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:

Disha H Jain

1BM23CS095

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 Disha H Jain (1BM23CS095) during the 5th Semester August 2025-December 2025.

Signature of the Faculty Incharge:

Name of your Batch Incharge and designation: Sonika Sharma D , Assistant Professor

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

Table of Contents

Class Diagram Requirements: Minimum 7 classes, 4 attributes and 4 operations in each class, associations, association name, association end names, multiplicity, association class, enumeration, qualified association, aggregation, composition, generalization, ordered, sequencing, multiplicity of attribute, brief description of the Class diagram
State Diagram: one simple state diagram, one advance state diagram (either Sub Machine or Nested State, include any one of the concurrency in your design) Requirements: minimum of 6 states for both the state diagram with state name, do activities, events, guard condition, brief description of State diagram
Use-Case Diagram: Minimum 5 use cases to be identified, and design one simple use case diagram and one advanced use case diagram(ie using include,extend and generalization relationship) and explanation as there in the solution manual.
Sequence Diagram: Write the scenario for any two use case transactions completely. and design the simple sequence diagram with a minimum of 5 objects and communication between them. Design advanced sequence diagrams using passive and transient objects. Brief explanation for each

Activity Diagram: Design the simple activity diagram showing the algorithm or workflow. Design the advanced activity diagram with swimlanes showing the responsible person for the swimlane and also write the brief explanation for both.

1. Hotel Management System

Problem Statement

A hotel needs a computerized system to manage its day-to-day operations efficiently. The current manual process used for managing room bookings, guest check-in/check-out, billing, and staff coordination is time-consuming, error-prone, and difficult to track. To overcome these problems, the hotel requires a Hotel Management System that automates major operational activities and improves service quality.

The system should allow customers to search for available rooms based on type, price, and date. It should support online and offline room reservations. When guests arrive, the receptionist should be able to check them in, assign rooms, and maintain guest records. Upon checkout, the system must

generate accurate bills that include room charges, additional services (restaurant, laundry, spa, etc.), and applicable taxes.

SRS-Software Requirements Specification

1. Introduction

The Hotel Management System (HMS) is intended to automate and streamline the operations of a hotel by enabling efficient management of room bookings, guest information, billing, staff coordination, and service tracking. The system aims to replace the hotel's existing manual processes, which are prone to errors, delays, and inconsistencies, with a computerized solution that ensures accuracy, reliability, and improved customer service. This document describes the functional and non-functional requirements, system features, constraints, and the overall purpose of the proposed system. The target users include hotel administrators, receptionists, housekeeping staff, service personnel, and customers. The SRS provides a detailed description of the system to developers and stakeholders and ensures a common understanding of the system's expected behavior.

1.1 Purpose

The purpose of the Hotel Management System is to simplify the management of hotel-related operations by providing a centralized platform that handles room reservations, customer check-in and check-out, billing, payment processing, and service management. The system is designed to reduce manual effort, minimize human errors, and enhance operational efficiency. It will also support hotel administrators by offering tools for generating reports, updating room and service information, and monitoring hotel performance. This SRS serves as a guideline for developers to design and implement the system according to the hotel's operational requirements.

1.2 Scope

The scope of the Hotel Management System includes functions such as room search, booking, reservation management, room allocation, guest registration, billing, payment handling, staff management, and report generation. Guests will be able to inquire about room availability and make reservations, either online or directly at the hotel. Receptionists will manage guest check-ins and check-outs, update room statuses, and handle cancellations. Additionally, the system will track housekeeping schedules, manage service requests such as laundry or food orders, and generate comprehensive invoices at the time of checkout. The system will also allow administrators to add new rooms, categorize room types, set pricing structures, update service charges, and view revenue and occupancy reports. The focus is on ensuring a smooth workflow for hotel operations while providing a stable, secure, and user-friendly interface.

1.3 Definitions, Acronyms, and Abbreviations

The term HMS refers to the Hotel Management System. GUI refers to the Graphical User Interface used for interacting with the system. Admin refers to a hotel administrator with elevated access privileges. Guest refers to a customer using the hotel's services. Receptionist refers to authorized hotel staff who handle customer-facing operations.

1.4 System Overview

The Hotel Management System will operate as a centralized application that integrates different

modules responsible for booking management, guest management, billing, service tracking, and reporting. Each module communicates with others to maintain consistent data and ensure coordinated operations. The system will maintain a database containing records of rooms, guests, transactions, staff information, and service logs. It will support real-time updates to room availability and will provide automated calculations for billing and tax processing. The system will run on hotel computer systems with a secure login mechanism to ensure authorized access.

2. Overall Description

2.1 Product Perspective

The Hotel Management System functions as a standalone software application designed to manage the internal processes of a hotel. It interacts with its database to retrieve and store information and provides a GUI interface for all user roles. The system maintains continuous updates of room availability and service records to prevent double bookings or mismanagement. All operations, from reservation to checkout, are interconnected so that information flows smoothly across modules, ensuring accuracy and consistency.

2.2 Product Functions

The primary functions of the system include enabling customers or receptionists to check room availability and make reservations, performing guest check-in by capturing personal information and assigning rooms, managing room status, and handling guest check-out along with automatic bill generation. The system also processes additional service requests and includes tax and discount calculations in final invoices. It supports cancellation of bookings, tracks service logs, and allows administrators to manage staff details, room details, and pricing information. Furthermore, the system can generate periodic reports providing insights into hotel performance, occupancy rates, and revenue trends.

2.3 User Characteristics

The system will be used by various categories of users, each with different levels of technical expertise. Hotel receptionists are expected to have basic computer skills and will use the system for booking, check-in, and check-out operations. Housekeeping and service staff will use the system minimally for updating room status or receiving service requests. Administrators are expected to have higher technical proficiency as they will manage service rates, staff data, room configurations, and access reports. Guests who interact with the system online are generally expected to be familiar with basic web usage.

2.4 Constraints

The system must operate within constraints such as reliable hardware, stable internet connectivity for online bookings, and secure authentication processes. Data accuracy must be maintained consistently, meaning database failures or unauthorized access must be avoided. The system must comply with hotel policies regarding data privacy and retention. Any downtime must be minimized because hotel operations depend heavily on real-time data. Additionally, the system's design must accommodate scalability in case the hotel expands its facilities or operations.

2.5 Assumptions and Dependencies

It is assumed that hotel staff will receive proper training before using the system. The system depends on the availability of a stable database and supporting technologies. It also assumes that customers provide accurate information during booking and that staff follow defined procedures for updating room and guest data. The system relies on local computer infrastructure and, if integrated online, requires reliable internet services.

3. Specific Requirements

3.1 Functional Requirements

The system must allow users to search for rooms and view availability based on room type, price, and booking dates. It must permit customers or receptionists to create reservations and store them in the database. During check-in, the system must record guest details, assign rooms, and update room status automatically. Service requests such as food, laundry, or housekeeping must be added to a guest's account, and these charges must be reflected in the bill. During checkout, the system must calculate total charges, including room rates, service charges, taxes, and discounts, and generate an invoice. The system must allow cancellation of bookings and update availability accordingly. Administrators must be able to add or remove rooms, update pricing information, create staff accounts, and access detailed performance reports. All users must authenticate using valid login credentials before accessing the system.

3.2 Non-Functional Requirements

The system must be reliable and ensure continuous availability during hotel working hours. It must respond quickly to user inputs, and operations like booking, check-in, and billing should be processed within a few seconds. The system interface must be simple and intuitive to ensure ease of use for non-technical staff. Security measures must be implemented to protect sensitive customer information and financial records. The system must be scalable to support the addition of more rooms or branches in the future. It must also support data backup and recovery mechanisms to prevent data loss.

3.3 Performance Requirements

The system should be able to handle multiple users concurrently without significant delays. Room availability updates should occur instantly after booking or check-in operations. The database should be optimized so that searches for rooms, guests, or reports are completed efficiently. Billing calculations must be accurate and performed in real time. The system should support hundreds of active records at any given time without degrading performance.

4. Conclusion

The Hotel Management System provides an efficient, reliable, and user-friendly platform for managing hotel operations. By replacing manual processes with an automated system, the hotel can improve accuracy, speed, and customer satisfaction. This SRS document outlines the complete set of requirements necessary for the successful design and implementation of the system. It ensures that developers, stakeholders, and users share a common understanding of the system's functional expectations and operational constraints.

