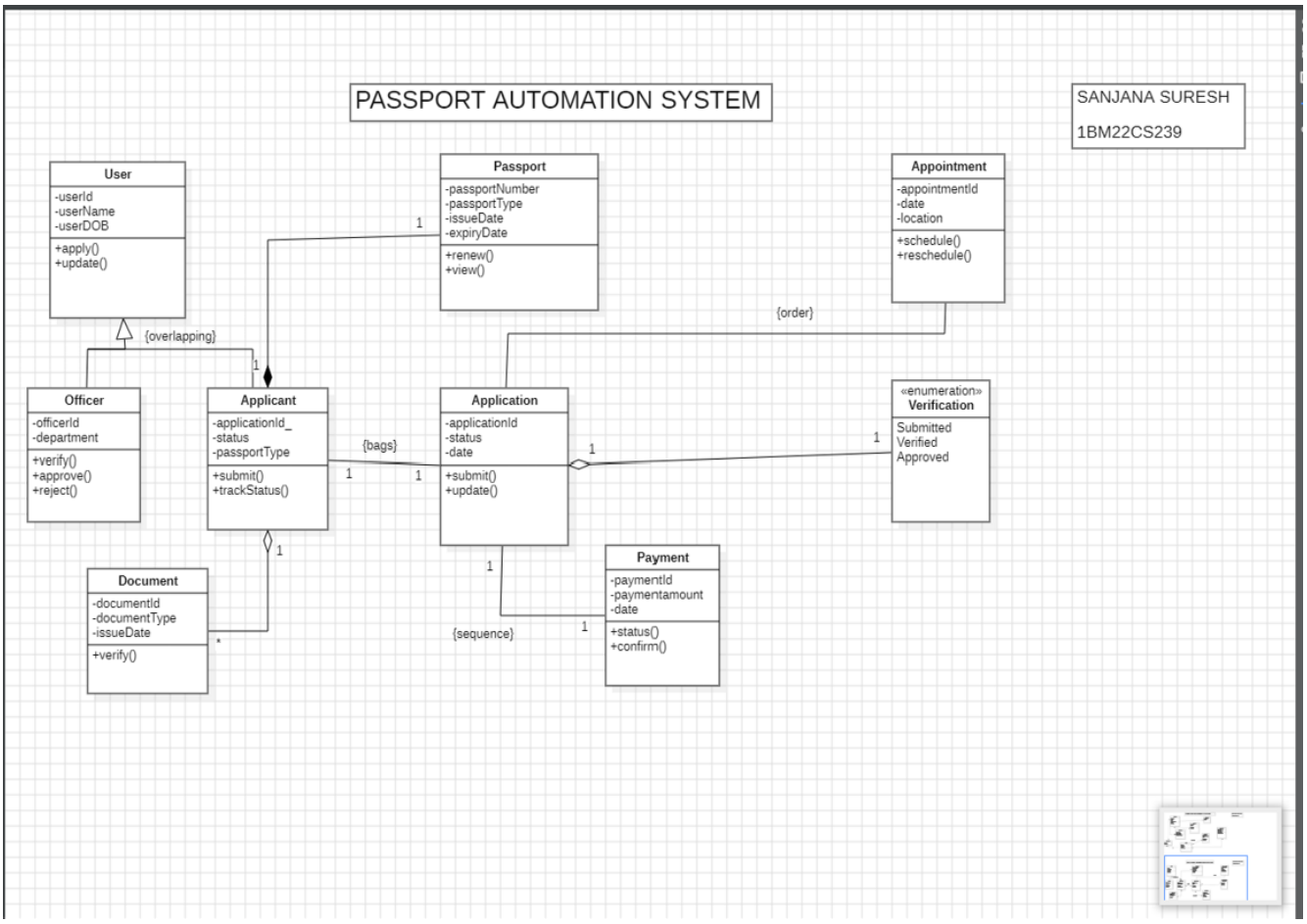


Fig5.1 Class Diagram for Passport Automation System

The UML class diagram illustrates a Passport Automation System. It shows classes like User, Passport, Appointment, Officer, Applicant, Application, Document, Verification, and Payment with their respective attributes and operations. Relationships are depicted, such as one-to-many associations between User and Passport, Applicant and Application, and Application and Verification. The diagram provides a visual representation of the system's structure and how different entities interact within the passport application and management process.

