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 6th Semester Laboratory

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Y Shamil Ahamed
(1BM21CS248)

Faculty In charge:

Sheetal. V. A

Assistant Professor

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

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 **Y Shamil Ahamed(1BM21CS248)** during the 6th Semester Mar-June-2024.

Signature of the Faculty Incharge:

Sheetal. V. A:

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

Table of Contents

1. HOTEL MANAGEMENT SYSTEM

1.1 Problem Statement

The Hotel Management System is a software solution designed to streamline hotel operations. It maintains information on guests, staff, and rooms, managing reservations,check-ins, and check-outs. The system tracks services provided to guests, handles billing and payments, and generates invoices. Additionally, it manages staff schedules, payroll, andperformance evaluations. The goal is to enhance operational efficiency, improve guest satisfaction, and provide insights into hotel performance.

1.2 Software Requirement Specification

1.2.1 Introduction

a. Purpose of this Document

The purpose of this document is to outline the requirements and specifications for the development of a Hotel Management System. This document serves as a guide for the development team to ensure all necessary functionalities are included and provides a clear understanding of the system's objectives.

b. Scope of this Document

This document describes the overall functioning and main objectives of the Hotel Management System. It details the value the system will provide to customers, including efficient hotel operations management, improved guest satisfaction, and comprehensive performance insights. It also outlines the estimated development cost and time required forthe project.

c. Overview

The Hotel Management System is a software application designed to manage hotel operations, including guest reservations, check-ins, check-outs, billing, and service management. The system also handles staff scheduling, payroll, and performance evaluation, providing a streamlined approach to hotel management.

1.2.2 General Description

The Hotel Management System is designed to streamline hotel operations by providing a centralized platform for managing guest information, room inventory, reservations, check-ins, check-outs, billing, and service tracking. It enhances operational efficiency by allowing easy access to guest details and room availability, facilitating accurate billing and payment processing, and managing staff schedules and payroll. The system is tailored for hotel staff, including receptionists, managers, housekeeping staff, and administrative personnel, ensuring that they can efficiently handle day-to-day tasks. Its features and benefits include improved guest satisfaction, reduced manual errors, and comprehensive performance reporting, making it an essential tool for modern hotel management.

1.2.3 Functional Requirements

a. Guest Information Management

- Store and manage guest details
- Retrieve guest information quickly

b. Reservation and Room Management

- Manage room inventory and availability
- Handle reservations, check-ins, and check-outs

c. Service Tracking

- Track services requested by guests
- Manage housekeeping and room service tasks

d. Billing and Payments

- Generate invoices for guests
- Process various payment methods

e. Staff Management

- Manage staff schedules and shifts
- Track staff performance and payroll

1.2.4 Interface Requirements

a. User Interfaces

- Intuitive graphical user interface for easy navigation

- Responsive design for use on various devices

b. Software Interfaces

- Integration with existing hotel management software (if any)
- Support for data import/export functionalities

c. Communication Interfaces

- Real-time notifications for staff and guests
- Email and SMS integration for alerts and confirmations

1.2.5 Performance Requirements

a. System Performance

- Quick response times for user actions
- Efficient data processing and retrieval

b. Resource Utilization

- Optimal memory usage
- Scalability to handle increasing data and user load

c. Error Handling

- Low error rate in data processing
- Robust error logging and notification mechanisms

1.2.6 Design Constraints

a. Technical Constraints

- Use of a specific programming language or framework
- Compatibility with existing hotel hardware and software

b. Regulatory Constraints

- Compliance with data protection and privacy regulations
- Adherence to industry standards for security and operations

1.2.7 Non-Functional Attributes

a. Usability

- User-friendly interface and easy navigation
- Comprehensive user documentation and support

b. Portability

- Support for multiple operating systems and devices

c. Reliability

- High system uptime and availability
- Redundant systems to ensure continuous operation

d. Security

- Secure data storage and access control
- Protection against unauthorized access and data breaches

e. Scalability

- Ability to scale with growing user and data demands
- Flexible architecture to accommodate future enhancements

1.2.8 Preliminary Schedule and Budget

a. Schedule

- Initial planning and requirement gathering: 1 month
- Design and architecture development: 2 months
- Implementation and coding: 4 months
- Testing and quality assurance: 2 months
- Deployment and training: 1 month

b. Budget

- Estimated development cost: ₹ 2,00,00,000
- Additional costs for hardware, training, and maintenance: ₹ 50,00,000