Fig 1.1 Class Diagram for Hotel Management System

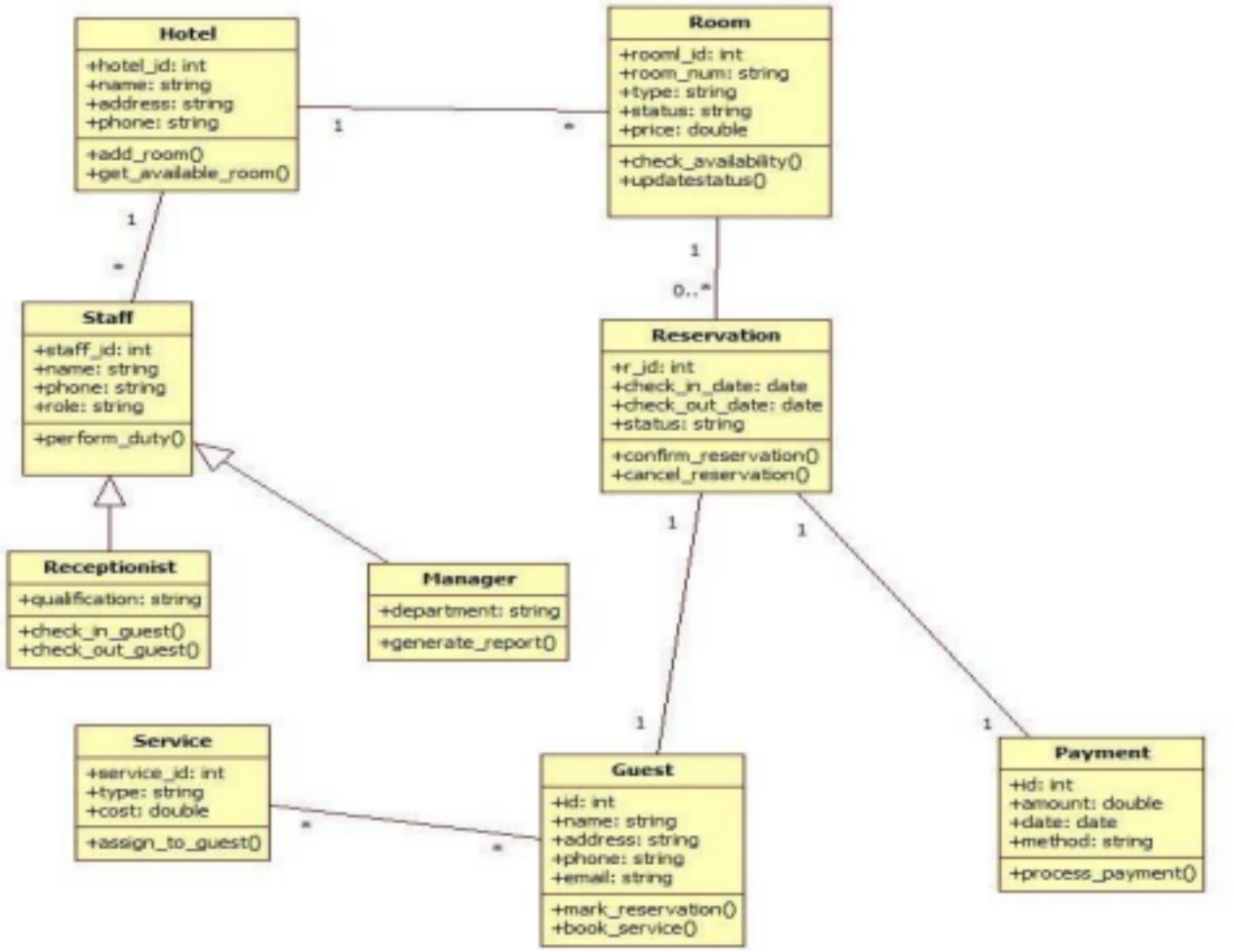


Fig 1.2 State Diagram for Hotel Management System

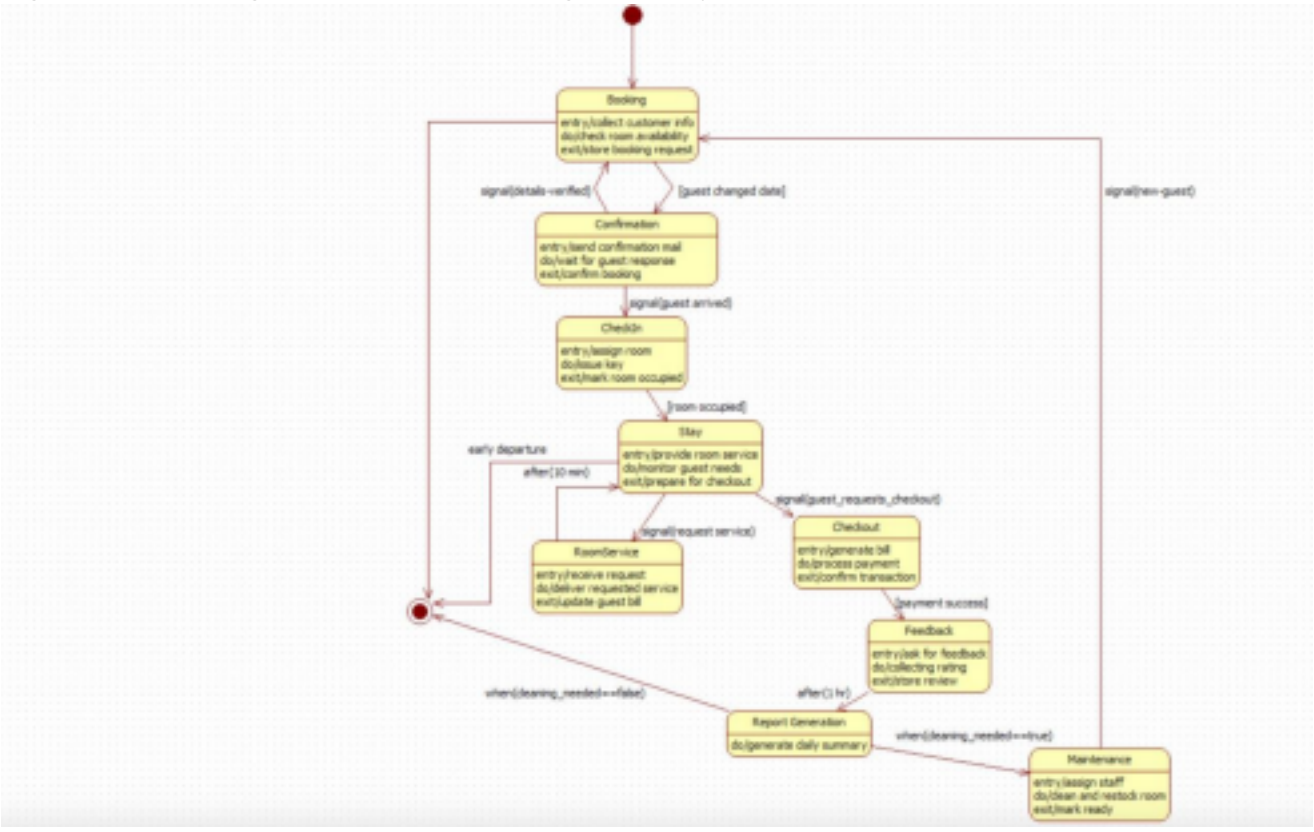


Fig 1.3 Use-Case Diagram for Hotel Management System

Fig 1.4 Sequence Diagram for Hotel Management System

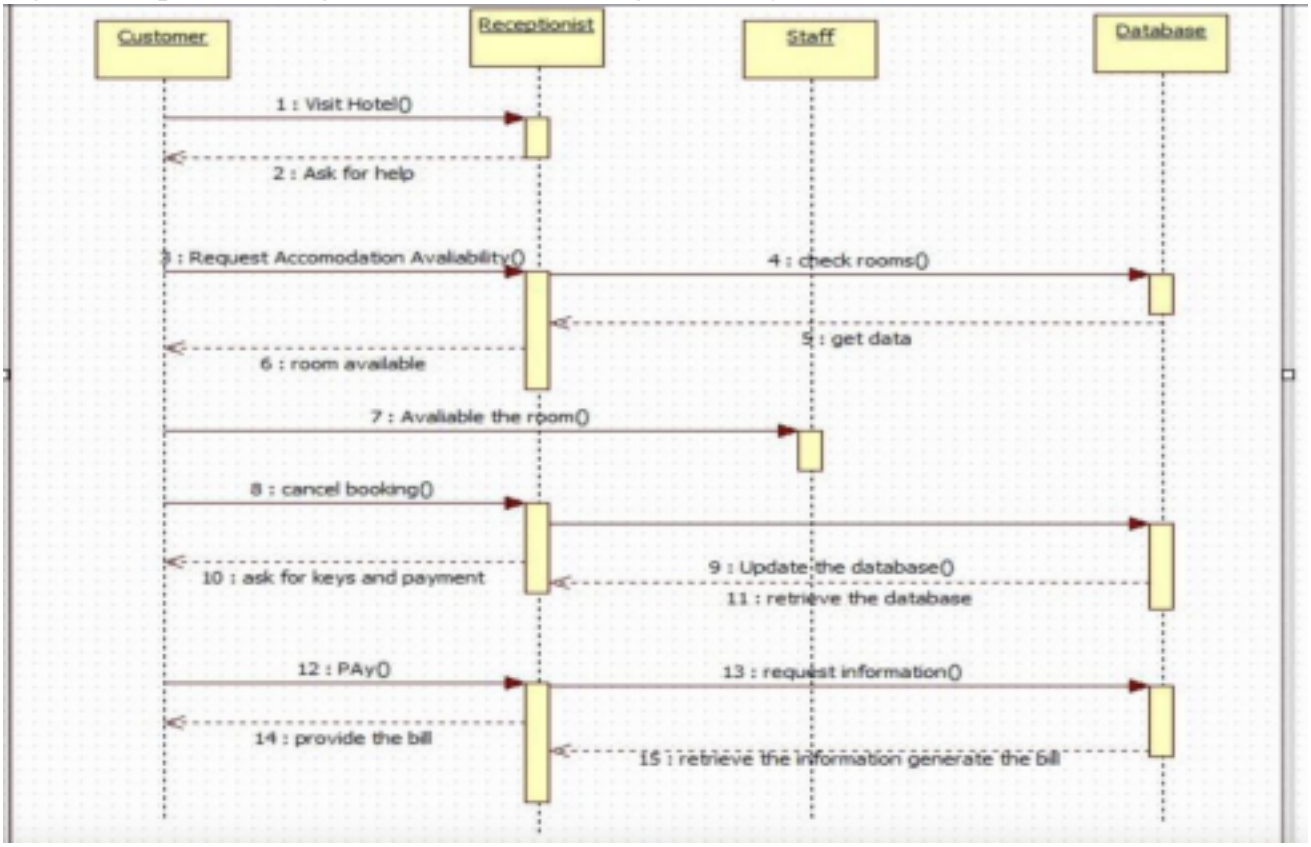


Fig 1.5 Activity Diagram for Hotel Management System

2. Credit Card Processing

Problem Statement

The Credit Card Processing System should allow cardholders to perform transactions such as purchases, cash withdrawals, balance inquiries, and bill payments. When a transaction is initiated, the system must verify card details, validate the cardholder's identity, check available credit limits, and ensure that the transaction is not fraudulent. The system must then approve or decline the transaction based on predefined rules and update the cardholder's account accordingly. Additionally, the system should generate periodic statements, track outstanding balances, calculate interest, and maintain accurate records of all transactions.

SRS-Software Requirements Specification

1. Introduction

The Credit Card Processing System (CCPS) is designed to automate and streamline the operations involved in processing credit card transactions. With the rapid increase in the use of credit cards for online and offline purchases, financial institutions must rely on a secure, accurate, and efficient system to manage authorizations, verifications, settlements, billing, and account maintenance. The current manual or partially automated methods often lead to delays, calculation errors, security vulnerabilities, and difficulties in maintaining large volumes of customer data and transaction records. This SRS document describes the functional and non-functional requirements of the Credit Card Processing System, ensuring a shared understanding among stakeholders, developers, and end users. The document outlines the purpose, scope, system behavior, constraints, and performance criteria required to implement a reliable and secure system.

1.1 Purpose

The primary purpose of the Credit Card Processing System is to provide a secure platform for processing credit card transactions, verifying cardholder information, authorizing payments, tracking purchases, generating bills, and maintaining customer accounts. The system aims to remove the inefficiencies of manual processing by offering automated validation, real-time authorization, fraud detection, and accurate financial record-keeping. This SRS document serves as a guiding framework for developers, testers, and stakeholders in understanding system objectives and requirements.

1.2 Scope

The Credit Card Processing System covers a broad range of functionalities that support the complete lifecycle of credit card usage. The system will validate card details, authenticate users, approve or decline transactions based on credit limits and account status, and maintain detailed transaction logs. It will allow customers to check balances, review statements, make payments, and report lost or stolen cards. Administrators will be able to register new customers, issue new cards, update credit limits, block cards, review suspicious activities, and generate financial and audit reports. The system will support merchant requests for transaction authorization and will update account balances in real time upon successful transactions. The system aims to ensure secure processing, high performance, and error-free functionality suitable for financial institutions.

1.3 Definitions, Acronyms, and Abbreviations

CCPS refers to the Credit Card Processing System. CVV refers to the Card Verification Value printed on credit cards. OTP refers to One-Time Password used for authentication. Cardholder refers

to the customer owning the credit card. Merchant refers to the business requesting payment authorization. Admin refers to users with elevated privileges to manage accounts and system activities.

1.4 System Overview

The Credit Card Processing System consists of several modules including customer management, card management, transaction processing, fraud detection, billing, payment handling, and reporting. These modules interact with a central database storing customer details, card information, transaction history, and system logs. When a transaction is initiated, the system validates the card details, checks the available credit, identifies any suspicious activity, and authorizes or declines the transaction. The system updates the account balance, logs transaction details, and reflects the changes in the billing cycle. The system supports both online and offline merchant transactions and ensures secure handling of sensitive data through encryption and authentication mechanisms.

2. Overall Description

2.1 Product Perspective

The Credit Card Processing System functions as a standalone software application used by financial institutions to manage credit card services. It integrates with merchant systems, customer service platforms, fraud detection tools, and banking networks. The system communicates with its internal database to retrieve and update cardholder records and processes authorization requests in real time. It ensures seamless communication between cardholders, merchants, and administrators through a secure, structured workflow that guarantees data consistency and operational accuracy.

2.2 Product Functions

The system will validate card numbers, CVV, and expiry dates, authenticate cardholders, authorize transactions, update credit limits, and maintain accurate transaction histories. It will generate monthly statements with itemized details of purchases, interest charges, late fees, and outstanding balances. Cardholders will be able to make payments through multiple channels such as online banking, mobile apps, or bank deposits. The system supports blocking of cards in case of theft or fraud and will trigger alerts for unusual or suspicious activity. Administrators will manage customer accounts, issue new cards, adjust credit limits, check fraud alerts, and generate summary reports for accounting or auditing purposes. The system ensures seamless communication between modules to provide a smooth and secure processing workflow.

2.3 User Characteristics

Users of the system include cardholders, merchants, customer service representatives, fraud analysts, system administrators, and financial auditors. Cardholders are expected to have basic digital literacy to access account information online. Merchants interact with the system indirectly through their terminals or payment gateways. Customer service representatives require moderate technical skills to manage customer accounts and address disputes. Administrators and fraud analysts must have a deeper understanding of financial operations and system controls due to their elevated responsibilities and access privileges.

2.4 Constraints

The system must ensure high levels of security because it handles sensitive personal and financial information. It must comply with industry standards such as PCI-DSS for data protection. The system

depends on stable network connectivity to process merchant transactions and must be able to handle high transaction volumes without failures. Strict authentication mechanisms, encryption protocols, and fraud detection rules must be applied. The system must also meet regulatory and legal requirements for data storage, privacy, auditing, and financial reporting.

2.5 Assumptions and Dependencies

It is assumed that merchants will provide accurate transaction data and that cardholders will protect their login credentials and report suspicious activity promptly. The system assumes that external banking networks and payment gateways used for transaction settlement are reliable and available. The system depends on a secure and scalable database to store all transaction records and user data. It also assumes that administrators and customer service staff are properly trained to use the system effectively.

3. Specific Requirements

3.1 Functional Requirements

The system must validate card information and authenticate users using PINs, passwords, or OTPs. It must authorize or decline transactions based on credit limits, card status, and fraud indicators. The system must update account balances after every transaction, store detailed transaction logs, and calculate monthly billing totals including interest and fees. It must allow customers to check their balance, view their statements, make payments, and dispute charges. The system must enable administrators to manage customer data, issue new cards, block or unblock cards, modify credit limits, and monitor fraudulent activity. It must also generate reports for auditing and financial analysis. All activities must be logged for future reference.

3.2 Non-Functional Requirements

The system must be highly secure, implementing strong encryption, authentication, and fraud detection methods. It must ensure high reliability and availability because financial transactions occur continuously. Performance must be efficient, with transaction authorization taking only a few seconds. The system should have a user-friendly interface and provide clear navigation for accessing account information. Scalability must be considered to support increasing numbers of users and transactions. Accuracy is essential, especially in interest calculations, balance updates, and financial reporting. Backup and recovery procedures must be in place to prevent data loss due to failures.

3.3 Performance Requirements

The system must handle multiple simultaneous transactions without delay. Authorization requests should be processed in real time, ensuring that merchants receive timely responses. Database operations such as transaction queries, account updates, and statement generations must be optimized for fast performance. The system should remain stable even under peak usage conditions, such as billing cycles or holiday shopping periods.

4. Conclusion

The Credit Card Processing System is designed to offer a secure, accurate, and efficient solution for handling credit card transactions and managing customer accounts. By automating verification, authorization, billing, and fraud detection, the system enhances performance and reduces the risks

associated with manual processing. This SRS document outlines clear functional and non-functional requirements necessary for developing a robust and scalable system that meets the needs of financial institutions, merchants, and cardholders.