State Diagram

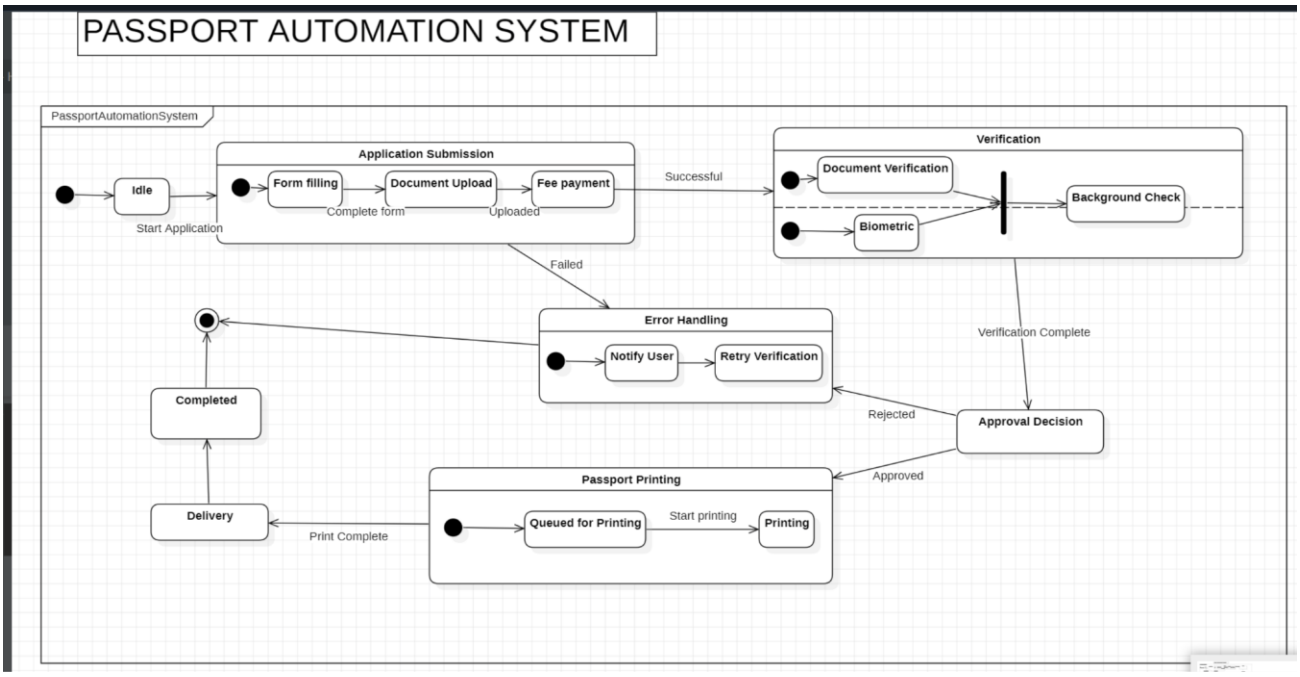


Fig5.2 State Diagram for Passport Automation System

The Activity Diagram illustrates the workflow of a Passport Automation System. It starts with the "Idle" state and transitions to "Application Submission" where the user fills the form, uploads documents, and pays the fee. Upon successful completion, it moves to "Verification" stage for document verification and background checks. If successful, it proceeds to "Approval Decision". If approved, the passport is printed and delivered. The diagram also includes "Error Handling" and retry mechanisms for failed stages.