1.3 Class Diagram

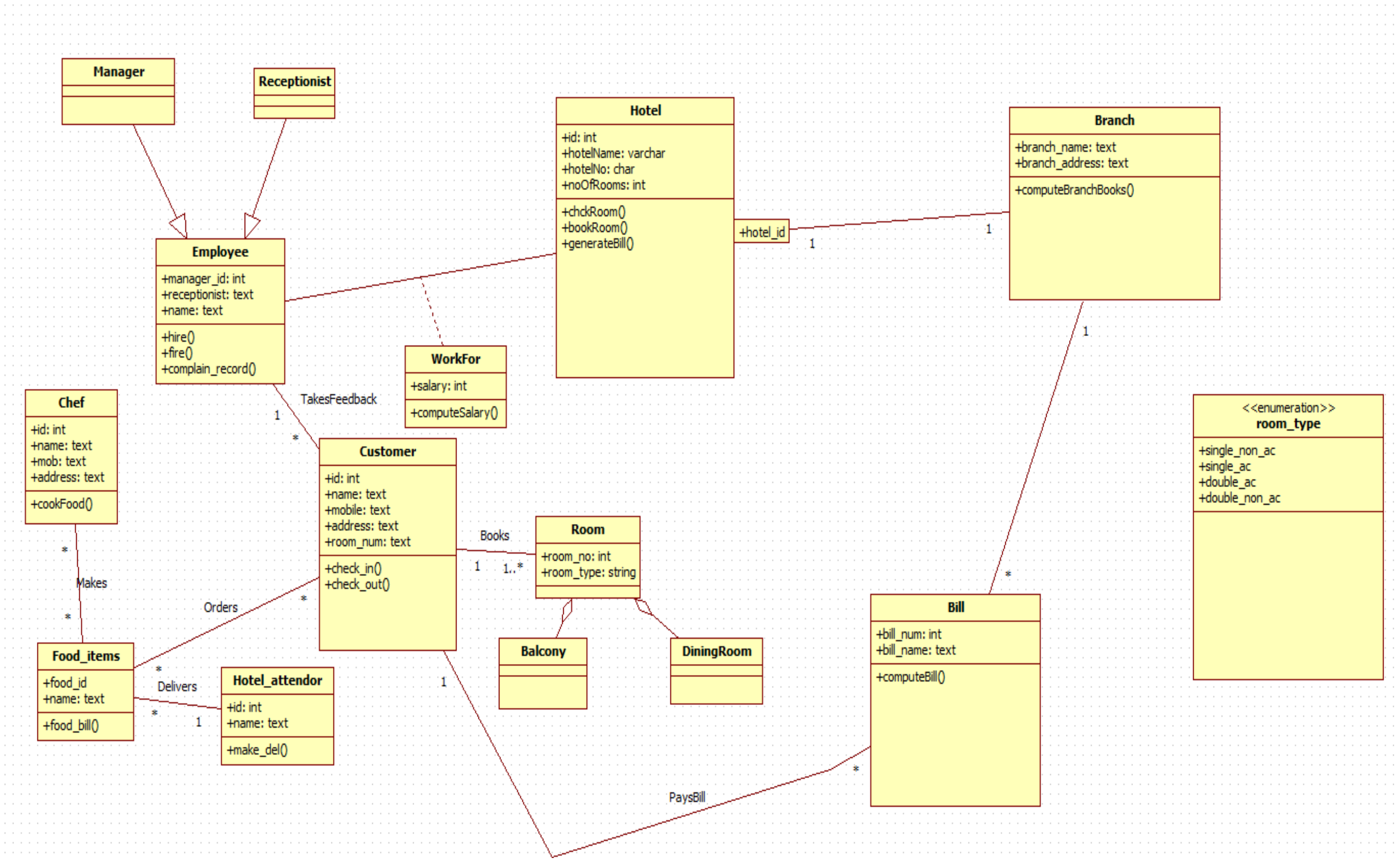


Fig. 1.1: Class Diagram Screenshot

1.4 State Diagram:

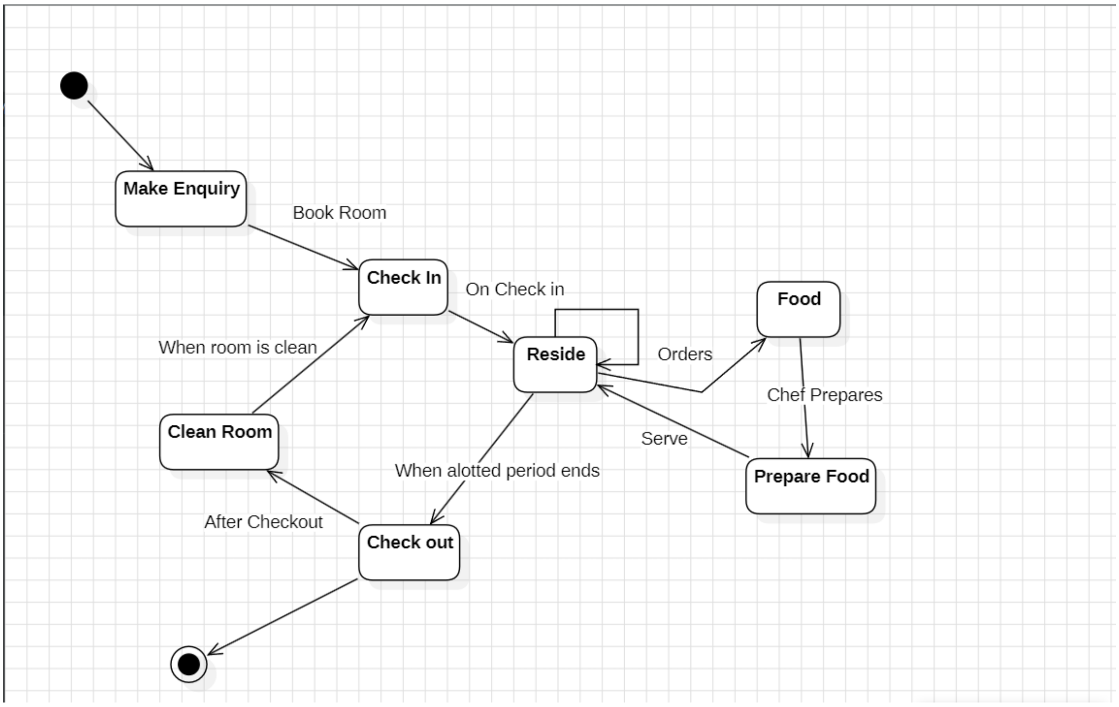


Fig. 1.2: State Diagram Screenshot

1.5 Use Case Diagram

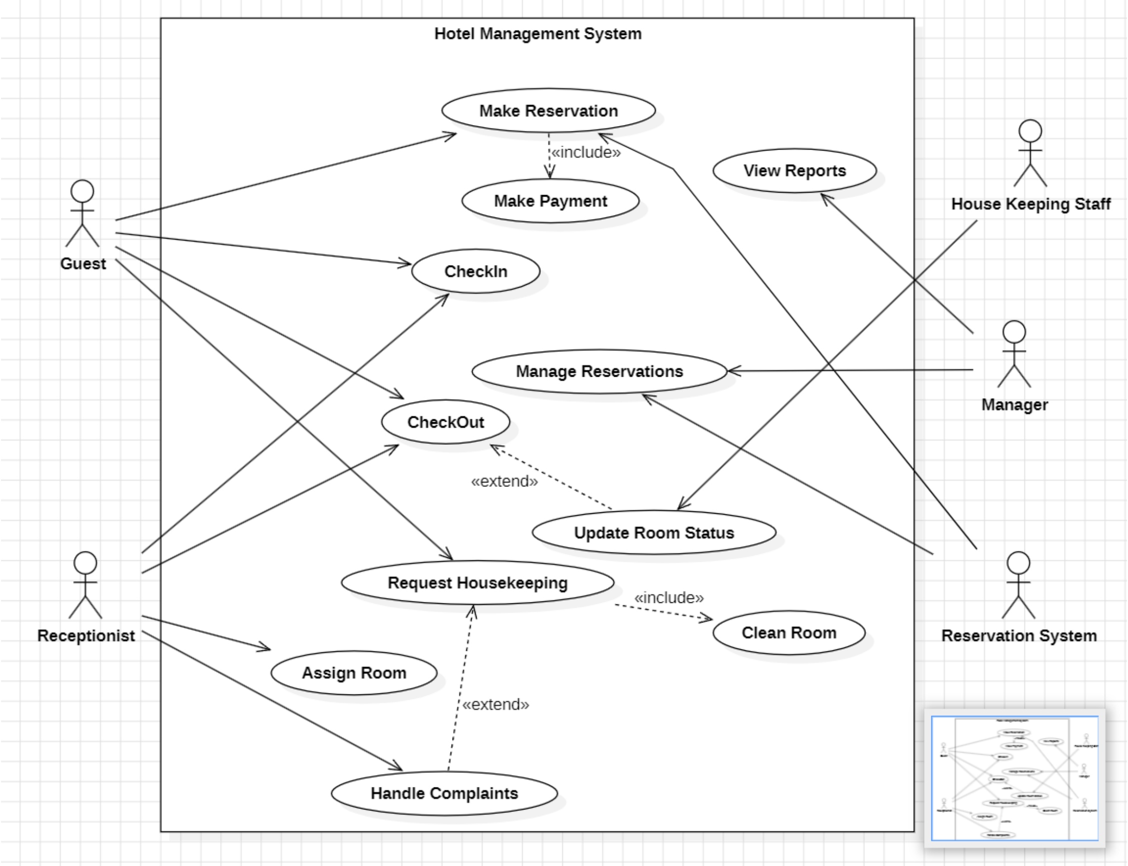


Fig. 1.3: Use Case Diagram Screenshot

1.6 Sequence Diagram

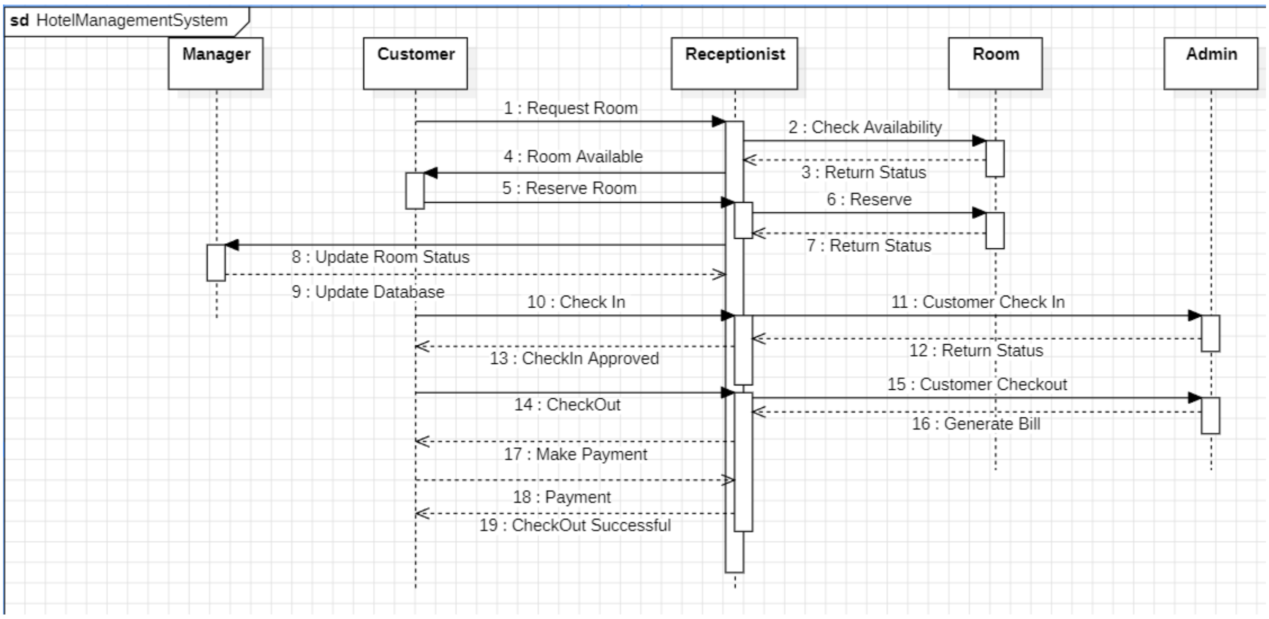


Fig. 1.4: Sequence Diagram Screenshot

1. 7 Activity Diagram

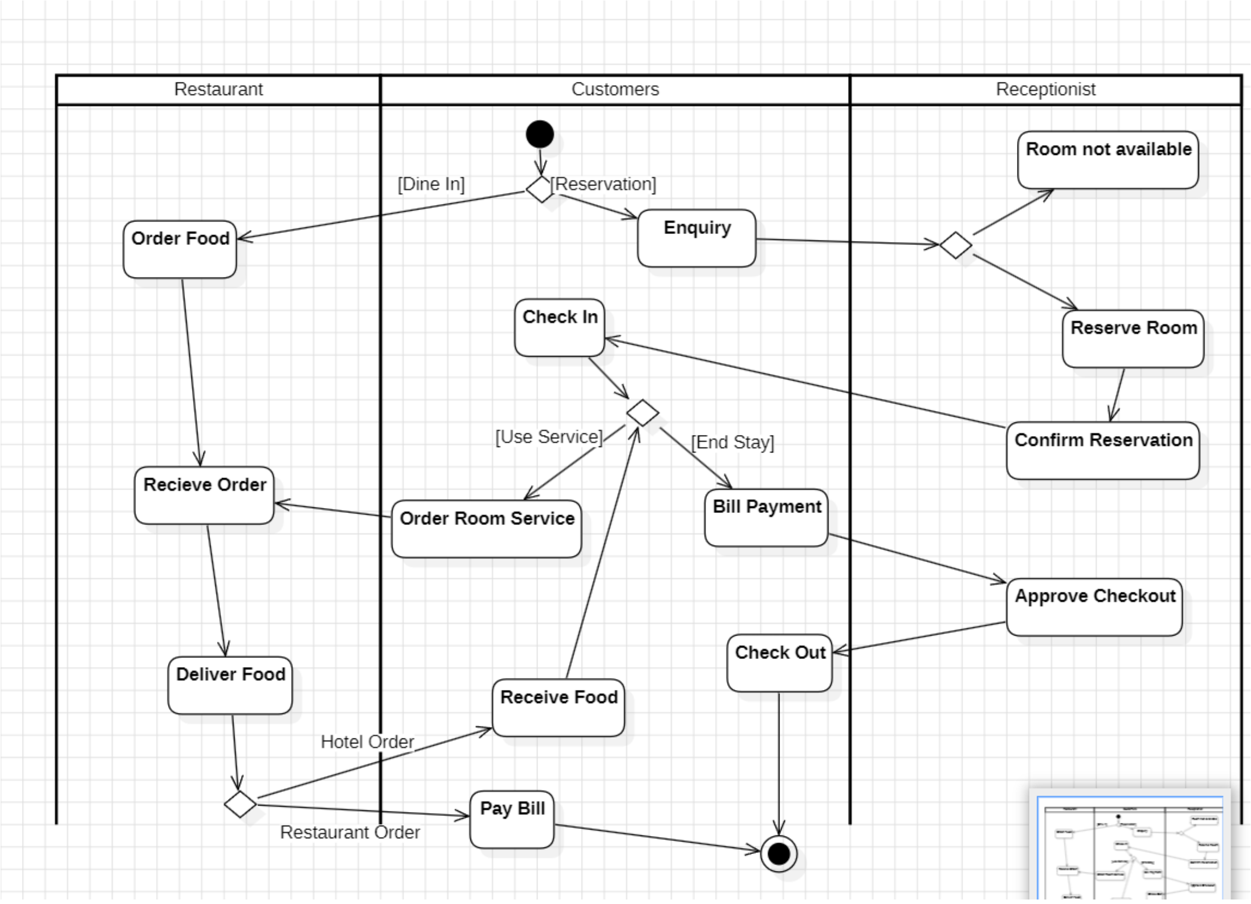


Fig. 1.5: Activity Diagram Screenshot

2. CREDIT CARD PROCESSING

2.1 Problem Statement

The Credit Card Processing System aims to address the challenges faced by businesses and financial institutions in securely managing credit card transactions. The system will focus on enhancing security measures, minimizing transaction processing times, simplifying user management, ensuring scalability, and maintaining compliance with industry standards. The primary goal is to provide a secure, efficient, and user-friendly platform for processing credit card payments, benefiting businesses, financial institutions, and users alike.

2.2 Software Requirement Specification

2.2.1 Introduction

a. Purpose of this Document

The purpose of this SRS document is to define the requirements for developing a Credit Card Processing System. It serves as a comprehensive guide for the development team, stakeholders, and other involved parties to understand the system's functionalities, interfaces, performance expectations, design constraints, and non-functional attributes.

b. Scope of this Document

This document encompasses the entire scope of the Credit Card Processing System, including its objectives, features, user interactions, system performance, security measures, and development timeline. It outlines the value proposition of the system to stakeholders, highlighting its importance in enhancing payment processing efficiency and security.

c. Overview

The Credit Card Processing System is designed to streamline credit card transactions for businesses, financial institutions, and users. It will provide a secure platform for authorizing, processing, and settling credit card payments, with a focus on user experience, data security, compliance with industry standards, and scalability to handle increasing transaction volumes.

2.2.2 General Description

The Credit Card Processing System will consist of the following key components and functionalities:

a. User Management

- Users can register, login, and manage their accounts.
- Account management includes adding, updating, and deleting credit card details.
- User authentication and access control mechanisms ensure secure access to accounts.

b. Transaction Processing

- Real-time authorization, processing, and settlement of credit card transactions.
- Support for multiple payment gateways and card networks.
- Transaction history tracking and reporting for users and administrators.

c. Security Measures

- Encryption and tokenization of sensitive credit card information.
- Compliance with Payment Card Industry Data Security Standard (PCI DSS) requirements.
- Regular security audits, vulnerability assessments, and intrusion detection mechanisms.

d. Interfaces

- Client Interface: Web and mobile interfaces for users to initiate and manage transactions.
- Employee Interface: Dashboard for employees to monitor transactions, handle disputes, and provide support.
- Admin Interface: Centralized dashboard for system administrators to manage settings, users, and reports.

2.2.3 Functional Requirements

a. User Account Management

- Users should be able to register with valid personal information and login securely.
- Users can add, update, and delete credit card details in their accounts.
- Account management actions should be logged for auditing purposes.

b. Transaction Processing

- The system must support real-time authorization and processing of credit card transactions.

- Transaction status updates should be visible to users and administrators.
- Automated settlement processes should be implemented based on transaction outcomes.

2.2.4 Interface Requirements

a. Client Interface

- Intuitive web and mobile interfaces for users to manage their accounts and initiate transactions.
- Responsive design for seamless user experience across devices.

b. Employee Interface

- Role-based access control for employees with different permissions (e.g., customer support, transaction monitoring).
- Dashboard with transaction tracking, dispute resolution tools, and customer communication features.

c. Admin Interface

- Centralized dashboard for administrators to manage system settings, user accounts, and transaction reports.
- Reporting tools for monitoring system performance, transaction volumes, and compliance metrics.

2.2.5 Performance Requirements

a. Response Time

- System response time for user actions should be less than 2 seconds.
- Transaction processing time should not exceed 5 seconds.

b. Scalability

- The system should handle a minimum of 1000 concurrent transactions.
- Scalable architecture to accommodate future growth and increased transaction volumes.

2.2.6 Design Constraints

a. Security Constraints

- Encryption standards (e.g., AES-256) for securing credit card data.
- Use of secure communication protocols (e.g., HTTPS) for data transmission.
- Regular security patches and updates to mitigate vulnerabilities.

b. Technology Stack

- Backend development using Java Spring Boot framework for robustness and scalability.
- Frontend development with React.js for responsive and dynamic user interfaces.
- Database management using PostgreSQL for data storage and retrieval.

2.2.7 Non-Functional Attributes

a. Security

- Data encryption at rest and in transit.
- Access control mechanisms to prevent unauthorized access to sensitive data.
- Security incident response and mitigation procedures.

b. Reliability

- System uptime of at least 99.9% to ensure uninterrupted service availability.
- Automated backup and recovery processes for data protection and disaster recovery.

c. Scalability

- Horizontal scalability to handle increased transaction volumes during peak periods.
- Load balancing and caching mechanisms for optimal performance.

2.2.8 Preliminary Schedule and Budget

a. Development Timeline

- Estimated development duration: 6 months.
- Iterative development approach with regular milestones and reviews.

b. Budget Estimate

- Development budget: ₹ 1,00,00,000.
- Includes costs for development, testing, deployment, and initial maintenance.