Fig 2.1 Class Diagram for Credit Card Processing

Fig 2.2 State Diagram for Credit Card Processing

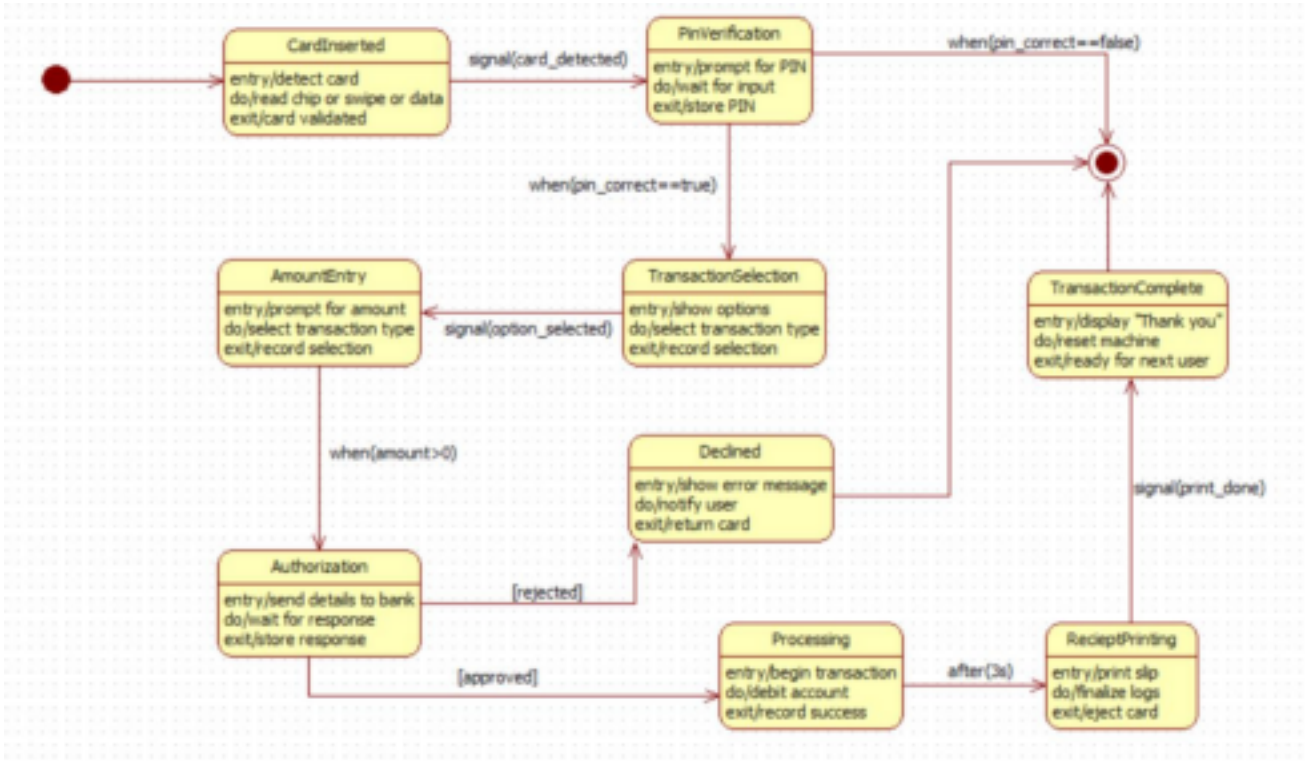


Fig 2.3 Use-Case Diagram for Credit Card Processing

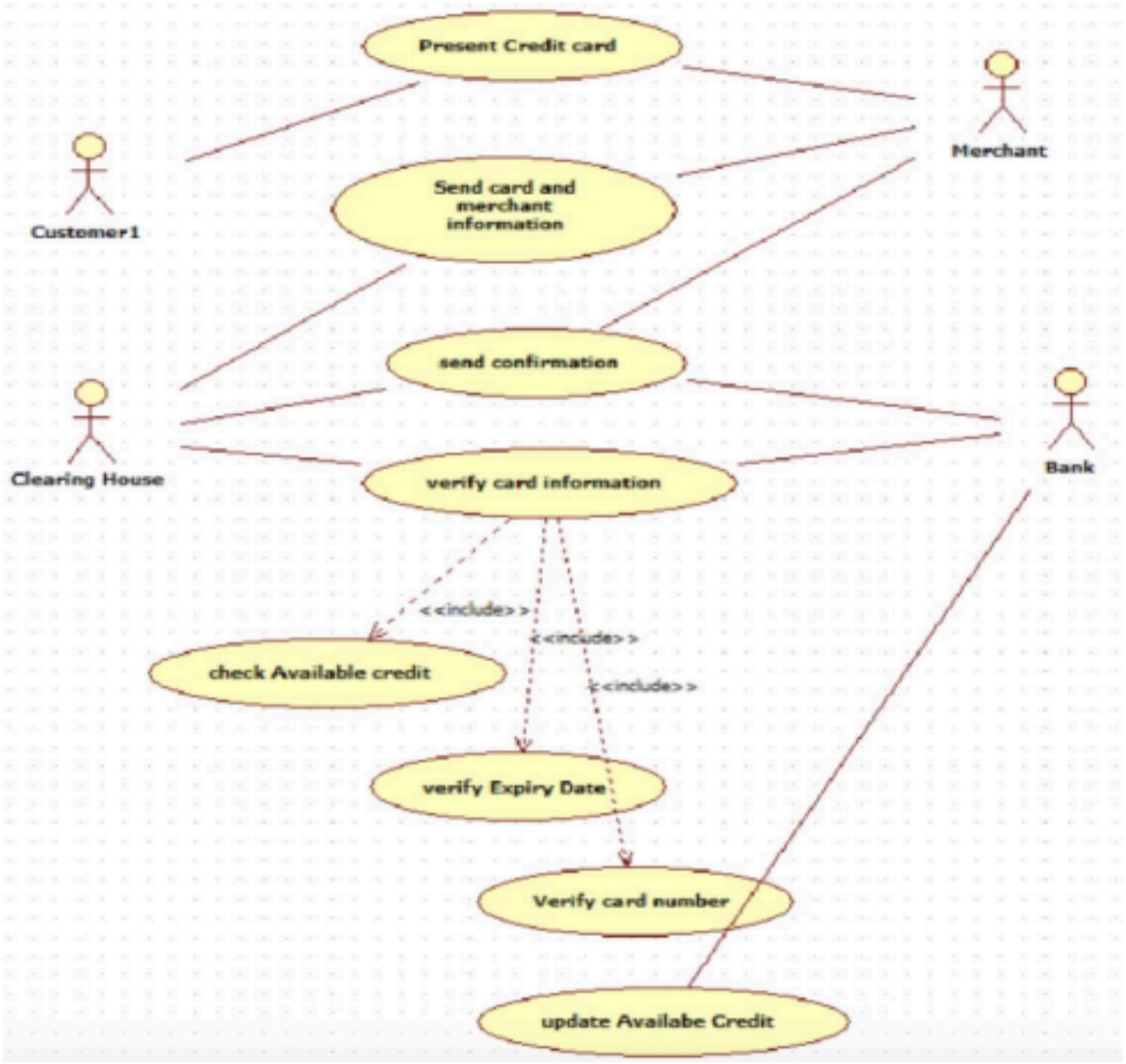


Fig 2.4 Sequence Diagram for Credit Card Processing

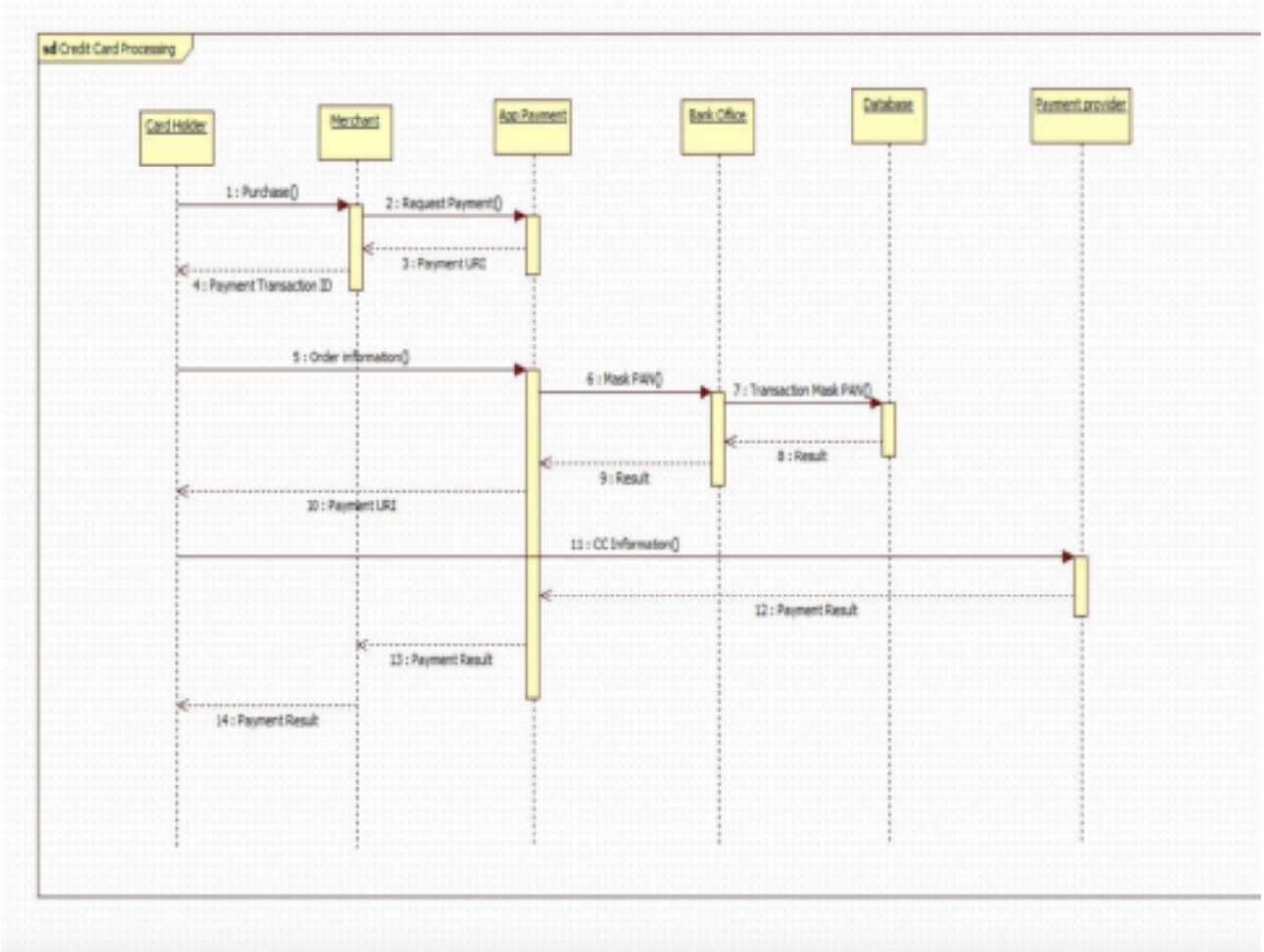
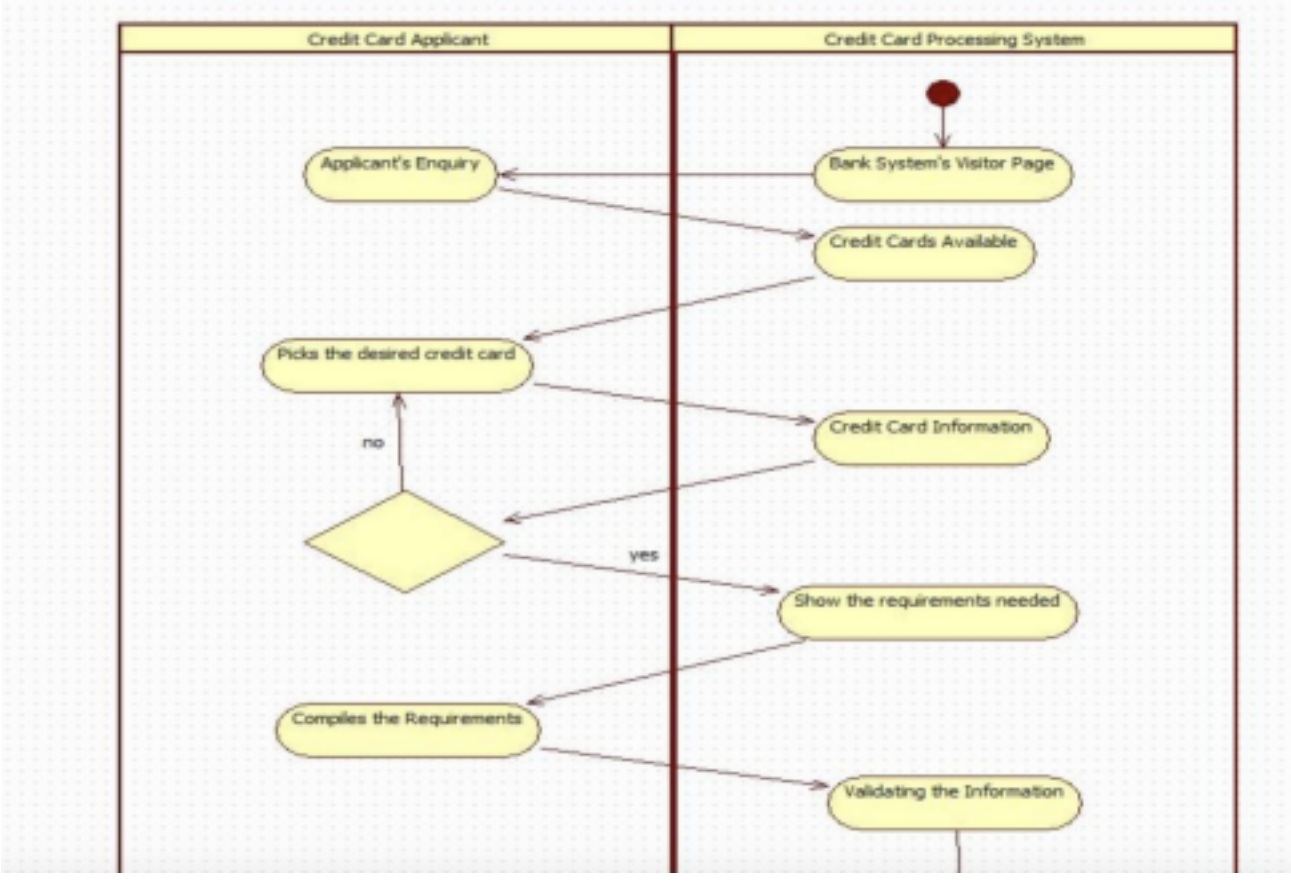
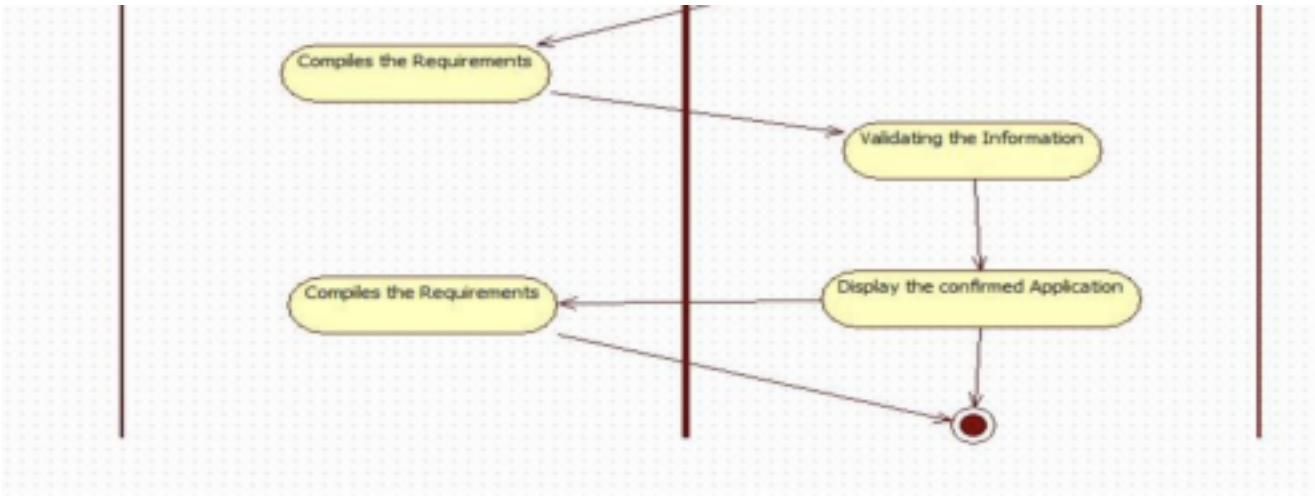


Fig 2.5 Activity Diagram for Credit Card Processing





3. Library Management System

Problem Statement

Libraries today manage large volumes of books, journals, digital media, and member records. Many libraries still rely on manual methods such as record books, paper-based membership files, and handwritten issue–return logs. These manual processes are time-consuming, prone to human errors, difficult to update, and inefficient when handling large numbers of users and resources. To overcome these limitations and modernize library operations, an automated Library Management System is required.