Sequence Diagram

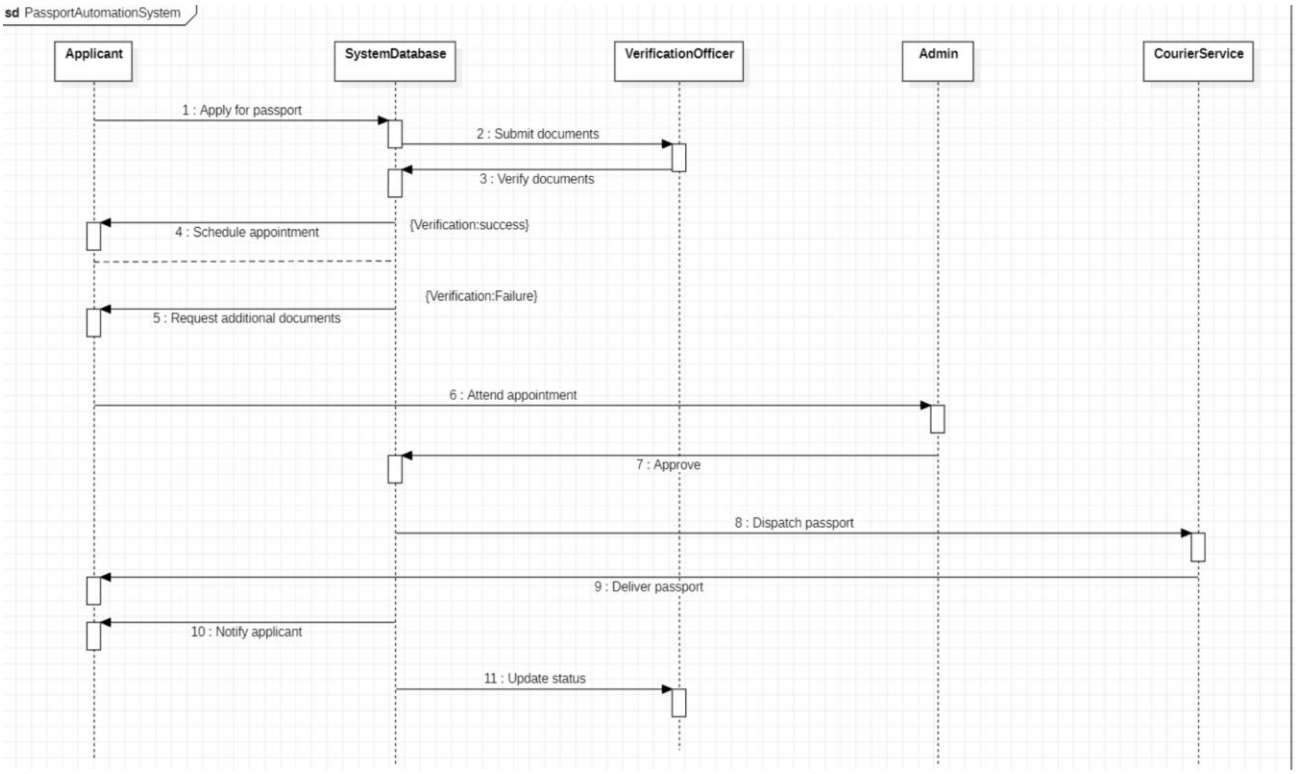


Fig5.3 Sequence Diagram for Passport Automation System

The sequence diagram illustrates the workflow of a passport application system. It starts with the Applicant applying for a passport and submitting documents to the System Database. The Verification Officer verifies the documents and schedules an appointment if successful. If verification fails, the Applicant is requested to submit additional documents. The Applicant attends the appointment, and the Verification Officer approves the application. The admin dispatches the passport, and the Courier Service delivers it to the Applicant. Finally, the System Database updates the application status, and the Applicant is notified. This diagram provides a step-by-step visualization of the interactions between different entities involved in the passport application process.