2.3 Class Diagram

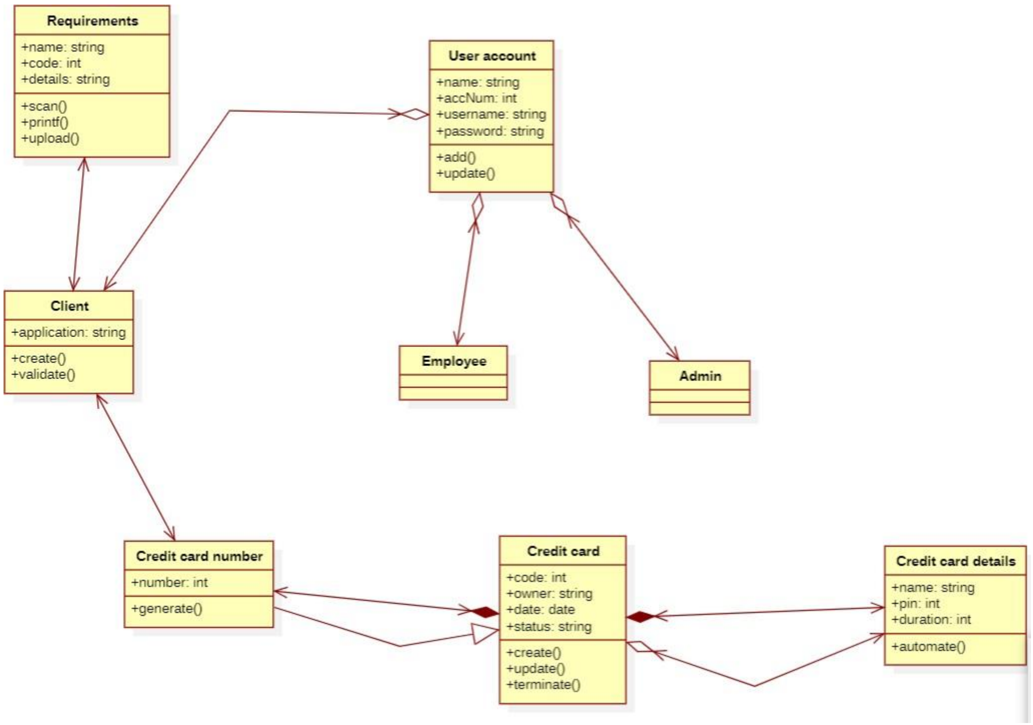


Fig. 2.1: Class Diagram Screenshot

2.4 State diagram

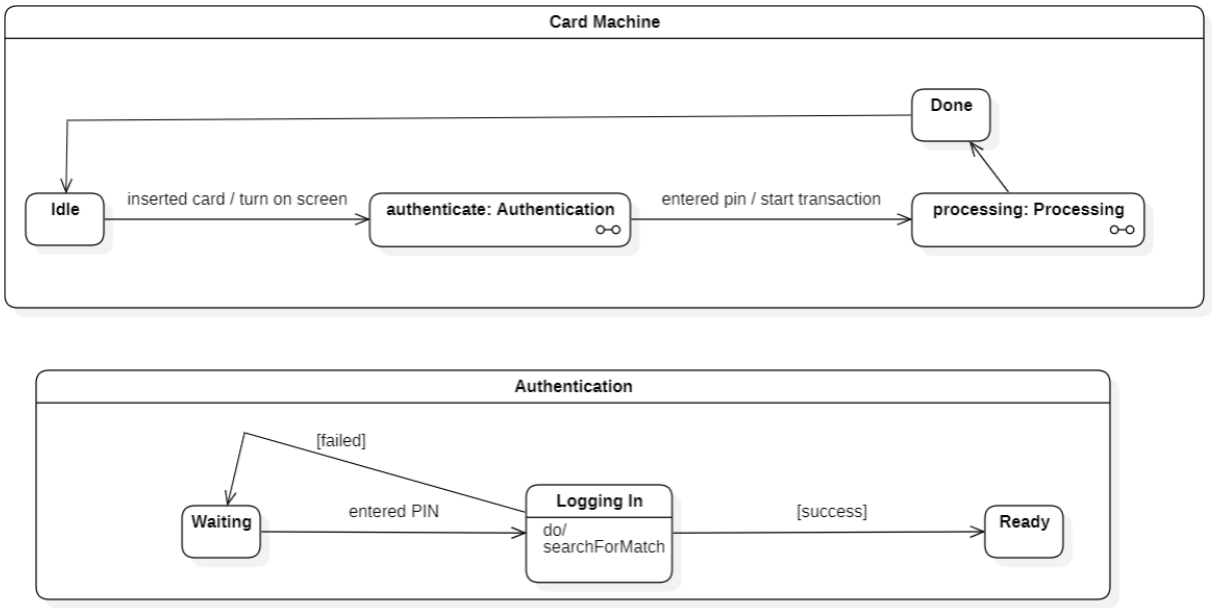


Fig. 2.2: a. State Diagram Screenshot

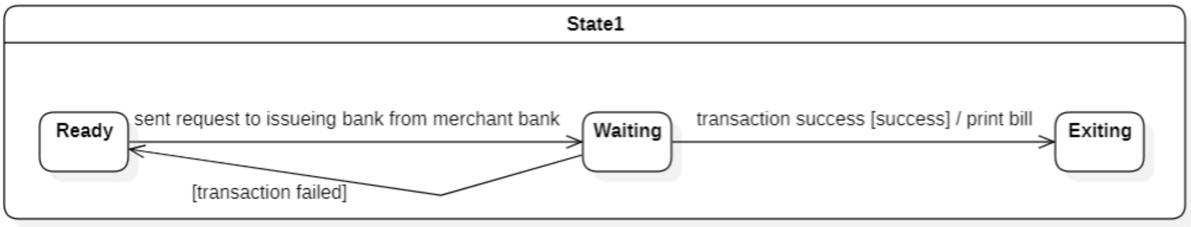

State1
Ready
sent request to issueing bank from merchant bank
Waiting
transaction success [success] / print bill
Exiting
[transaction failed]

