

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

# Object Oriented Modelling and Design

*Submitted by*

**UTSAV BOSE (1BM20CS219)**

*in partial fulfillment for the award of the degree of*  
**BACHELOR OF ENGINEERING**  
*in*  
**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**  
(Autonomous Institution under VTU)  
**BENGALURU-560019**  
**May-2022 to July-2022**

**B. M. S. College of Engineering,**  
**Bull Temple Road, Bangalore 560019**  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “**Object Oriented Modelling and Design**” carried out by **UTSAV BOSE(1BM20CS219)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Object Oriented Modelling and Design - (20CS6PCOMD)**work prescribed for the said degree.

**Dr.Pallavi G B**  
Assistant Professor  
of CSE  
BMSCE, Bengaluru

**Dr. Jyothi S Nayak**  
Professor and Head Department  
Department of CSE  
BMSCE, Bengaluru

## Index Sheet

Sl. No.	Experiment Title	Page No.
1	Hotel Management System	
2	Credit Card Processing	
3	Library Management System	
4	Stock Maintenance System	
5	Passport Automation System	
6	Railway ManagementSystem	
7	Online shopping System	

## Course Outcome

CO1	Ability to apply the knowledge of class, State & Interaction Modelling using Unified Modeling Language to solve a given problem.
CO2	Ability to analyze a System for a given requirement using Unified Modeling language.
CO3	Ability to design a given system using high level strategy.
CO4	Ability to conduct practical experiment to solve a given problem using Unified Modeling language.

# **EXPERIMENT - 01**

## **HOTEL MANAGEMENT**

### **PROBLEM STATEMENT :**

Managing a hotel can be a complex and challenging task. Hotel managers need to keep track of room availability, reservations, check-ins and check-outs, housekeeping, inventory, and customer service. They also need to ensure that their hotel complies with all relevant regulations and provides a high-quality experience for their guests. In addition, managers need to be able to monitor their hotel's financial performance and make data-driven decisions to improve profitability. It is required to design and build a hotel management system that meets these requirements and provides a seamless experience for both hotel staff and guests.

Furthermore, hotel managers need to comply with various regulations, such as fire safety codes, food safety regulations, and labor laws. They need to ensure that their staff is trained to provide high-quality customer service and that their hotel provides a safe and comfortable environment for guests.

To handle all these tasks effectively, hotel managers often use a hotel management system, which is a software application designed to streamline hotel operations and provide real-time data to managers. The system can help managers keep track of room availability, reservations, and guest requests, as well as manage housekeeping schedules and inventory.

A hotel management system can also provide financial reporting and analytics, allowing managers to monitor their hotel's revenue, expenses, and profitability. This information can be used to make data-driven decisions about pricing, promotions, and marketing strategies to improve their hotel's financial performance.

Overall, a hotel management system can help hotel managers run their hotel more efficiently, provide better customer service, and improve their bottom line.

# Software Requirement Specification(SRS)

## 1 Introduction :

**1.1 Purpose of this Document :** The purpose of this Software Requirements Specification (SRS) document is to define the functional and nonfunctional requirements for the Hotel Management System. This document outlines the features, functions, and constraints of the system.

**1.2 Scope of this document :** The Hotel Management System is intended to automate the management process for hotels. The system will allow hotels to track the room availability, reservations, check-in and check-out, housekeeping, inventory, and customer services. The system will also generate reports and provide analytics to help hotels make informed decisions.

**1.3 Overview :** A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also provides a comprehensive outline of the requirements for a Hotel Management System, and it serves as a guide for developers and stakeholders to ensure that the system meets the needs of its users.

## 2 General description :

A Hotel Management System is a software application that helps hotels manage their daily operations more efficiently. It is designed to provide a seamless experience for guests from the moment they make a reservation to the moment they check out. The system typically includes a range of features, such as room availability management, reservation management, check-in and check-out management, housekeeping management, inventory management, customer service management, and financial reporting and analytics.

By using a Hotel Management System, hotel staff can easily manage reservations, room assignments, and customer information, which reduces the chances of overbooking or double booking. It also helps to streamline the check-in and check-out process, which saves time for both guests and staff. The system provides real-time updates on room availability, which helps staff to make informed decisions and provide accurate information to guests.

### **3 Functional Requirements :**

- **User Management:** The system should allow the hotel staff to create and manage user accounts with different access levels, such as admin, manager, receptionist, and housekeeping staff.
- **Reservation Management:** The system should allow guests to make online reservations, and hotel staff to manage and track the reservation status, room availability, and room assignment.
- **Room Management:** The system should allow hotel staff to manage room inventory, assign rooms to guests, track room status, and handle room service requests.
- **Billing and Payment:** The system should generate bills for guests, handle multiple payment methods, and integrate with payment gateways.
- **Guest Management:** The system should store guest information, track guest history, and manage guest requests and complaints.
- **Housekeeping Management:** The system should allow the housekeeping staff to view the room cleaning schedule, update room status, and track housekeeping supplies.
- **Inventory Management:** The system should allow hotel staff to manage and track inventory of supplies, food, and beverages, and generate reports on inventory usage and stock levels.
- **Reports:** The system should generate various reports, including occupancy reports, revenue reports, and guest history reports.
- **Security:** The system should ensure the security of guest information and transactions, and comply with data privacy regulations.

#### **4 Interface Requirements :**

- **User Interface:** The user interface should be intuitive and user-friendly, allowing users to navigate through the system easily. The interface should have a clean and modern design, with easy-to-read fonts and color schemes that are pleasing to the eye.
- **Dashboard Interface:** The dashboard interface should provide an overview of critical information such as the occupancy rate, reservation status, and room availability. The interface should have customizable widgets that display relevant information based on the user's role and access level.
- **Reservation Interface:** The reservation interface should allow users to make and manage reservations easily. The interface should display available room types, rates, and amenities, with options to filter and sort results. Users should be able to select a room, set check-in and check-out dates, and view the total cost of the reservation.
- **Room Management Interface:** The room management interface should allow hotel staff to view and manage room inventory, assign rooms to guests, and handle room service requests. The interface should display room status, including occupied, vacant, and under maintenance. Users should be able to update room status, view room details, and assign housekeeping tasks.
- **Billing and Payment Interface:** The billing and payment interface should allow hotel staff to generate bills for guests, handle multiple payment methods, and integrate with payment gateways. The interface should display the total cost of the stay, additional charges such as room service and taxes, and provide options for splitting the bill. Users should be able to process payments, print receipts, and view payment history.
- **Guest Management Interface:** The guest management interface should allow hotel staff to view guest information, track guest history, and manage guest requests and complaints. The interface should display guest details such as name, contact information, and reservation history. Users should be able to update guest information, manage requests, and handle complaints.
- **Housekeeping Management Interface:** The housekeeping management interface should allow housekeeping staff to view the room cleaning schedule, update room status, and track housekeeping supplies. The interface should display the room status, cleaning schedule, and

room details. Users should be able to update room status, view cleaning instructions, and manage inventory.

- **Inventory Management Interface:** The inventory management interface should allow hotel staff to manage and track inventory of supplies, food, and beverages, and generate reports on inventory usage and stock levels. The interface should display inventory items, stock levels, and usage history. Users should be able to update inventory levels, view usage reports, and manage supply orders.

## **5 Performance Requirements :**

- **Response Time:** The hotel management system should have a fast response time to ensure users can perform tasks quickly. The system should respond to user requests within a reasonable time frame, typically less than 3 seconds.
- **System Availability:** The hotel management system should be available to users at all times. The system should have a high uptime percentage, typically greater than 99%, to ensure users can access the system when needed.
- **Scalability:** The hotel management system should be scalable, allowing it to handle an increasing number of users and data without performance degradation. The system should be able to handle peak loads during high demand periods such as holidays and events.
- **Concurrent User Capacity:** The hotel management system should be able to handle a high number of concurrent users without performance degradation. The system should be able to handle at least 100 users simultaneously without affecting response time.
- **Data Processing Speed:** The hotel management system should have fast data processing speed to ensure users can perform tasks quickly. The system should be able to handle large data volumes without affecting response time.
- **Security:** The hotel management system should be secure to ensure data confidentiality, integrity, and availability. The system should use encryption to protect sensitive data, have access control mechanisms to restrict unauthorized access, and have a backup and recovery plan in case of data loss.



- Integration: The hotel management system should be able to integrate with other systems, such as payment gateways, property management systems, and customer relationship management systems. Integration ensures seamless data exchange and reduces manual data entry, leading to better system performance.

## **6 Design Constraints :**

- Security: The system must have robust security features to protect guest information and maintain privacy. This includes authentication, authorization, and access control mechanisms.
- Scalability: The system must be scalable to support a large number of users and guests. It should be able to handle peak loads during high season periods without performance degradation.
- Reliability: The system should be highly reliable and available 24/7. It must have mechanisms to handle system failures and recover quickly from them.
- Usability: The system should be easy to use for both staff and guests. The interface should be intuitive, and the system should be designed with a user-centered approach.
- Integration: The system should be able to integrate with other hotel systems such as billing, accounting, inventory, and guest services. It should be able to exchange data seamlessly with these systems.
- Accessibility: The system should be accessible to users with disabilities. It should comply with accessibility standards such as WCAG 2.1 to ensure that everyone can use the system.
- Compliance: The system must comply with local and international regulations such as GDPR, PCI-DSS, and HIPAA. It should also comply with industry standards such as ISO 27001.
- Performance: The system must have high performance to support real-time operations such as room reservations, check-ins, check-outs, and housekeeping. It should be able to process requests quickly and efficiently.
- Customization: The system should be customizable to meet the unique needs of each hotel. It should allow hotels to configure settings, policies, and workflows according to their requirements.

- Cost-effectiveness: The system should be cost-effective and provide a good return on investment for the hotel. It should have a low total cost of ownership and be able to generate revenue for the hotel through upselling and cross-selling.

### **7 Non-Functional Attributes :**

Usability: The system should be user-friendly and easy to learn, with a simple and intuitive interface.

Performance: The system should be able to handle a large number of users and transactions, and respond quickly to user requests.

Reliability: The system should be reliable and available 24/7, with minimal downtime for maintenance and updates.

Scalability: The system should be scalable to meet the needs of hotels of different sizes and types, and able to handle future growth and expansion.

Security: The system should be secure and protect sensitive information from unauthorized access or data breaches.

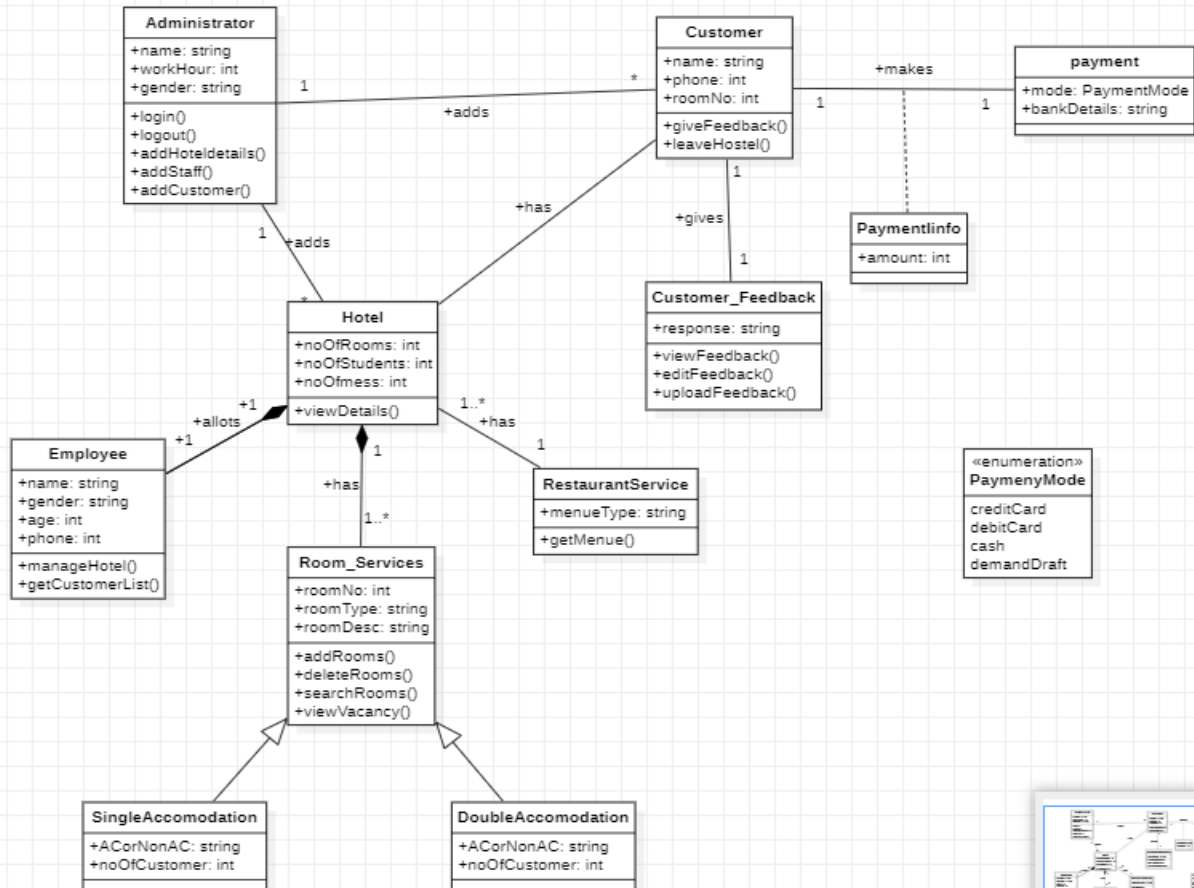
Compatibility: The system should be compatible with different hardware and software platforms, and support integration with third-party applications and services

### **8 Preliminary Schedule and Budget :**

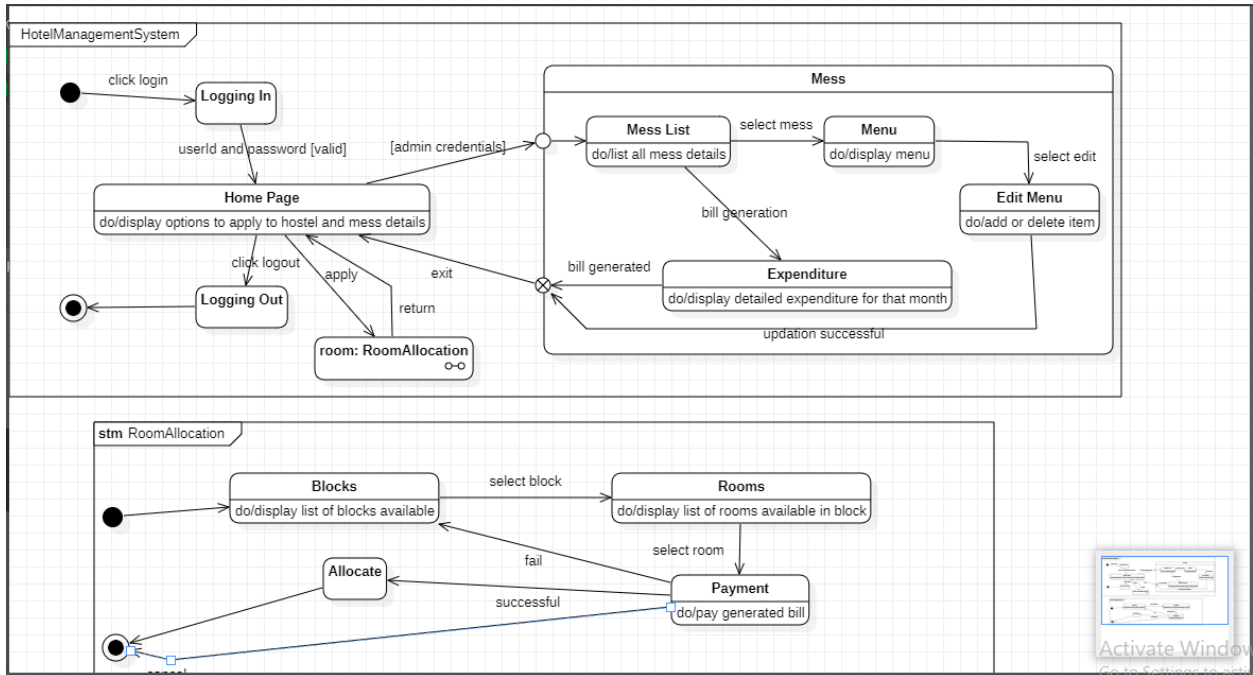
BUDGET : 50,000 SCHEDULE :

- Requirements gathering: 2 weeks
- System design: 3 weeks
- Software development: 12 weeks
- Hardware procurement and setup: 2 weeks
- Testing and quality assurance: 4 weeks
- Training and documentation: 2 weeks
- Deployment and go-live: 2 weeks

