

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

# OBJECT ORIENTED MODELLING AND DESIGN

*Submitted by*  
**Jai Shankar K S**  
**(1BM20CS062)**

*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**  
**March-2023 to July-2023**

**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**" was carried out by **Jai Shankar K S (1BM20CS062)**, who is a 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 2023. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Modeling And Design - (20CS6PCOMD)** work prescribed for the said degree.

**Prof. Syed Akram**  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

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

## Index Sheet

<b>Sl. No.</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	Hotel Management System	1-10
2	Credit Card Processing	11-20
3	Library Management System	21-31
4	Stock Maintenance System	32-43
5	Passport Automation System	44-53
6	Railway Reservation System	54-64
7	Online Shopping System	65-75

## Course Outcome

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

# **1. Hotel Management System**

## **Problem Statement:**

The current manual system used for managing hotel operations is inefficient, error-prone, and time-consuming. There are numerous challenges faced by hotel managers, including managing guest reservations, inventory management, billing, and reporting. The lack of an integrated system also leads to data duplication, inconsistencies, and delays in information sharing between departments.

Moreover, the current system does not provide a seamless guest experience, resulting in guest dissatisfaction and negative reviews. The manual processes also require significant human resources, which increases operational costs and reduces profitability.

To address these challenges, a Hotel Management System is required that will automate and integrate all hotel operations, providing a more efficient and effective solution for managing the hotel. The software must be designed to meet the specific needs of the hotel industry, providing a comprehensive and user-friendly solution for managing guest reservations, inventory management, billing, and reporting. The software must also be secure, reliable, and scalable to meet the needs of growing hotels.

# **Software Requirement Specification (SRS)**

## **1 Introduction**

### **1.1 Purpose of this document**

The purpose of this document is to provide a detailed description of the requirements for the development of a Hotel Management System. This document will serve as a reference for the development team, stakeholders, and users, to ensure that the software meets their needs and requirements.

### **1.2 Scope of this document**

The Hotel Management System will be an online software solution for managing various operations in a hotel, such as reservations, room management, billing, inventory management, and reporting. The software will be designed to improve the overall efficiency and accuracy of hotel operations, while also enhancing the guest experience. The development cost and time required will depend on the scope and complexity of the project.

### **1.3 Overview**

The Hotel Management System will provide an integrated solution for hotel operations. The software will include features such as online reservations, check-in and check-out management, room management, billing and payment processing, inventory management, and reporting. The system will be designed to be user-friendly and accessible through web browsers.

## **2 General Description**

The Hotel Management System will provide the following general functions:

### **2.1. Objective of the User:**

The objective of the Hotel Management System is to provide an easy-to-use, efficient, and accurate solution for managing hotel operations.

### **2.2 User Characteristics:**

The Hotel Management System will be designed for use by hotel staff, including front desk agents, managers, and administrators. The software will also be accessible to guests for online reservations, check-in and check-out, and payment processing.

### **2.3 Features and Benefits:**

The Hotel Management System will include features such as: Online reservations Room management Check-in and check-out management Billing and payment processing Inventory management Reporting The benefits of the system will include improved efficiency, accuracy, and guest experience.

### **2.4 User Community:**

The Hotel Management System will be used by hotel staff and guests, as well as administrators and stakeholders.

## **3 Functional Requirements**

The functional requirements of the Hotel Management System will include: Online reservations: The software will allow guests to book rooms online, with the ability to view room availability and rates. Room management: The system will allow hotel staff to manage room assignments, room types, and room availability. Check-in and check-out management: The software will allow hotel staff to manage guest check-in and check-out processes, including room assignments and billing. Billing and payment processing: The system will allow for accurate and efficient billing and payment processing, including the ability to process multiple payment types.

Inventory management: The software will allow for inventory management of hotel supplies, such as towels, linens, and toiletries. Reporting: The system will provide detailed reports on hotel operations, including occupancy rates, revenue, and expenses.

## **4 Interface Requirements**

The Hotel Management System will include interfaces for the following:

- Web-based user interface for hotel staff and guests
- Payment processing interfaces for multiple payment types
- Inventory management interfaces for tracking and managing inventory levels

## **5 Performance Requirements**

The Hotel Management System will be designed to perform efficiently under the following conditions:

- High traffic periods: The software must be able to handle high volumes of online reservations and guest check-ins and check-outs during peak periods.
- Data processing: The system must be able to process large volumes of data quickly and accurately.
- System availability: The software must be available to users at all times, with minimal downtime for maintenance and upgrades.