Fig. 2.2: b. State Diagram Screenshot

2.5 Use Case Diagram

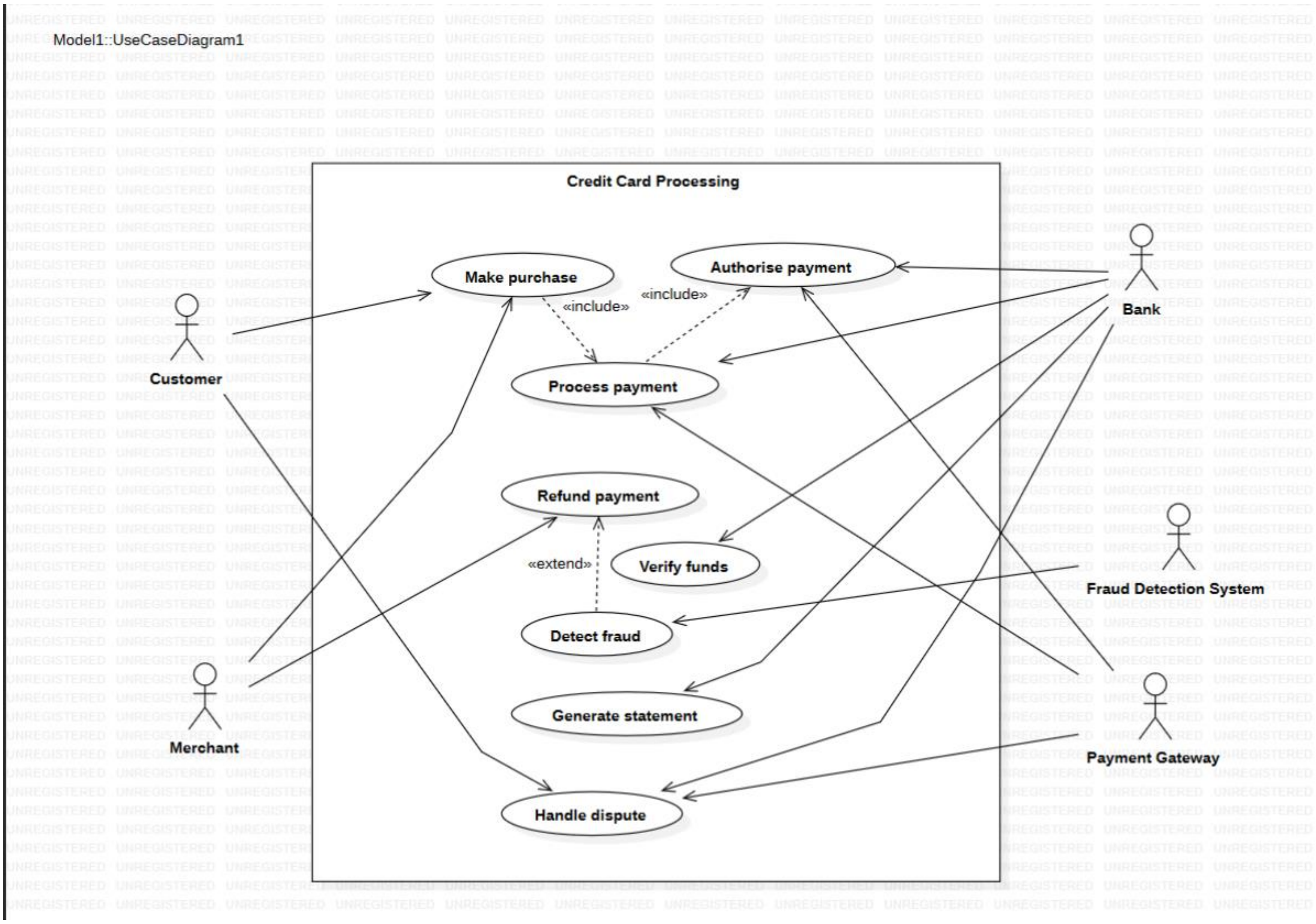


Fig. 2.3: Use Case Diagram Screenshot

2.6 Sequence diagram

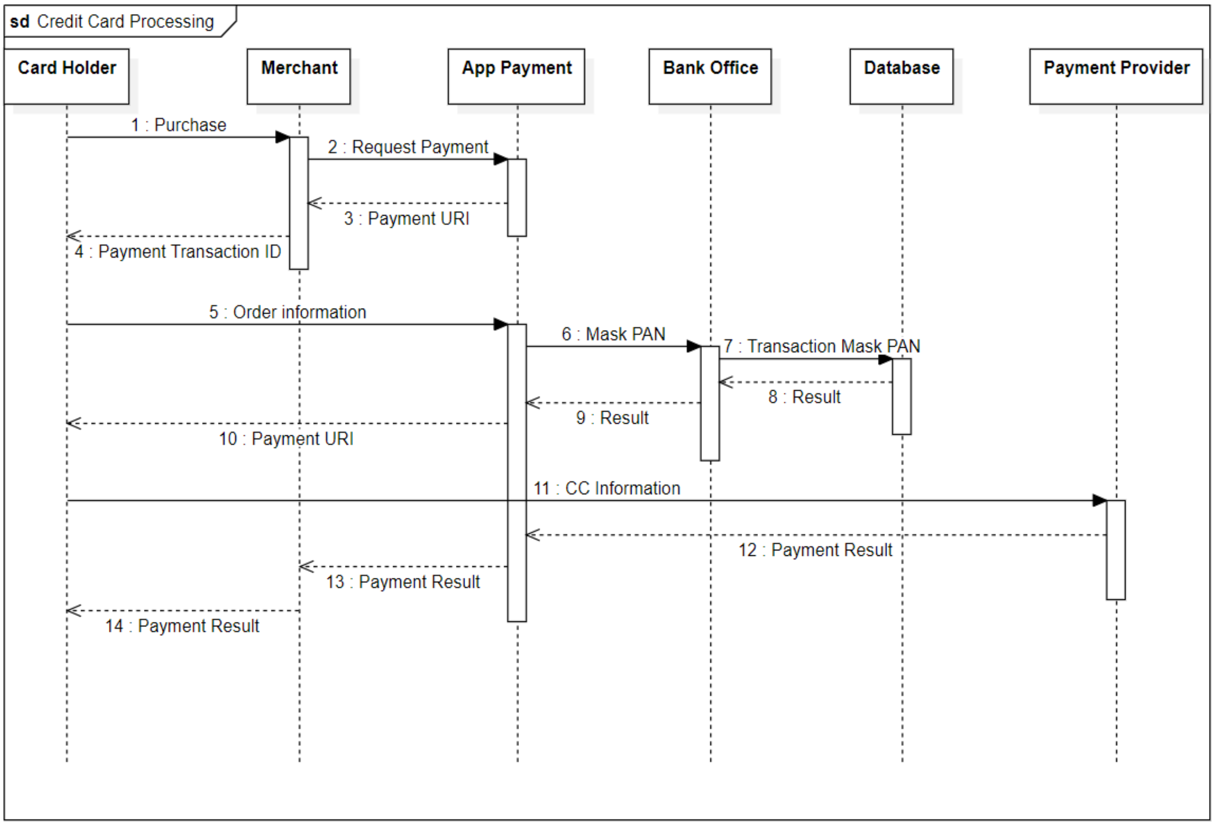


Fig. 2.4: Sequence Diagram Screenshot

2.7 Activity Diagram

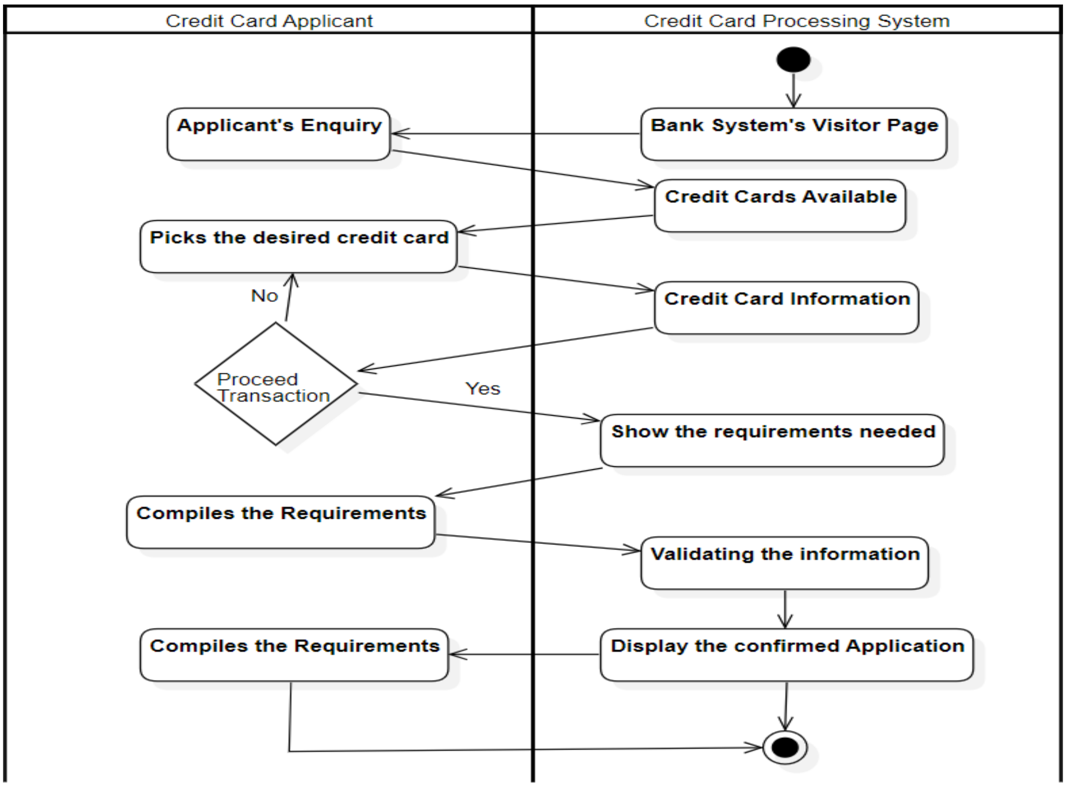

Credit Card Applicant
Credit Card Processing System
Applicant's Enquiry
Bank System's Visitor Page
Credit Cards Available
Picks the desired credit card
Credit Card Information
No
Proceed Transaction
Yes
Show the requirements needed
Compiles the Requirements
Validating the information
Compiles the Requirements
Display the confirmed Application

Fig. 2.5: Activity Diagram Screenshot

3. LIBRARY MANAGEMENT SYSTEM

3.1 Problem Statement

Libraries face significant challenges in managing their operations efficiently due to thereliance on manual processes and outdated systems. These issues include difficulties incataloging books, managing member records, tracking borrowing and returning transactions, and generating insightful reports. The manual handling of these tasks often leads to errors, delays, and a poor user experience for both library staff and members. There is a need for an automated, user-friendly, and reliable Library Management System (LMS) that streamlines these operations, enhances accuracy, and provides real-time access to information, ultimately improving the overall efficiency and effectiveness of library services.

3.2 Software Requirement Specification

3.2.1 Introduction

a. Purpose of this Document

The purpose of this document is to provide a detailed description of the Library Management System (LMS). It aims to outline the system’s functionalities, interfaces, performance requirements, and constraints to ensure a clear understanding of the system's objectives and development process. This document serves as a guide for stakeholders, developers, and users, ensuring that all requirements are met for an efficient and effective library management solution.

b. Scope of this Document

This document encompasses the overall functionality and objectives of the LMS, detailing how it will manage library operations, including book cataloging, member management, borrowing and returning processes, and report generation. It will provide value to library staff and members by automating and streamlining library operations. Additionally, it includes an overview of the estimated development cost and the time required to implement the system.

c. Overview

The Library Management System is a software application designed to manage the operations of a library. It will facilitate the management of books, members, and transactions (borrowing and returning books). The system aims to enhance the efficiency of library operations, reduce manual errors, and provide real-time data on library activities.

3.2.2 General Description

The Library Management System will serve the primary function of managing library resources and operations. It will cater to the needs of librarians and library members, providing features such as book cataloging, member management, transaction handling, and reporting. The system is crucial for maintaining organized records and offering a seamless library experience.

a. User Characteristics

- **Librarians**: Manage the entire system, including book cataloging, member registration, and overseeing transactions.
- **Members**: Can search for books, borrow and return books, and check their transaction history.

b. Features and Benefits

- **Book Cataloging:** Easy entry and management of book details.
- **Member Management:** Efficient handling of member registrations and updates.
- **Transaction Management:** Automated tracking of book borrow and return activities.
- **Reporting:** Generation of various reports for better insights into library operations

c. Importance

The LMS is essential for the smooth operation of a library. It minimizes manual work, reduces errors, and provides quick access to information, thus enhancing the overall user experience.

3.2.3 Functional Requirements

The functional requirements detail the specific operations the LMS will perform, ranked by priority.

a. Book Cataloging

- Adding, updating, and deleting book records.
- Searching and filtering books by various criteria (title, author, genre).

b. Member Management

- Registering new members.
- Updating member information.
- Deleting member records.

c. Transaction Handling

- Recording book borrow and return transactions.
- Managing due dates and overdue notifications.
- Tracking transaction history for each member.

d. Reporting

- Generating reports on borrowed books, overdue books, and member activity.
- Providing summary statistics on library operations.

3.2.4 Interface Requirements

The LMS will have several interfaces for effective communication and operation.

a. User Interface

- A user-friendly graphical interface for librarians and members.
- Accessible via web browsers and potentially mobile applications.

b. Database Interface

Secure and efficient interaction with the backend database for storing and retrieving data.

c. External Interfaces

- Integration with email services for sending notifications.
- Potential integration with external library networks for inter-library loan services.

3.2.5 Performance Requirements

a. Response Time

- The system should respond to user actions (e.g., searching for a book) within 2 seconds.

b. Memory Usage

- Efficient memory usage to handle a large number of records without performance degradation.

c. Error Rate

- The system should have a minimal error rate, aiming for less than 0.01% in transaction processing.

3.2.6 Design Constraints

a. Hardware Constraints

- Must be compatible with existing library hardware, such as computers and barcode scanners.

b. Software Constraints

- Use of a specific database management system (e.g., MySQL, PostgreSQL).
- Compatibility with standard operating systems (Windows, macOS, Linux).

c. Regulatory Constraints

- Compliance with data protection regulations, ensuring the security and privacy of member information.

3.2.7 Non-Functional Attributes

a. **Security**

- Implement robust authentication and authorization mechanisms.
- Ensure data encryption for sensitive information.

b. **Portability**

- The system should be portable across different operating systems and devices.

c. **Reliability**

- High reliability with minimal downtime, aiming for 99.9% availability.

d. **Reusability**

- Modular design to allow reuse of components in other library-related applications.

e. **Application Compatibility**

- Compatible with other applications used in library operations, such as cataloging software.

f. **Data Integrity**

- Ensure data integrity with regular backups and validation checks.

g. **Scalability**

- The system should be scalable to accommodate the growing number of books and members.

3.2.8 Preliminary Schedule and Budget

a. **Schedule**

- **Phase 1: Requirements Gathering and Analysis** (2 months)
- **Phase 2: Design** (1 month)
- **Phase 3: Development** (3 months)
- **Phase 4: Testing** (2 months)
- **Phase 5: Deployment** (1 month)

- **Total Duration**: 9 months

b. Budget

- **Personnel Costs**: ₹ 50,00,000
- **Software and Tools**: ₹ 10,00,000
- **Hardware Costs**: ₹ 5,00,000
- **Miscellaneous Expenses**: ₹ 5,00,000
- **Total Estimated Cost**: ₹ 70,00,000

This document outlines the comprehensive requirements and plans for the Library Management System, ensuring clarity and alignment among all stakeholders.

3.3 Class Diagram

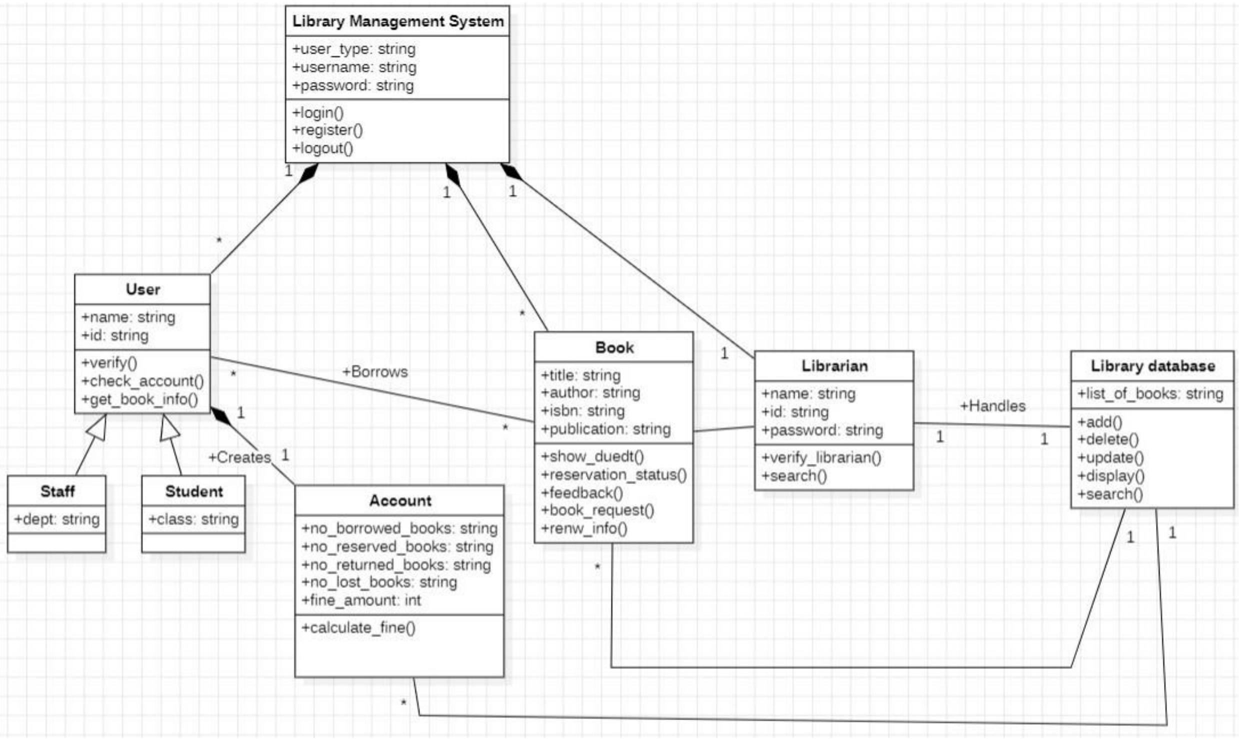
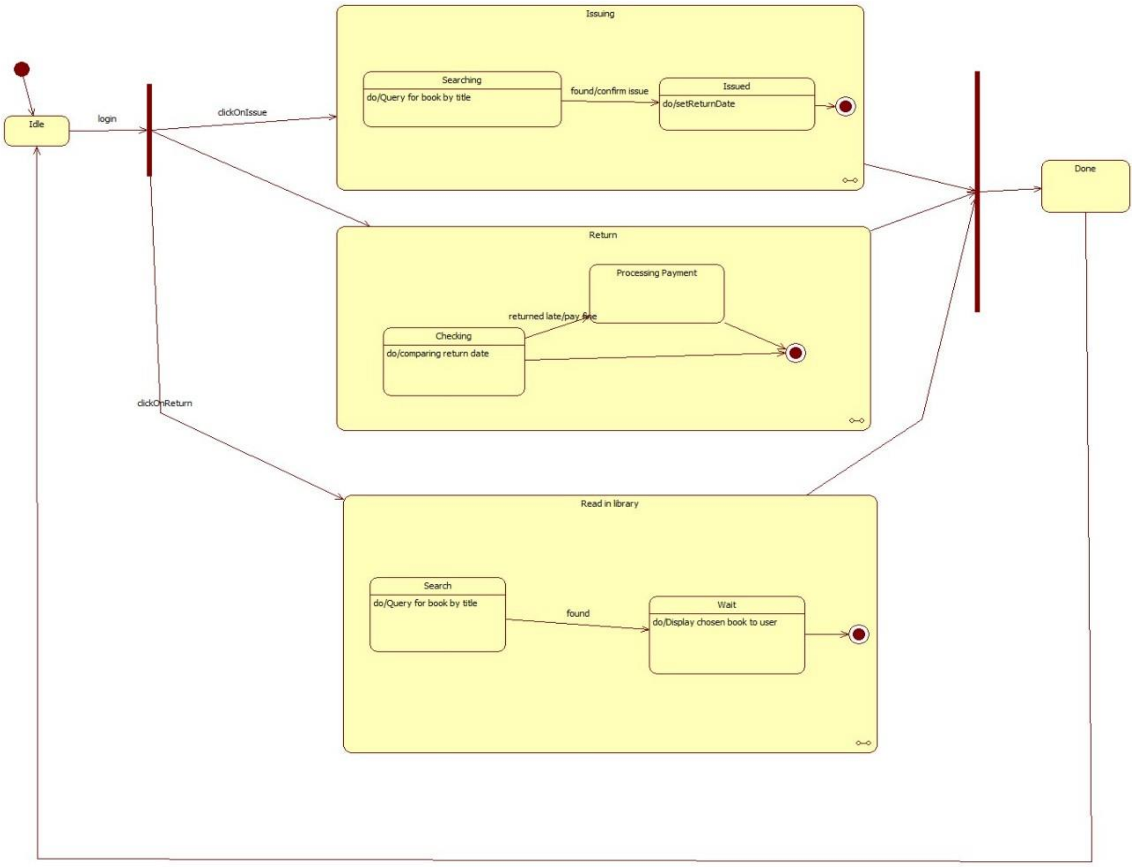