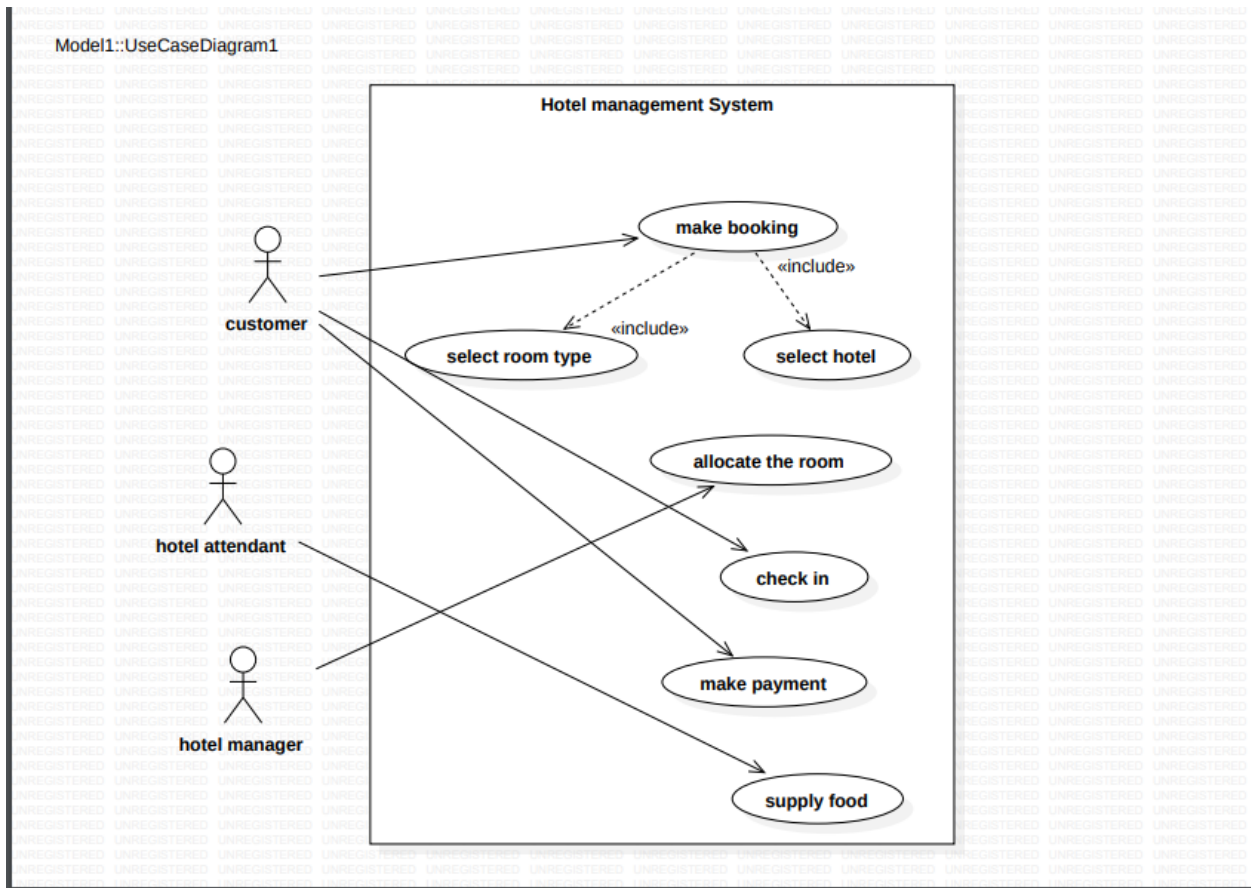
# CLASS DIAGRAM



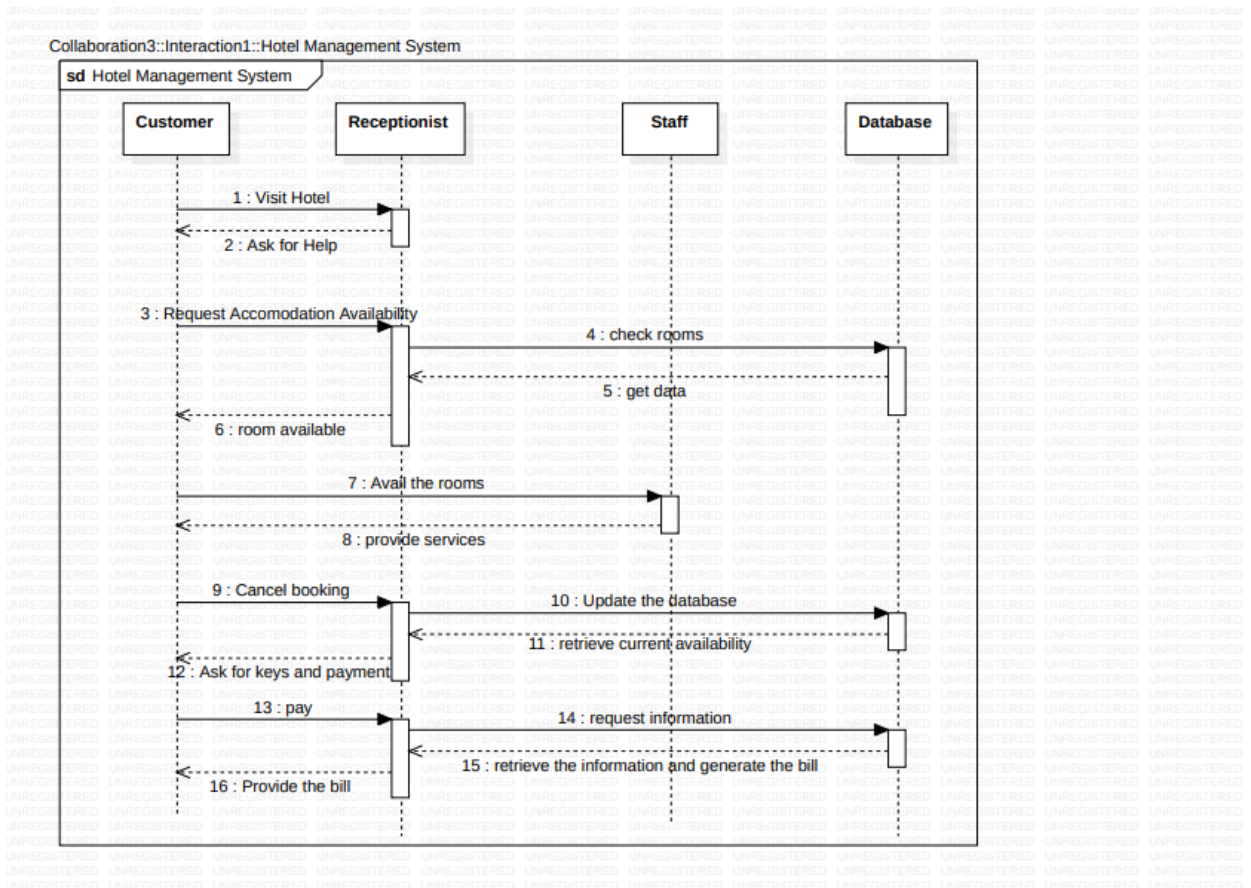
# STATE DIAGRAM



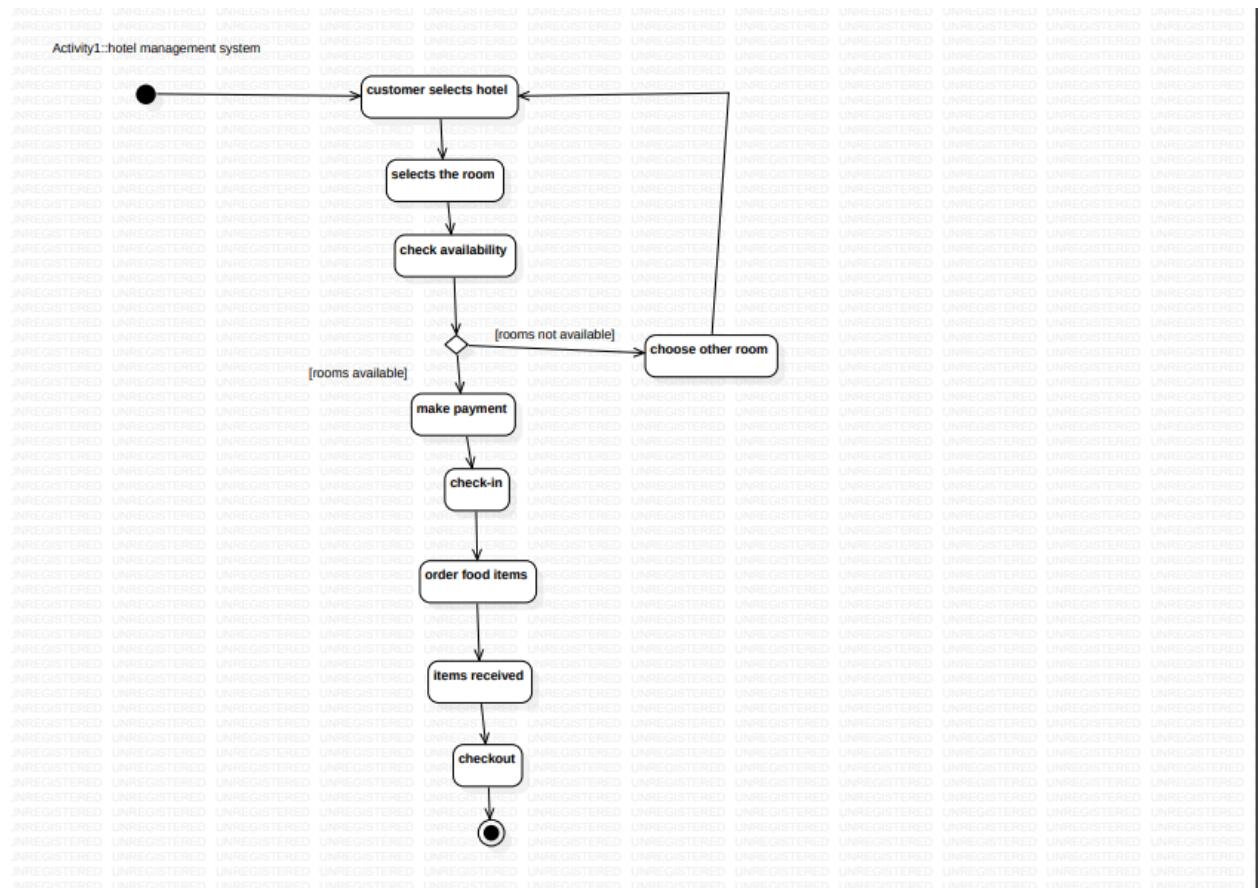
# USE CASE DIAGRAM



# SEQUENCE DIAGRAM



# ACTIVITY DIAGRAM



# **Experiment – 02**

## **Credit Card Processing**

### **2.1 PROBLEM STATEMENT:**

Credit card processing through offline involves the merchant collecting order information storing this in a database on your site, and entering it using their on-site merchant credit card processing system. Takes time to manually enter credit card information for each order.

This solution creates following cons: ·

Insecure – there is a possibility that a skilled hacker could break into the database and steal an entire list of credit card numbers, thereby damaging the merchant’s reputation with current client.

- There is a higher risk of customer charge backs with no signature
- Higher risk of fraud for using stolen credit cards
- Many discerning online shoppers will not give their credit card to an “untrusted” online.

So there is a need of online and trusted credit card processing.

### **2.2 SOFTWARE REQUIREMENT SPECIFICATION:**

#### **2.2.1 INTRODUCTION**

The Credit Card Processing System is designed to facilitate the processing of credit card transactions securely and efficiently. This system will be used by merchants and financial institutions to process credit card transactions in real-time. This document outlines the requirements and specifications for the Credit Card Processing System.

#### **PURPOSE OF THE DOCUMENT**

When customers complete their shopping cart, their credit card is preauthorized and the order is entered into Sales Order. Credit Card Processing dials out and obtains a credit card payment. Within five minutes the customer receives an e-mail receipt.



## **SCOPE OF THE DOCUMENT**

- Automatically connects to your financial network for credit card authorizations and settlements
- Integrates with Sales Order, Accounts Receivable, and e-Business Manager
- Support for dial-up (modem) connections or secure Internet connections through TCP/IP and SSL
- Compliant with Visa and MasterCard Electronic Commerce Indicator (ECI) regulations
- Multiple address verification options available

## **OVERVIEW**

Credit card processing through offline involves the merchant collecting order information, storing this in a database on your site, and entering it using their on-site merchant credit card processing system. Takes time to manually enter credit card information for each order. This solution creates following cons:

- Insecure – there is a possibility that a skilled hacker could break into the database and steal an entire list of credit card numbers, thereby damaging the merchant’s reputation with current client.
- There is a higher risk of customer charge backs with no signature
- Higher risk of fraud for using stolen credit cards
- Many discerning online shoppers will not give their credit card to an “untrusted” online merchant

### **2.2.2 GENERAL DESCRIPTION**

A credit card processing system is a software-based platform that facilitates the electronic authorization, verification, and settlement of credit card transactions. It enables merchants to process credit card payments securely and efficiently from customers using their credit or debit cards. The credit card processing system works by securely transmitting customer payment information to the card issuer or the card network, which then approves or declines the transaction based on factors such as available credit, fraud risk, and other security checks. Once approved, the payment is settled and funds are transferred from the customer's account to the merchant's account.

### **2.2.3 FUNCTIONAL REQUIREMENTS**

1.     Payment Processing: The system should be able to process credit card payments and verify that the card is valid and has sufficient funds for the transaction. The system should be able to handle transactions in multiple currencies.
2.     User Authentication: The system should require customers to authenticate themselves before making a transaction. The system should be able to verify the customer's identity and the authenticity of the credit card being used.
3.     Transaction Monitoring: The system should provide real-time transaction monitoring and reporting to both customers and merchants. The system should be able to identify suspicious activity and flag potential fraud.
4.     Integration with E-commerce Platforms: The system should be easy to integrate with various e-commerce platforms such as Shopify, WooCommerce, and Magento. The integration should be seamless and require minimal effort from the merchant.
5.     Secure Transactions: The system should be designed to ensure the security of all transactions. The system should encrypt all data transmitted between the customer, merchant, and the payment gateway. The system should comply with industry-standard security protocols such as PCI DSS.
6.     Refund Management: The system should allow merchants to process refunds for transactions. The system should be able to process partial and full refunds and update the customer's account accordingly.

### **2.2.4 INTERFACE REQUIREMENTS**

Interface Requirements: The credit card processing system shall be designed with a user- friendly interface to facilitate easy and efficient use by the end-user. The following are the interface requirements:

1.     Login Interface: The login interface shall provide a secure and reliable means of access to the system. Users shall be required to provide their unique login credentials before accessing the system.

2.     Dashboard Interface: The dashboard interface shall display a summary of the user's account, including available credit, transaction history, and pending payments.
3.     Payment Interface: The payment interface shall facilitate secure and reliable credit card payments. Users shall be required to provide their credit card information, including card number, expiry date, and CVV code, to complete the payment.
4.     Transaction Interface: The transaction interface shall display a detailed summary of each transaction, including the date and time of the transaction, the amount charged, and the transaction status.
5.     Report Interface: The report interface shall provide users with the ability to generate reports on their credit card usage, including spending patterns, transaction history, and account balances.
6.     Settings Interface: The settings interface shall allow users to configure their account settings, including personal information, billing preferences, and notification settings.
7.     Help and Support Interface: The help and support interface shall provide users with access to documentation, FAQs, and support channels to assist with any issues they may encounter while using the system.

### **2.2.5 PERFORMANCE REQUIREMENTS**

1.     Response Time: The system should have a maximum response time of 2 seconds for all transactions. This includes the time taken to verify the user's identity, authenticate the credit card, and process the transaction.
2.     Transaction Volume: The system should be capable of handling at least 1000 transactions per second during peak times. This includes processing, authorization, and settlement of transactions.
3.     System Availability: The system should have a minimum uptime of 99.99%. This means the system should be available and accessible to users for at least 99.99% of the time.

4. Network Latency: The system should have a maximum network latency of 100 milliseconds for all transactions. This includes the time taken to transmit data between the user's device and the system.
5. Data Throughput: The system should be capable of processing at least 10 GB of data per day. This includes transaction data, user data, and system logs.
6. Concurrent Users: The system should be capable of handling at least 10,000 concurrent users at peak times. This includes users accessing the system via mobile devices, web browsers, and other applications.
7. Scalability: The system should be scalable to handle increasing transaction volumes and user loads. The system should be able to add additional resources, such as servers or processing power, to accommodate increased demand.
8. Error Rates: The system should have a low error rate of less than 0.01%. This includes errors such as failed transactions, timeouts, and system failures.
9. Security: The system should be secure and comply with industry standards such as PCI DSS. This includes protecting user data, preventing fraud, and ensuring system integrity.

#### **2.2.6 NON-FUNCTIONAL REQUIREMENTS**

1. Reliability: The system should be designed to ensure high availability and reliability. The system should be able to handle large volumes of transactions without downtime.
2. Performance: The system should be able to process transactions quickly and efficiently. The system should have low response times and be able to handle high levels of traffic.
3. Scalability: The system should be able to scale horizontally and vertically to handle increasing transaction volumes. The system should be able to add additional resources as needed without downtime.
4. User-Friendly: The system should be designed to be user-friendly and easy to use for both customers and merchants. The system should have a simple and intuitive interface that is easy to navigate.

### **2.2.7 DESIGN CONSTRAINTS**

1. Security: The system must comply with Payment Card Industry Data Security Standards (PCI-DSS) and other relevant security regulations. The system must encrypt all sensitive data, including credit card numbers, and store it securely.
2. Scalability: The system must be scalable to handle a large volume of transactions. The system must be designed to handle peak loads and scale up or down as needed.
3. Availability: The system must be highly available and have a reliable failover mechanism in place to ensure that transactions can be processed even in the event of a failure.
4. Compatibility: The system must be compatible with different types of credit cards and payment gateways. The system must also be compatible with different types of merchant accounts.
5. Performance: The system must be designed for optimal performance, with response times that meet or exceed industry standards.
6. User Interface: The user interface must be intuitive and easy to use, with clear navigation and easy-to-understand instructions. The system must also be accessible to users with disabilities.

### **2.2.8 PRELIMINARY SCHEDULE AND BUDGET**

#### **Schedule:**

Requirements gathering and analysis - 2 weeks Design and architecture - 4 weeks Development and coding - 12 weeks

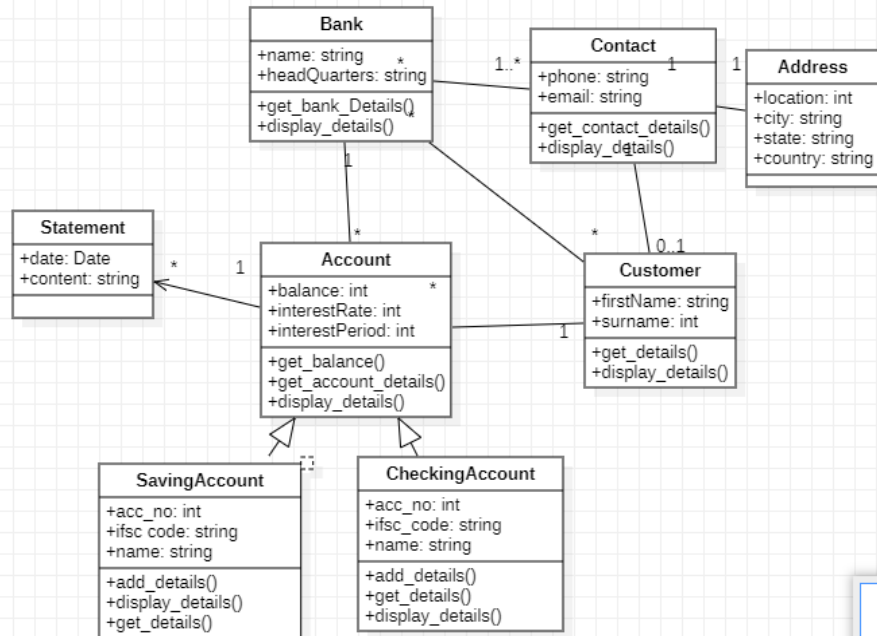
Testing and quality assurance - 4 weeks Deployment and training - 2 weeks

Total time estimate: 24 weeks or 6 months Budget:

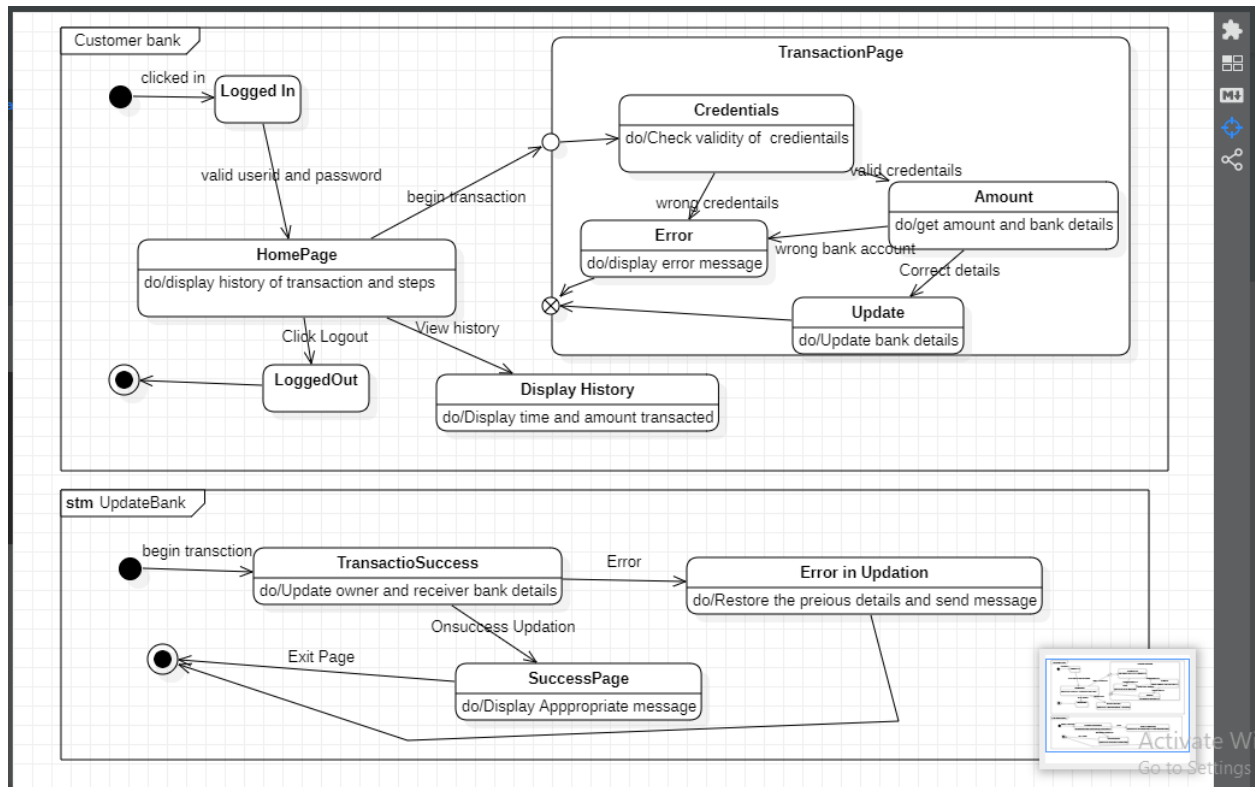
Salaries for project team (developers, testers, designers, project manager) - \$400,000 Hardware and software infrastructure - \$50,000

Third-party payment gateway integration - \$20,000 Miscellaneous expenses (travel, training, etc.) - \$30,000 Total budget estimate: \$500,000