The Library Management System should enable librarians to efficiently manage the acquisition, cataloging, tracking, and distribution of books and other resources. The system must allow members to search for books, check availability, borrow and return items, place reservations, and receive notifications for due dates or overdue fines. Librarians should be able to register new members, maintain member details, add or remove books, update catalog information, manage book categories, handle book issues, enforce return deadlines, calculate fines for late returns, and generate various reports related to inventory and member activity.

SRS-Software Requirements Specification

1. Introduction

The Library Management System (LMS) is intended to modernize and automate the operations of a library that manages a large collection of books, journals, magazines, and digital resources. Traditional manual methods of maintaining member information, book catalogs, and borrowing records often lead to inefficiencies, errors, and difficulties in tracking library activities. This system aims to provide a computerized platform that enables efficient handling of book circulation, member registration, catalog maintenance, reservation management, fine calculation, and report generation. The SRS document describes the purpose, scope, features, functional requirements, and non-functional expectations of the system. It serves as a guide for developers, stakeholders, and users, ensuring clarity regarding system behavior and implementation goals.

1.1 Purpose

The primary purpose of the Library Management System is to provide a simplified, efficient, and accurate way to manage library resources and member interactions. The system will automate tasks such as book search, issue, return, reservation, catalog updating, fine calculation, and member account tracking. This eliminates manual bookkeeping, reduces human errors, and improves the library's operational productivity. The SRS outlines all the requirements that developers must follow to design, develop, and deploy the system effectively.

1.2 Scope

The Library Management System covers all major library operations including management of books, members, issue–return transactions, and fines. The system will allow members to search for books by title, author, subject, or category and check availability in real time. Librarians will be able to register new members, update profiles, add new books to the system, categorize them, track borrowed books, manage reservations, and maintain accurate records of returned or overdue books. The system will also automatically calculate fines for late returns and maintain a comprehensive transaction history for all members. Administrators will have the ability to manage users, control access levels, update system settings, and generate reports on library usage, inventory status, and member activity. By automating these tasks, the system improves reliability, accuracy, and service efficiency.

1.3 Definitions, Acronyms, and Abbreviations

LMS stands for Library Management System. ISBN refers to the International Standard Book Number. User refers to a member or librarian interacting with the system. Admin refers to a privileged user responsible for overall management and configuration. Issue means borrowing a book. Return means giving back a borrowed book. Reservation refers to placing a request for a book that is currently unavailable.

1.4 System Overview

The Library Management System consists of interconnected modules such as catalog management, member management, circulation management, reservation tracking, fine calculation, and reporting. These modules work with a centralized database that stores book records, member details, and transaction histories. The system ensures that book availability is updated instantly after every issue, return, or reservation. It supports multiple users simultaneously and provides a user-friendly interface to access book catalogs and records. Security mechanisms such as login authentication and controlled access ensure that only authorized users perform sensitive operations. The system aims to provide smooth, error-free operations and enhance the overall management of library resources.

2. Overall Description

2.1 Product Perspective

The Library Management System functions as a standalone application used inside a library or institution. It integrates its modules with a shared database to ensure real-time updates and coordinated operations. The system replaces manual filing systems by digitally storing all information related to books, members, and transactions. It communicates with users through a graphical interface and supports operations such as book search, borrowing, returning, and fine processing. The system is designed to be flexible and scalable so that additional modules or features

can be added in the future.

2.2 Product Functions

The system provides various functions including adding new books to the catalog, editing book details, categorizing resources, registering new members, updating member profiles, issuing books, processing returns, updating book status, tracking overdue items, calculating fines, managing reservations, and generating library usage reports. Users can search the database for books based on multiple criteria. Librarians can verify member eligibility before issuing books and enforce borrowing rules such as limits and due dates. The system maintains transaction logs to ensure transparency and accountability.

2.3 User Characteristics

Users of this system include librarians, members, and administrators. Librarians typically have moderate computer skills and will use the system extensively for day-to-day operations. Members may have basic knowledge of using search functionality to find books and check their borrowing status. Administrators will have higher-level access and require a deeper understanding of library policies and system controls to manage user accounts, update system settings, and handle maintenance tasks.

2.4 Constraints

The system must ensure accurate and timely updates to the database because multiple users may access it simultaneously. It must operate on reliable hardware and require secure login authentication to prevent unauthorized access. The system must follow organizational policies regarding borrowing limits, return deadlines, and fine amounts. The database must remain consistent even under high usage, and the system must operate properly during library working hours. Backup and recovery mechanisms must be available to prevent data loss.

2.5 Assumptions and Dependencies

It is assumed that librarians and administrators will receive proper training before using the system. The system assumes that users input accurate data while registering members or adding books. It also depends on the reliability of the network and database server for multi-user access. The system may rely on barcode scanners or RFID devices if integrated for automated book identification.

3. Specific Requirements

3.1 Functional Requirements

The system must allow members to search for books using keywords such as title, author, subject, or category, and display results along with availability status. It must allow librarians to register new members, assign unique member IDs, and update or delete member accounts. The system must enable librarians to add, edit, or remove books from the catalog and record details such as ISBN, title, author, edition, and category. It must support book issue operations where the system verifies member eligibility, updates book status to “issued,” generates due dates, and records transaction details. The system must allow book return operations, update availability status, calculate overdue fines if applicable, and log the transaction. It must also allow members to reserve books that are currently unavailable and notify them when the book becomes available. Administrators must be able to generate statistical reports such as total books issued, most borrowed books, member activity

logs, and inventory status. The system must ensure secure login for all users and maintain detailed logs of all operations.

3.2 Non-Functional Requirements

The system must be reliable and capable of handling continuous usage throughout library hours. It must provide fast response times for book searches and transaction processing. Usability is important; therefore, the interface must be simple, intuitive, and accessible to users with basic computer knowledge. The system must implement security features such as authentication, role-based access control, and data encryption for protecting sensitive information. It must be scalable to support an expanding collection of resources and increasing numbers of users. The system must also support regular data backup and quick recovery in case of system failures.

3.3 Performance Requirements

The system must perform search queries quickly and return results within a few seconds. It must support multiple concurrent users without performance degradation. Book issue and return operations must reflect instantly in the database to avoid inconsistencies. The system should process fines and generate reports efficiently, even when database records grow in size. It must remain stable under peak usage conditions such as beginning or end of academic semesters.

4. Conclusion

The Library Management System is designed to automate and improve the management of library resources by reducing manual efforts and increasing accuracy and operational efficiency. This SRS document outlines all the functional, non-functional, and performance requirements necessary to build a robust system that meets the expectations of librarians, administrators, and members. The system ensures reliability, security, and ease of use, ultimately modernizing library operations and enhancing user experience.