Fig. 3.1: Class Diagram Screenshot

3.4 State Diagram



3.5 Use case diagram

Fig. 3.2: State Diagram
Screenshot

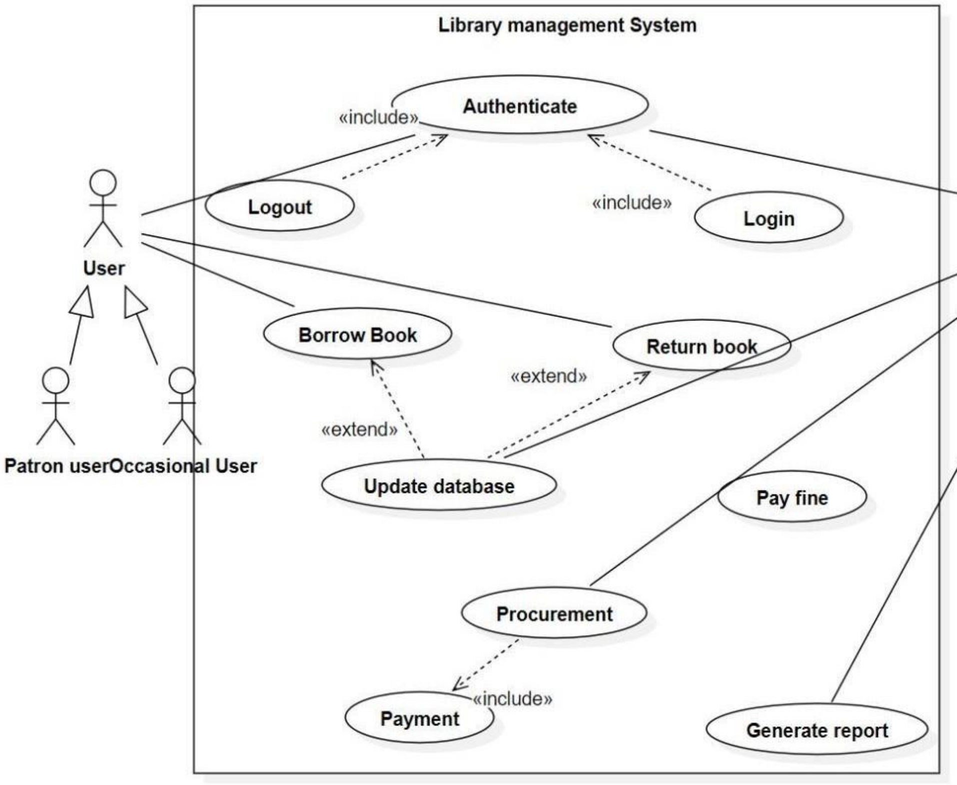


Fig. 3.3: Use Case Diagram Screenshot

3.6 Sequence Diagram

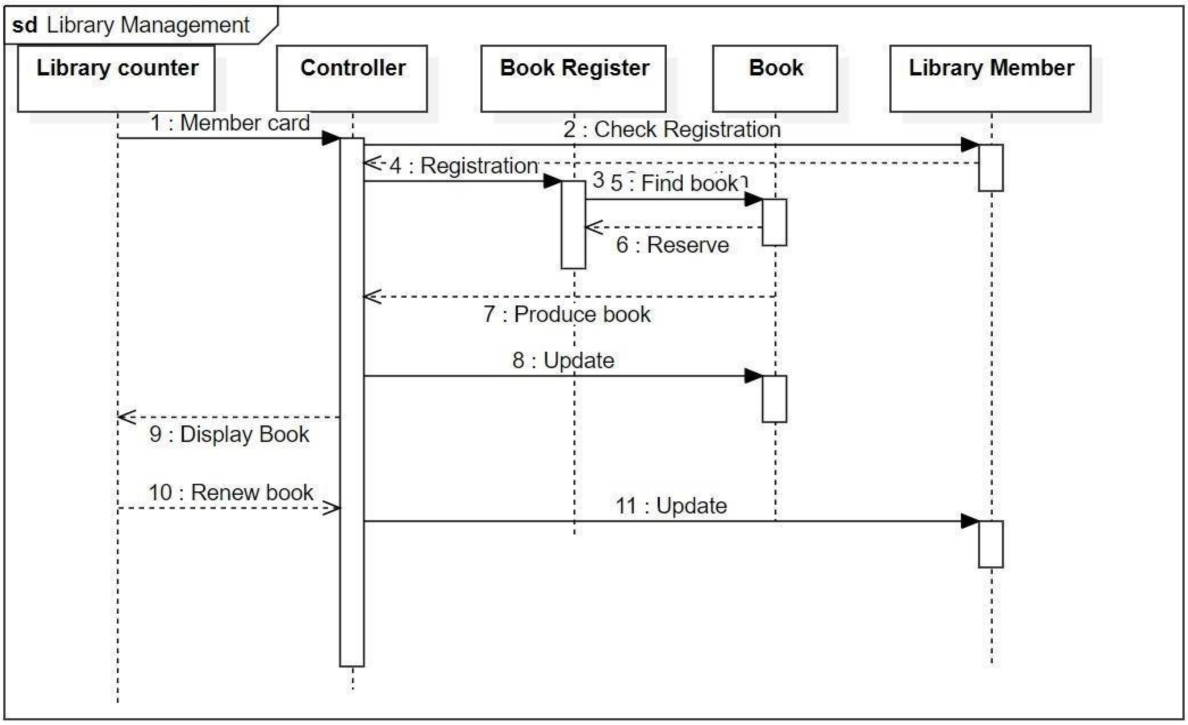


Fig. 3.4: Sequence Diagram Screenshot

3.7 Activity diagram

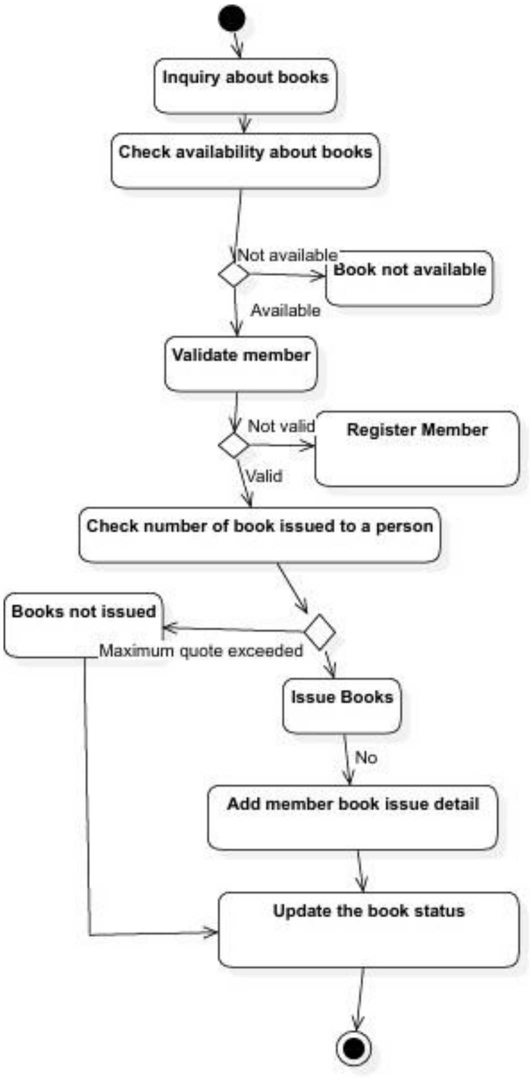


Fig. 3.5: Activity Diagram Screenshot

4. STOCK MAINTENENCE SYSTEM

4.1 Problem Statement

Organizations face significant challenges in managing their stock efficiently due to thereliance on manual processes and outdated systems. These challenges include difficulties in accurately tracking stock levels, managing inventory movement, and generating insightful reports. Manual handling of these tasks often leads to errors, delays, and inefficiencies in stock management. There is a need for an automated, user-friendly, and reliable StockManagement System (SMS) that streamlines these operations, enhances accuracy, and provides real-time access to stock information, ultimately improving organizationalperformance.

4.2 Software Requirement Specification

4.2.1 Introduction

a. Purpose of this Document

The purpose of this document is to define the requirements for the development of a Stock Management System (SMS). It provides a detailed description of the system's functionalities, interfaces, performance requirements, and constraints to guide stakeholders, developers, and users throughout the development process.

b. Problem Statement

Organizations face significant challenges in managing their stock efficiently due to manual processes and outdated systems. These challenges include difficulties in accurately tracking stock levels, managing inventory movement, and generating insightful reports. Manual handling of these tasks often leads to errors, delays, and inefficiencies in stock management. There is a need for an automated, user-friendly, and reliable Stock Management System that streamlines these operations, enhances accuracy, and provides real-time access to stock information, ultimately improving organizational performance.

c. Scope of this Document

This document encompasses the overall functionality and objectives of the StockManagement System, detailing how it will manage stock items, track inventory movement, generate reports, and facilitate efficient stock management. It includes an overview of the estimated development cost and time required for system implementation.

d. Overview

The Stock Management System is a software application designed to streamline the management of stock items, track inventory movement, and provide insights into stock operations. It aims to enhance the efficiency and accuracy of stock management processes, ultimately improving organizational performance.

4.2.2 General Description

a. User Characteristics

Stock Manager: Responsible for managing stock items, updating inventory, and generating reports.

b. Features and Benefits

Stock Item Management: Easy addition, update, and deletion of stock items. Inventory Tracking: Real-time monitoring of stock movement (inward and outward). Reporting: Generation of stock reports for analysis and decision-making.

c. Importance

The SMS is essential for organizations to maintain optimal stock levels, reduce stockouts, and improve inventory management efficiency.

4.2.3 Functional Requirements

a. Stock Item Management

- Adding new stock items with details such as name, quantity, unit price, etc.

- Updating existing stock items including quantity and price.
- Deleting stock items from the inventory.

b. Inventory Tracking

- Recording stock movements (inward and outward) with timestamps.
- Tracking stock levels and generating alerts for low stock.

c. Reporting

- Generating reports on stock levels, stock movements, and inventory turnover.
- Providing summary statistics for analysis and decision-making.