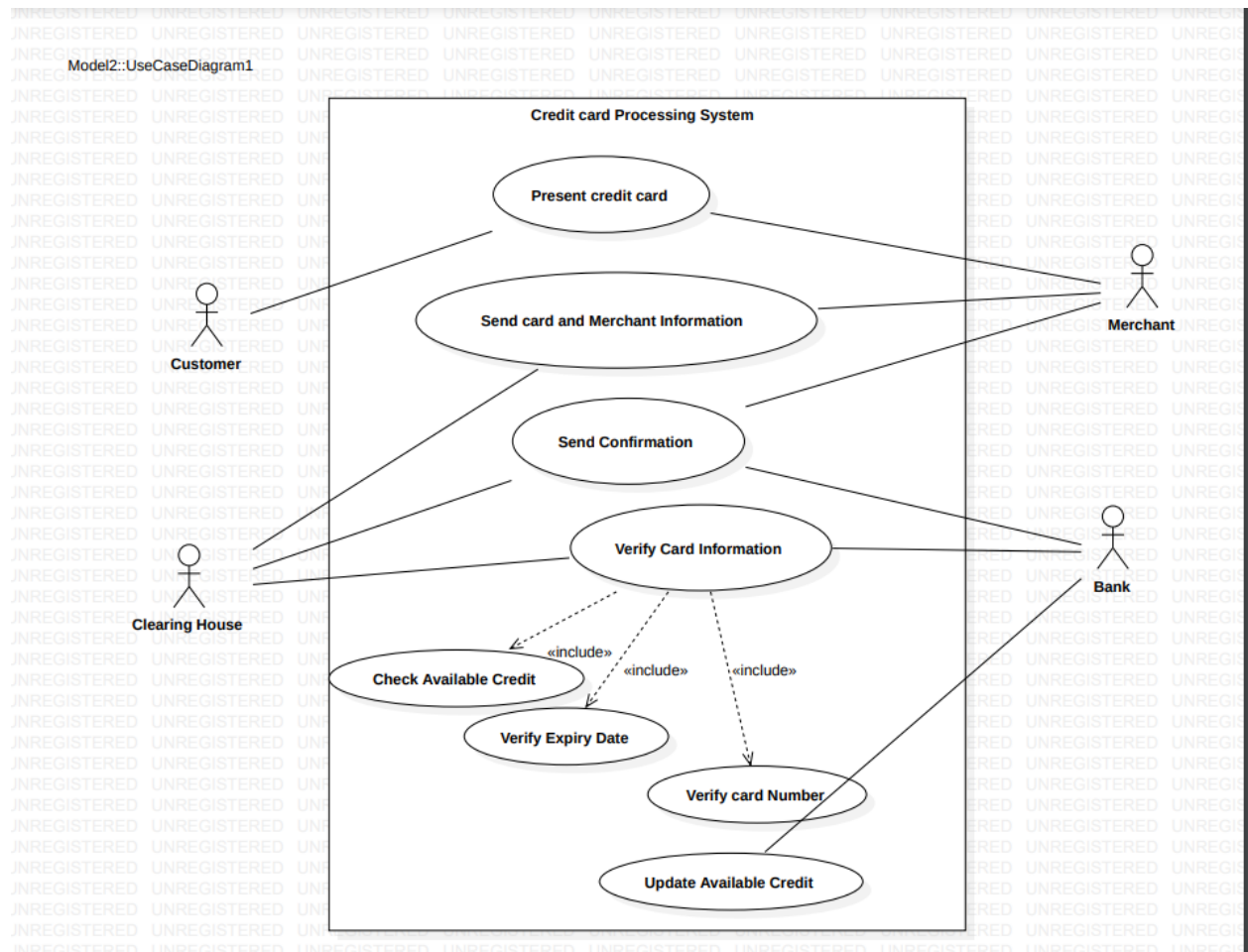
# CLASS DIAGRAM



# STATE DIAGRAM

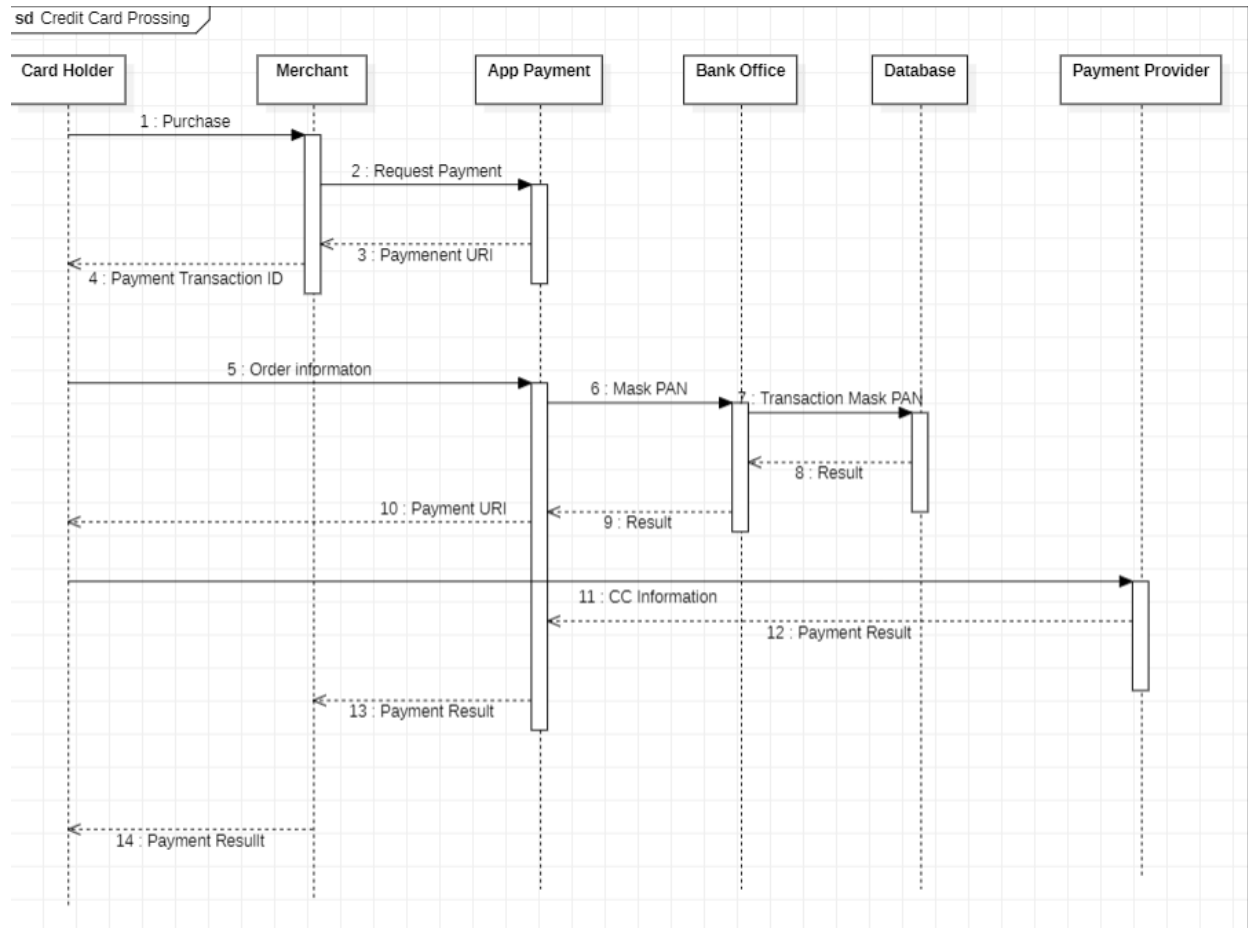


# USE CASE DIAGRAM

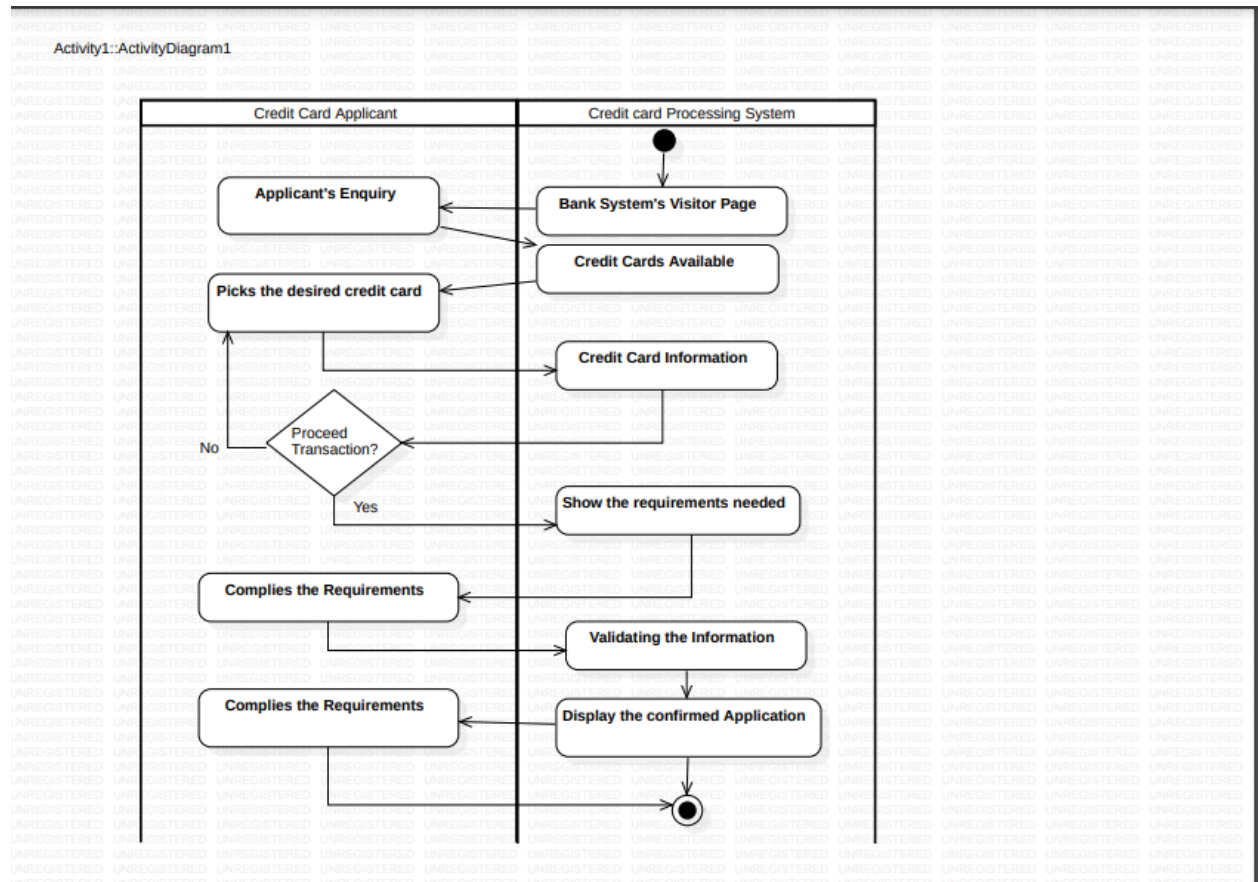




# SEQUENCE DIAGRAM



# ACTIVITY DIAGRAM



## **EXPERIMENT - 03**

### **LIBRARY MANAGEMENT SYSTEM**

#### **PROBLEM STATEMENT:**

A library management system is a software solution designed to manage and automate the day-to-day operations of a library. The problem statement for such a system can be

summarized as follows:

Libraries are essential institutions that serve as resources for learning, research, and entertainment. However, managing a library can be a complex and time-consuming task, particularly when it comes to tasks such as cataloging, tracking inventory, managing loans, and handling fines and fees. These tasks can be overwhelming for library staff and can lead to inefficiencies, errors, and delays.

To address these challenges, a library management system is needed to streamline and automate many of these processes. The system should be user-friendly, reliable, and efficient, enabling library staff to easily manage their collections, track inventory, and handle loans, returns, and fines. Additionally, the system should provide patrons with a seamless experience, allowing them to search for and check out books, access digital resources, and manage their accounts online.

The ultimate goal of a library management system is to improve the overall efficiency and effectiveness of the library, enhancing its ability to serve its patrons and fulfill its mission as a vital community resource.

#### **SOFTWARE REQUIREMENT SPECIFICATION(SRS)**

##### **1 Introduction:**

**1.1 Purpose of this Document** - The purpose of this Software Requirement Specification (SRS) document is to describe the functional and nonfunctional requirements of the Library Management System (LMS). This system is designed to manage the operations of a library including book cataloguing, book borrowing and returning, user management, and generating reports.

**1.2 Scope of this document** – The Library Management System will allow librarians to manage the library resources and users efficiently. The system will be able to manage book information, user information, and lending information. The system will be available for use by library staff and registered users. The system will have the following functionality: Book cataloging, Book borrowing and returning, User management, Report generation

**1.3 Overview** – A library management system is a software application designed to automate and manage the tasks and operations of a library. The main purpose of a library management system is to help librarians manage library resources more efficiently and effectively.

The system typically includes modules for cataloging, circulation, acquisitions, and administration. The cataloging module is used to manage the library's collection, including books, periodicals, media, and other materials. It allows librarians to add, modify, and delete items in the collection and track their availability and location.

The circulation module handles the loaning and returning of library materials. It tracks the status of borrowed items, manages fines and fees, and generates reports on circulation activity. The acquisitions module is used to manage the procurement of new materials for the library. It tracks purchase orders, invoices, and budgets, and helps librarians manage their collection development process.

Overall, a library management system is a critical tool for librarians to manage the vast amount of information and resources in their libraries and provide efficient and effective services to their patrons.

## **2 General description:**

A library management system is a software solution that helps libraries manage their day-to-day operations. It is designed to automate many of the tasks that librarians perform manually, such as cataloging, inventory management, circulation, and patron management. The system can be used by librarians and staff to manage the library's collection, track books and other resources, and handle circulation tasks such as checking out and returning items. A typical library management system will have several modules or components that work together to provide a comprehensive solution. These modules may include: Cataloging, Circulation, Patron management, Reporting, Digital resources.

### **3 Functional Requirements:**

A library management system is a software application that helps to manage the operations of a library. The functional requirements of a library management system include:

- **User Management:** The system should allow the librarian to create and manage user accounts. This includes registering new users, updating user information, and deleting user accounts.
- **Book Management:** The system should allow the librarian to manage the books in the library. This includes adding new books, updating book information, and removing books from the library.
- **Cataloging:** The system should provide a catalog of books in the library that can be easily searched by the users. The catalog should contain information such as the author, title, publication date, ISBN, and availability status of each book.
- **Circulation:** The system should manage the circulation of books in the library. This includes checking out and returning books, maintaining a record of who has borrowed each book, and managing fines for overdue books.
- **Reservation:** The system should allow users to reserve books that are currently checked out. The system should also notify users when reserved books become available.
- **Reporting:** The system should provide reports on various aspects of library operations, such as circulation statistics, overdue books, and popular books.
- **Security:** The system should have appropriate security measures in place to ensure that user information and library data are protected from unauthorized access.
- **Interlibrary Loan:** The system should allow users to request books from other libraries in case the required book is not available in the library.
- **Database Management:** The system should have a robust database management system to store and retrieve information in an efficient manner.
- **Integration:** The system should be able to integrate with other systems used by the library, such as financial systems or RFID systems used for book tracking.

**4 Interface Requirements:** Interface requirements of a library management system are related to the user interface design and how it interacts with the users. Some of the key interface requirements for a library management system include:

- User-friendly interface: The system should have an easy-to-use interface that can be easily navigated by users. It should have a simple and intuitive design that enables users to quickly and easily locate the information they need.
- Search functionality: The system should provide a powerful search feature that allows users to search for books by various criteria such as author, title, keyword, publication date, and subject. The search results should be presented in a clear and organized manner.
- Book details display: The system should display the detailed information about each book in a clear and concise manner. This includes the book cover image, author, title, publication date, ISBN, and availability status.
- Borrowing and returning books: The system should have a simple and easy-to-use interface for borrowing and returning books. Users should be able to check out books, renew them, and return them with minimal effort.
- Notifications: The system should provide notifications to users about the status of their library account, such as overdue books, reserved books, and pending fines.
- Accessibility: The system should be accessible to all users, including those with disabilities. It should comply with accessibility standards and provide features such as screen readers and keyboard navigation.
- Multilingual support: The system should provide multilingual support to cater to users from different regions and speaking different languages. It should provide an option to switch the language of the interface.
- Mobile compatibility: The system should be compatible with mobile devices such as smartphones and tablets, providing a mobile-friendly interface for users on the go.
- Customization: The system should provide customization options that allow users to adjust the interface according to their preferences, such as changing the font size or color scheme.

- Integration: The system should be designed to integrate with other systems used by the library, such as RFID systems or payment systems, providing a seamless user experience.

5 Performance Requirements: Performance requirements for a library management system would depend on the specific needs and objectives of the library. However, here are some general performance requirements that could be considered:

- Response time: The library management system should be able to respond quickly to user requests. This includes searching for books, checking out books, and other transactions.
- Concurrent users: The system should be able to handle a large number of users simultaneously. Libraries can have a high volume of users during peak hours, and the system should be able to handle this without slowing down.
- Scalability: The system should be scalable, allowing for future growth and expansion of the library's collection and user base. This means that the system should be able to handle increasing amounts of data and users without affecting performance.
- Reliability: The system should be reliable, with minimal downtime and errors. This is important to ensure that users can access the system when they need to and that library staff can perform necessary tasks without interruption.
- Security: The system should be secure, protecting the library's data and the privacy of its users. This includes ensuring that only authorized users can access the system and that user data is encrypted and protected.
- Data storage and retrieval: The system should be able to efficiently store and retrieve large amounts of data, such as information about the library's collection and user data.
- Reporting and analytics: The system should be able to generate reports and analytics on library usage, such as the number of books checked out, popular books, and other metrics. This is important for library staff to make informed decisions about the library's collection and operations.

**6 Design Constraints:** There can be several design constraints for a library management system, such as:

- **Budget:** The budget allocated for the development and implementation of the system can be a major constraint. The design must be cost-effective and not exceed the budget.
- **Hardware and Software:** The design must take into account the hardware and software limitations of the library's infrastructure. For example, if the library has old computers, the system should be designed to work with them, without requiring significant hardware upgrades.
- **Interoperability:** The library management system may need to interact with other systems, such as the library's website, cataloging systems, or other third-party software. The design must ensure that the system is compatible with these other systems and can integrate with them seamlessly.
- **Accessibility:** The system must be accessible to all users, including those with disabilities. The design must take into account the accessibility guidelines and ensure that the system is accessible to users with different abilities.
- **Security:** The system must be secure and protect the privacy of user data. The design must take into account security measures such as encryption, access control, and data backup.
- **User Experience:** The design must provide an intuitive and user-friendly experience for library staff and patrons. The system should be easy to navigate, and the design should take into account user feedback and input to improve the user experience.
- **Time Constraints:** The design must take into account the time constraints for the development and implementation of the system. The design should prioritize the most important features and functionalities to ensure that the system can be implemented within the specified timeframe.

Overall, the design constraints for a library management system can vary depending on the library's specific needs and constraints. It is important to consider these constraints during the design phase to ensure that the system meets the library's requirements and can be implemented within the specified budget and timeframe.



## **7 Non-Functional Attributes:**

Non-functional requirements for a library management system might include:

- Performance: The system should be able to handle a large number of simultaneous users, and should respond quickly to user requests.
- Availability: The system should be available 24/7, with minimal downtime for maintenance and upgrades.
- Security: The system should be secure, with appropriate access controls, encryption, and authentication mechanisms to protect user data.
- Reliability: The system should be reliable, with a low probability of failure or errors.
- Scalability: The system should be able to handle growth in terms of both users and data.
- Maintainability: The system should be easy to maintain, with clear documentation and well-structured code.
- Usability: The system should be easy to use, with a user-friendly interface and clear instructions for common tasks.
- Compatibility: The system should be compatible with a range of hardware and software platforms.
- Interoperability: The system should be able to communicate with other systems, such as library catalog systems or databases.
- Accessibility: The system should be accessible to users with disabilities, with appropriate support for assistive technologies.

## **8 Preliminary Schedule and Budget: Preliminary Schedule:**

- Requirements Gathering: 2 weeks
- System Design: 4 weeks
- Development: 12 weeks
- Testing and Quality Assurance: 4 weeks

- Deployment: 2 weeks
- User Training and Documentation: 1 week
- Total Time: 25 weeks

**Preliminary Budget:**

- Salaries and Wages: ₹5,00,000
- Hardware and Software: ₹50,000
- Testing and Quality Assurance: ₹25,000
- User Training and Documentation: ₹10,000
- Contingency (10% of total budget): ₹58,500
- Total Budget: ₹6,43,500

# CLASS DIAGRAM

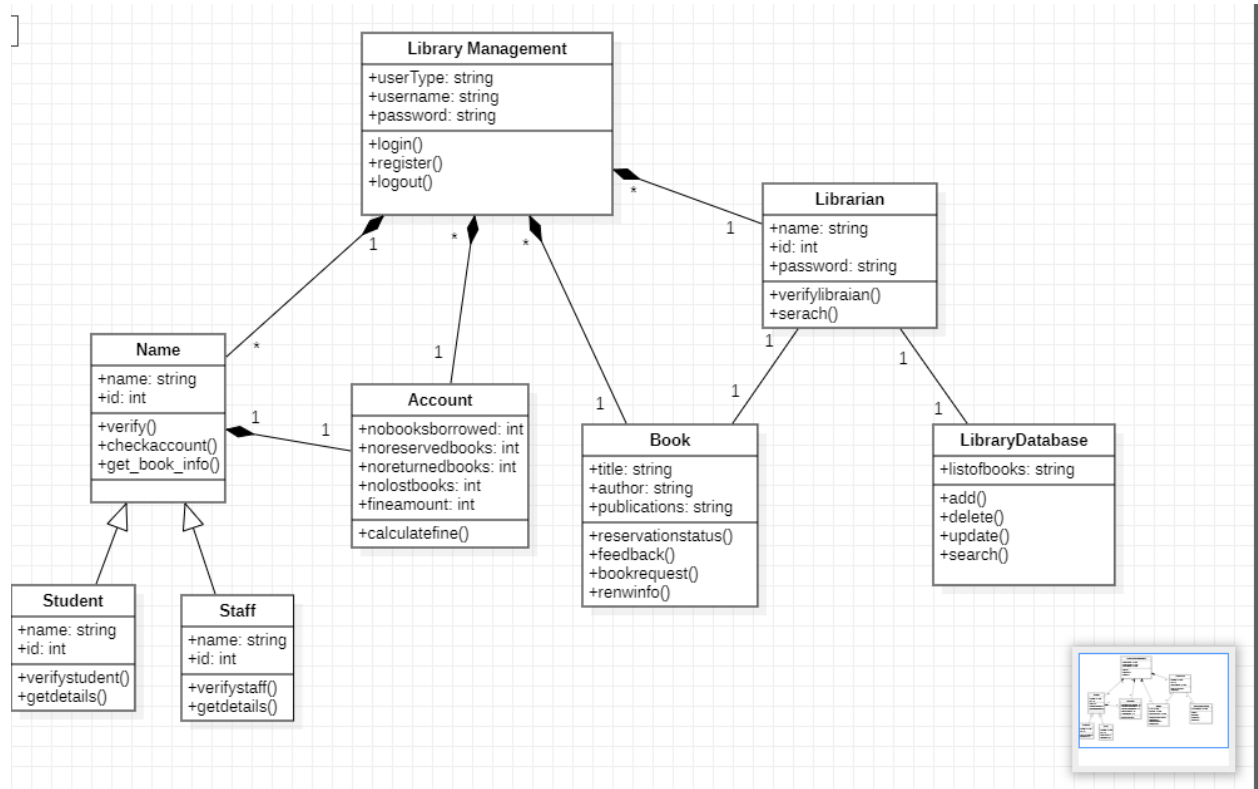
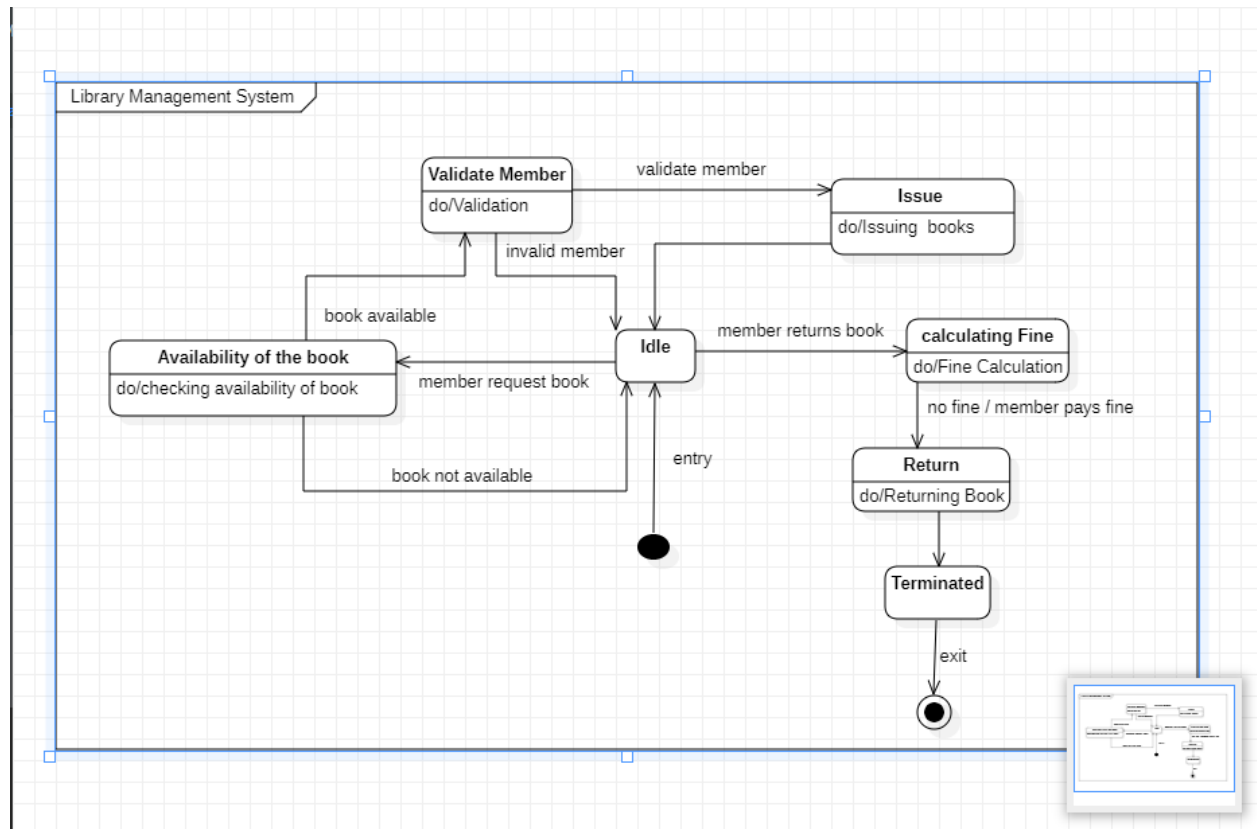