Fig 3.1 Class Diagram for Library Management System

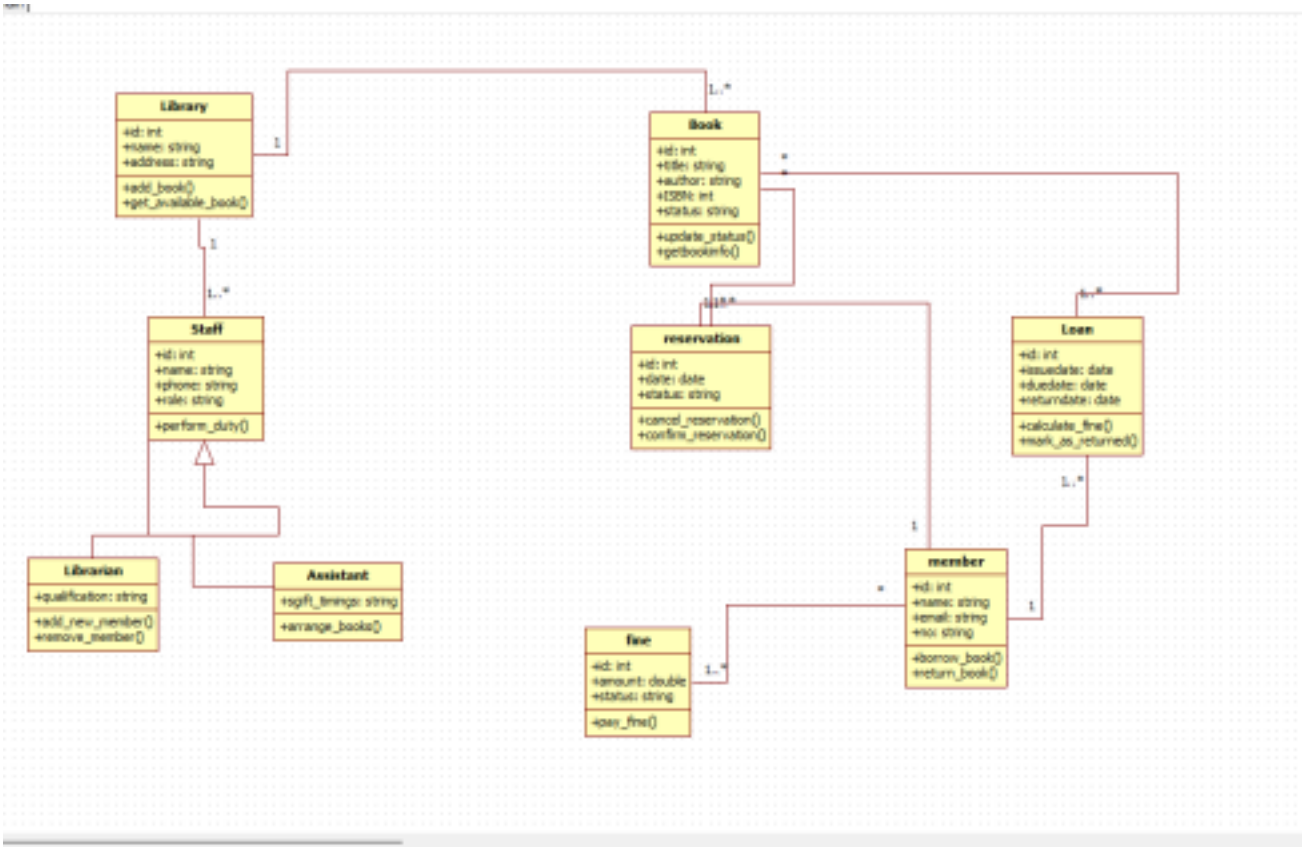


Fig 3.2 State Diagram for Library Management System

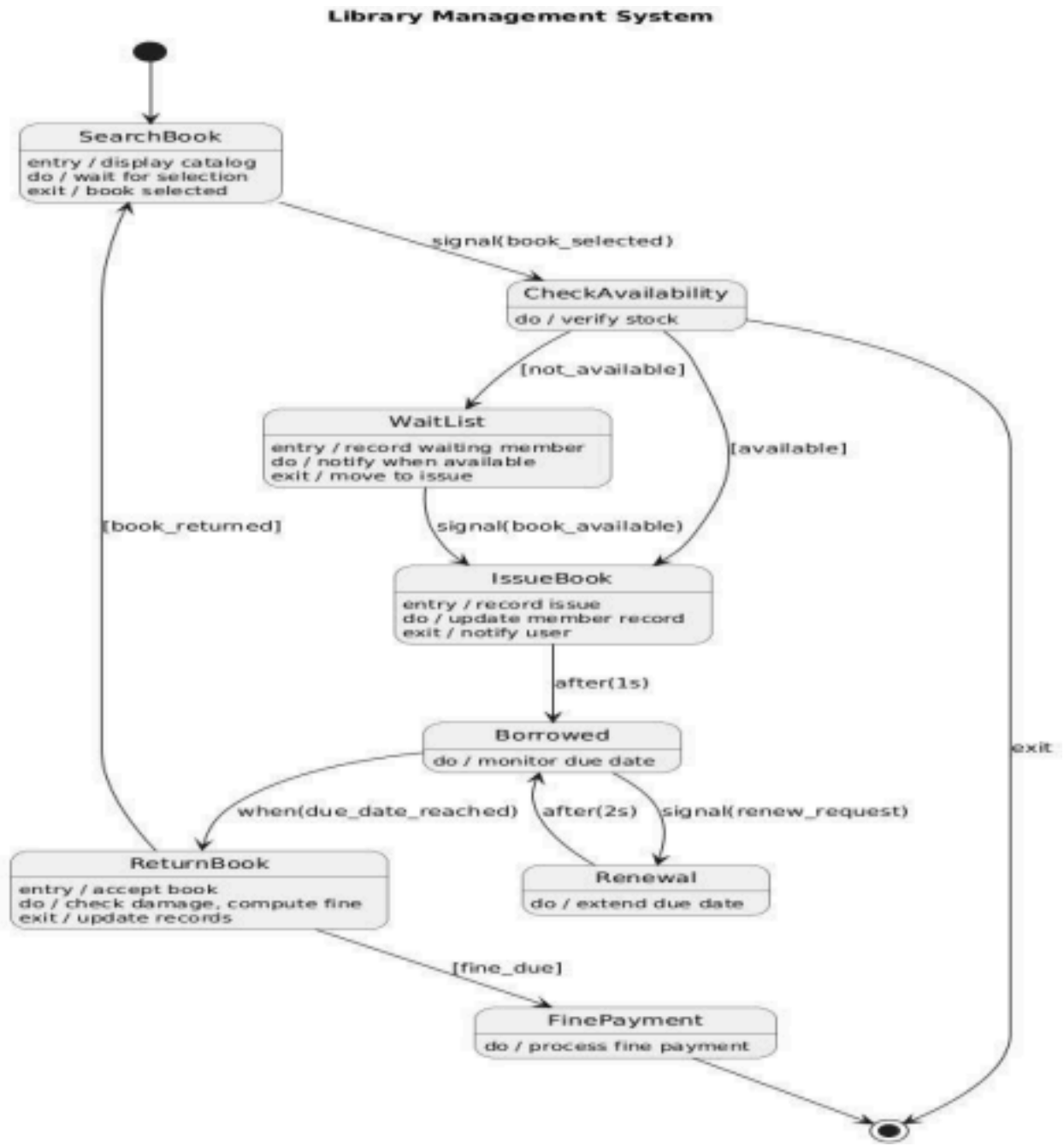


Fig 3.3 Use-Case Diagram for Library Management System

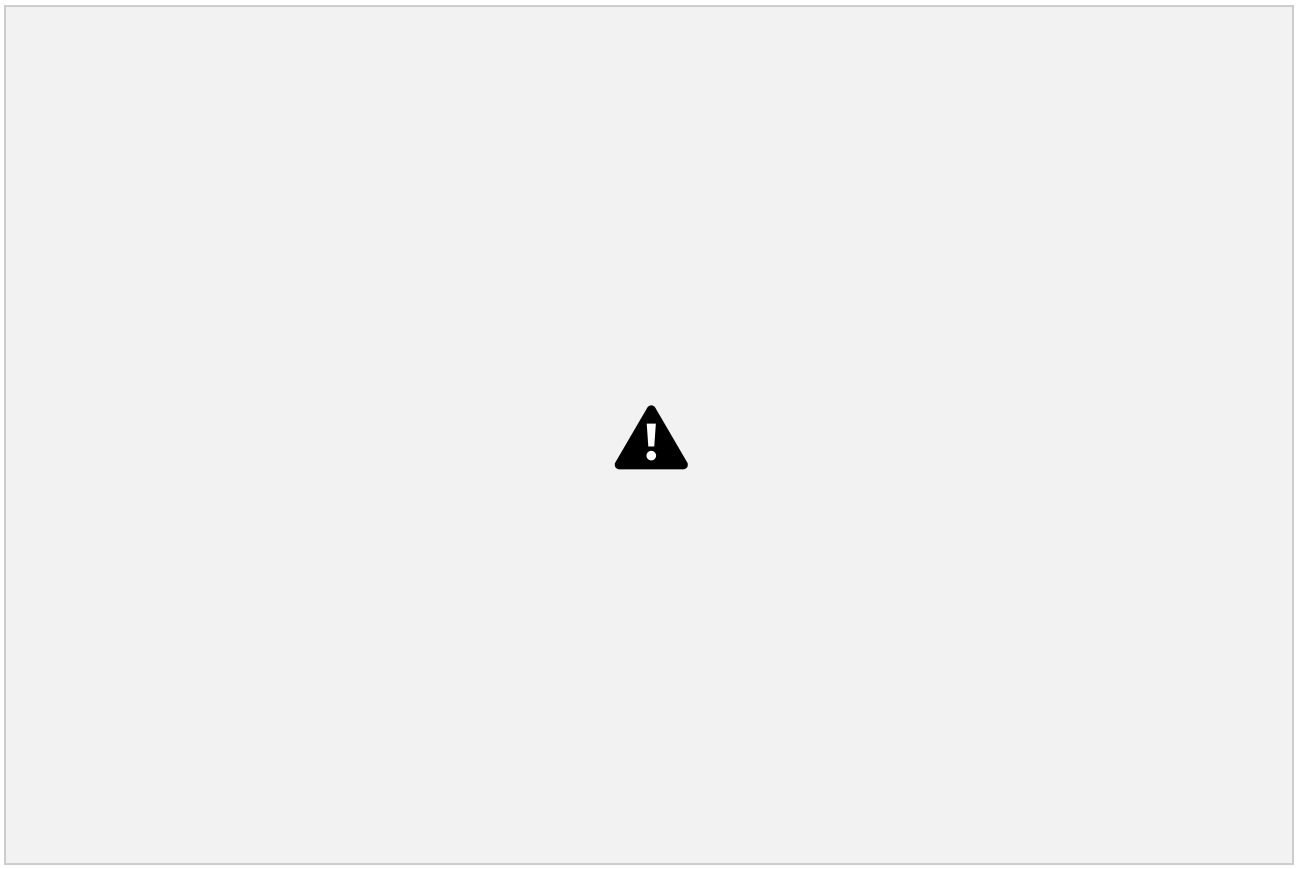


Fig 3.4 Sequence Diagram for Library Management System

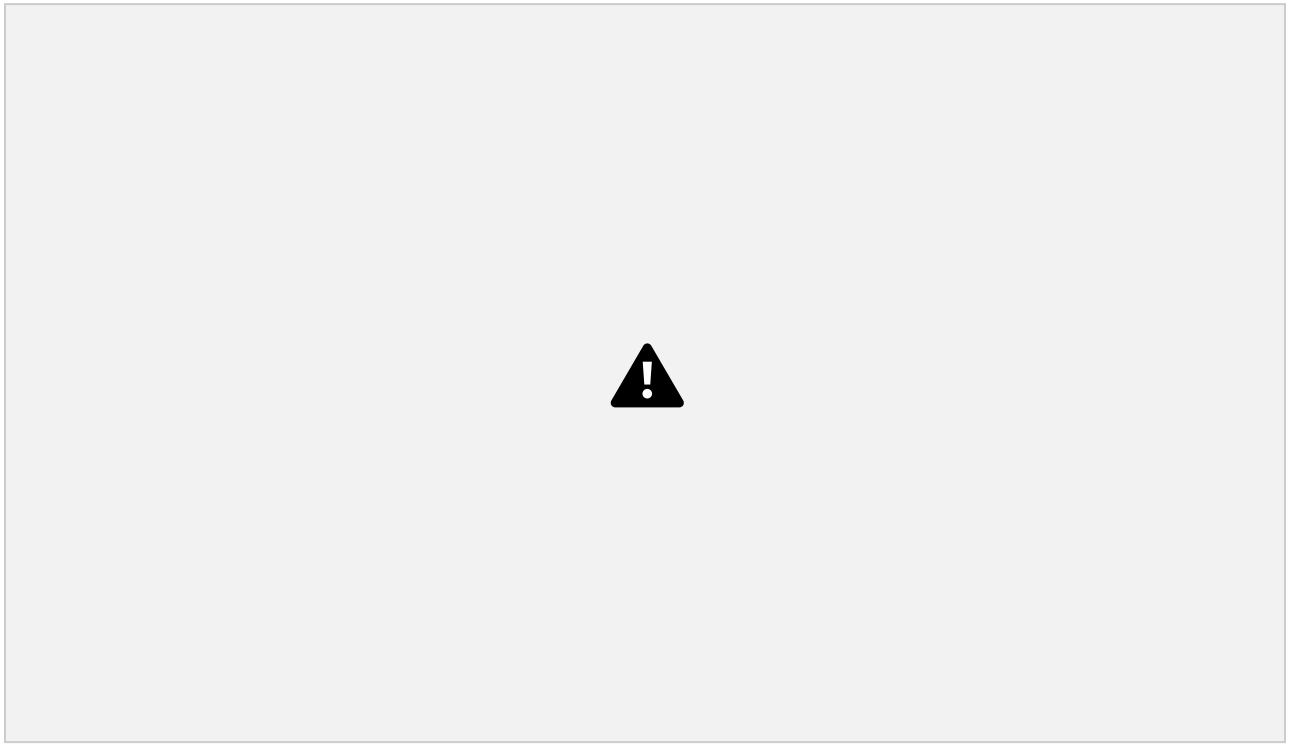
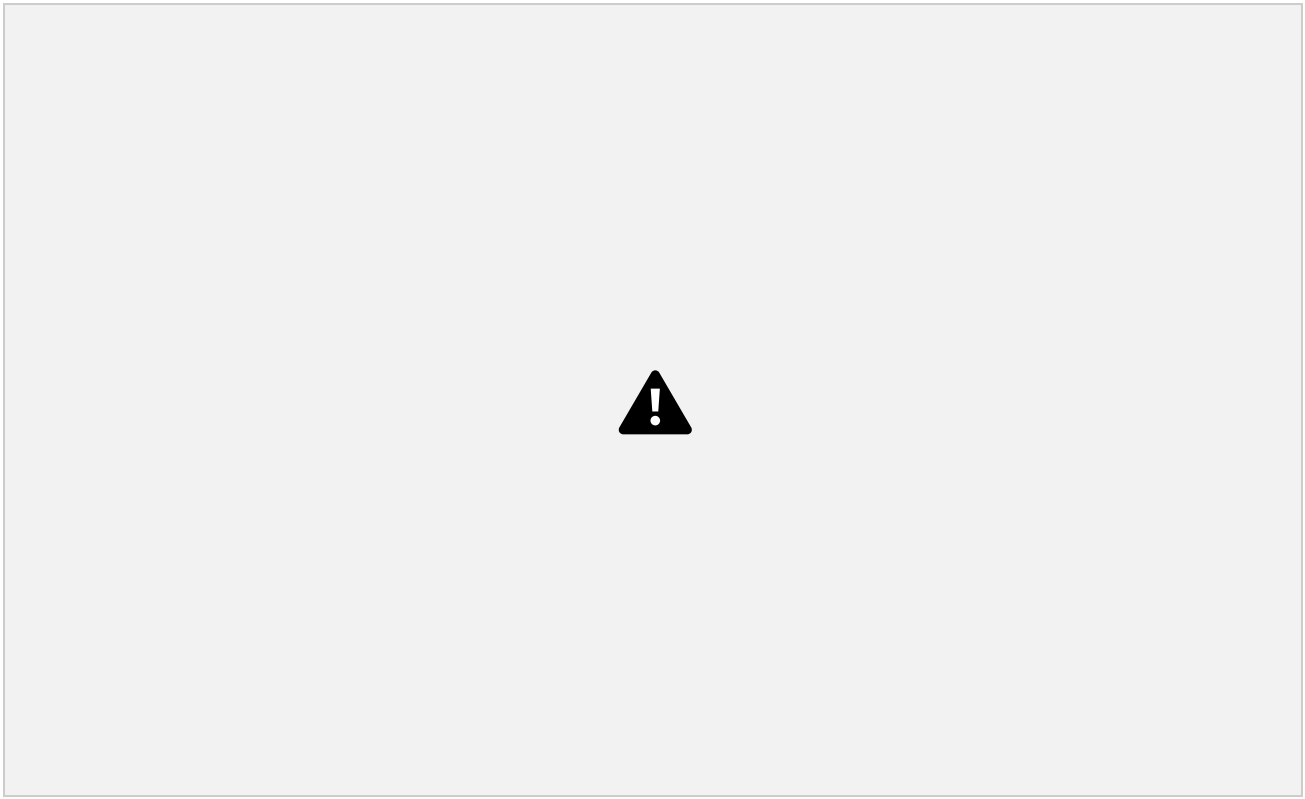
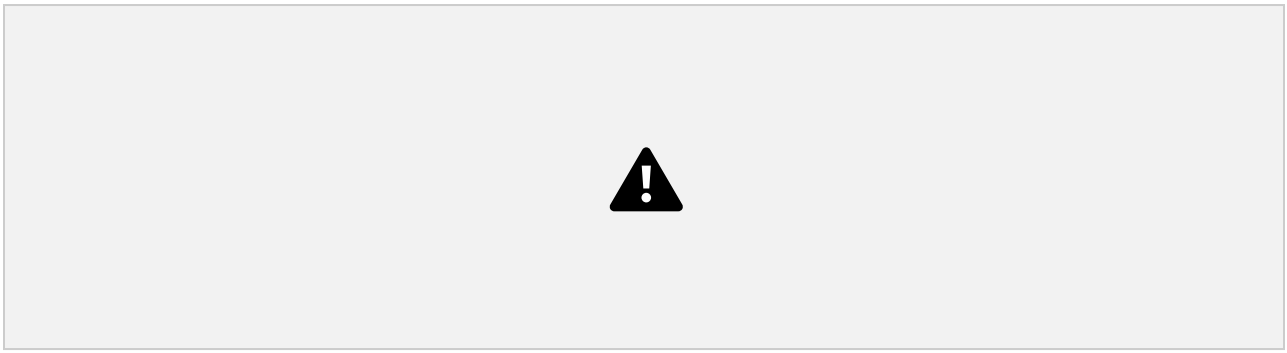


Fig 3.5 Activity Diagram for Library Management System





4. Stock Maintenance System

Problem Statement

The Stock Maintenance System should enable businesses to track stock quantities, monitor product movements, and maintain updated records of items entering or leaving the inventory. The system must allow users to add new stock items, categorize products, update quantities, record purchases, process stock issues, and generate alerts when stock levels fall below a specified threshold. It should also support search and inquiry functions so that users can quickly check availability, item details, supplier information, and stock status in real time. The system must maintain historical records of stock transactions, including purchases, sales, returns, damaged goods, and adjustments, ensuring transparency and traceability.

SRS-Software Requirements Specification

1. Introduction

The Stock Maintenance System (SMS) is designed to automate and simplify the management of inventory within an organization. Many businesses still rely on manual or spreadsheet-based

processes to record stock levels, track item movements, and monitor shortages. These traditional methods are prone to inaccuracies, delays, duplication of entries, and difficulties in maintaining real-time visibility of stock. To address these limitations, the Stock Maintenance System provides a computerized solution that ensures accurate tracking of inventory, efficient stock updates, and timely notifications regarding stock availability. This SRS document outlines the purpose, scope, functional and non-functional requirements, system design overview, and performance expectations of the proposed system.

1.1 Purpose

The purpose of the Stock Maintenance System is to provide an automated platform that helps organizations record, manage, and control their inventory effectively. The system will maintain accurate stock levels, monitor item additions and withdrawals, track supplier information, and allow users to view and update stock records in real time. By eliminating manual tracking, the system reduces errors, enhances productivity, and provides management with reliable data for decision-making. This SRS document serves as a guideline for developers, stakeholders, and users, ensuring a common understanding of the system's requirements and functionality.