4.2.4 Interface Requirements

a. User Interface

- Intuitive graphical interface for stock managers.
- Accessibility via web browsers and potentially mobile applications.

b. Database Interface

- Secure interaction with the backend database for storing and retrieving stock data.

c. External Interfaces

- Integration with email services for sending notifications.
- Potential integration with suppliers' systems for automated stock replenishment.

4.2.5 Performance Requirements

a. Response Time

- The system should respond to user actions (e.g., adding a stock item) within 2 seconds.

b. Memory Usage

- Efficient memory usage to handle a large number of stock items without performance degradation.

c. Error Rate

- The system should have a minimal error rate, aiming for less than 0.01% in stock recording.

4.2.6 Design Constraints

a. Hardware Constraints

- Compatibility with existing hardware infrastructure.

b. Software Constraints

- Use of a specific database management system (e.g., MySQL, PostgreSQL).
- Compatibility with standard operating systems (Windows, macOS, Linux).

4.2.7 Non-Functional Attributes

a. Security

- Implement robust authentication and authorization mechanisms.
- Ensure data encryption for sensitive stock information.

b. Portability

- The system should be portable across different operating systems and devices.

c. Reliability

- High reliability with minimal downtime, aiming for 99.9% availability.

d. Reusability

- Modular design to allow reuse of components in other inventory-related applications.

e. Data Integrity

- Ensure data integrity with regular backups and validation checks.

f. Scalability

- The system should be scalable to accommodate the growing number of stock items and transactions.

4.2.8 Preliminary Schedule and Budget

a. Schedule

- Phase 1: Requirements Gathering and Analysis (1 month)
- Phase 2: Design (1 month)
- Phase 3: Development (3 months)
- Phase 4: Testing (1 month)
- Phase 5: Deployment (1 month)
- Total Duration: 7 months

b. Budget

- Personnel Costs: ₹ 40,00,000
- Software and Tools: ₹ 15,00,000
- Hardware Costs: ₹ 5,00,000
- Miscellaneous Expenses: ₹ 5,00,000
- Total Estimated Cost: ₹ 65,00,000

This document outlines the comprehensive requirements and plans for the Stock Management System, ensuring clarity and alignment among all stakeholders.