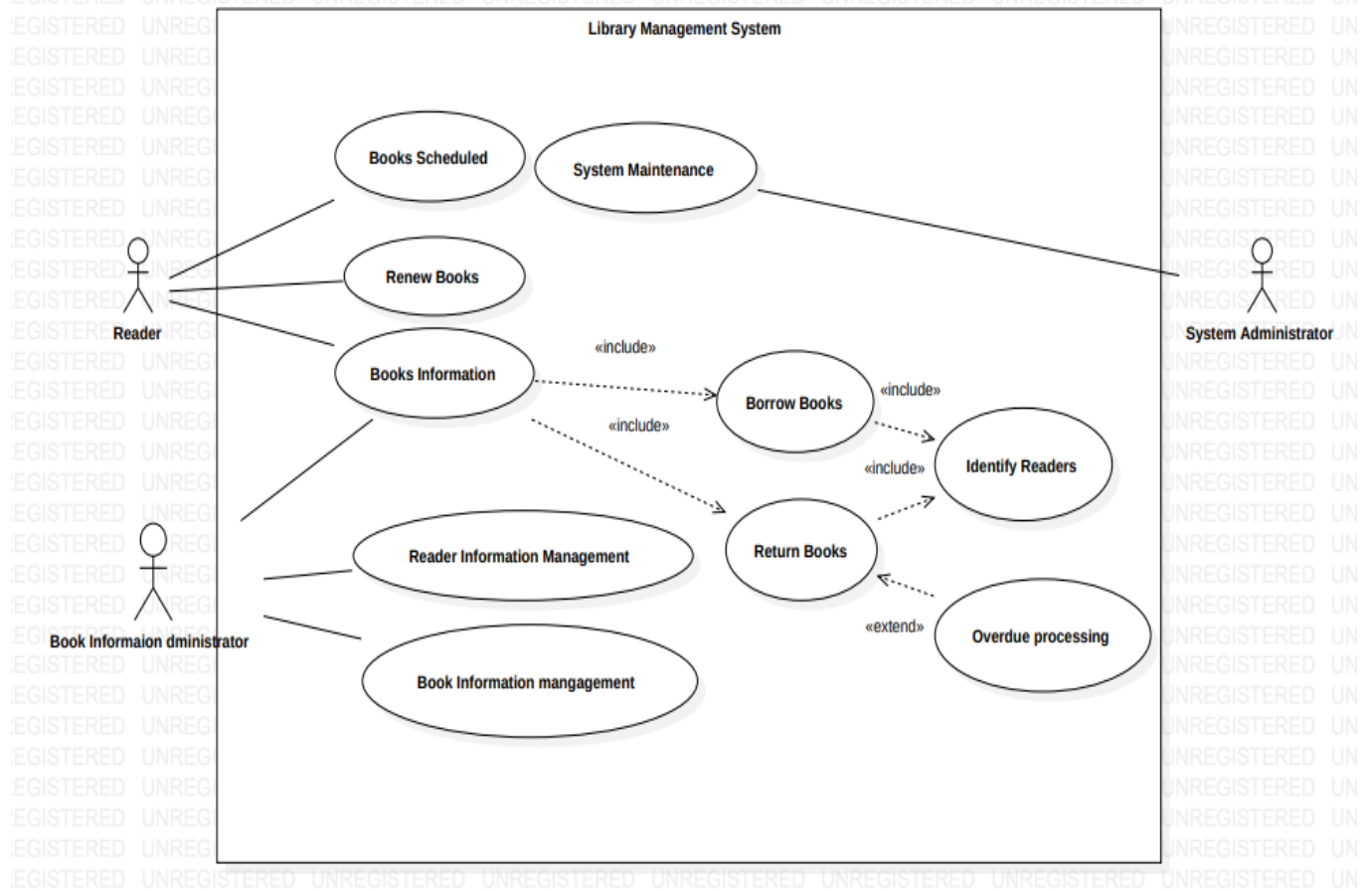
FIG. LIBRARY MANAGEMENT SYSTEM

# STATE DIAGRAM

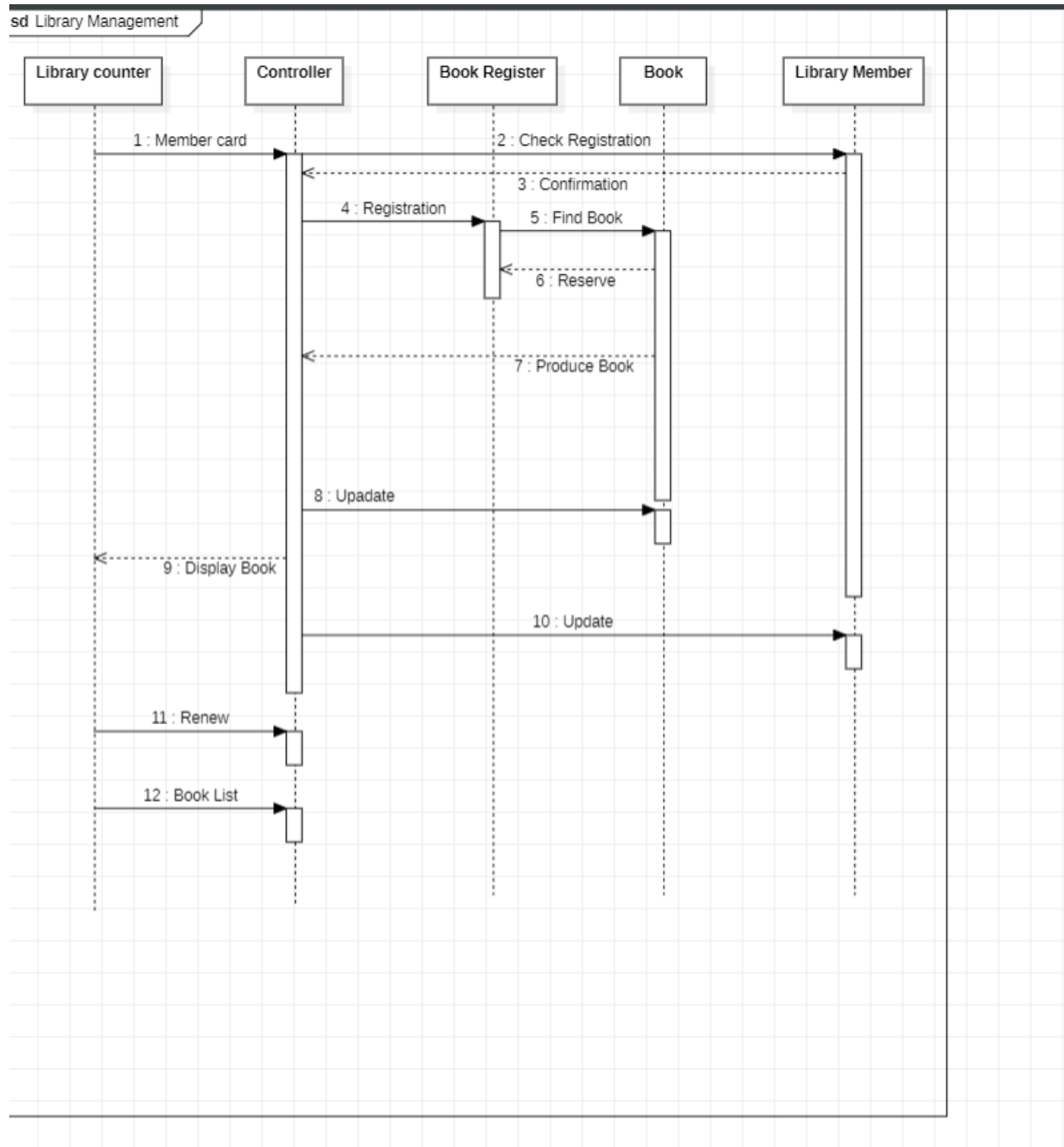


# USE CASE DIAGRAM

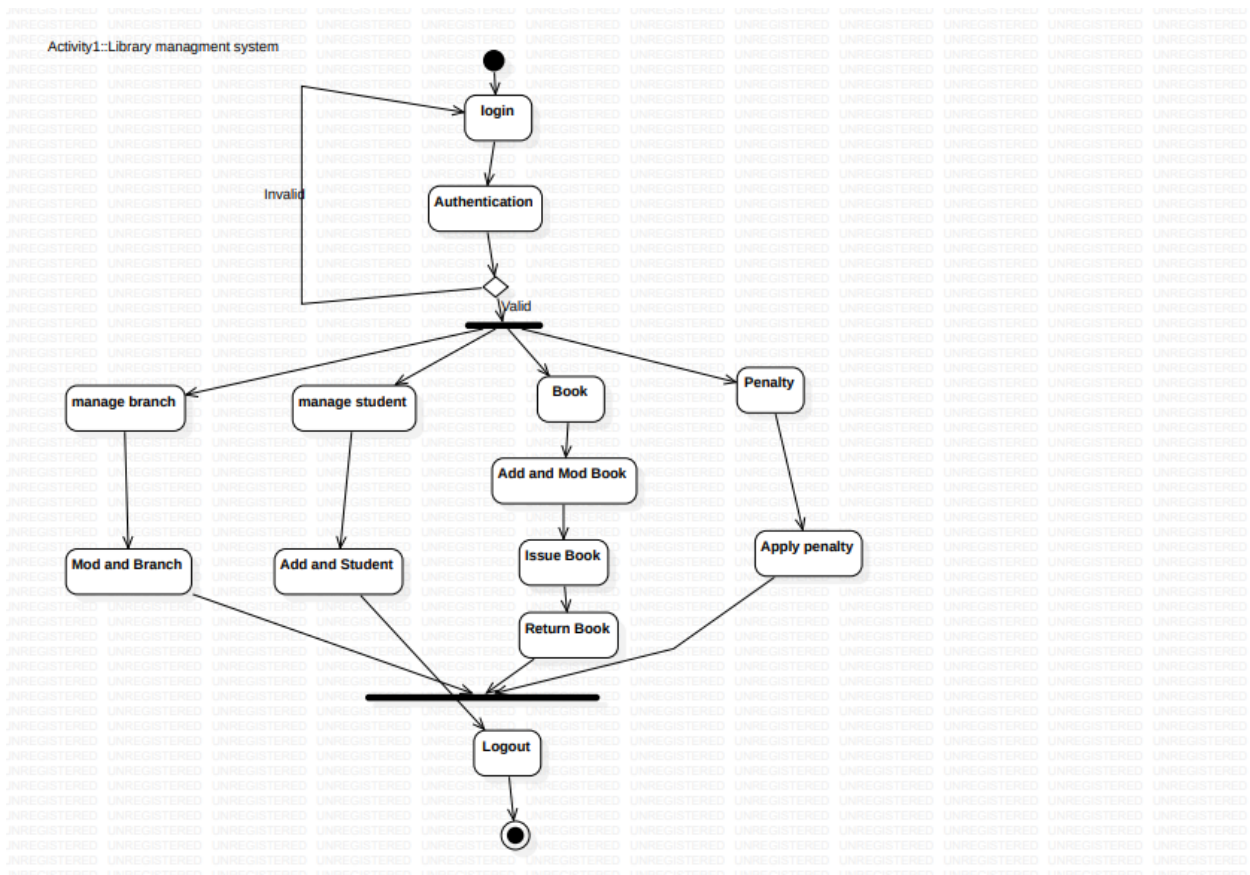
Model1: Library Management System



# SEQUENCE DIAGRAM



# ACTIVITY DIAGRAM



## **EXPERIMENT - 04**

### **STOCK MAINTENANCE SYSTEM**

#### **PROBLEM STATEMENT :**

The retail industry faces a challenge in managing inventory levels and ensuring stock availability. The current system of manual tracking of stock levels is not only time-consuming but also prone to errors. This creates issues like overstocks, stockouts, and missed sales opportunities, which affect the business's profitability. To overcome these issues, a stock maintenance system is needed that can automate the inventory management process and provide real-time information on stock levels, reorder points, and lead times. The system should also be able to track the movement of stock items, manage purchase orders, and generate reports on sales and inventory levels. The system should integrate with the point-of-sale system and accounting software to ensure accurate and up-to-date records. The business wants to improve the efficiency of the inventory management process and reduce the risk of stockouts and overstocks. The system should be user-friendly and accessible to all staff members who need to access it.

Implementing a stock maintenance system is critical for the business as it will help in reducing costs, improving customer satisfaction, and increasing sales. The system will allow the business to accurately track inventory levels and ensure that stock is available when customers want to buy it. By avoiding stockouts and overstocks, the business can increase revenue, reduce storage costs, and improve profitability. The system will also provide insights into inventory trends and demand patterns, allowing the business to make informed decisions about stock ordering and pricing. The system will save time and effort for staff members who are currently involved in manual tracking of inventory levels. By automating the process, they can focus on other tasks that require human intervention. The system will be a valuable asset to the business, enabling it to streamline operations and achieve greater efficiency.

#### **Software Requirement Specification(SRS)**

##### **1. Introduction :**

1.1. Purpose of this Document: The SRS document outlines the necessary features and functionalities that the stock maintenance system should possess, such as the ability to add and



manage products, track stock levels, receive stock alerts, generate stock reports, and more. Additionally, the document outlines any design constraints and performance requirements that may impact the system's development.

1.2. Scope of this document : The scope of the document for a stock maintenance system includes defining the functional and non-functional requirements of the software application. The document aims to provide a clear understanding of the scope of the project, the expected outcome, and the limitations of the system.

1.3. Overview : The stock maintenance system is a software application designed for managing inventory levels and stock availability in a retail business. The system will automate the inventory management process and provide real-time information on stock levels, reorder points, and lead times. The system will also track the movement of stock items, manage purchase orders, and generate reports on sales and inventory levels. The system will integrate with the point-of-sale system and accounting software to ensure accurate and up-to-date records.

## **2. General description :**

A stock maintenance system is a software application designed to manage inventory levels and stock availability in a retail business. The system automates the inventory management process and provides real-time information on stock levels, reorder points, and lead times. It also tracks the movement of stock items, manages purchase orders, and generates reports on sales and inventory levels. The system integrates with the point-of-sale system and accounting software to ensure accurate and up-to-date records.

The stock maintenance system helps businesses to streamline their inventory management processes and reduce costs associated with stockouts and overstocking. With this system, businesses can optimize their inventory levels, minimize stockouts, and ensure that the right products are always available at the right time. The system also helps businesses to reduce the risk of losses due to theft, spoilage, or obsolescence, by enabling them to track the movement of stock items and generate reports on inventory levels. Overall, the stock maintenance system is an essential tool for businesses that want to improve their inventory management processes and ensure that they always have the right products available to meet customer demand.

### **3. Functional Requirements :**

3.1. Inventory Management: The system will allow the user to add, update and delete stock items. The user can also view the current stock level, reorder point, and lead time for each stock item. The system will generate automatic notifications when stock levels reach the reorder point.

3.2. Purchase Order Management: The system will allow the user to create and manage purchase orders for stock items. The user can also track the status of purchase orders and receive notifications when they are received.

3.3. Reporting: The system will generate reports on sales and inventory levels. The user can also view historical data on stock levels and sales.

3.4. Integration: The system will integrate with the point-of-sale system and accounting software to ensure accurate and up-to-date records.

### **4. Interface Requirements :**

4.1. User Interface: The user interface of the stock maintenance system should be easy to use and navigate. It should have clear and concise labels and buttons that are easily recognizable to the user. The user interface should also be customizable, allowing users to modify the layout to suit their preferences.

4.2. Point-of-Sale Integration: The stock maintenance system should integrate seamlessly with the point-of-sale system. This integration will ensure that the inventory levels are updated in real-time when a sale is made. It should also ensure that the stock levels are adjusted automatically when a return is processed.

4.3. Accounting Software Integration: The stock maintenance system should also integrate with the accounting software. This integration will ensure that the inventory levels and costs are accurately reflected in the accounting records. It should also ensure that the system generates accurate reports on sales and inventory levels.

4.4. Reporting: The system should have a reporting module that allows users to generate reports on sales and inventory levels. The reports should be customizable, allowing users to select the data that they want to include. The system should also provide users with the ability to export reports in a variety of formats, including Excel, PDF, and CSV.

4.5. Mobile Compatibility: The system should be compatible with mobile devices to allow users to access it from anywhere. The system should have a responsive design that adjusts to different screen sizes and resolutions. It should also have a mobile app that users can download from the app store.

## **5. Performance Requirements :**

5.1. Speed: The stock maintenance system should be able to handle large amounts of data and perform operations quickly. It should be able to retrieve information from the database and display it on the user interface in a matter of seconds. The system should also be able to process transactions quickly to ensure that the inventory levels are updated in real-time.

5.2. Scalability: The system should be scalable and able to handle an increasing number of users and transactions. As the business grows, the system should be able to accommodate more data and transactions without affecting performance.

5.3. Reliability: The stock maintenance system should be reliable and available at all times. The system should be able to handle unexpected events such as power outages, hardware failures, and network issues without losing data or causing data corruption.

5.4. Security: The system should have robust security features to prevent unauthorized access to the system and protect sensitive data. It should use encryption to secure data in transit and at rest. The system should also have access controls to ensure that only authorized users have access to the system.

5.5. Compatibility: The system should be compatible with different hardware and software configurations to ensure that it can be used in different environments. It should work with different operating systems, browsers, and mobile devices.

5.6. Data Integrity: The stock maintenance system should maintain the integrity of data by ensuring that data is accurate, consistent, and up-to-date. The system should also have backup and recovery mechanisms in place to protect data in case of data loss or corruption.

## **6. Design Constraints :**

6.1. Hardware Requirements: The stock maintenance system should be designed to work on a variety of hardware platforms, including desktops, laptops, and mobile devices. The system should also be able to work with different operating systems and browsers.

6.2. Integration: The stock maintenance system should be designed to integrate with other systems such as point-of-sale systems and accounting software. The system should be able to exchange data with these systems to ensure that inventory levels and costs are accurate.

6.3. Security: The system should be designed with robust security features to protect against unauthorized access and data breaches. It should use encryption to secure data in transit and at rest. The system should also have access controls to ensure that only authorized users have access to the system.

6.4. Usability: The stock maintenance system should be designed with usability in mind. The user interface should be intuitive, easy to use, and customizable. The system should also provide users with clear and concise feedback to ensure that they understand the system's operations.

6.5. Scalability: The system should be designed to accommodate future growth and increasing data volumes. The system should be scalable and able to handle an increasing number of users and transactions.

6.6. Maintenance: The system should be designed to be easily maintainable. The system should be modular to allow for easy updates and changes. The system should also be designed to be compatible with future updates to hardware and software.

## **7. Non-Functional Attributes :**

7.1. Performance: The system should be able to handle a large volume of stock items and transactions without affecting system performance.

7.2. Usability: The system should be user-friendly and accessible to all staff members who need to access it.

7.3. Security: The system should ensure the confidentiality and integrity of data. User authentication and access control mechanisms should be implemented to prevent unauthorized access to the system.

## **8. Preliminary Schedule and Budget :**

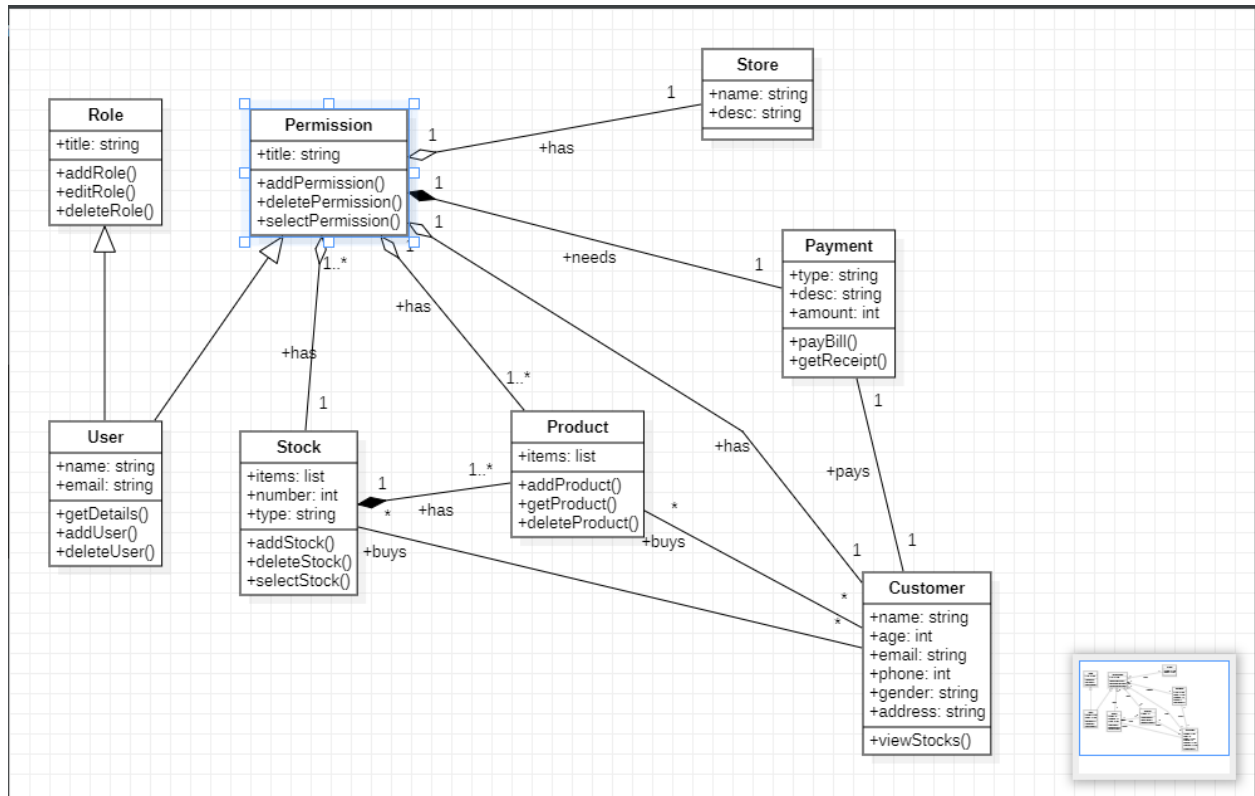
### **8.1. Preliminary Schedule:**

- Requirements Gathering: 2 weeks
- System Design: 4 weeks
- Development: 12 weeks
- Testing and Quality Assurance: 4 weeks
- Deployment: 2 weeks
- User Training and Documentation: 1 week
- Total Time: 25 weeks