1.2 Scope

The scope of the Stock Maintenance System includes maintaining a structured inventory database containing product details, stock levels, pricing, supplier information, and transaction history. The system allows users to add new products, update quantities, record purchases, track issued items, and categorize stock based on type or department. It alerts users when stock levels fall below a predefined minimum and helps avoid stockouts or overstocking. The system enables authorized personnel to search for items quickly, verify availability, generate inventory reports, and track stock adjustments caused by damage, returns, or discrepancies. Administrators can manage user accounts, assign roles, configure stock categories, and oversee overall system operations. The system aims to enhance efficiency, accuracy, and transparency in inventory control.

1.3 Definitions, Acronyms, and Abbreviations

SMS refers to Stock Maintenance System. SKU stands for Stock Keeping Unit. User refers to staff members who interact with the system. Admin refers to individuals with elevated access to manage users and system settings. Stock In refers to items added to inventory. Stock Out refers to items issued or removed from inventory.

1.4 System Overview

The Stock Maintenance System is composed of several integrated modules, including stock entry, stock monitoring, stock movement tracking, user management, supplier management, and reporting. A centralized database stores all information related to products, suppliers, stock transactions, and threshold settings. Users interact with the system through a graphical interface that supports adding, updating, and managing stock details. The system ensures that every stock transaction—whether purchase, issue, return, or adjustment—is recorded accurately along with timestamps and responsible users. Notifications, alerts, and reports assist in maintaining optimal stock levels and improving organizational workflow.

2. Overall Description

2.1 Product Perspective

The Stock Maintenance System operates as a standalone inventory management application used by organizations such as warehouses, retail stores, manufacturing units, and offices. It integrates its modules with a shared database to ensure real-time updates and consistent information across all users. The system replaces traditional manual processes and enhances accuracy by providing automated validation, instant updates, and detailed transaction histories. The architecture is designed to be flexible and scalable, allowing integration with other business systems in the future.

2.2 Product Functions

The system provides essential functions such as adding new stock items, updating item details, recording stock in and stock out transactions, managing supplier data, and categorizing inventory. It allows users to check product availability, view stock levels, set minimum thresholds, and receive alerts when items run low. The system maintains transaction records for purchases, issues, damaged items, lost items, and returns. It also supports generating reports such as stock summaries, daily transaction logs, and shortage reports. Administrators can manage user roles, configure system settings, and monitor overall inventory performance.

2.3 User Characteristics

Users of the system include inventory staff, storekeepers, purchasing officers, managers, and administrators. Most users require only basic computer knowledge to operate the system, as it provides a simple and intuitive interface. Administrators require a higher understanding of system functions because they manage user permissions, stock categories, and system configurations. Managers may use the system primarily for viewing reports and monitoring stock status.

2.4 Constraints

The system must ensure accurate and consistent updates to the inventory database, especially when multiple users perform transactions simultaneously. It must provide secure user authentication to prevent unauthorized access or manipulation of stock data. The system must comply with organizational policies related to stock management, procurement, and data privacy. It should operate efficiently on the available hardware, and data backup mechanisms must be implemented to avoid data loss in case of system failures.

2.5 Assumptions and Dependencies

It is assumed that users will be trained to operate the system and provide accurate data when adding or updating stock. The system assumes reliable network connectivity for multi-user environments and depends on database stability for maintaining large volumes of data. The system may also depend on external barcode scanners or labeling equipment if integrated for automated identification of stock items.

3. Specific Requirements

3.1 Functional Requirements

The system must allow users to add new products to the inventory by entering details such as SKU, name, category, quantity, supplier, and price. It must support updating product information and modifying stock levels based on purchase receipts or issued items. Each stock transaction must be recorded with the date, time, and responsible user. The system must allow users to categorize items, search for products using various filters, and view current stock levels. It must generate alerts when stock levels fall below predefined thresholds. The system should allow administrators to manage user accounts, assign roles, define minimum stock levels, and modify settings. It must support generating detailed reports showing stock balances, transactions, shortages, received goods, and item history.

3.2 Non-Functional Requirements

The system must be reliable and provide consistent performance during working hours. It must be usable with a clean, intuitive interface for operators with basic technical skills. Security is essential; therefore, the system must implement login authentication, access restrictions, and data protection mechanisms. The system must be scalable to accommodate an increasing number of products and users. It must ensure fast response times for searches, updates, and report generation. The system must also support regular backups and allow recovery of data after system failures.

3.3 Performance Requirements

The system must perform stock searches quickly and return results within seconds. It should update stock levels instantly after each transaction, ensuring real-time accuracy. The system must support numerous concurrent users without reducing performance. Report generation should be efficient, even when handling large datasets. The system must remain stable during peak operational hours and handle frequent updates without errors.

4. Conclusion

The Stock Maintenance System is designed to streamline inventory management and improve operational efficiency by automating stock tracking, updating, and reporting. This SRS document provides a comprehensive description of the system's purpose, functionalities, requirements, and performance expectations. By ensuring accuracy, security, and usability, the system will help organizations maintain optimal stock levels, reduce losses, and support better decision-making.