4.3 Class Diagram

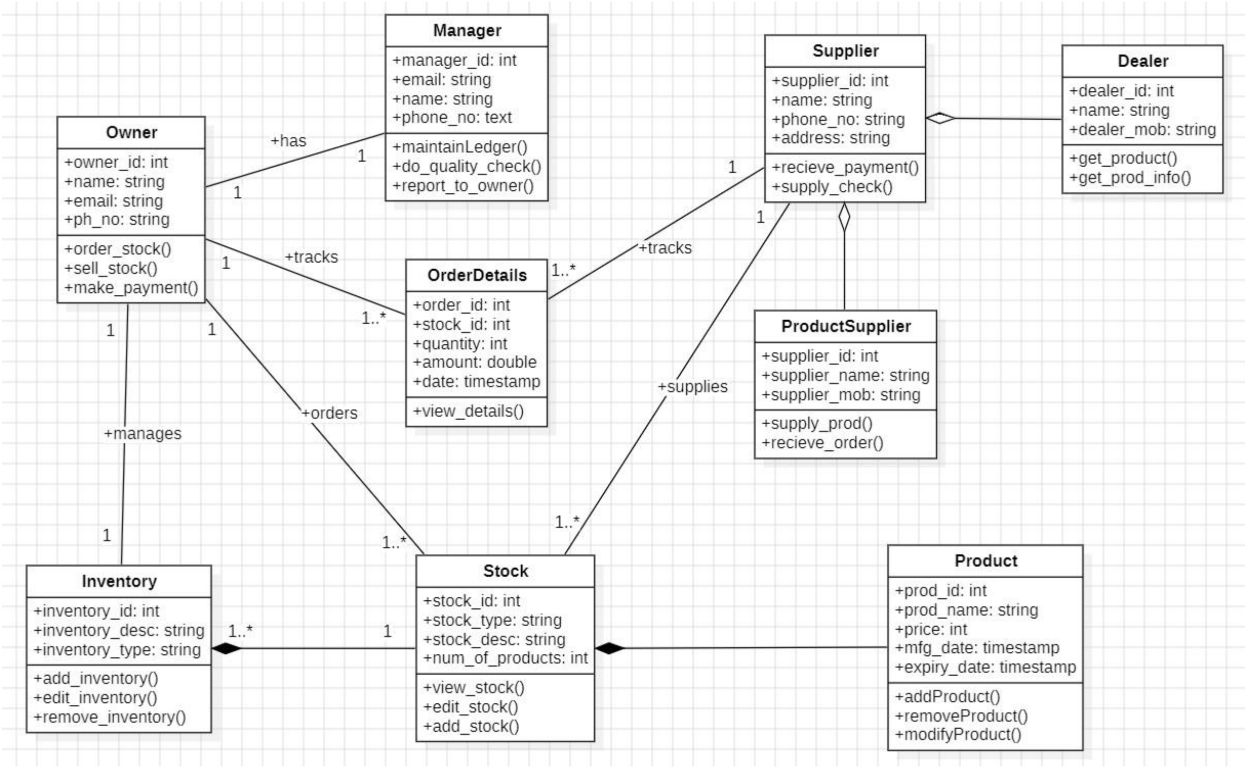


Fig. 4.1: Class Diagram Screenshot

4.4 State Diagram

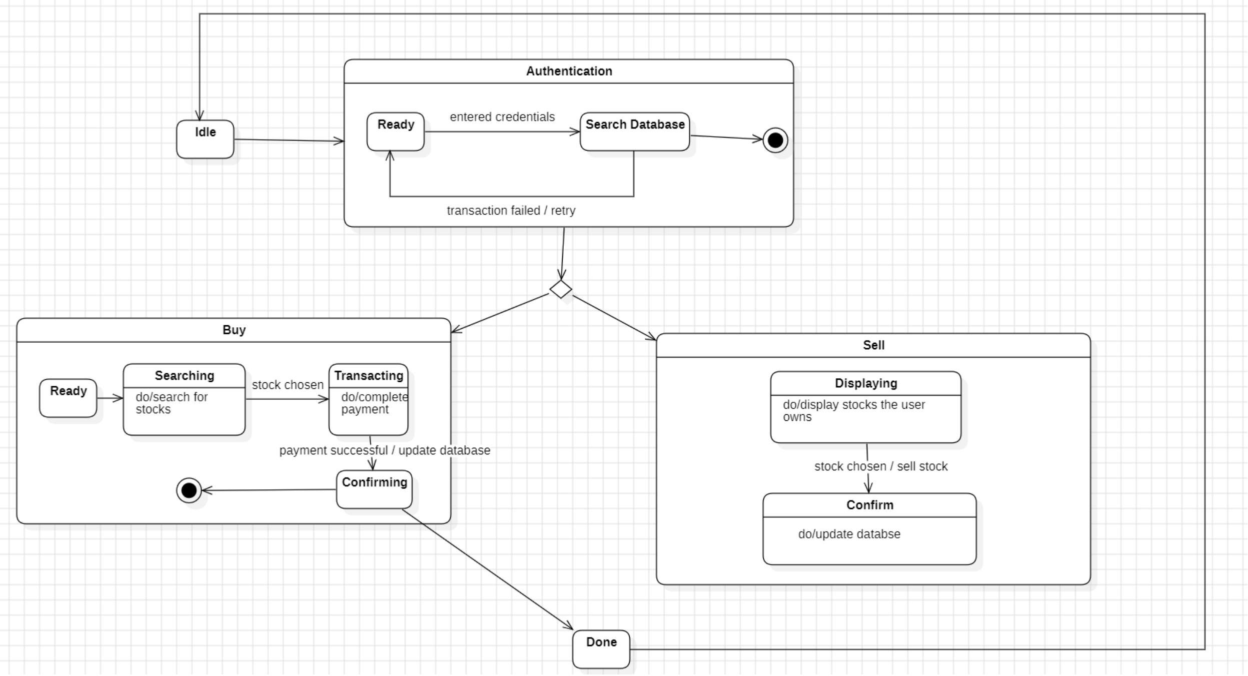


Fig. 4.2: State Diagram Screenshot

4.5 Use Case Diagram

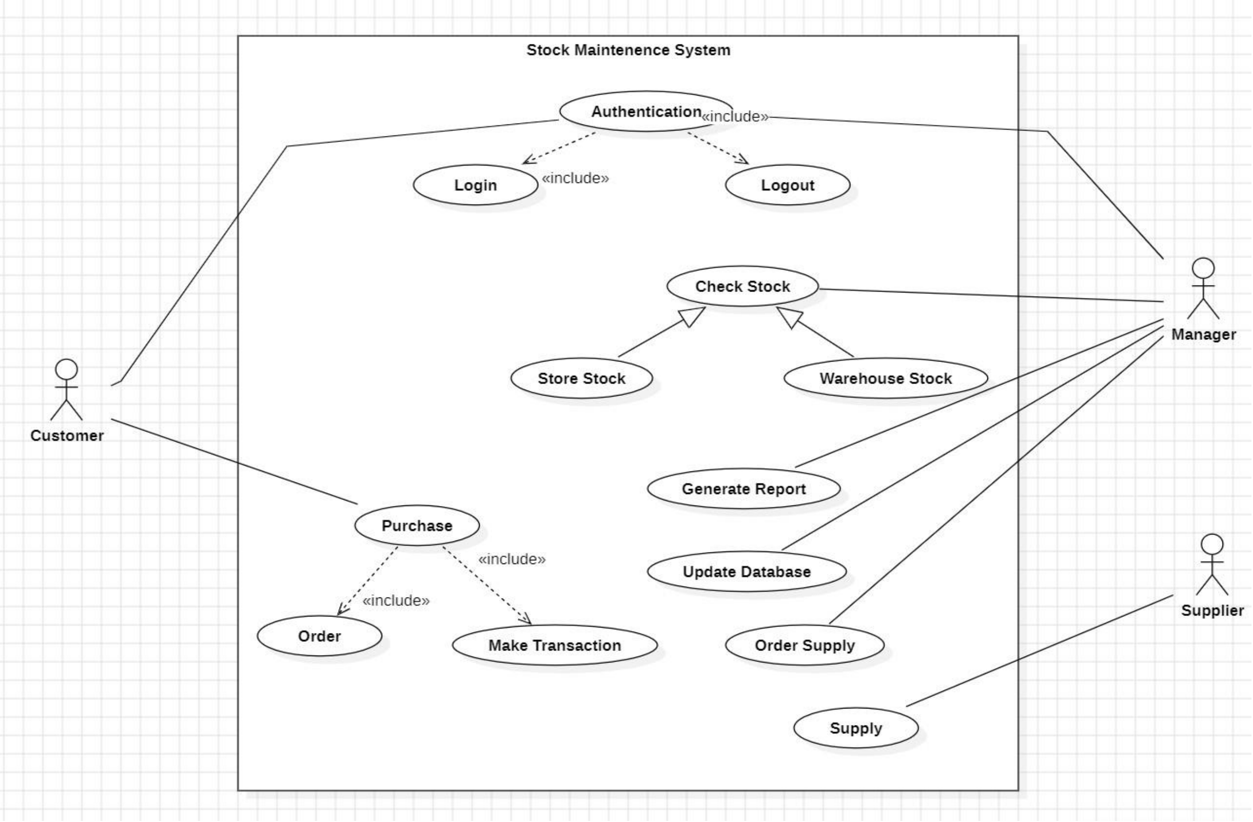


Fig. 4.3: Use Case Diagram Screenshot

4.6 Sequence Diagram

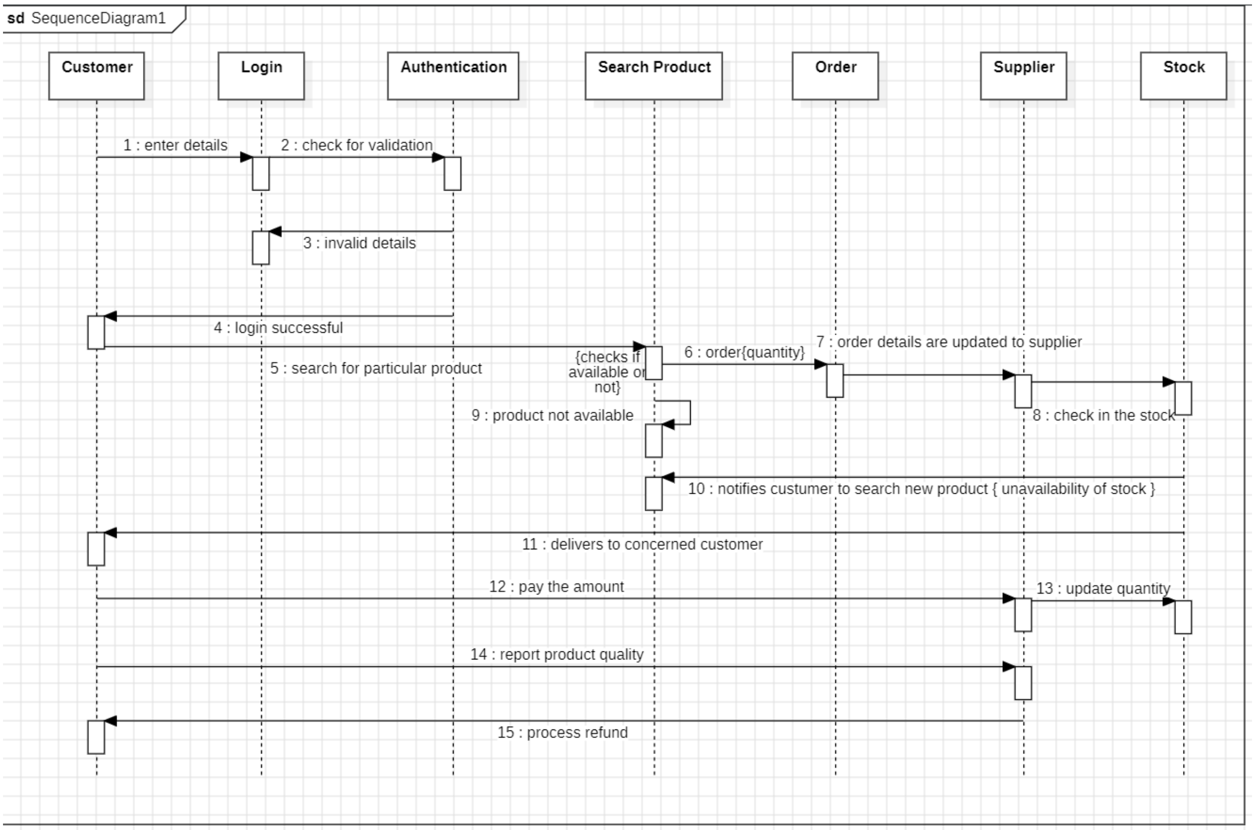

sd SequenceDiagram1
Customer
Login
Authentication
Search Product
Order
Supplier
Stock
1 : enter details
2 : check for validation
3 : invalid details
4 : login successful
5 : search for particular product
{checks if available or not}
6 : order{quantity}
7 : order details are updated to supplier
8 : check in the stock
9 : product not available
10 : notifies custumer to search new product { unavailability of stock }
11 : delivers to concerned customer
12 : pay the amount
13 : update quantity
14 : report product quality
15 : process refund

Fig. 4.4: Sequence Diagram Screenshot

4.7 Activity Diagram

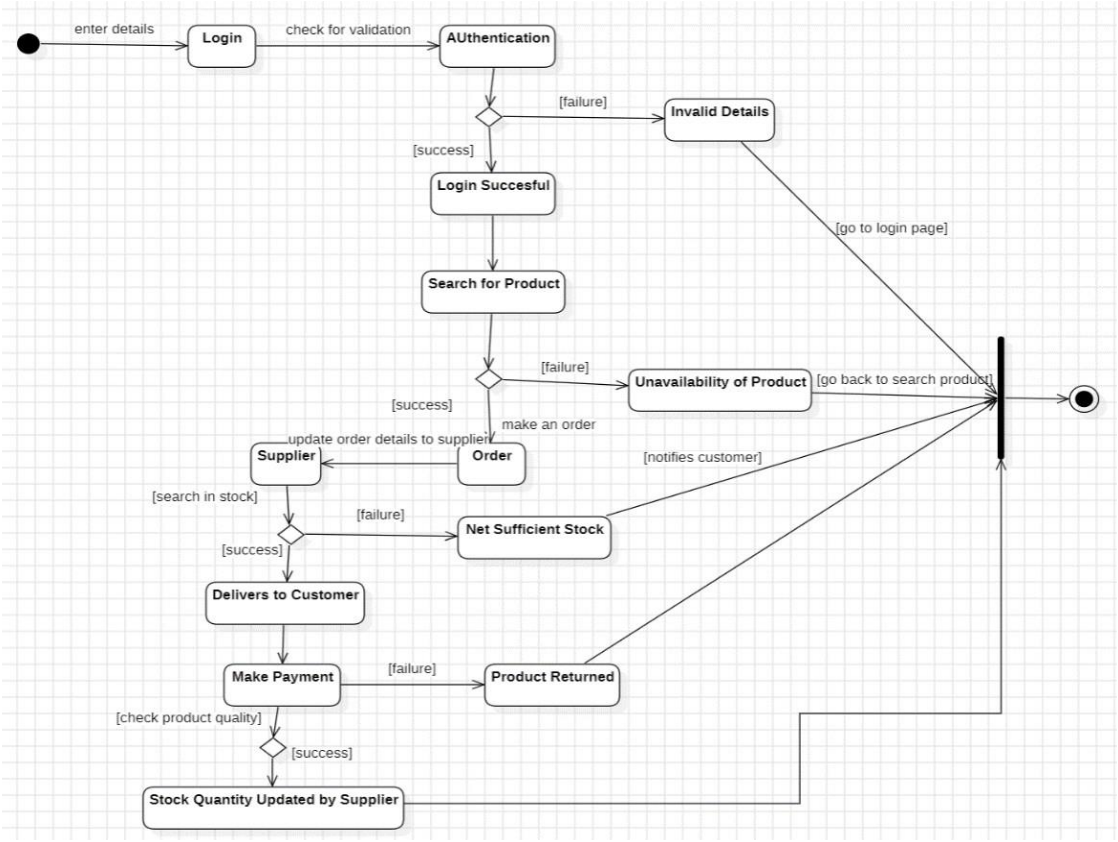


Fig. 4.5: Activity Diagram Screenshot

5. PASSPORT AUTOMATION SYSTEM

5.1 Problem Statement

Passport Automation System is used in the effective dispatch of passport to all of theapplicants. This system adopts a comprehensive approach to minimize the manual work andschedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name,address etc.,) filled by the applicant whose testament is verified for its genuineness by thePassport Automation System with respect to the already existing information in the database. This forms the first and foremost step in the processing of passport application. After the first round of verification done by the system,the information is in turn forwarded to theregional administrator's (Ministry of External Affairs) office. The application is then processed manually based on the report given by the system, and any forfeiting identified can make the applicant liable to penalty as per the law.The system forwards the necessary details to the police for its separate verification whosereport is then presented to the administrator. After all the necessary criteria have been met,the original information is added to the database and the passport is sent to the applicant.

5.2 Software Requirement Specification

5.2.1 Introduction

a. Purpose of this Document

The purpose of this document is to define the requirements for the development of a Passport Automation System (AMS). It provides a detailed description of the system's functionalities, interfaces, performance requirements, and constraints to guide stakeholders, developers, and users throughout the development process.

b. Problem Statement

Passport Automation System is used in the effective dispatch of passport to all of theapplicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner. The core of the system is to get the online registration form (with details such as name, address etc.,) filled by the applicant whose testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database.

c. Scope of this Document

The System provides an online interface to the user where they can fill in their personal The authority concerned with the issue of passport can use this system to reduce his workload and process the application in a speedy manner. Provide a communication platform between the applicant and the administrator.Transfer of data between the Passport Issuing Authority and the Local Police for verification of applicant's information.

d. Overview

SRS includes two sections overall description and specific requirements - Overall description will describe major role of the system components and inter-connections. Specific requirements will describe roles & functions of the actors

5.2.2 General Description

a. Product Perspective

The PAS acts as an interface between the 'applicant' and the 'administrator'. This

system tries to make the interface as simple as possible and at the same time not risking the

security of data stored in. This minimizes the time duration in which the user receives the

passport.

b. Software Requirements

- Front End Client - The applicant and Administrator online interface is built using Microsoft Visual Basic 6.0.
- Back End – MS Access database.

c. Hardware Interface

The server is directly connected to the client systems. The client systems have access

to the database in th**e server.**

d. System Functions

- Secure Registration of information by the Applicants.

- Message box for Passport Application Status Display by the Administrator.
- Administrator can generate reports from the information and is the only authorized personnel to add the eligible application information to the database.

e. User Characteristics

- Applicant - They are the people who desires to obtain the passport and submit the information to the database.
- Administrator - He has the certain privileges to add the passport status and to approve the issue of passport. He may contain a group of persons under him to verify the documents and give suggestion whether or not to approve the dispatch of passport.
- Police - He is the person who upon receiving intimation from the PAS, perform a personal verification of the applicant and see if he has any criminal case against him before or at present. He has been vetoed with the power to decline an application by suggesting it to the administrator if he finds any discrepancy with the applicant. He communicates via this PAS.

5.2.3 Functional Requirements

a. Applicant Functions

- The system shall allow applicants to register for an account using a secure method.
- The system shall provide an online application form that captures all mandatory applicant information.
- The system shall allow applicants to upload required documents in various formats (e.g., PDF, JPG).
- The system shall integrate with a secure payment gateway for online fee collection.
- The system shall allow applicants to schedule appointments for biometric data capture (if applicable).
- The system shall provide a real-time dashboard for applicants to track application status.
- The system shall enable secure communication between applicants and administrators through an internal messaging system.

b. Administrator Functions

- The system shall provide administrators with a secure login mechanism.

- The system shall display a list of submitted applications for review.
- The system shall allow administrators to approve or reject applications with justification.
- The system shall integrate with government databases for verification of applicant information.
- The system shall provide secure storage and management of uploaded documents.
- The system shall facilitate passport printing and issuance management.
- The system shall allow administrators to create and manage user accounts (applicants & administrators).
- The system shall provide administrators with tools for system configuration and report generation.

5.2.6 Design Constraints

- The applicants require a computer to submit their information.
- Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.
- The user has to be careful while submitting the information. Much care is required.
- The Applicants and Administrator must have basic knowledge of computers and English Language.
- The applicants may be required to scan the documents and send.

5.2.7 Non-Functional Attributes

a. Performance Requirements

- The system response time for applicant actions (e.g., form submission, document upload) should be within 5 seconds on average.
- The system should be able to handle a peak load of X applications per hour (define based on expected volume).

b. Security Requirements

- The system shall employ industry-standard security practices to protect user data and prevent unauthorized access.
- All data transmission shall be encrypted using secure protocols (e.g., HTTPS).
- The system shall implement strong user authentication mechanisms.

- The system shall have provisions for user access control and role-based permissions.

c. Usability Requirements

- The user interface shall be intuitive and user-friendly for applicants with varying technical skills.
- The system shall provide clear instructions and informative messages to guide users through the application process.
- The system shall be accessible to users with disabilities in accordance with relevant accessibility standards.