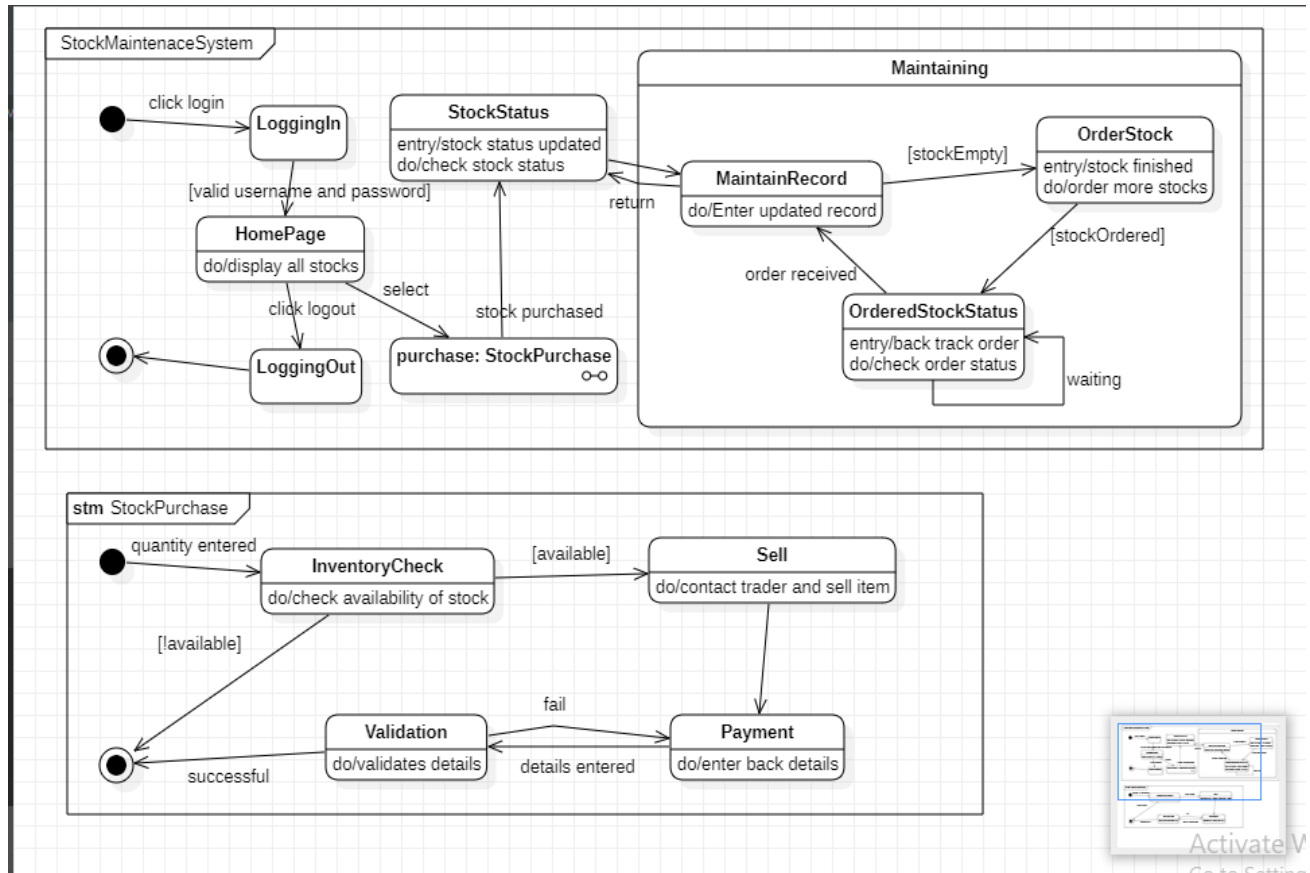
### **8.2. Preliminary Budget:**

- Salaries and Wages: ₹5,00,000
- Hardware and Software: ₹50,000
- Testing and Quality Assurance: ₹25,000
- User Training and Documentation: ₹10,000
- Contingency (10% of total budget): ₹58,500
- Total Budget: ₹6,43,500

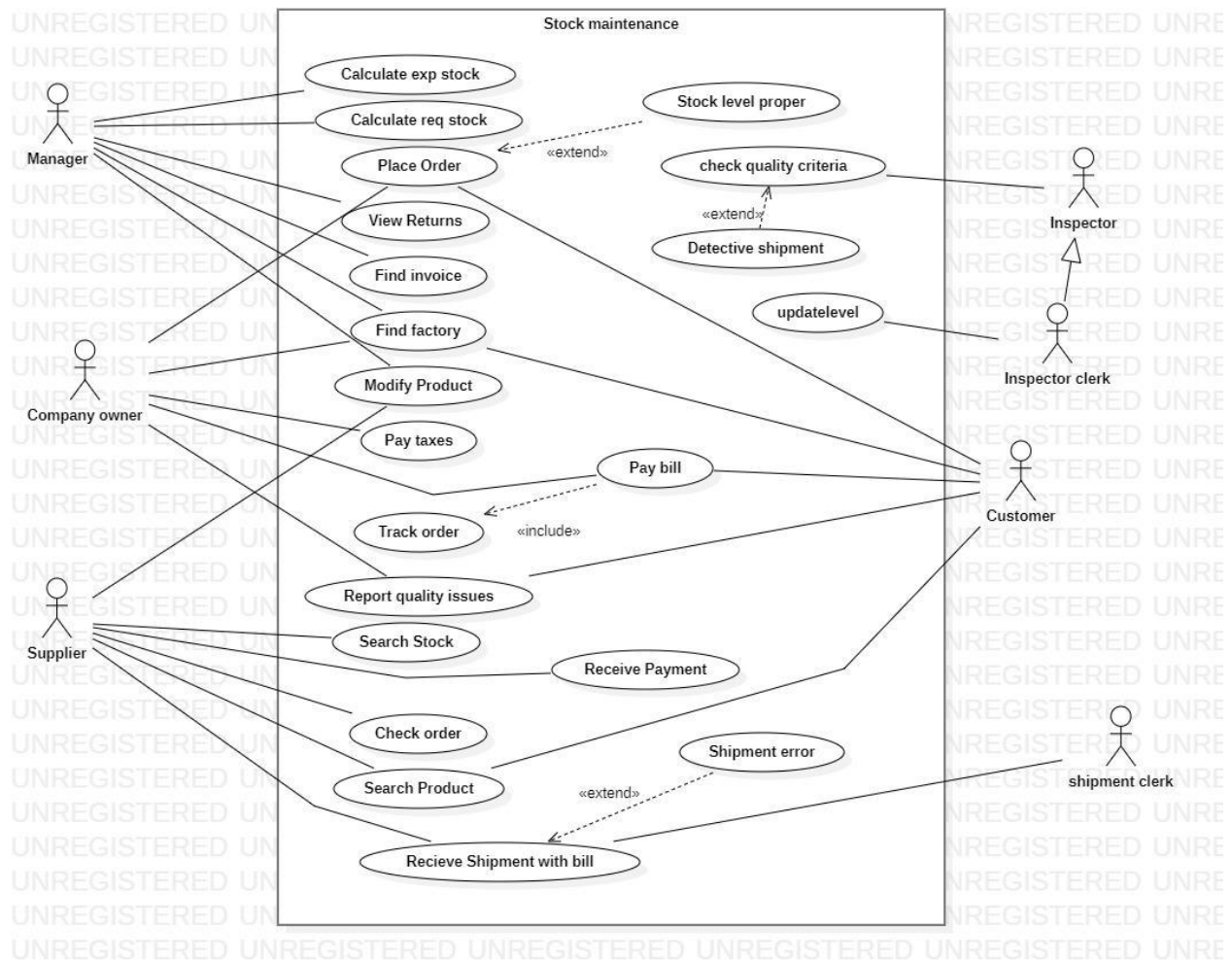
# CLASS DIAGRAM



# STATE DIAGRAM

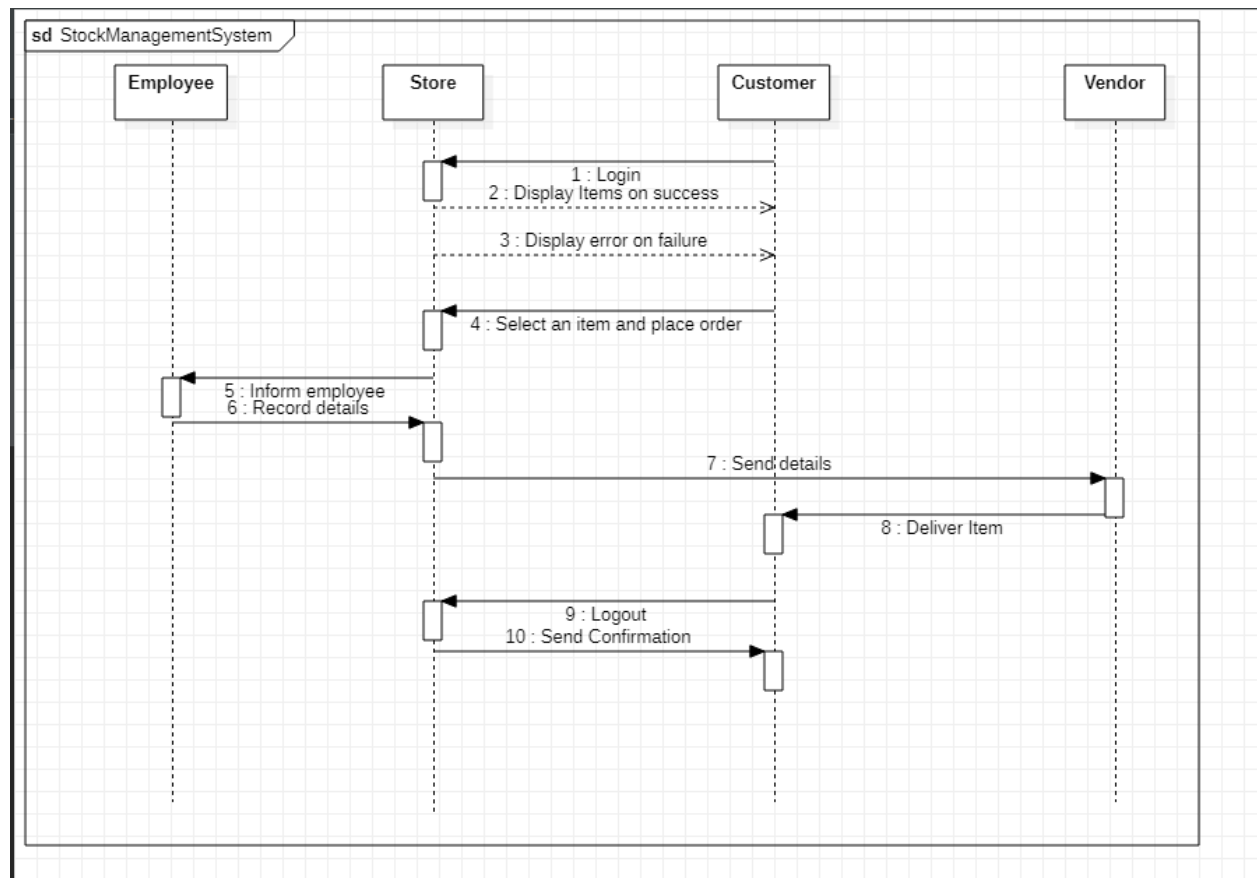


# USE CASE DIAGRAM

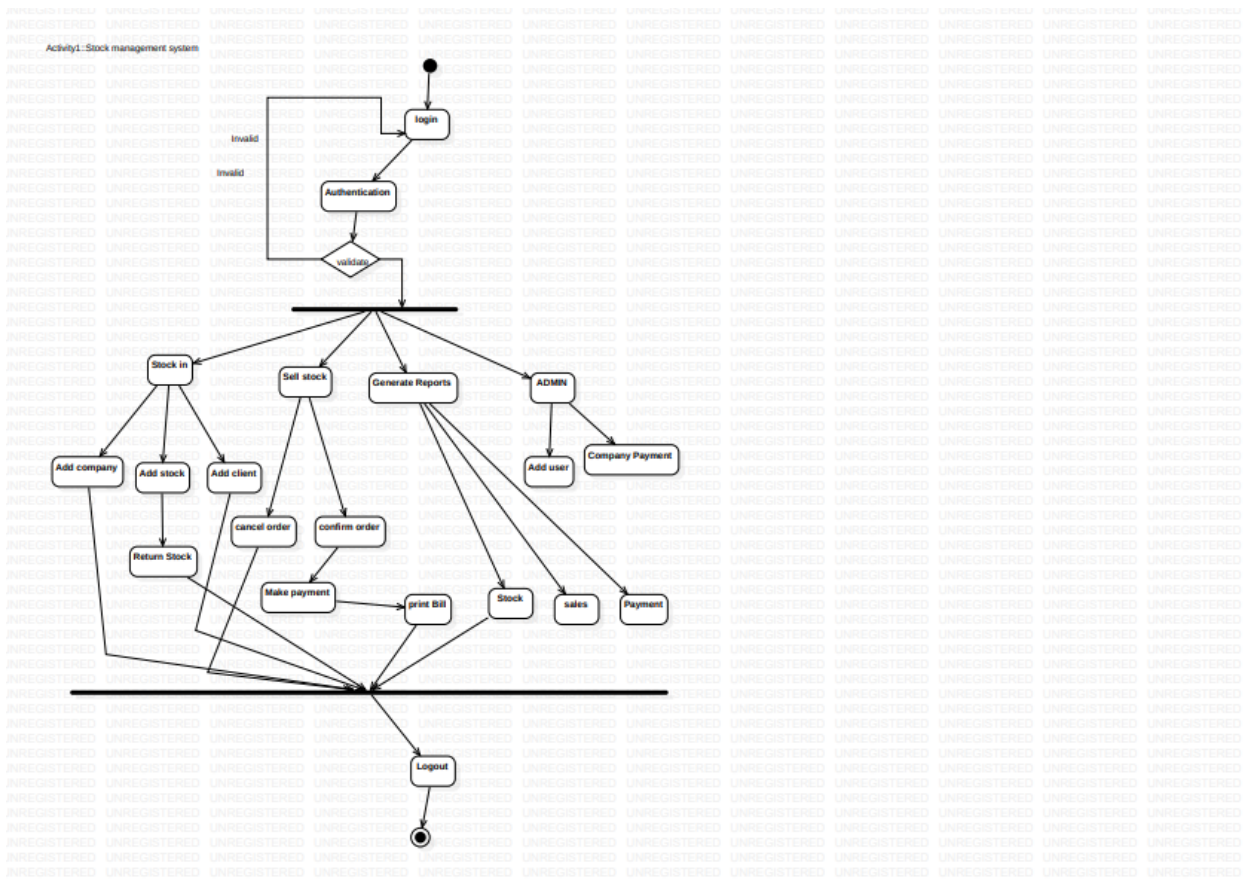




## SEQUENCE DIAGRAM



# ACTIVITY DIAGRAM



## **EXPERIMENT - 05**

### **PASSPORT AUTOMATION SYSTEM**

#### **PROBLEM STATEMENT :**

The traditional process of obtaining a passport involves multiple steps that can be time-consuming and frustrating for citizens. Typically, an applicant must visit a government office to fill out a paper application form and provide various supporting documents, such as proof of identity and address. The process can involve long wait times, as well as multiple visits to the office if any mistakes or omissions are found in the application.

The problem is the long and cumbersome process involved in obtaining a passport through traditional methods. The manual process often involves long wait times, multiple visits to government offices, and complex paperwork. This can lead to frustration and delays for citizens who need a passport urgently, such as for travel or employment purposes.

The Passport Automation System aims to simplify and digitize the passport application process, making it easier and more accessible for citizens. It aims to eliminate the need for physical visits to government offices by allowing citizens to complete the application process online.

Additionally, it aims to reduce wait times and provide a faster turnaround for passport issuance. Furthermore, the manual process is prone to errors and inefficiencies, which can result in delays in processing and issuance of passports. This can be a significant problem for citizens who need their passport quickly, such as for urgent travel plans or employment opportunities abroad.

The Passport Automation System seeks to address these problems by providing a digital platform for passport applications. Citizens can complete the entire application process online, including filling out the application form and uploading supporting documents. This eliminates the need for physical visits to government offices, saving time and reducing the risk of errors.

Additionally, the system aims to provide faster processing and issuance times for passports. By automating the process, the system can quickly verify the information provided by the applicant.

and issue passports in a timely manner. This can be a significant improvement over the manual process, which can take several weeks or even months to complete.

Overall, the Passport Automation System aims to provide a more convenient, efficient, and accessible way for citizens to obtain passports. By streamlining the process and eliminating inefficiencies, the system can help to reduce wait times and provide faster turnaround times for passport issuance, ultimately benefiting citizens and promoting ease of travel

## **Software Requirement Specification(SRS)**

### **1 Introduction :**

1.1 Purpose of this Document : The purpose of this Software Requirements Specification (SRS) document is to define the comprehensive set of functional and nonfunctional requirements for the Passport Automation System. This document outlines the features, functions, and constraints of the system.

1.2 Scope of this document : The Passport Automation System will be a web-based application that allows citizens to apply for passports online. The system will support the following features: Citizen registration and login , Passport application form with required fields and supporting document upload , Payment processing for passport fees , Application status tracking,Passport issuance and delivery tracking,Admin panel for managing applications and users,Reporting and analytics for system usage and performance.

1.3 Overview : A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also provides a comprehensive outline of the requirements for a Passport Automation System, and it serves as a guide for developers and stakeholders to ensure that the system meets the needs of its users.

### **2 General description :**

Passport Automation System is a digital platform designed to automate and streamline the passport application process. The system aims to provide a convenient and efficient way for citizens to apply for passports online, track their application status, and receive their passport in a timely manner.

Passport Automation System aims to provide a more convenient, efficient, and accessible way for citizens to obtain passports. By streamlining the process and eliminating inefficiencies, the system can help to reduce wait times and provide faster turnaround times for passport issuance, ultimately benefiting citizens and promoting ease of travel.

### **3 Functional Requirements :**

- **Citizen Registration and Login :** The system shall allow citizens to register for an account using their personal information. The system shall require citizens to verify their email address and mobile number during registration. The system shall allow citizens to log in to their account using their email address and password.
- **Passport Application Form :** The system shall provide an online application form for citizens to apply for a passport. The application form shall include required fields such as name, date of birth, gender, and contact information and also fields for uploading supporting documents such as proof of identity and address. The system shall validate the information provided by the applicant and notify the user of any errors or missing information.
- **Payment Processing :** The system shall allow citizens to pay the passport fee using a secure payment gateway. The system shall support multiple payment methods, such as credit/debit cards and net banking. The system shall generate a payment receipt for the user upon successful payment.
- **Application Status Tracking :** The system shall allow citizens to track the status of their passport application. The system shall provide real-time updates on the application status, such as "Application Submitted," "Under Review," and "Passport Issued." The system shall notify the user via email and SMS when there is a status update on their application.
- **Passport Issuance and Delivery Tracking :** The system shall track the issuance and delivery of passports. The system shall notify the user via email and SMS when their passport is issued and dispatched. The system shall provide tracking information for passport delivery.
- **Admin Panel :** The system shall provide an admin panel for government officials to manage passport applications and users. The admin panel shall allow officials to view and manage applications, such as approving or rejecting applications and updating application

status. The admin panel shall allow officials to view and manage users, such as resetting passwords and disabling accounts.

- Reporting and Analytics : The system shall provide reporting and analytics features to track system usage and performance. The system shall generate reports on application volume, payment processing, and user demographics. The system shall provide analytics on system performance, such as application processing time and payment success rate.

#### **4 Interface Requirements :**

- User Interface : The user interface of the Passport Automation System shall be user-friendly and intuitive, allowing citizens to easily navigate through the system. The system shall provide clear and concise instructions for each step of the passport

application process to guide users through the process. The system shall provide a feedback mechanism for users to report any issues or errors they encounter while using the system. The system shall provide an option for users to change their password and update their personal information.

- Database Interface : The system shall use a database to store and manage user data, passport application data, and other relevant information. The database interface shall be secure, preventing unauthorized access to the database and ensuring the integrity of the data. The system shall be able to retrieve and update data from the database in real-time to ensure that users have access to the latest information.

- Payment Gateway Interface : The system shall provide multiple payment gateway options for citizens to pay their passport application fee online. The payment gateway interface shall be secure, ensuring the confidentiality of user payment information. The system shall provide real-time confirmation of payment for users to verify the status of their payment.

- Document Verification Interface : The system shall provide a document verification interface for verifying the authenticity of supporting documents provided by applicants. The document verification interface shall be secure, preventing unauthorized access to the documents and ensuring the privacy of user information. The system shall provide real-time feedback on the status of document verification to users.

## **5 Performance Requirements :**

- **Response Time:** The system should have a fast response time to ensure users can perform tasks quickly. The system should respond to user requests within a reasonable time frame, typically less than 3 seconds.
- **System Availability:** The system should be available to users at all times. The system should have a high uptime percentage, typically greater than 99%, to ensure users can access the system when needed.
- **Scalability:** The system should be scalable, allowing it to handle an increasing number of users and data without performance degradation. The system should be able to handle peak loads during high demand periods such as holidays and events.
- **Concurrent User Capacity:** The system should be able to handle a high number of concurrent users without performance degradation. The system should be able to handle at least 100 users simultaneously without affecting response time.
- **Data Processing Speed:** The system should have fast data processing speed to ensure users can perform tasks quickly. The system should be able to handle large data volumes without affecting response time.
- **Security:** The system should be secure to ensure data confidentiality, integrity, and availability. The system should use encryption to protect sensitive data, have access control mechanisms to restrict unauthorized access, and have a backup and recovery plan in case of data loss.
- **Integration:** The system should be able to integrate with other systems, such as payment gateways, property management systems, and customer relationship management systems. Integration ensures seamless data exchange and reduces manual data entry, leading to better system performance.

## **6 Design Constraints :**

- **Legal and Regulatory Requirements:** The system must comply with all relevant legal and regulatory requirements related to passport issuance and management.

- Security: The system must be designed with strong security features to ensure that personal information and other sensitive data are protected against unauthorized access, theft, or misuse.
- Scalability: The system must be designed to accommodate future growth and expansion of the passport office, and should be able to handle a large volume of users and data.
- User Interface: The system must have an intuitive and easy-to-use interface that can be used by a wide range of users, including those with limited computer skills.
- Performance: The system must be able to perform quickly and efficiently to minimize waiting times for users.
- Reliability: The system must be designed with a high level of reliability, to ensure that it is available and functioning properly at all times.
- Compatibility: The system must be compatible with a wide range of hardware and software systems, including the operating system used by the passport office.
- Maintainability: The system must be designed with easy maintenance and support in mind, to minimize downtime and repair costs.
- Accessibility: The system must be designed to be accessible to all users, including those with disabilities.
- Data Integrity: The system must ensure the accuracy and integrity of data, and must be able to detect and prevent errors and inconsistencies.

## **7 Non-Functional Attributes :**

- Performance: The system must be able to handle a large volume of transactions efficiently, with minimal delay or waiting times for users.
- Reliability: The system must be highly reliable, with a high level of availability and minimal downtime.
- Security: The system must be designed with strong security features to protect sensitive information and prevent unauthorized access.



- Usability: The system must be user-friendly and easy to navigate, with clear instructions and guidance for users.
- Compatibility: The system must be compatible with a wide range of hardware and software systems, and must be able to integrate with other systems as necessary.
- Scalability: The system must be able to handle future growth and expansion of the passport office, with the ability to add new features and functionality as needed.
- Maintainability: The system must be easy to maintain and support, with clear documentation and tools for troubleshooting and problem resolution.
- Accessibility: The system must be designed to be accessible to all users, including those with disabilities or special needs.
- Performance under load: The system must be able to handle a high volume of users and transactions without slowing down or crashing.
- Data privacy: The system must comply with all relevant data privacy regulations and protect personal information from unauthorized access or disclosure.