Fig 4.1 Class Diagram for Stock Maintenance System

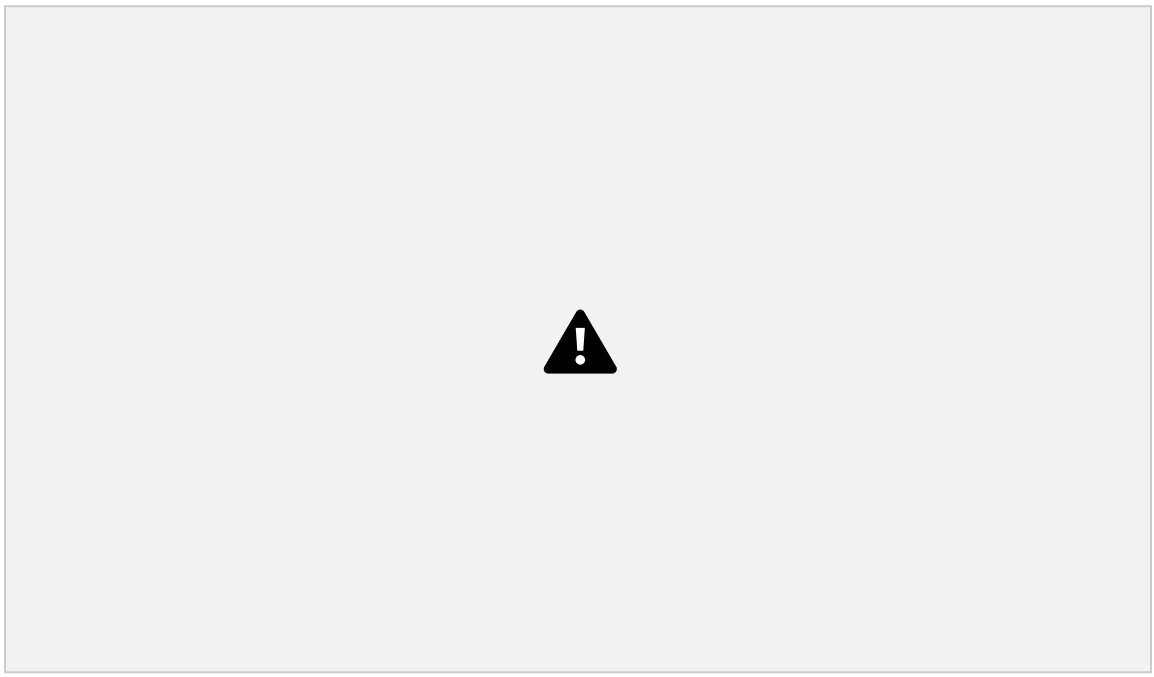


Fig 4.2 State Diagram for Stock Maintenance System

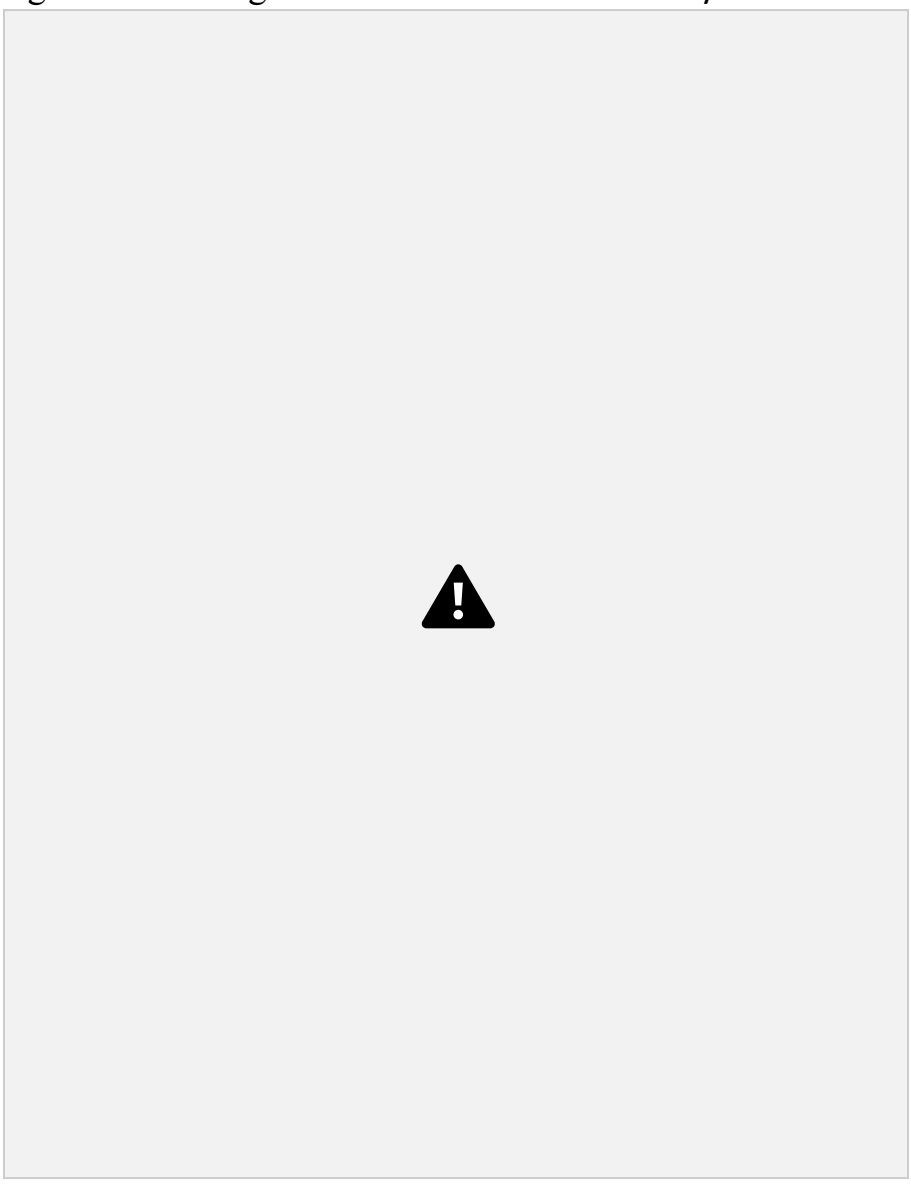


Fig 4.3 Use-Case Diagram for Stock Maintenance System

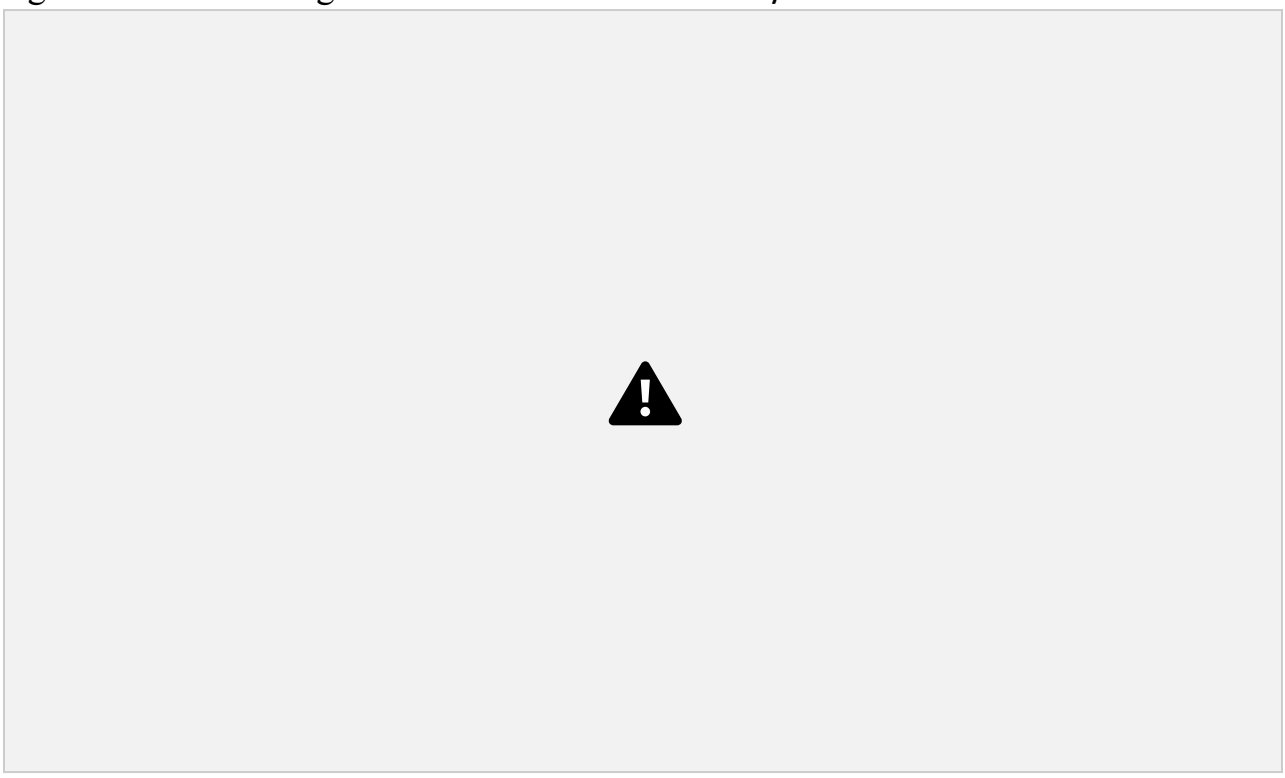


Fig 4.4 Sequence Diagram for Stock Maintenance System

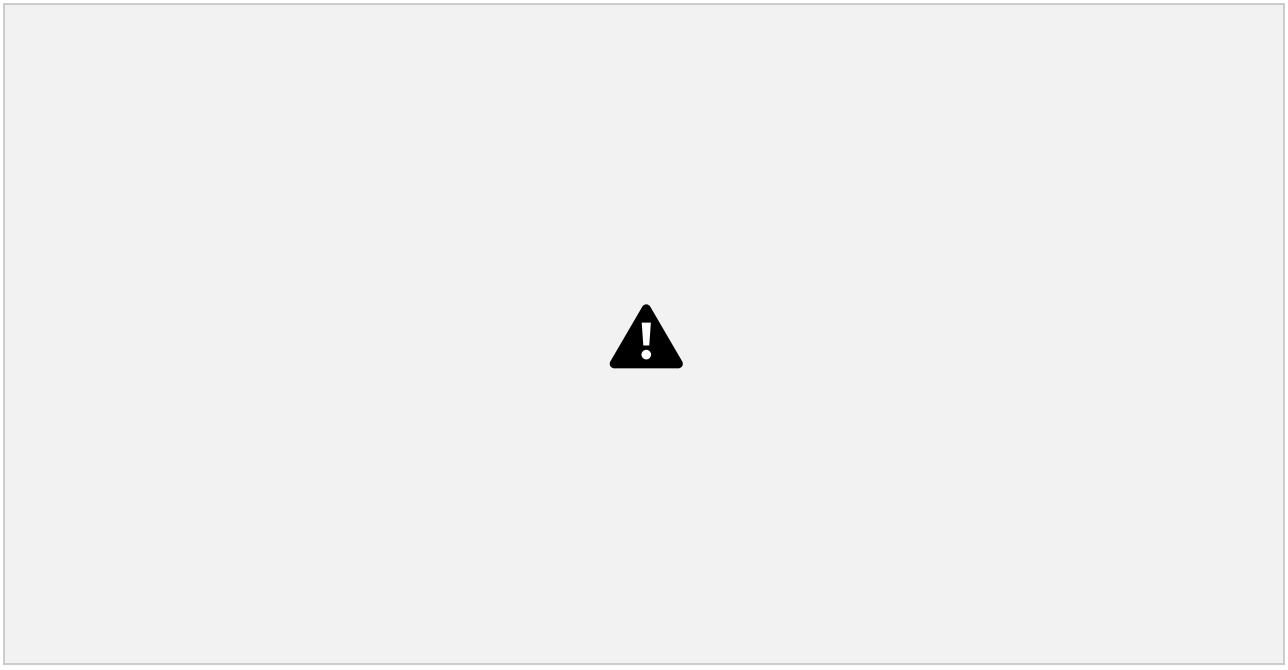
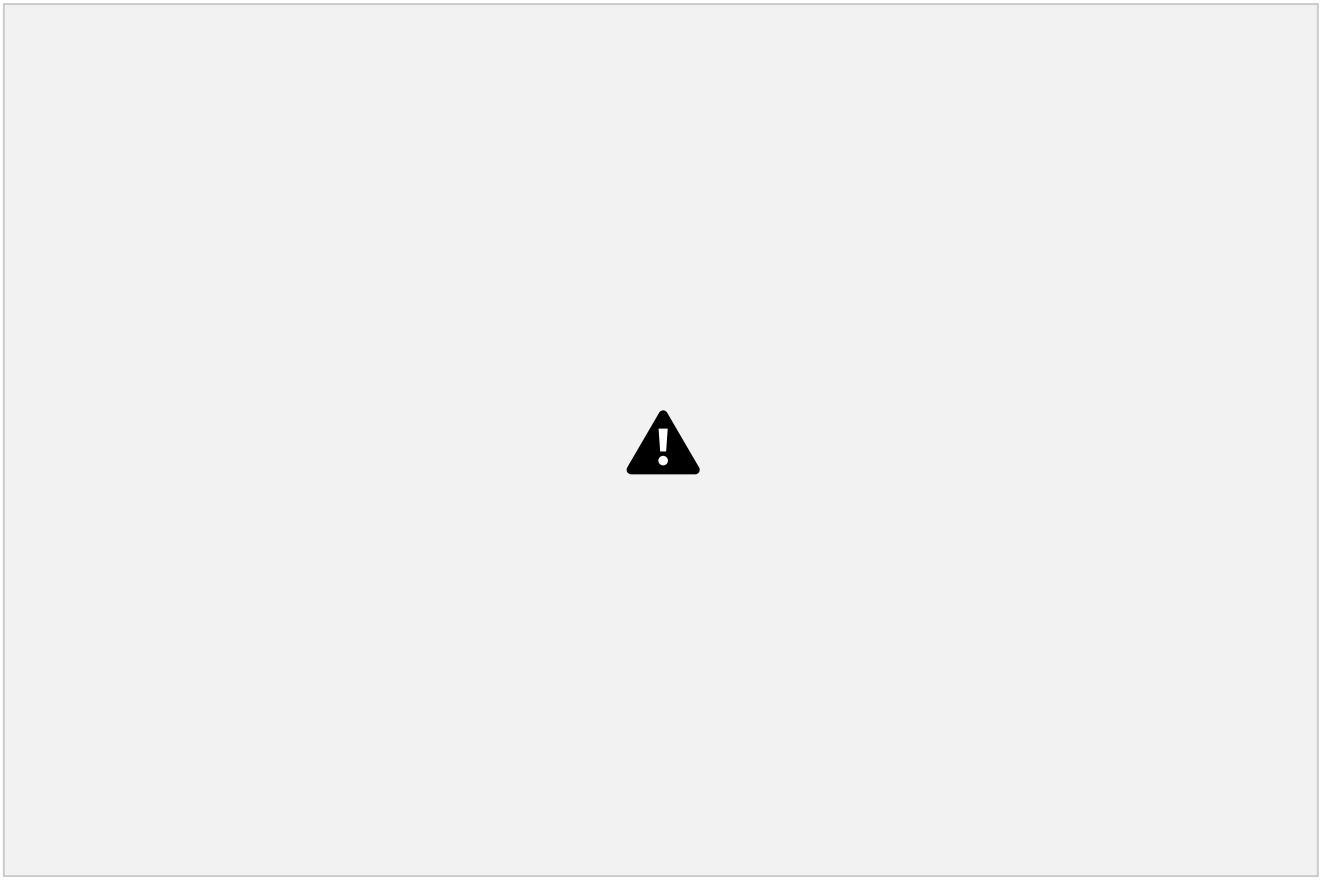


Fig 4.5 Activity Diagram for Stock Maintenance System



5. Passport Automation System
Problem Statement

The process of passport application and issuance in many regions involves several manual steps such as filling out physical forms, submitting documents in person, waiting in long queues, and undergoing slow verification processes. These traditional methods often lead to delays, inconsistencies in data, missing documents, difficulty in tracking application status, and a lack of transparency for applicants. To address these challenges and improve the efficiency of passport-related services, a computerized Passport Automation System is required.

The Passport Automation System must automate the major functions involved in applying for, processing, and issuing passports. Applicants should be able to submit their applications online, upload required documents, schedule appointments, and track the status of their application at each stage. The system must validate applicant information, verify supporting documents, and provide alerts or notifications if corrections or additional documents are needed. It must support various stages of processing, including police verification, document scanning, fee payment, background checks, and final approval.

SRS-Software Requirements Specification

1. Introduction

The Passport Automation System is designed to streamline, digitize, and enhance the process of applying for, verifying, and issuing passports. Traditional passport processing involves various manual steps such as filling paper forms, submitting documents at government offices, undergoing physical verification, and waiting for long periods due to slow processing and inefficient coordination among departments. These manual practices often result in delays, inaccurate records,

misplaced files, difficulty in tracking applications, and limited accessibility for applicants. The Passport Automation System aims to overcome these limitations by offering an online, automated, efficient, and transparent platform for managing all passport-related activities. This SRS document defines the purpose, scope, system functions, constraints, and requirements necessary for developing the Passport Automation System.

1.1 Purpose

The purpose of the Passport Automation System is to provide applicants with a user-friendly and accessible platform for submitting passport applications, uploading documents, scheduling appointments, paying fees, and tracking application status. The system also assists passport officials in validating data, verifying documents, coordinating police verification, processing applications, issuing passports, and maintaining secure records. By automating these tasks, the system reduces manual workload, increases accuracy, minimizes delays, and improves transparency. This SRS document provides detailed specifications to guide developers, testers, and stakeholders through the design and implementation of the system.

1.2 Scope

The scope of the Passport Automation System covers the complete lifecycle of passport services including new passport applications, renewals, re-issues, status tracking, verification procedures, and passport delivery. The system enables applicants to fill out online forms, upload required documents, schedule appointments for biometric authentication, and make fee payments digitally. It allows passport officials to validate applicant data, verify supporting documents, perform background checks, and communicate with police departments for address verification. The system maintains records of each application, stores applicant information securely, and generates digital or printed passports once approval is complete. Administrators can manage users, configure system settings, generate reports, monitor workloads, and ensure compliance with legal and security policies. The system aims to modernize the passport processing workflow, reduce human error, and improve service efficiency.

1.3 Definitions, Acronyms, and Abbreviations

PAS refers to the Passport Automation System. Applicant refers to the individual applying for or renewing a passport. Officer refers to a passport processing official. Admin refers to system administrators with elevated privileges. Biometrics includes fingerprints and photographs captured during verification. Police Verification refers to the background verification performed by the police department.

1.4 System Overview

The Passport Automation System consists of several modules that work together, including the applicant registration module, application submission module, document verification module, biometric and appointment module, police verification module, processing and approval module, payment module, and passport generation module. These modules interact with a centralized secure database that stores applicant details, documents, verification records, and passport data. The system ensures accurate and real-time updates to application status, facilitates communication between departments, and enables applicants to track progress at each step. Security and privacy mechanisms are implemented to protect sensitive personal information.

2. Overall Description

2.1 Product Perspective

The Passport Automation System functions as a standalone web-based application integrated with government databases, police departments, and payment gateways. It replaces manual paperwork and decentralized processing with a centralized and automated system. The system uses digital forms, secure data storage, and workflow automation to coordinate the passport process from application submission to final issuance. It is scalable to support large volumes of applications and flexible enough to incorporate future enhancements.

2.2 Product Functions

The system supports numerous functions such as applicant registration, secure login, online application submission, document upload, appointment scheduling, biometric recording, online fee payment, status tracking, and passport delivery notifications. For officials, it offers document verification, background checks, approval or rejection of applications, passport generation, and report creation. The system keeps detailed logs of all operations, stores documents securely, and generates receipts and acknowledgments automatically. Administrators can manage user access, monitor departmental activities, generate statistical reports, update system configurations, and oversee the overall functioning of the system.

2.3 User Characteristics

Users of the system include applicants, passport officers, police personnel, and system administrators. Applicants may have basic computer or smartphone knowledge and require an intuitive interface to apply for passports. Passport officers require moderate technical understanding to verify applications, update statuses, and process approvals. Police personnel use the system to access applicant details, record verification results, and forward reports. Administrators have higher technical knowledge and manage system permissions, security policies, and configurations.

2.4 Constraints

The system must comply with government rules, data privacy legislation, and security policies. It must ensure the security of sensitive personal information, including biometric data, by implementing strong authentication, encryption, and access control mechanisms. The system depends on stable network connectivity for online operations and must integrate with external services such as police verification systems and payment gateways. It must operate reliably with minimum downtime as applicants rely heavily on timely updates. Hardware such as biometric devices must be compatible with the system.

2.5 Assumptions and Dependencies

It is assumed that all users have access to a stable internet connection when using the system. The system assumes that applicants provide accurate and truthful information during submission. Passport officers and police personnel are assumed to follow proper procedures and use the system responsibly. The system depends on external services such as payment gateways and verification APIs which must remain functional for uninterrupted service.

3. Specific Requirements

3.1 Functional Requirements

The system must allow applicants to register with unique login credentials and submit online applications by entering personal details and uploading required documents. It must validate entered data, verify document formats, and store all information securely. The system should allow applicants to schedule appointments for biometric collection and pay passport fees online. It must enable applicants to track application status at every stage and receive notifications for updates, additional document requests, or approvals. Officials must be able to view applications, verify data, update statuses, request police verification, and review reports. Police officials must be able to access the verification module, conduct background checks, and submit verification results electronically. The system must generate unique passport numbers, produce the passport document once the application is approved, and update records in the central database. Administrators must be able to add or remove users, configure system parameters, generate reports, and ensure smooth operation.

3.2 Non-Functional Requirements

The system must be secure, implementing multi-factor authentication, encryption, and strict access policies to protect personal and biometric data. It must provide high availability and minimal downtime as applicants may access it at any time. The system must be easy to use, offering a clean, intuitive interface that guides applicants and officials through various processes. Performance must be optimized to handle multiple concurrent applications without delays. Scalability is essential to manage large volumes of applicants during peak seasons. The system must also maintain reliability through proper error handling, data backup, and disaster recovery mechanisms.

3.3 Performance Requirements

The system should process application submissions, payment validations, and status updates quickly, returning results within a few seconds. It must handle high traffic during peak periods without affecting performance. Database operations such as retrieving applications, updating verification statuses, and generating reports must be efficient. The system should support a large number of simultaneous users and maintain smooth performance across all modules.

4. Conclusion

The Passport Automation System is designed to bring efficiency, accuracy, and transparency to passport-related services by digitizing and automating the entire process. This SRS document outlines the system's purpose, scope, user expectations, constraints, functional and non-functional requirements, and performance criteria. By following these specifications, developers can build a secure, scalable, and user-friendly system that significantly improves the passport application and issuance process for both applicants and officials.