Use Case Diagram

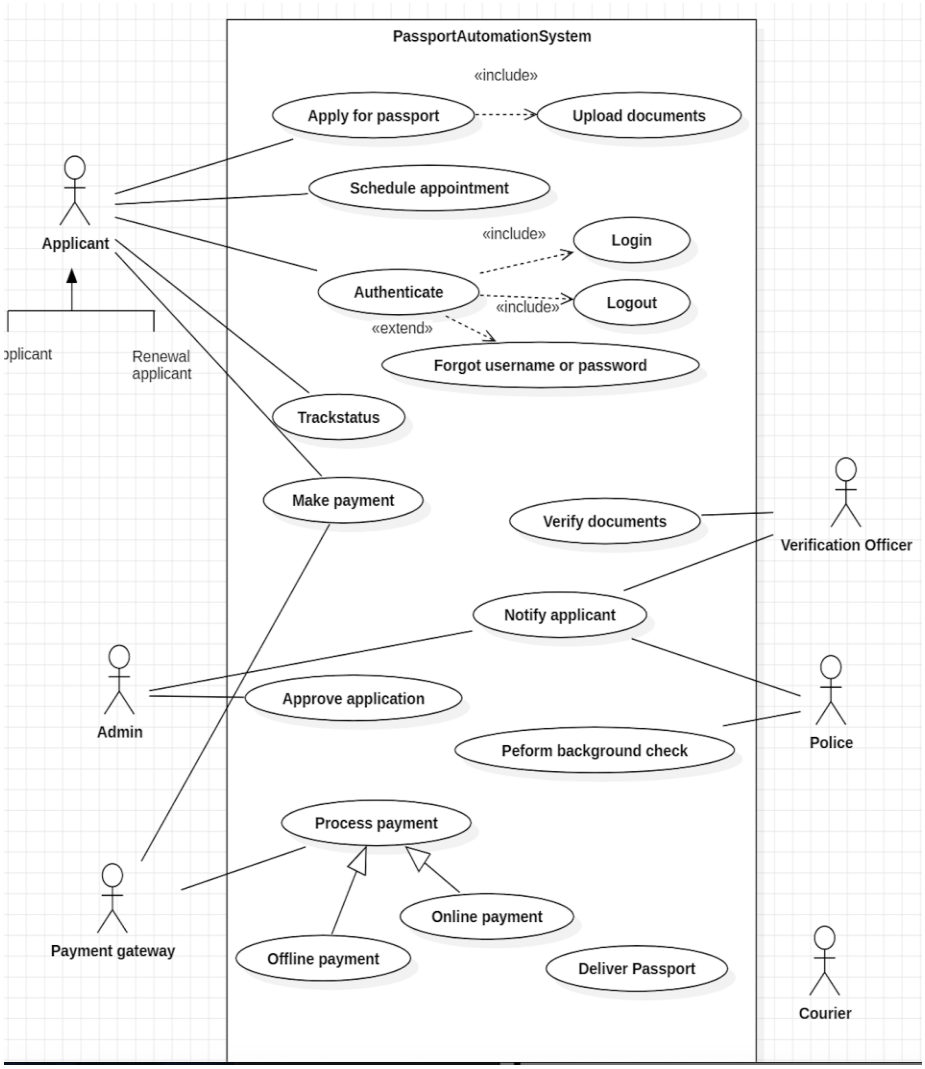


Fig5.4 Use Case Diagram for Passport Automation System

The Use Case Diagram illustrates the functionalities of a Passport Automation System. It shows various use cases like Apply for passport, Schedule appointment, Make payment, Verify documents, Approve application, and Deliver Passport. The diagram also includes relationships like "include" and "extend" to represent optional or alternative behaviors. For example, "Upload documents" is included in "Apply for passport" and "Forgot username or password" extends "Authenticate". The system interacts with different actors such as Applicant, Verification Officer, Police, Courier, Payment Gateway, and Admin. This diagram provides a high-level overview of the system's capabilities and the interactions between different entities in the passport application process.

Activity Diagram

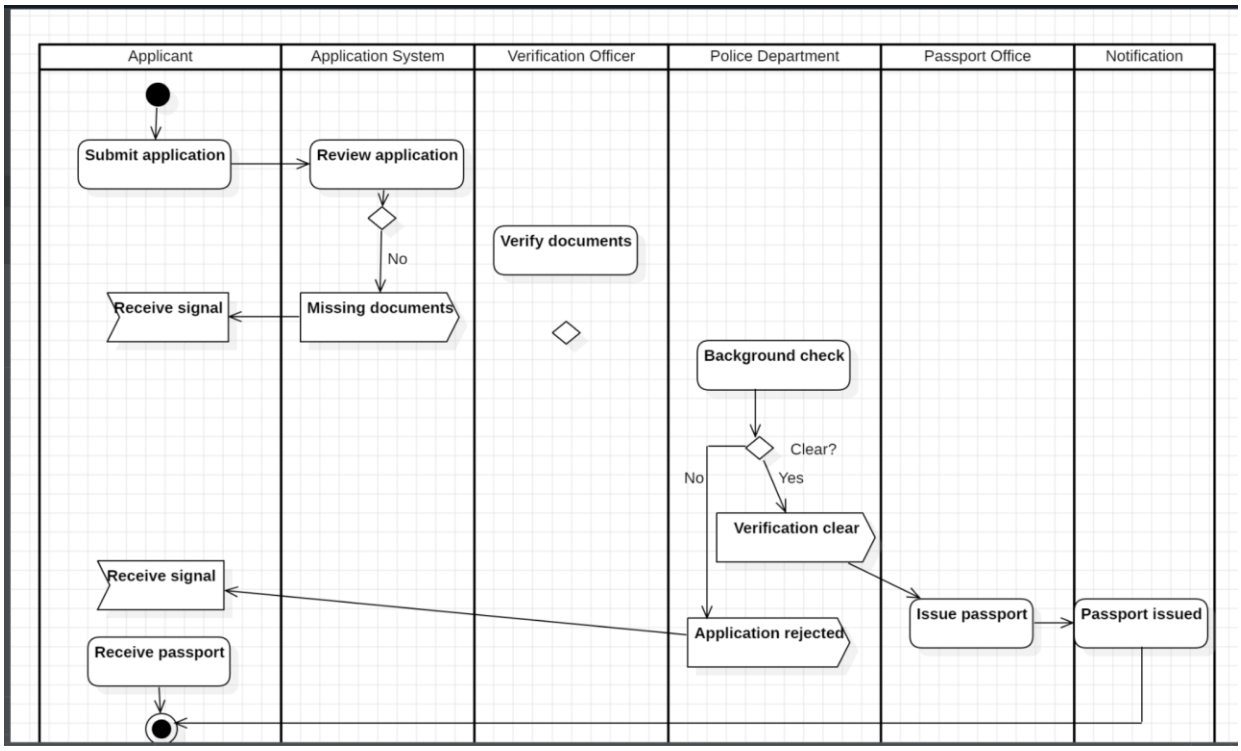


Fig5.5 Activity Diagram for Passport Automaton System

The Activity Diagram illustrates the workflow of a passport application process. It starts with the Applicant submitting the application. The Application System reviews it and verifies the documents. If documents are missing, the Applicant is notified and asked to submit them. The Police Department conducts a background check. If clear, the Verification is cleared and the Passport Office issues the passport. If the background check is not clear or the application is rejected, the process ends. This diagram provides a visual representation of the key steps and decisions involved in obtaining a passport