5.2.8 Preliminary Schedule and Budget

a. Schedule

- Phase 1: Requirements Gathering and Analysis (1 month)
- Phase 2: Design (1 month)
- Phase 3: Development (3 months)
- Phase 4: Testing (1 month)
- Phase 5: Deployment (1 month)
- Total Duration: 7 months

b. Budget

Personnel Costs:	₹ 40,00,000
Software and Tools:	₹ 15,00,000
Hardware Costs:	₹ 5,00,000
Miscellaneous Expenses:	₹ 5,00,000
Total Estimated Cost:	₹ 65,00,000

5.3 Class Diagram

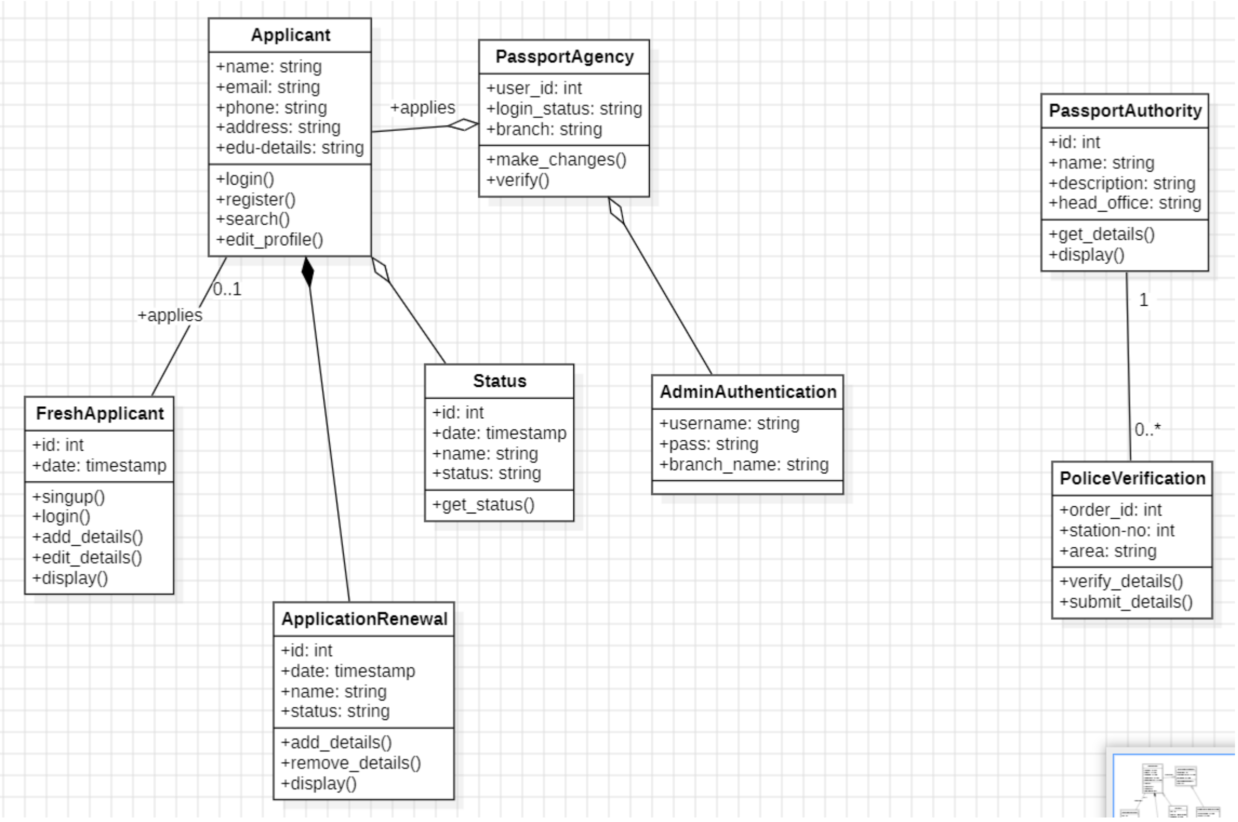


Fig. 5.1: Class Diagram Screenshot

5.4 State Diagram

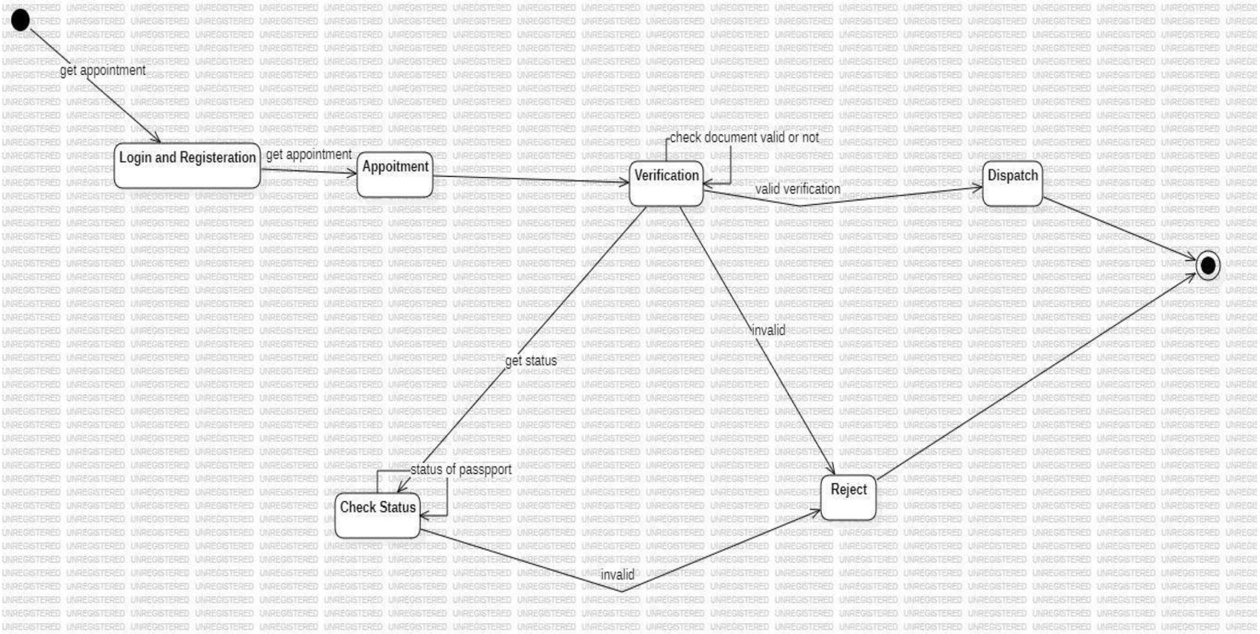


Fig. 5.2: State Diagram Screenshot

5.5 Use Case Diagram

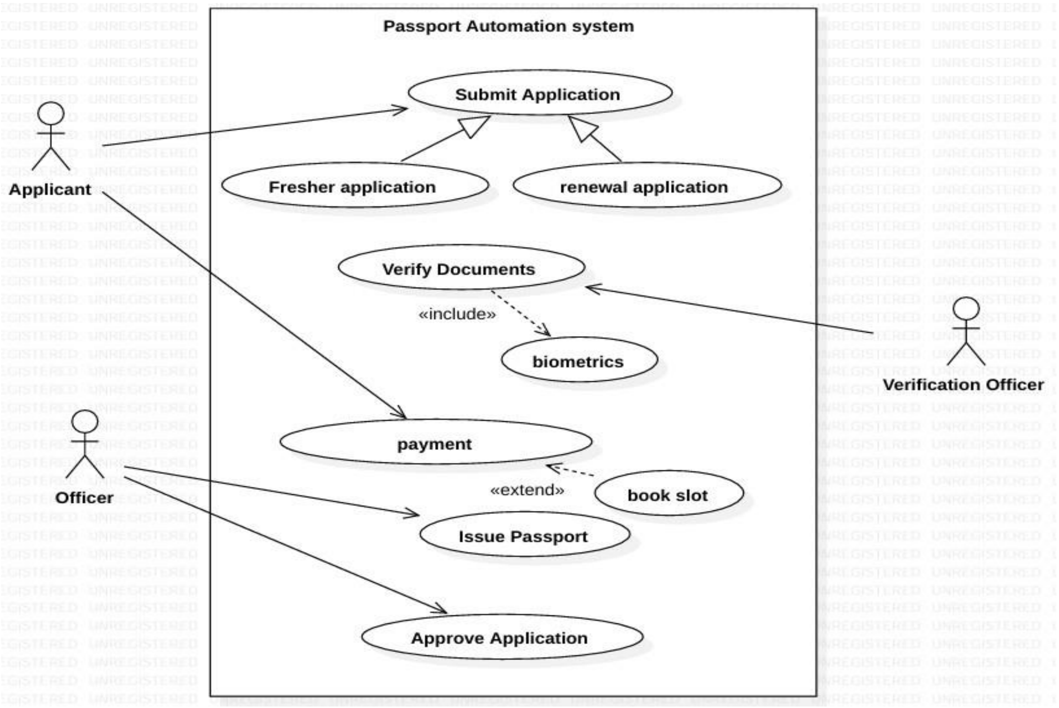


Fig. 5.3: Use Case Diagram Screenshot

5.6 Sequence Diagram

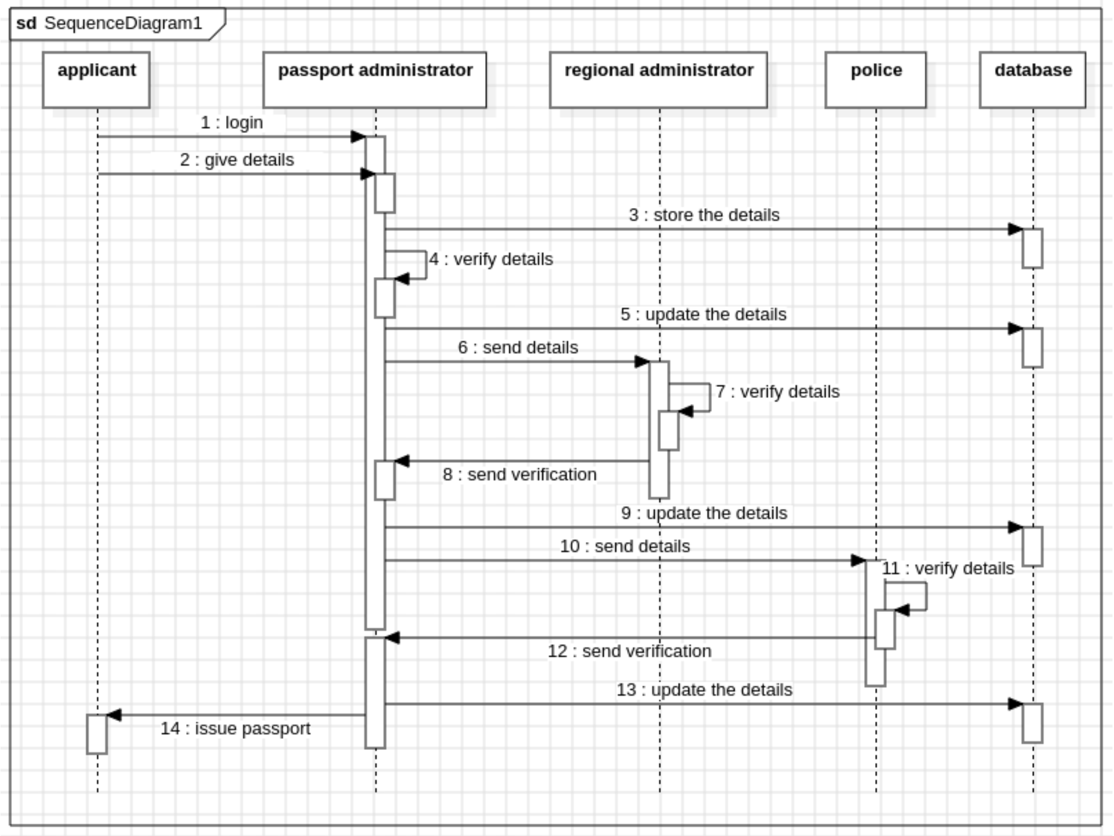


Fig. 5.4: Activity Diagram Screenshot

5.7 Activity Diagram

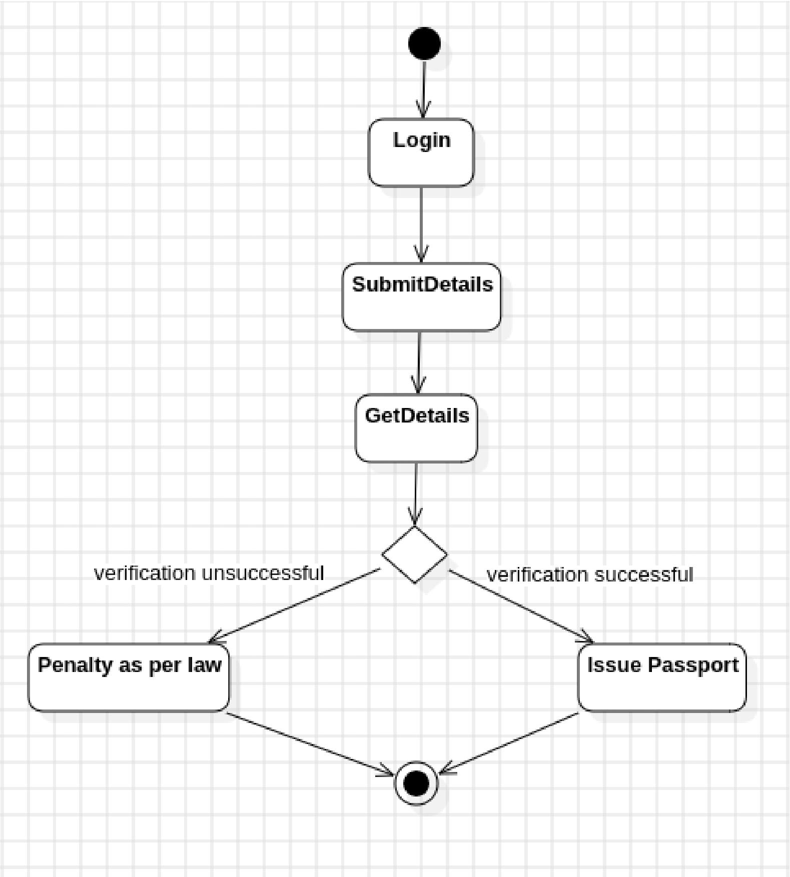


Fig. 5.5: Activity Diagram Screenshot