Fig 5.1 Class Diagram for Passport Automation System

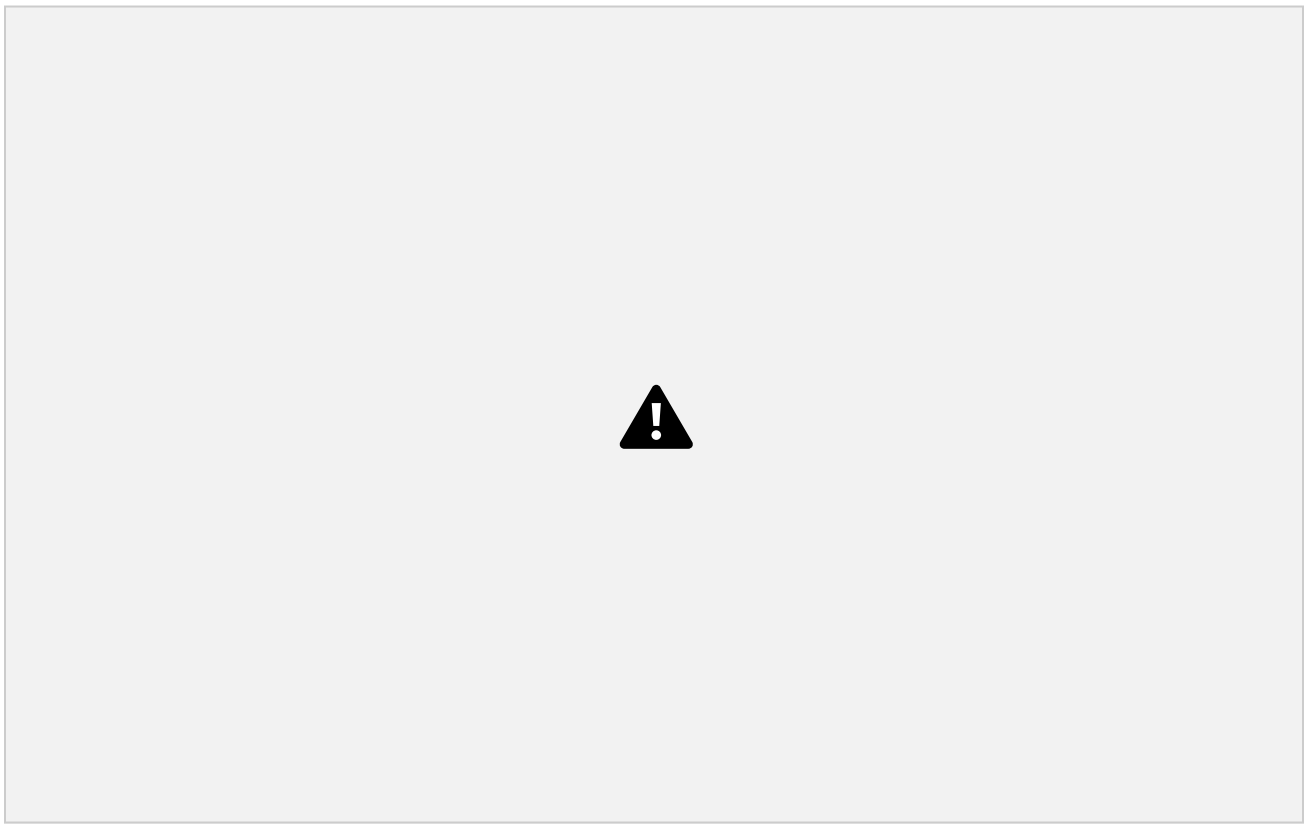


Fig 5.2 State Diagram for Passport Automation System

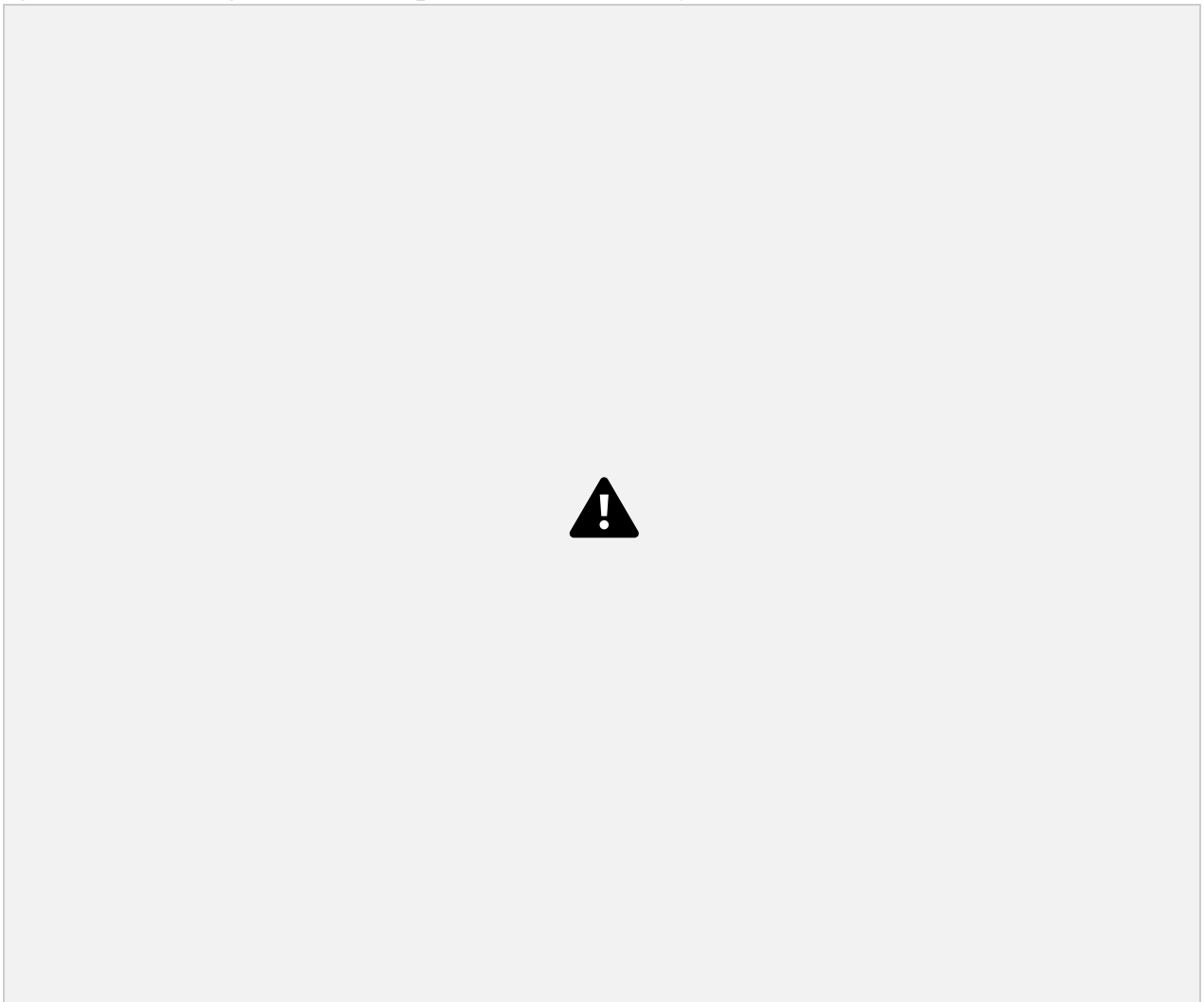


Fig 5.3 Use-Case Diagram for Passport Automation System

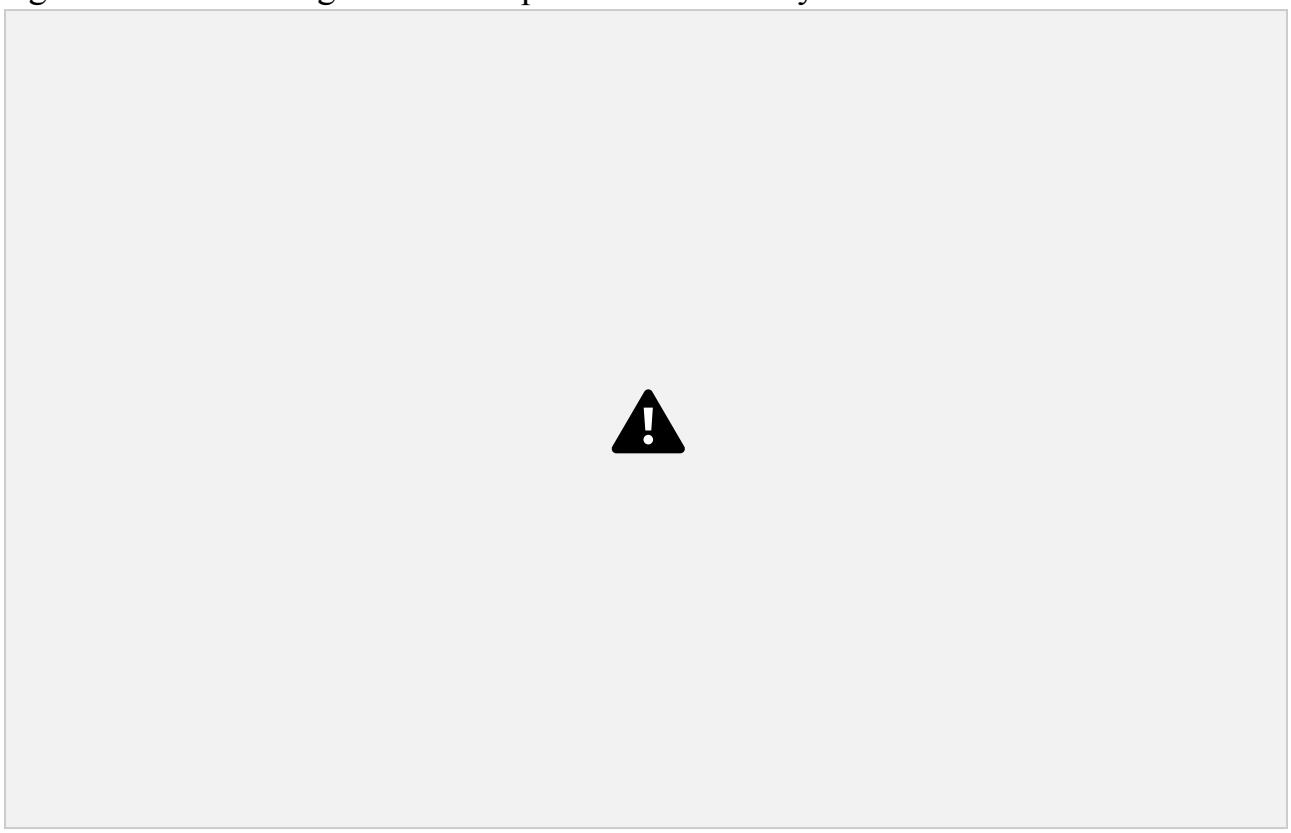


Fig 5.4 Sequence Diagram for Passport Automation System

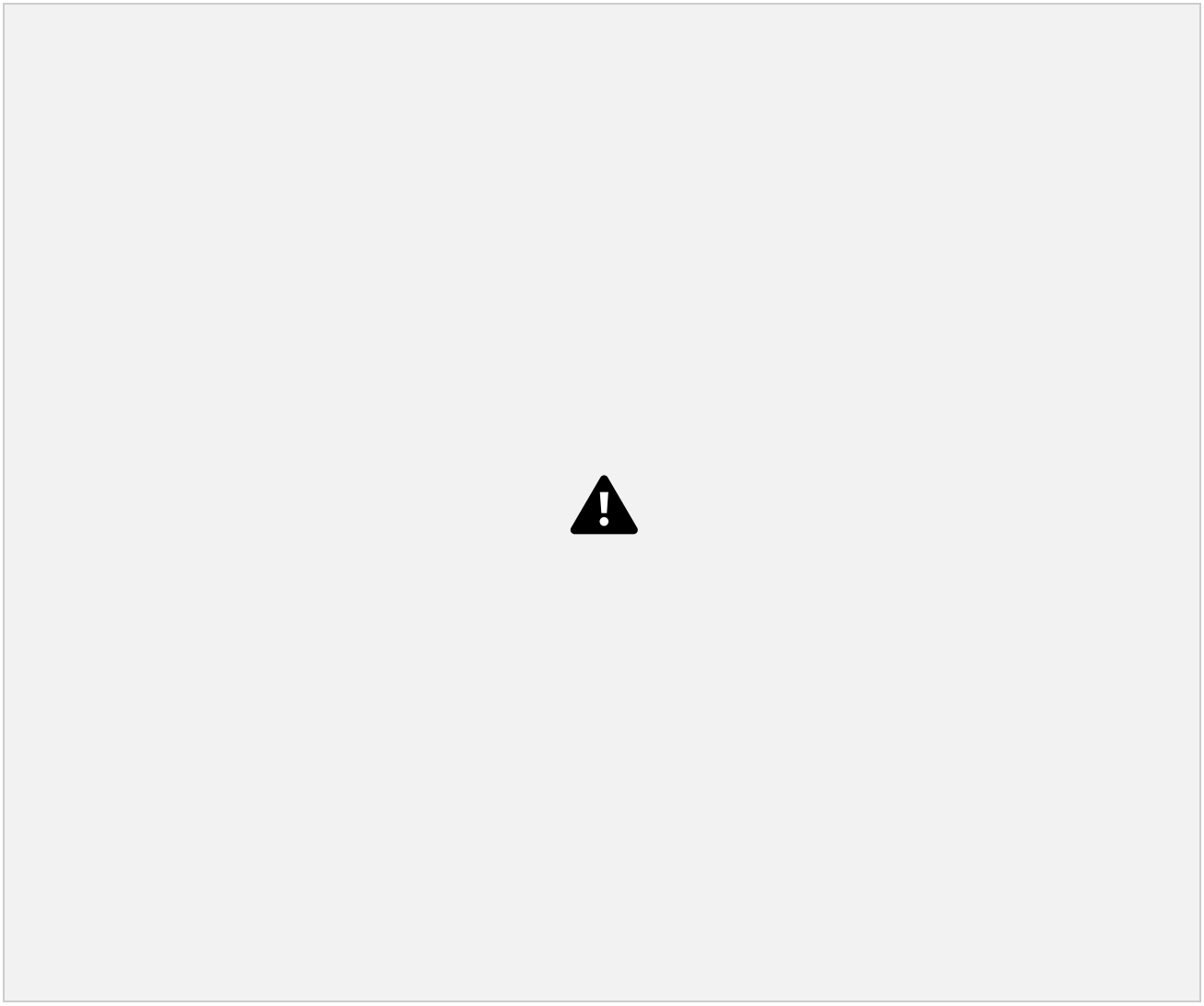


Fig 5.5 Activity Diagram for Passport Automation System