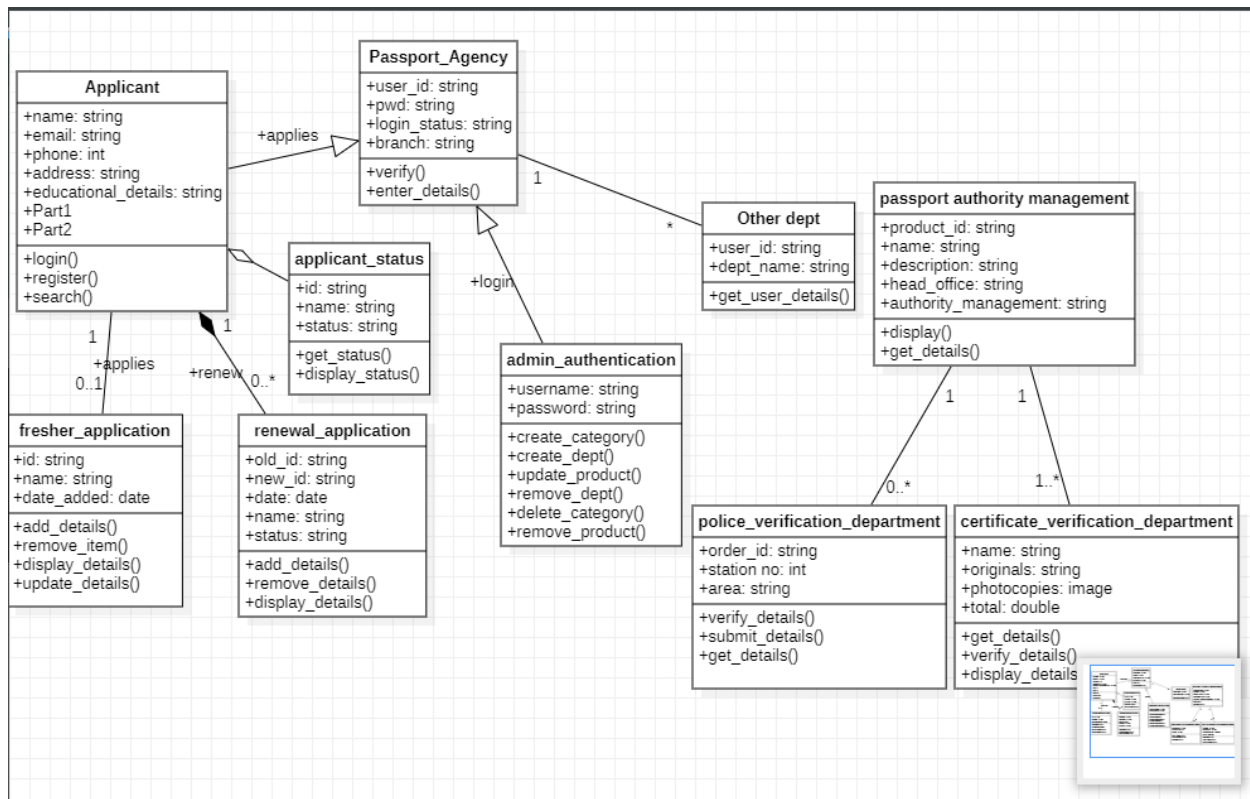
## **8 Preliminary Schedule and Budget : Preliminary Schedule:**

- Requirements Gathering: 2 weeks
- System Design: 4 weeks
- Development: 12 weeks
- Testing and Quality Assurance: 4 weeks
- Deployment: 2 weeks
- User Training and Documentation: 1 week
- Total Time: 25 weeks

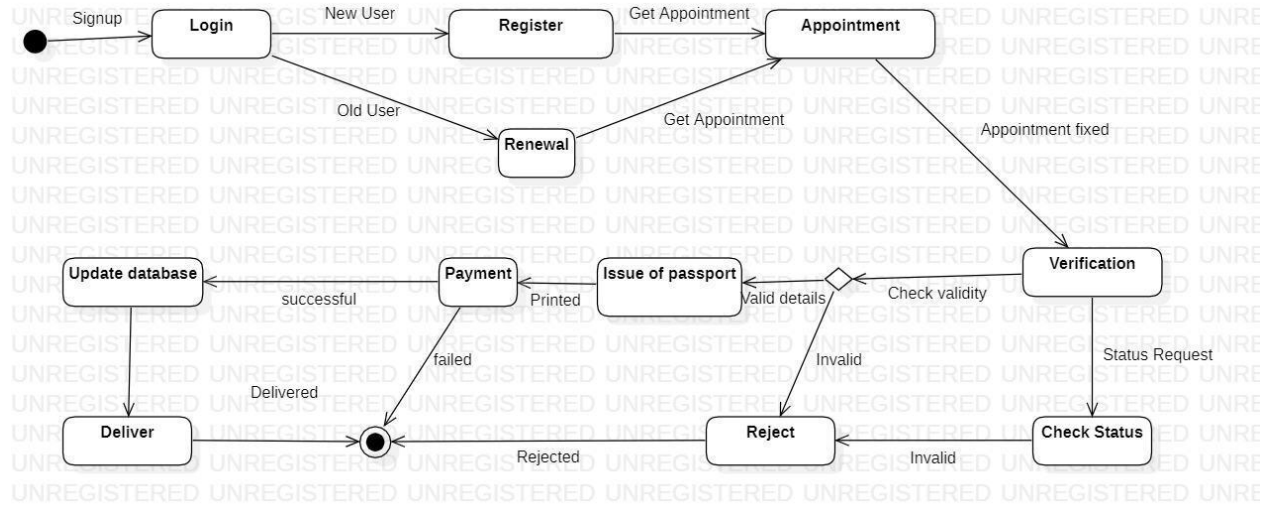
## Preliminary Budget:

- Salaries and Wages: ₹5,00,000
- Hardware and Software: ₹50,000
- Testing and Quality Assurance: ₹25,000
- User Training and Documentation: ₹10,000
- Contingency (10% of total budget): ₹58,500
- Total Budget: ₹6,43,500

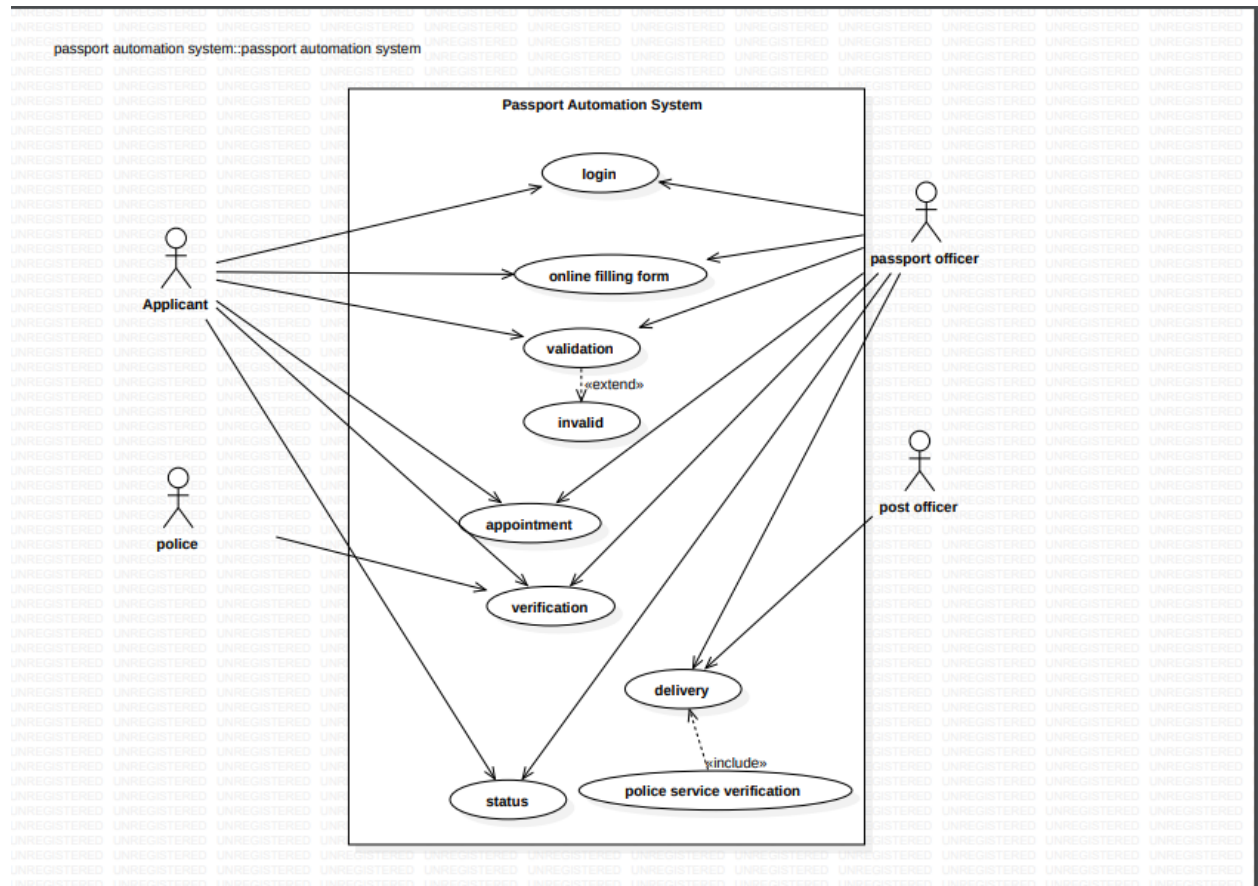
## CLASS DIAGRAM



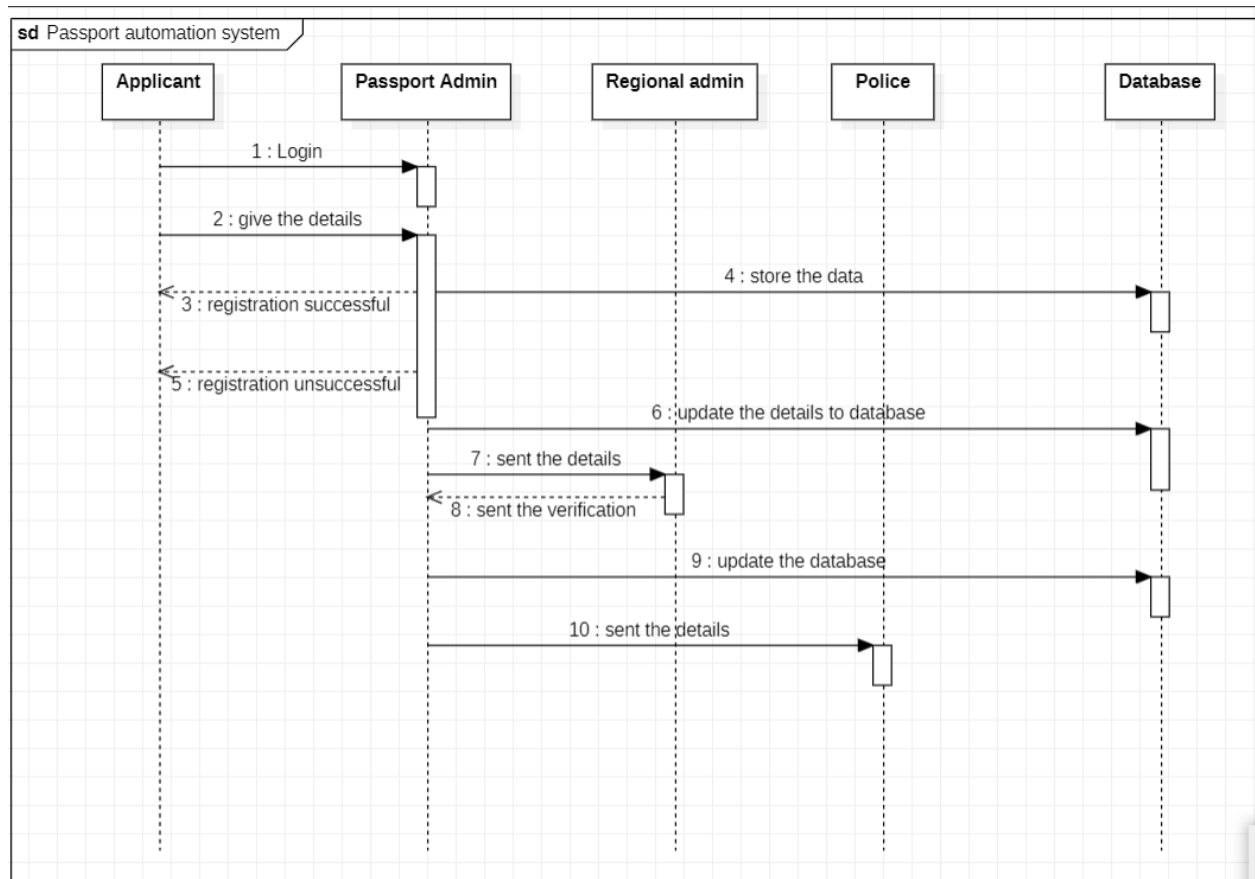
## STATE DIAGRAM



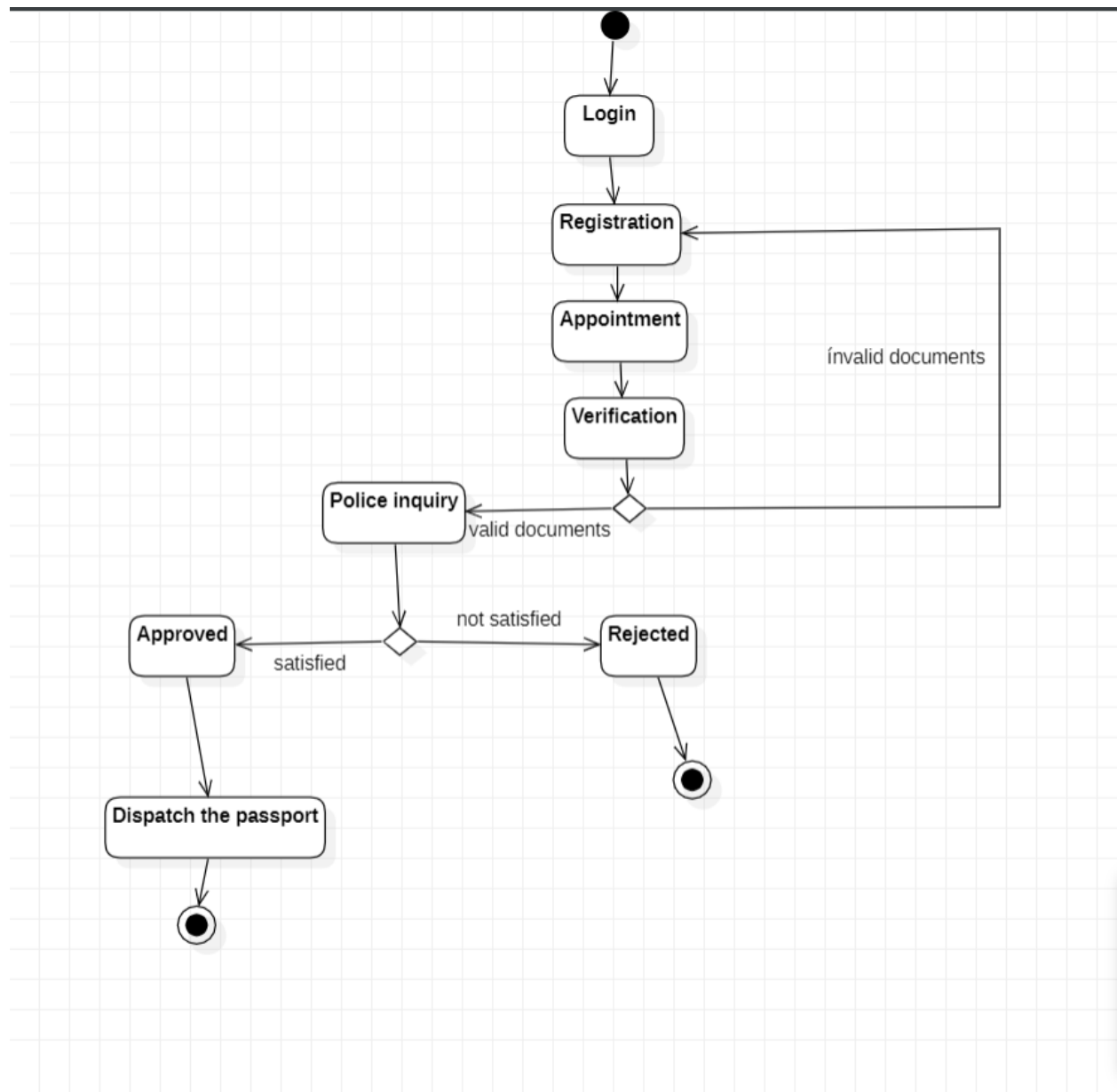
# USE CASE DIAGRAM



# SEQUENCE DIAGRAM



## ACTIVITY DIAGRAM



## **Experiment – 06**

### **Railway Reservation System**

#### **2.1 PROBLEM STATEMENT:**

The current manual reservation process for train tickets is inefficient and time-consuming, causing inconvenience and long queues for customers. There is also a lack of transparency and real-time information, making it difficult for customers to plan their travel in advance. This results in a sub-optimal customer experience, which impacts the reputation of the railway system.

Therefore, the aim of this project is to develop an automated Railway Reservation System that will streamline the ticket booking process and provide real-time information to customers. The system should be user-friendly, secure, and scalable to handle large volumes of transactions. It should also have features such as online payment, seat availability, and ticket cancellation to enhance the customer experience. The ultimate goal is to provide a hassle-free and efficient booking process for passengers and improve the overall efficiency of the railway system.

#### **2.2 SOFTWARE REQUIREMENT SPECIFICATION:**

##### **2.2.1 INTRODUCTION**

Railway Reservation System is a computerized system designed to simplify and automate the process of booking and managing train tickets. It enables passengers to book, modify or cancel their train tickets from the comfort of their homes or offices. The system provides a convenient and efficient way for passengers to plan their travel, check availability, and make payments online. This system also helps railway authorities to manage train schedules, allocate seats and berth, and generate reports on passenger bookings and revenue earned. With the help of modern technologies like cloud computing and mobile applications, the Railway Reservation System has evolved into a robust and reliable platform that benefits both passengers and railway authorities alike.

## **PURPOSE OF THE DOCUMENT**

The purpose of a railway reservation system is to facilitate the booking and management of train tickets for passengers. It is a software system that enables passengers to check train schedules, seat availability, and fares, as well as reserve and purchase tickets online or through a ticketing agent.

The system is designed to provide an efficient and convenient way for passengers to book their train tickets, without having to physically visit a ticket counter or booking office. It also allows railway authorities to manage and track ticket sales, passenger information, and train schedules, and make data-driven decisions to improve the overall efficiency and profitability of the railway system.

## **SCOPE OF THE DOCUMENT**

- **Simplified Booking Process:** A railway reservation system can simplify the booking process for passengers by allowing them to book tickets online, thereby reducing the need to visit a railway station physically.
- **Efficient Resource Utilization:** With a railway reservation system, railway authorities can better manage the available resources, such as seats, coaches, and trains
- **Real-Time Information:** A railway reservation system can provide passengers with real-time information about the availability of seats, the status of trains, and the expected arrival and departure times
- **Revenue Generation:** A railway reservation system can help railway authorities to generate more revenue by offering various incentives, discounts, and packages to passengers.
- **Data Analytics:** A railway reservation system can collect and analyze data about passenger preferences, travel patterns, and booking behavior.
- **Overall,** a railway reservation system can bring significant benefits for both passengers and railway authorities, making it an attractive proposition for development and implementation.



## **OVERVIEW**

The Railway Reservation System is a computerized system that allows passengers to book train tickets online. The system is designed to simplify the process of booking tickets and to reduce the long queues at the ticket counters. It also provides a platform for passengers to check train schedules, seat availability, and fares. The Railway Reservation System is a crucial part of the Indian Railways, as it handles millions of ticket bookings every day. It has made the process of booking train tickets faster, more efficient, and more convenient for passengers.

### **2.2.2 GENERAL DESCRIPTION**

A Railway Reservation System is an automated system that manages the process of booking and canceling train tickets. It is designed to make the booking process easier and more efficient for both passengers and railway staff.

The system is typically web-based, allowing passengers to check train schedules, availability of seats, and prices, and book their tickets online. It can also be accessed through mobile applications or at railway station counters.

The Railway Reservation System usually includes a database that stores information about train schedules, routes, seat availability, and ticket prices. When a passenger books a ticket, the system checks for seat availability and reserves the requested seat.

### **2.2.3 FUNCTIONAL REQUIREMENTS**

1. User registration: The system should allow users to create accounts and register with their personal details such as name, email, phone number, and address.
2. Search and booking: The system should provide a search interface to help users find trains based on their departure and arrival cities, travel dates, and preferences. Users should be able to select a train and book tickets for themselves or other passengers.
3. Seat selection: The system should allow users to choose their preferred seats or berth type based on availability.

4.      Payment gateway integration: The system should integrate with a payment gateway to facilitate secure online transactions and accept payments through various modes such as credit/debit cards, net banking, and mobile wallets.
5.      Ticket confirmation: The system should generate a ticket after the booking is completed and send it to the user's registered email address or mobile number.
6.      Cancellation and refund: The system should allow users to cancel their tickets and initiate a refund based on the cancellation policies of the railway operator.
7.      Train schedule and status: The system should display the train schedules, arrival and departure times, and real-time status updates of trains to keep passengers informed about delays and cancellations.
8.      User feedback: The system should provide a feedback mechanism to collect user reviews and ratings to improve the overall service quality.
9.      Admin dashboard: The system should provide an administrative dashboard for railway operators to manage the train schedules, seat availability, pricing, and other related tasks.
10.     Reporting and analytics: The system should provide reporting and analytics features to generate insights into user behavior, booking trends, and revenue performance.

#### **2.2.4 INTERFACE REQUIREMENTS**

The interface requirements for a railway reservation system depend on various factors such as the platform, user demographics, and the overall system architecture. However, here are some general requirements that can be considered for the system's interface:

1.      User-Friendly Interface: The railway reservation system should have a user-friendly interface that can be easily navigated by users of all ages and backgrounds. It should be easy to understand and use.
2.      Login/Register Page: The system should have a login/register page where users can create an account and log in to the system. This page should be secure and easy to use.

3.      **Booking Page:** The booking page should have all the necessary fields to enter the journey details such as source, destination, date, and class of travel. It should also have an option to choose the mode of payment.
4.      **Seat Availability:** The system should display the seat availability status based on the journey details entered by the user. It should also allow users to select their preferred seats.
5.      **Payment Gateway:** The system should have a secure payment gateway integrated to allow users to make payments online. It should support multiple payment modes such as net banking, credit/debit cards, and mobile wallets.
6.      **Cancellation and Refund:** The system should have an easy-to-use cancellation and refund process. It should allow users to cancel their tickets and get refunds if applicable.
7.      **User Dashboard:** The system should have a user dashboard that displays the user's booking history, upcoming journeys, and other relevant information.
8.      **Help and Support:** The system should have a dedicated help and support section that can assist users in case of any issues or queries.
9.      **Mobile Compatibility:** The system should be mobile-friendly and have a responsive design that can adapt to different screen sizes.
10.     **Security:** The system should have strong security measures in place to protect users' data and prevent unauthorized access. It should also comply with relevant data protection laws and regulations.