## **6 Design Constraints**

The following design constraints will apply to the Hotel Management System:

- Use of secure payment processing protocols to ensure the safety and security of guest data.

Integration with existing hotel systems, such as accounting and inventory management software.

Compliance with data privacy and security regulations, such as GDPR and PCI-DSS.

## **7 Non-Functional Attributes**

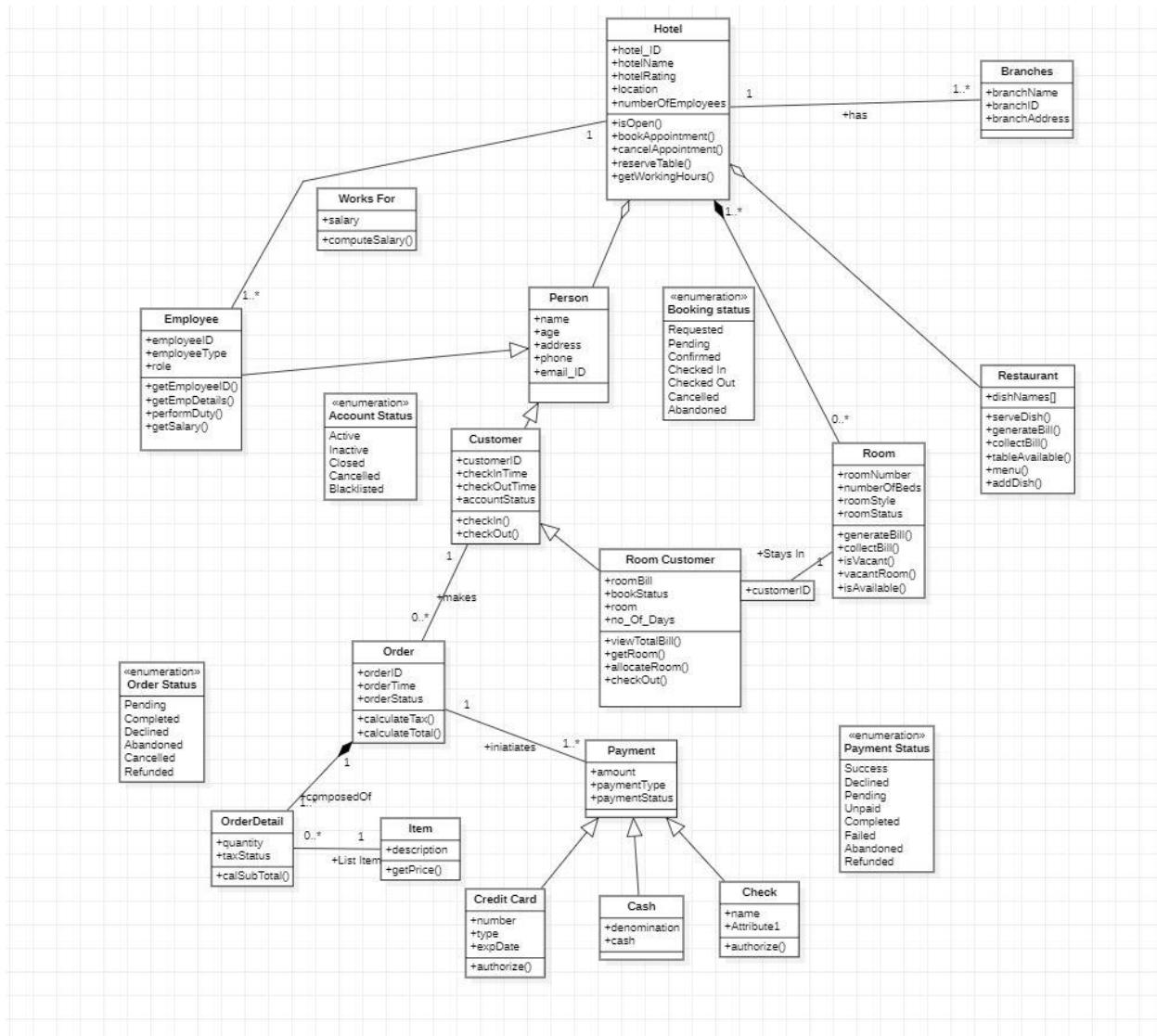
The following non-functional attributes are required for the Hotel Management System:

- Security: The system must ensure the confidentiality, integrity, and availability of guest data, as well as protect against unauthorized access and data breaches.
- Portability: The software must be compatible with multiple web browsers and operating systems.
- Reliability: The system must be reliable, with minimal downtime or system errors.
- Reusability: The software code should be designed for reusability and modularity, to facilitate future updates and maintenance.
- Application compatibility: The software must be compatible with existing hotel systems and software.
- Data integrity: The system must ensure the accuracy and consistency of data across all modules and interfaces.
- Scalability capacity: The software must be designed to scale to accommodate increasing volumes of data and users.

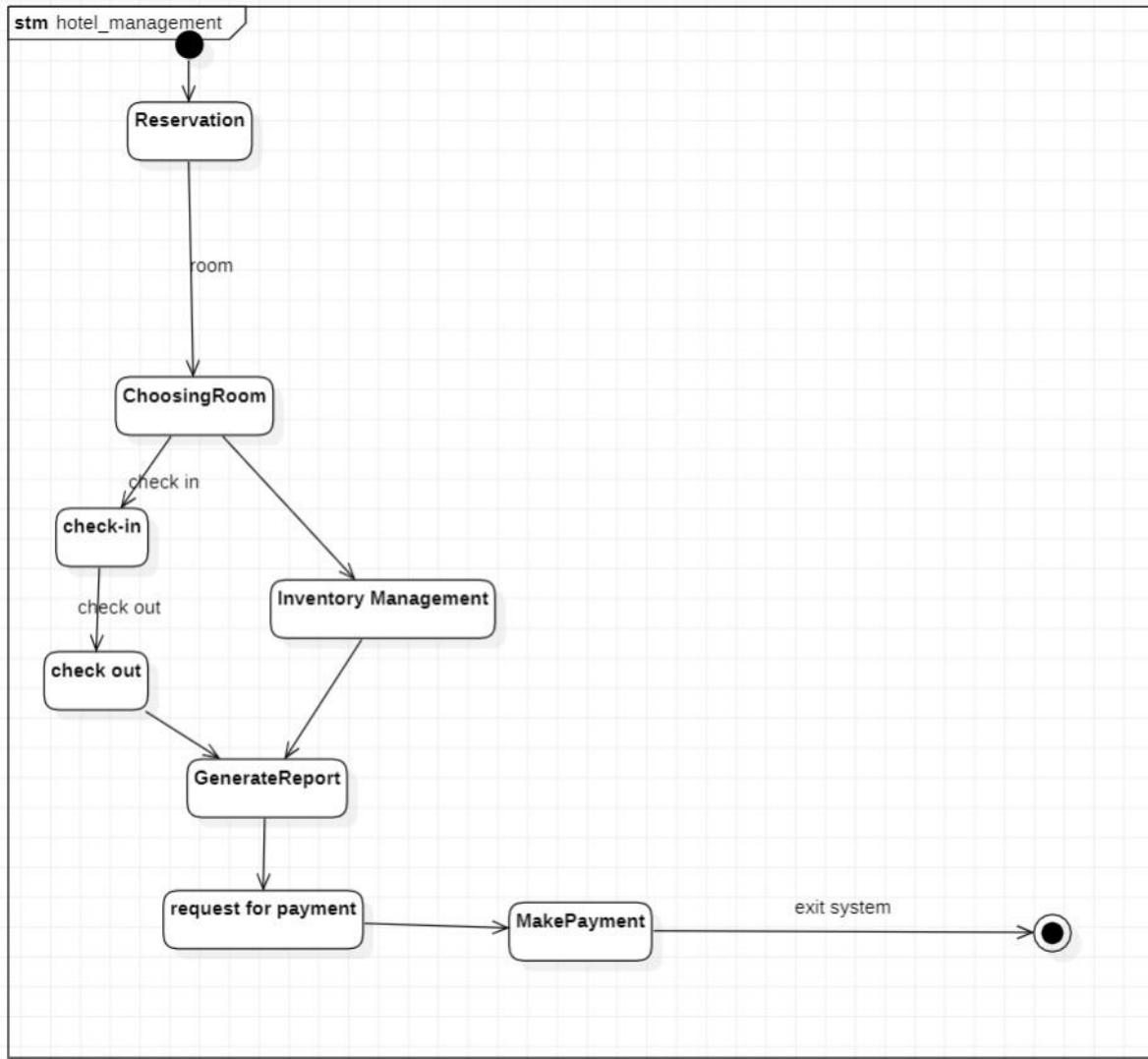
## **8 Preliminary Schedule and Budget**