### **2.2.5 PERFORMANCE REQUIREMENTS**

The railway reservation system must meet the following performance requirements:

1.      **Response time:** The system must respond quickly to user requests for booking, cancellation, and modification of tickets. Response time should be less than a few seconds.
2.      **Capacity:** The system must be able to handle a large number of concurrent users without slowing down or crashing. The capacity should be sufficient to handle peak traffic during holiday seasons and festivals.

3. Reliability: The system must be highly reliable and available 24/7. It should have backup systems in place to ensure continuity of service in case of hardware or software failures.
4. Security: The system must be secure and protect user data from unauthorized access, modification, or theft. It should use encryption and authentication mechanisms to ensure data confidentiality and integrity.
5. Scalability: The system must be able to scale up or down based on changing user demands. It should be easy to add new servers, storage, or network resources to meet increased demand.
6. Availability: The system must be highly available and provide uninterrupted service to users. It should have a robust disaster recovery plan in place to ensure business continuity in case of a natural or man-made disaster.
7. User experience: The system must provide a seamless and intuitive user experience to users. It should be easy to navigate, with clear instructions and error messages.
8. Performance monitoring: The system should have performance monitoring and logging mechanisms in place to track usage patterns, identify bottlenecks, and optimize performance.
9. Integration: The system should be able to integrate with other systems and applications, such as payment gateways, accounting software, and customer relationship management (CRM) systems.

### **2.2.6 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements for a railway reservation system are the requirements that do not specify the functionality of the system but instead specify how the system should perform. Some of the non-functional requirements for a railway reservation system are:

1. Performance: The system should be able to handle a large number of users simultaneously without compromising the response time.
2. Scalability: The system should be scalable to accommodate future growth in terms of the number of users and the volume of data.

3. Reliability: The system should be highly reliable and available at all times, with minimal downtime.
4. Security: The system should be highly secure to prevent unauthorized access, data theft, and malicious attacks.
5. Usability: The system should be easy to use and understand for all users, including those with disabilities.
6. Maintainability: The system should be easy to maintain and upgrade, with minimal disruption to the users.
7. Compatibility: The system should be compatible with different hardware and software platforms.
8. Interoperability: The system should be able to communicate and exchange data with other systems.
9. Accessibility: The system should be accessible to users with different devices and operating systems.
10. Performance under stress: The system should perform well under stress, such as during peak hours or in case of hardware or software failures.

### **2.2.7 DESIGN CONSTRAINTS**

As a language model, I can provide some general design constraints for a Railway Reservation System. These are:

1. Scalability: The system should be designed to handle a large number of users simultaneously. The design should be scalable to handle a large number of transactions and data.
2. Reliability: The system should be highly reliable to ensure that bookings and cancellations are processed accurately and in a timely manner. The system should be designed to handle hardware and software failures with minimal impact on users.

3. Security: The system should be secure to ensure that user data is protected from unauthorized access. The system should have appropriate authentication and authorization mechanisms to protect user data.
4. Performance: The system should be designed to perform well under heavy load. The design should be optimized to minimize response times and maximize throughput.
5. Usability: The system should be user-friendly and easy to use. The system should be designed with a clear and intuitive interface that is easy to navigate.
6. Compatibility: The system should be compatible with different devices, platforms, and browsers to ensure that users can access the system from anywhere using their preferred device.
7. Accessibility: The system should be designed to be accessible to users with disabilities. The system should be designed with accessibility features such as screen readers and keyboard navigation.
8. Maintainability: The system should be easy to maintain and update. The system should be designed with a modular and extensible architecture that allows for easy modification and addition of new features.
9. Cost-effectiveness: The system should be designed to be cost-effective. The design should be optimized to minimize hardware and software costs while still providing high-quality service to users.
10. Compliance: The system should comply with all relevant laws and regulations related to data privacy, security, accessibility, and usability.

### **2.2.8 PRELIMINARY SCHEDULE AND BUDGET**

Schedule:

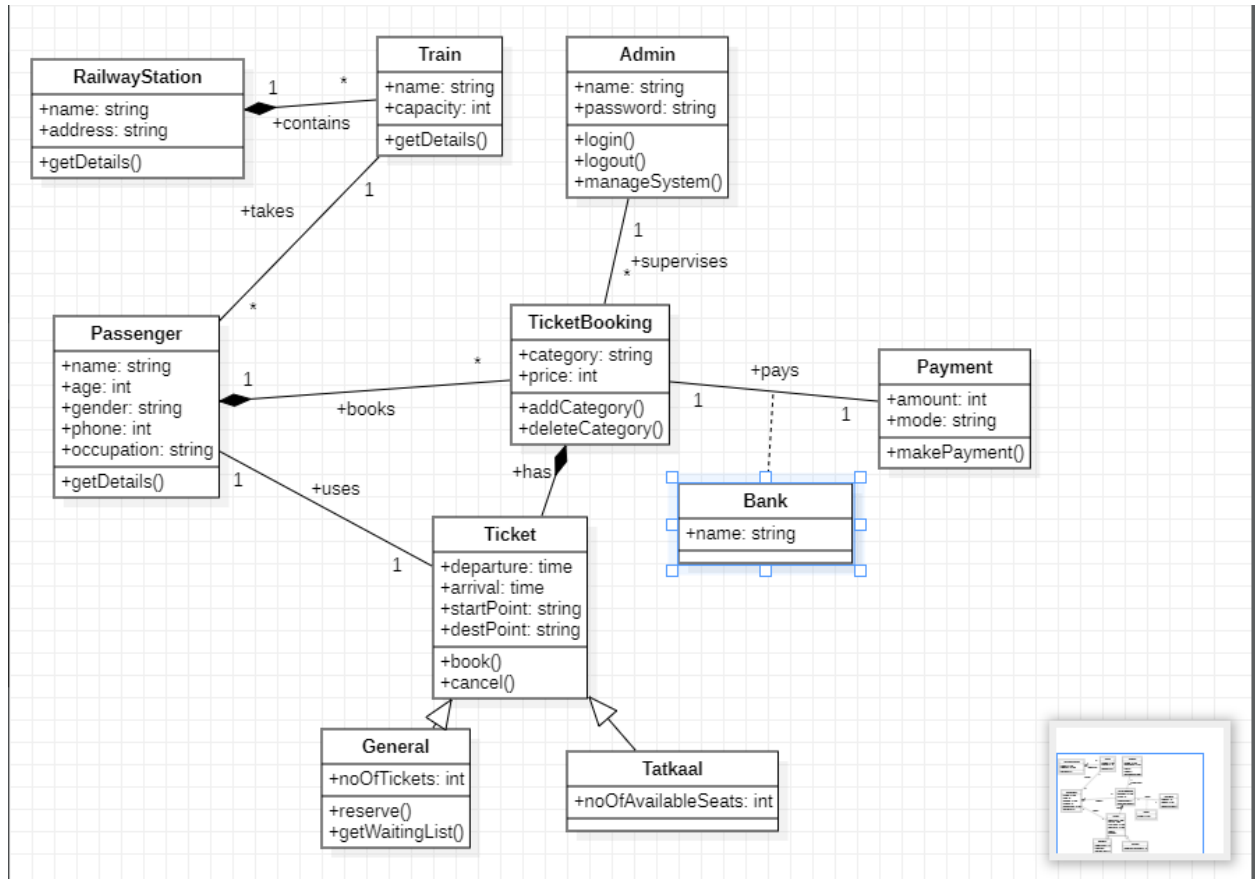
Requirements gathering: 2-4 weeks System design: 4-6 weeks Development and testing: 12-20 weeks Deployment and integration: 2-4 weeks User acceptance testing: 4-6 weeks Training and documentation: 2-4 weeks Maintenance and support: ongoing

Budget:

Salaries and benefits for development team: \$200,000 - \$500,000 Hardware and software:  
\$50,000 - \$100,000

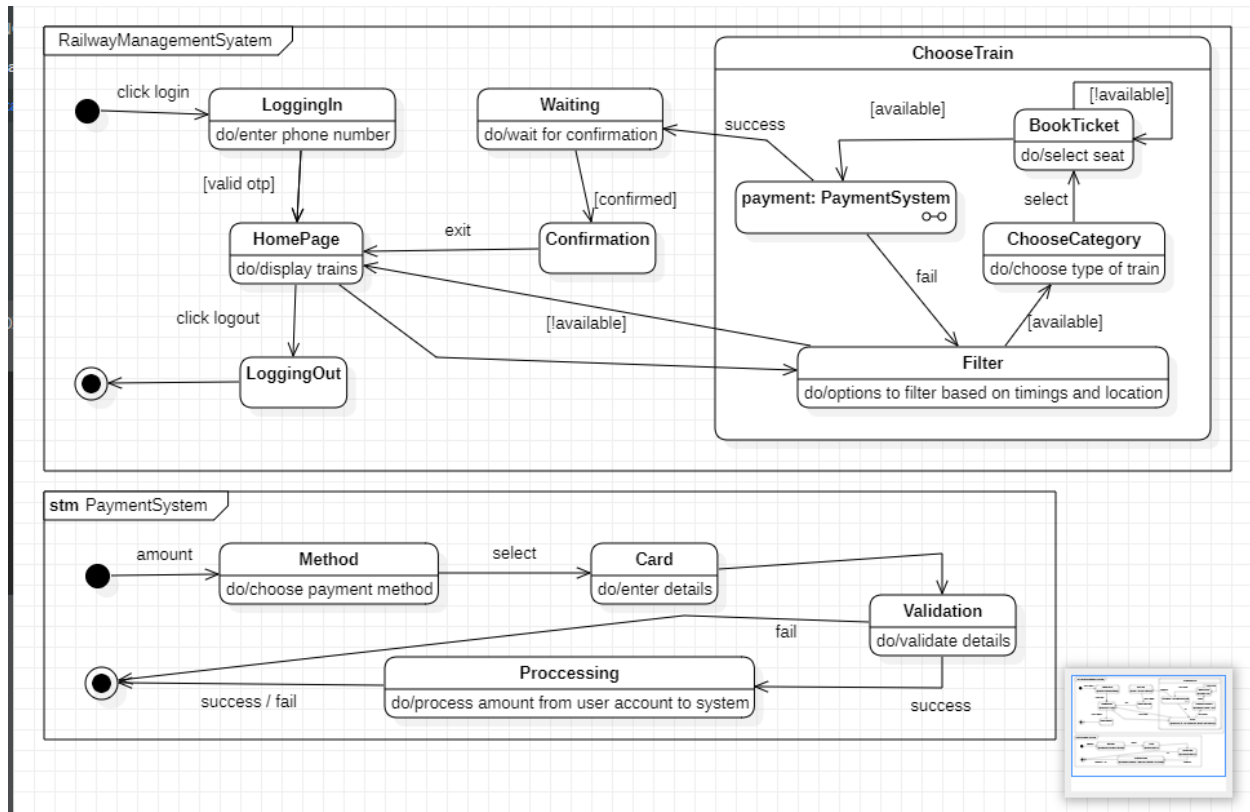
Third-party services (e.g. testing, deployment, training): \$50,000 - \$100,000 Contingency  
(10-20% of total budget): \$35,000 - \$100,000

# CLASS DIAGRAM

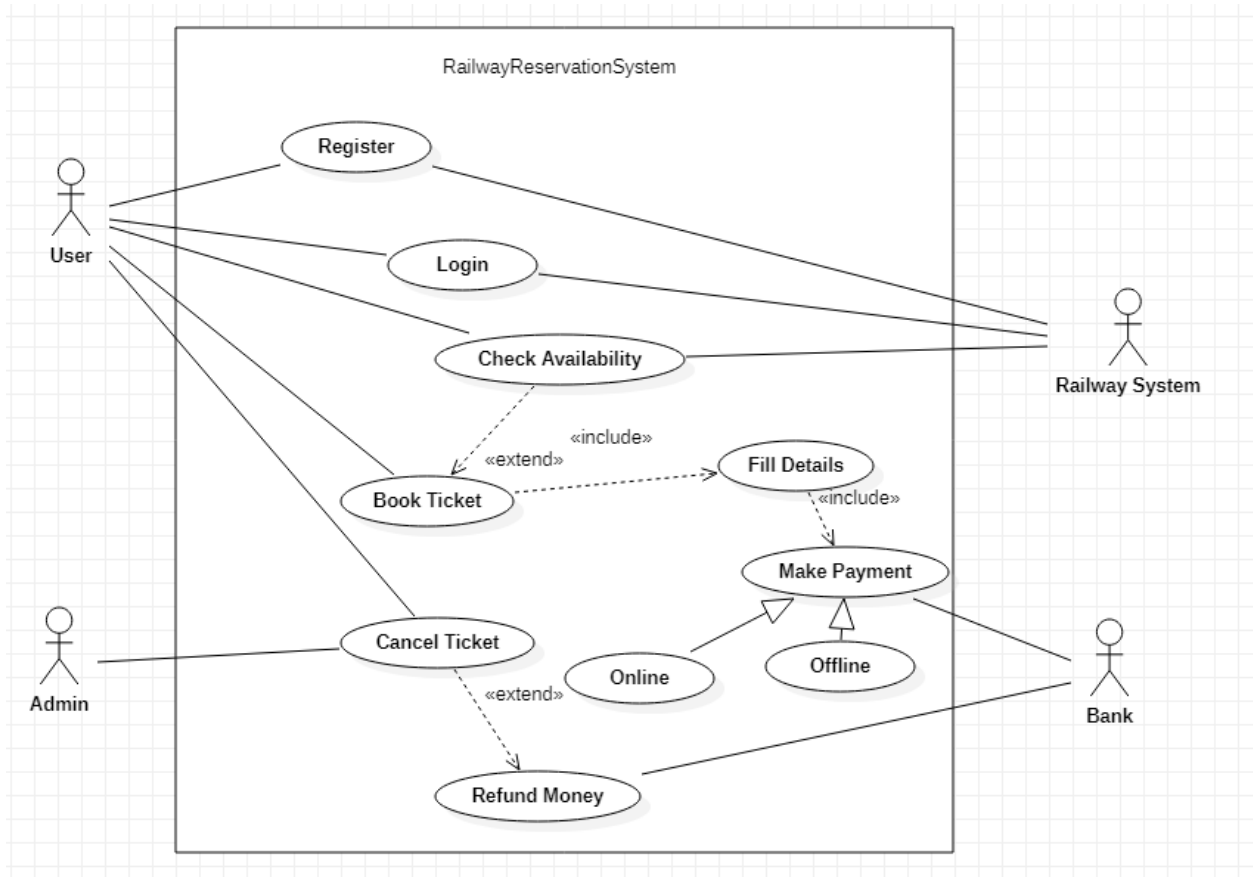




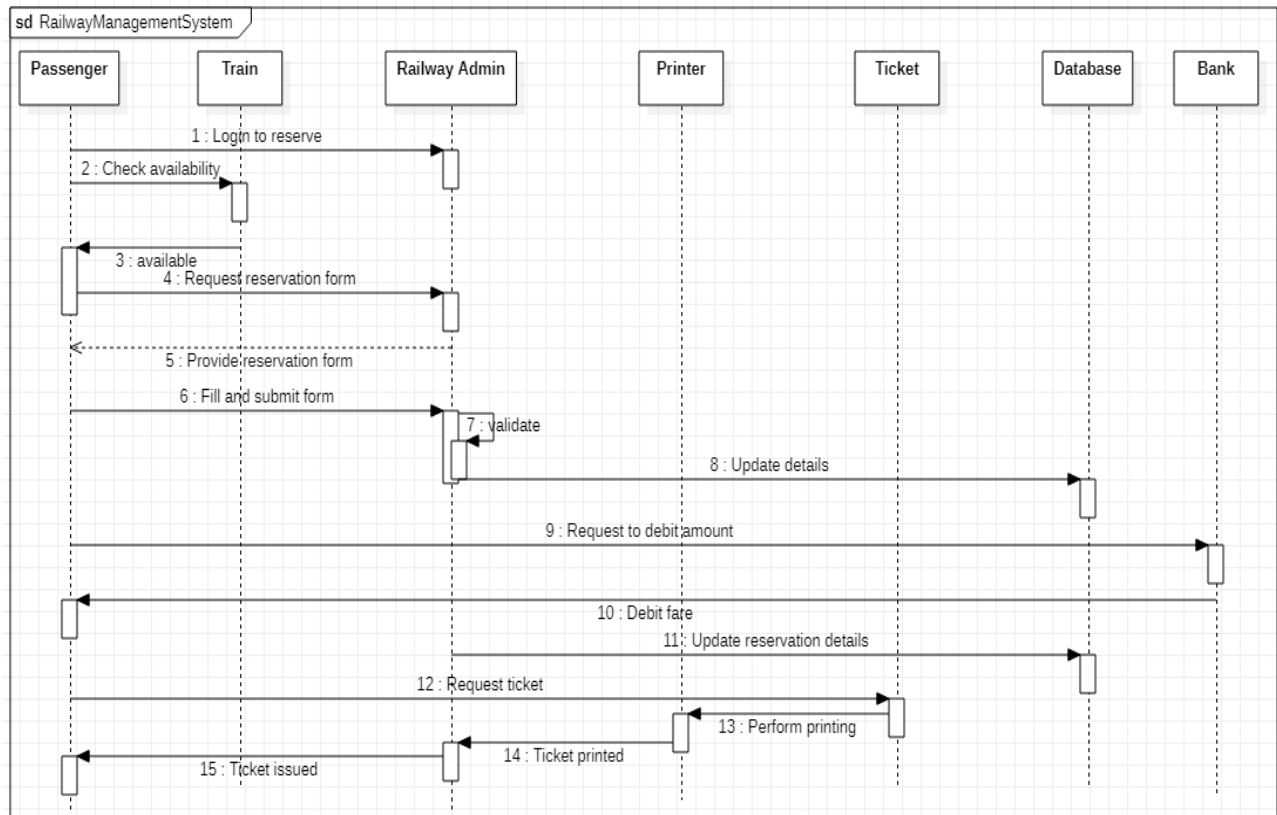
# STATE DIAGRAM



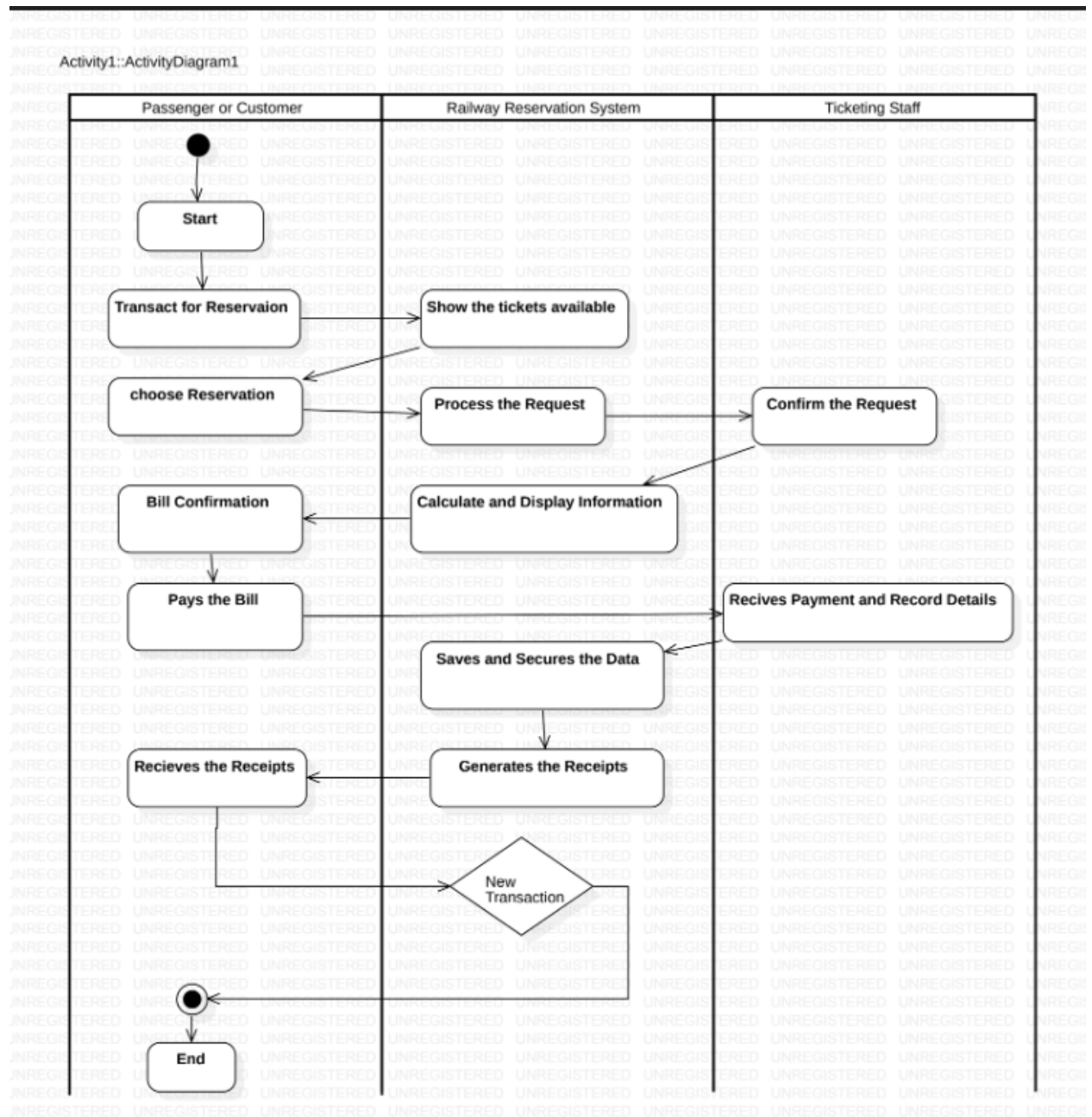
# USE CASE DIAGRAM



# SEQUENCE DIAGRAM



# ACTIVITY DIAGRAM



## **Experiment – 07**

### **Online shopping System**

#### **2.1 PROBLEM STATEMENT:**

The rise of e-commerce has led to an increasing demand for online shopping systems. However, many online shopping systems suffer from several issues, such as poor website design, complicated checkout processes, inadequate search functionality, and unreliable delivery services. These problems often result in frustrated customers, reduced sales, and negative brand perception.

Therefore, the problem statement is to develop an online shopping system that provides a seamless and convenient shopping experience for customers. The system should have a user-friendly interface, easy navigation, efficient search functionality, secure payment gateway, and reliable delivery services. Additionally, the system should be able to handle high traffic and ensure data security and privacy.

The proposed online shopping system aims to enhance customer satisfaction, increase sales, and improve the overall brand reputation. By addressing the issues faced by existing online shopping systems, the proposed system can attract new customers, retain existing ones, and establish a competitive advantage in the e-commerce industry.

#### **2.2 SOFTWARE REQUIREMENT SPECIFICATION:**

##### **2.2.1 INTRODUCTION**

Online shopping systems, also known as e-commerce platforms, are digital marketplaces where consumers can purchase goods and services over the internet. These systems have become increasingly popular in recent years due to their convenience, accessibility, and ease of use. An online shopping system typically involves a website or mobile application where users can browse and purchase products from a variety of vendors. Transactions are processed electronically, and products are shipped directly to the consumer's address. Online shopping systems have revolutionized the retail industry, enabling consumers to shop from anywhere at

any time, and giving businesses access to a global customer base. However, these systems also present challenges related to security, privacy, and fraud prevention.

## **PURPOSE OF THE DOCUMENT**

The purpose of an online shopping system is to provide customers with a convenient and efficient way to purchase goods and services over the internet. With an online shopping system, customers can browse through a wide selection of products, compare prices and features, make purchases, and track their orders all from the comfort of their own homes. For businesses, an online shopping system can provide a cost-effective way to reach a wider audience and increase sales. It also allows businesses to gather valuable data about their customers and their buying habits, which can be used to improve marketing strategies and customer service. Overall, the purpose of an online shopping system is to provide a seamless and enjoyable shopping experience for customers while streamlining the buying process for businesses.

## **SCOPE OF THE DOCUMENT**

The scope of an online shopping system is vast and can vary depending on the business needs and goals. However, here are some common features and benefits of an online shopping system:

- **Convenience:** Online shopping systems provide customers with the convenience of shopping from anywhere and at any time without the need to physically visit a store.
- **Wider reach:** An online shopping system enables businesses to expand their reach beyond their physical location and cater to customers worldwide.
- **Cost-effective:** Operating an online shopping system can be less costly than running a brick- and-mortar store.
- **Personalization:** Online shopping systems can use customer data to provide personalized shopping experiences, including recommendations and targeted promotions.
- **Faster transactions:** Online shopping systems enable quick and easy transactions, allowing customers to purchase products with just a few clicks.
- **Inventory management:** Online shopping systems can help businesses manage their inventory more efficiently and accurately.

- Analytics and insights: Online shopping systems can provide businesses with valuable data and insights about customer behavior, sales trends, and product performance.

## **OVERVIEW**

An online shopping system is a web-based application that enables customers to purchase goods and services over the internet. The system includes various components such as a website, a database, a payment gateway, and an order management system.

The website is the front-end component that customers interact with. It displays the products, prices, and other details about the items for sale. Customers can search for products, add them to their shopping cart, and proceed to checkout. The website should also include features like product reviews, ratings, and recommendations to help customers make informed decisions.

### **2.2.2 GENERAL DESCRIPTION**

An online shopping system is a web-based platform that allows customers to purchase products or services over the internet. This system typically consists of a website or application where customers can browse products, add items to their shopping cart, and check out by making a payment through various payment methods.

The online shopping system typically includes features such as product search and filtering, product categorization, product reviews and ratings, shopping cart management, payment processing, order tracking, and customer support. It also provides sellers with tools to manage their inventory, pricing, and shipping.

### **2.2.3 FUNCTIONAL REQUIREMENTS**

The functional requirements of an online shopping system typically include the following features:

1. User Registration and Login: The system should provide user registration and login functionality to allow customers to create accounts and securely access the system.
2. Product Catalog: The system should include a product catalog that displays all available products, their descriptions, prices, and images. Customers should be able to browse and search for products based on different categories, such as brand, type, color, size, etc.

3. Shopping Cart: The system should allow customers to add products to their shopping cart and view the contents of the cart at any time. The cart should also display the total price and any applicable taxes, shipping fees, or discounts.
4. Checkout: The system should provide a checkout process that allows customers to review their orders, enter their shipping and billing information, choose their payment method, and confirm their purchase.
5. Payment Gateway: The system should integrate with one or more payment gateways to process payments securely and efficiently. Payment options may include credit/debit cards, PayPal, Stripe, etc.
6. Order Management: The system should allow administrators to manage and track orders, update order status, and notify customers of order updates and shipping information.
7. User Profile: The system should allow customers to view and update their profile information, including their contact details, shipping addresses, and order history.
8. Feedback and Reviews: The system should provide a feedback and review system to allow customers to rate and review products, as well as provide feedback on their shopping experience.
9. Search Functionality: The system should provide an advanced search functionality to allow customers to quickly find products based on their specific criteria.
10. Admin Panel: The system should provide an admin panel to manage product listings, user accounts, orders, and other system settings. Administrators should be able to add, edit, and delete products, as well as manage user accounts and view reports and analytics.

#### **2.2.4 INTERFACE REQUIREMENTS**

The interface requirements for an online shopping system may vary depending on the specific needs and goals of the system. However, here are some general interface requirements that can help improve the user experience:

1. Intuitive and user-friendly interface: The interface should be easy to navigate, with clear and concise labels, and easily identifiable buttons.



2. Responsive design: The interface should be responsive and accessible across a range of devices and screen sizes, including mobile phones and tablets.
3. Clear product display: The product images should be clear and high-quality, with detailed descriptions of each item.
4. Shopping cart functionality: The interface should allow users to easily add items to their shopping cart and modify the quantity of items in their cart.
5. Secure payment gateway: The interface should ensure secure transactions by incorporating a trusted payment gateway with strong encryption and verification processes.
6. Order tracking: The interface should enable users to track the status of their orders, including shipping and delivery details.
7. Customer support: The interface should provide users with easy access to customer support, including email, phone, and chat options.
8. Personalization: The interface should provide personalized recommendations and offers based on the user's browsing and purchase history.
9. Social media integration: The interface should allow users to share products on social media platforms and read reviews from other customers.
10. Multiple language support: The interface should support multiple languages to cater to a diverse user base.

#### 2.2.5 PERFORMANCE REQUIREMENTS

Performance requirements of an online shopping system are critical to ensure that the system can handle the expected traffic and provide a satisfactory user experience. Here are some key performance requirements:

1. Scalability: The system should be able to handle a large number of concurrent users and transactions without affecting performance. It should also be able to scale up or down as needed to accommodate changes in traffic.
2. Response time: The system should respond quickly to user requests, such as page loads, searches, and checkout processes. Ideally, the response time should be less than 3 seconds.

3. Availability: The system should be available 24/7 with minimal downtime for maintenance or upgrades.
4. Security: The system should be secure and protect user data from unauthorized access, hacking, and other cyber threats.
5. Reliability: The system should be reliable and ensure that transactions are completed accurately and without errors.
6. Load testing: The system should undergo load testing to ensure that it can handle the expected traffic levels and can maintain performance during peak times.
7. Usability: The system should be user-friendly and easy to navigate, with clear instructions and intuitive design.
8. Mobile compatibility: The system should be mobile-responsive and provide an optimized user experience on smartphones and tablets.

#### **2.2.6 NON-FUNCTIONAL REQUIREMENTS**

Non-functional requirements for an online shopping system can include:

1. Usability: The system should be easy to use and navigate, with a clear and intuitive user interface.
2. Performance: The system should be able to handle a large number of simultaneous users, with minimal downtime or slow response times.
3. Security: The system should be secure, with measures in place to protect user data and prevent unauthorized access.
4. Reliability: The system should be reliable, with a high level of uptime and minimal errors or failures.
5. Scalability: The system should be able to scale up or down to accommodate changes in user traffic and usage patterns.
6. Accessibility: The system should be accessible to users with disabilities, with features such as text-to-speech and high-contrast modes.

7.      Compatibility: The system should be compatible with different devices, browsers, and operating systems, to ensure that all users can access it.
8.      Maintainability: The system should be easy to maintain, with clear documentation, modular code, and regular updates to address bugs and security issues.
9.      Performance: The system should load quickly and be able to handle a large number of simultaneous users without slowing down.
10.     Availability: The system should be available to users 24/7 with minimal downtime for maintenance or upgrades.

### **2.2.7 DESIGN CONSTRAINTS**

There are several design constraints that should be considered when developing an online shopping system. Some of the most important ones are:

1.      Usability: The system should be easy to use and navigate, with a user-friendly interface that allows customers to quickly find and purchase the products they need.
2.      Security: Online shopping systems must be designed with strong security measures to protect customer data and financial information. This may include encryption, secure login protocols, and regular security audits.
3.      Scalability: As the number of customers and products grows, the online shopping system must be able to handle increasing traffic and transaction volumes without slowing down or crashing.
4.      Performance: The system should be designed to deliver fast page load times and minimal lag during the checkout process, in order to ensure a smooth and efficient shopping experience.
5.      Compatibility: The system should be designed to work seamlessly across multiple platforms and devices, including desktop computers, mobile devices, and tablets.
6.      Accessibility: The system should be designed to be accessible to users with disabilities, such as those who are visually impaired or have mobility impairments.

7. Integration: The system should be able to integrate with other key systems, such as inventory management and shipping systems, in order to streamline the entire online shopping process.

### **2.2.8 PRELIMINARY SCHEDULE AND BUDGET**

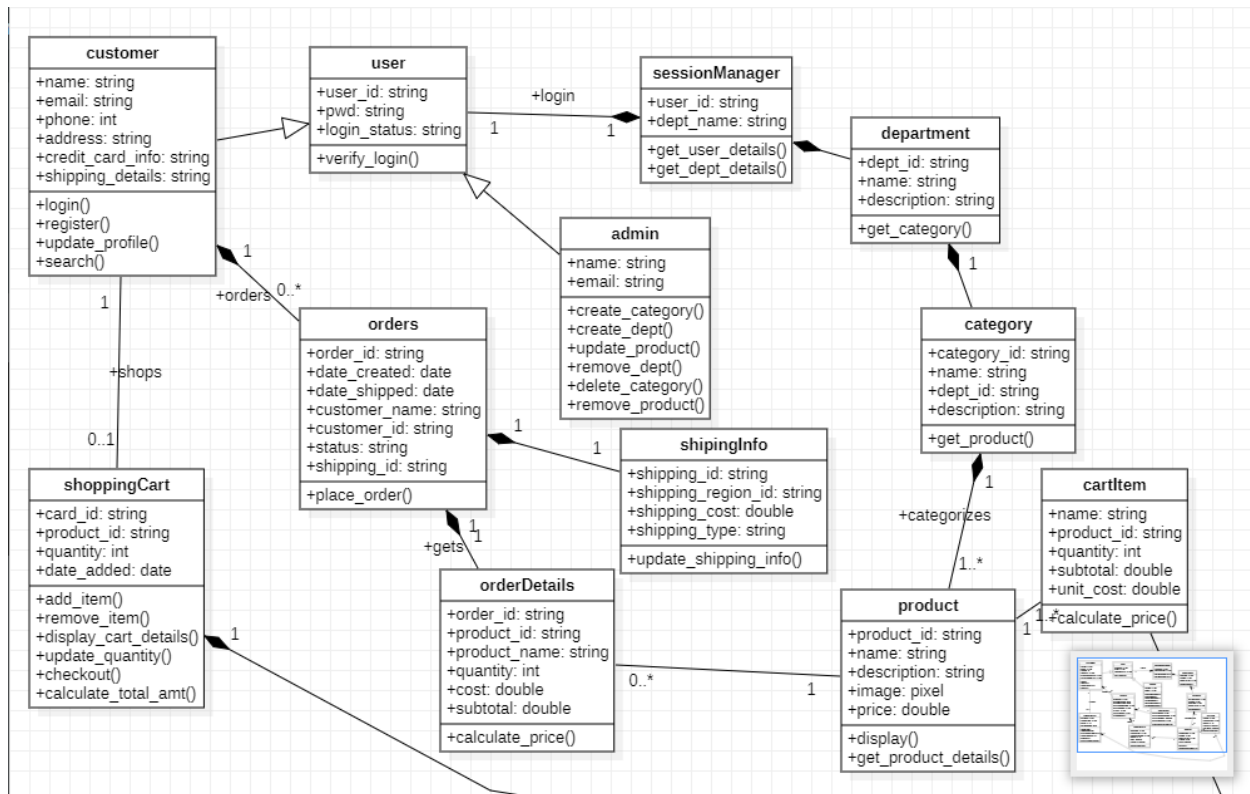
#### **SCHEDULE:**

1. Requirements gathering and analysis (1-2 weeks)
2. Design and development of the website and backend systems (8-12 weeks)
3. Integration with payment gateways and other third-party services (2-4 week)
4. Testing and quality assurance (2-4 weeks)
5. Launch and deployment (1-2 weeks)
6. Ongoing maintenance and updates (as needed)

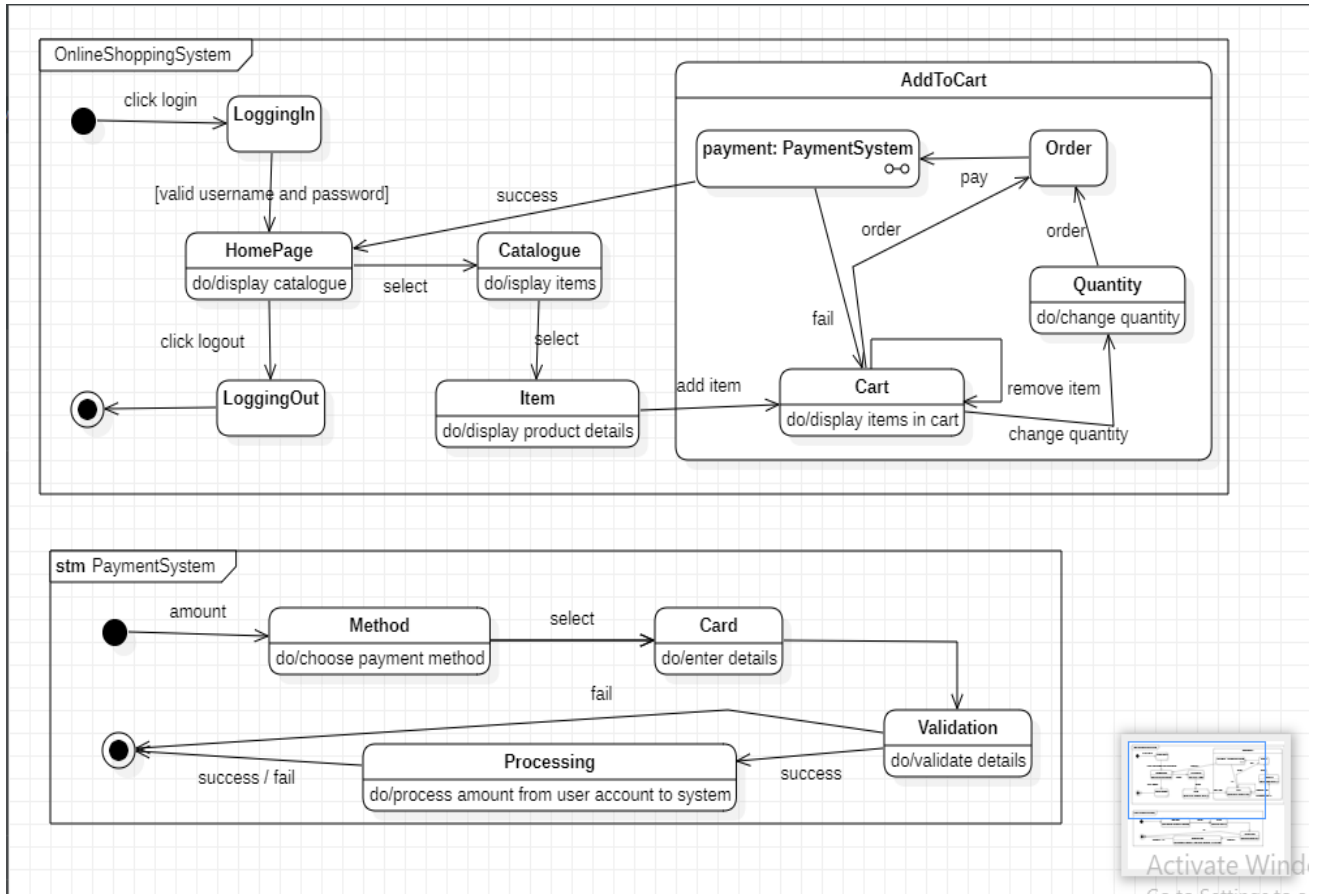
#### **Preliminary Budget:**

1. Salaries and wages for development team: \$100,000 - \$500,000+
2. Hardware and software costs: \$10,000 - \$50,000+
3. Marketing and advertising: \$5,000 - \$50,000+
4. Hosting and domain fees: \$1,000 - \$5,000+
5. Payment processing fees: 2.9% + \$0.30 per transaction
6. Ongoing maintenance and support: 10-20% of development costs per year

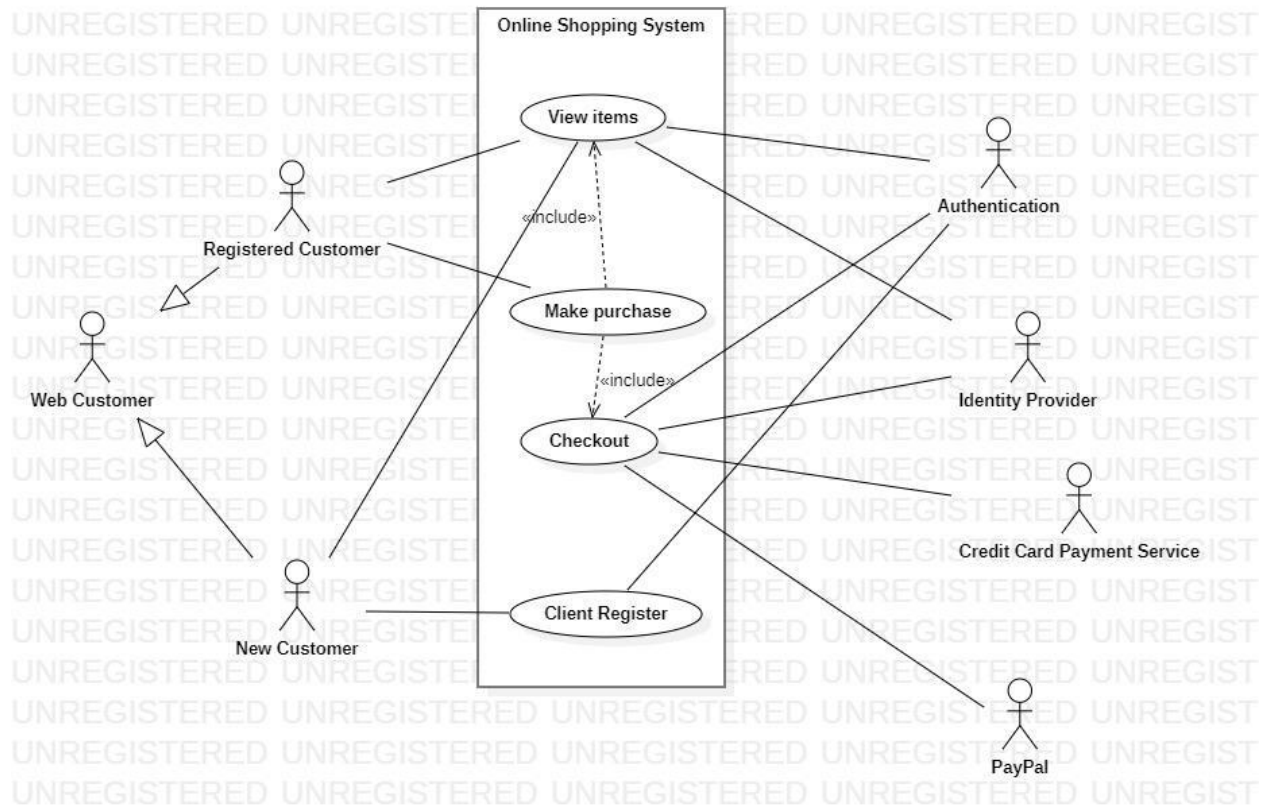
# CLASS DIAGRAM



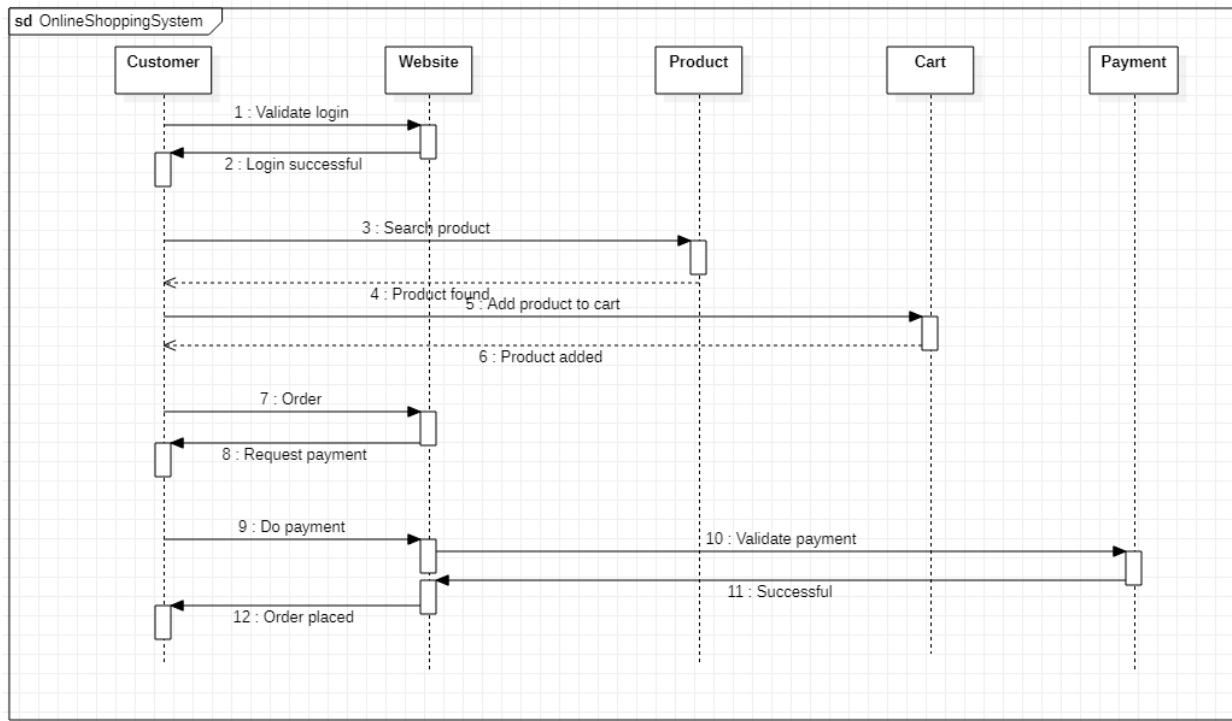
# STATE DIAGRAM



# USE CASE DIAGRAM



# SEQUENCE DIAGRAM





# ACTIVITY DIAGRAM