The development of the Hotel Management System is expected to take approximately 12 months, with an estimated budget of \$500,000. The development team will work in an agile development environment, with regular sprints and iterations to ensure the software meets the requirements of stakeholders and users. In conclusion, the Hotel Management System will provide a comprehensive and integrated solution for managing hotel operations, improving efficiency, accuracy, and guest experience. The software will be designed to meet the functional and non-functional requirements of stakeholders and users, with a focus on security, reliability, and scalability.

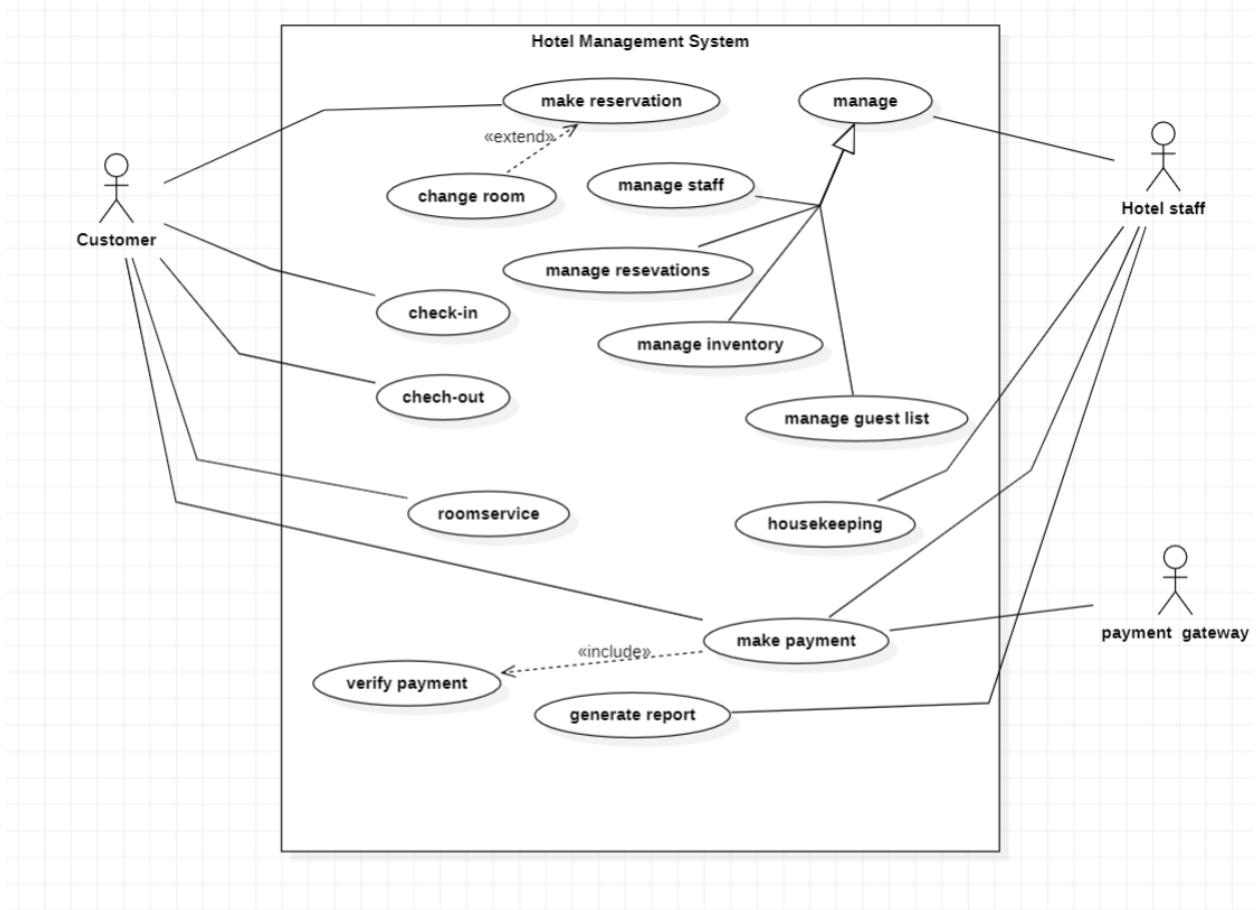
# Class Diagram



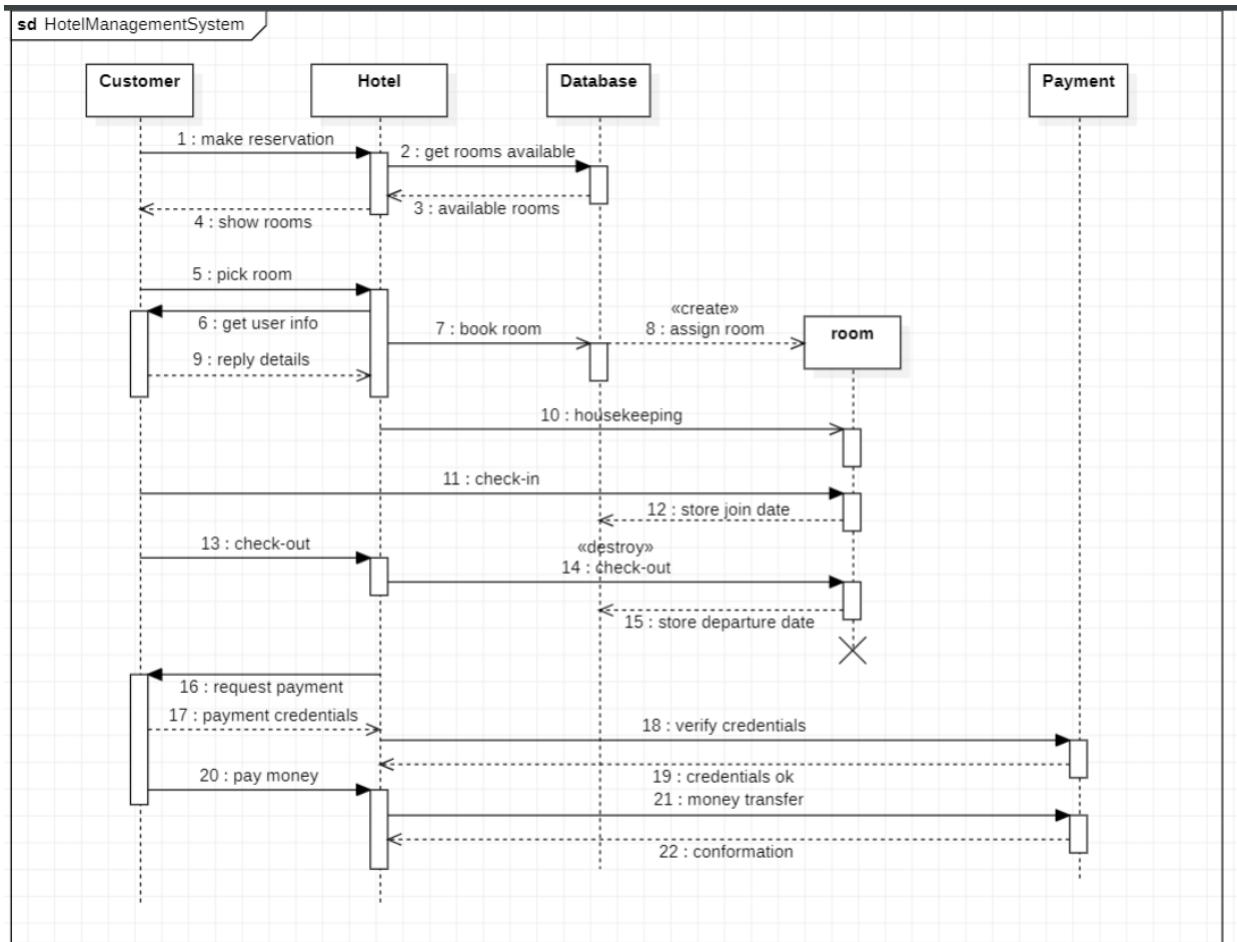
## State Diagram



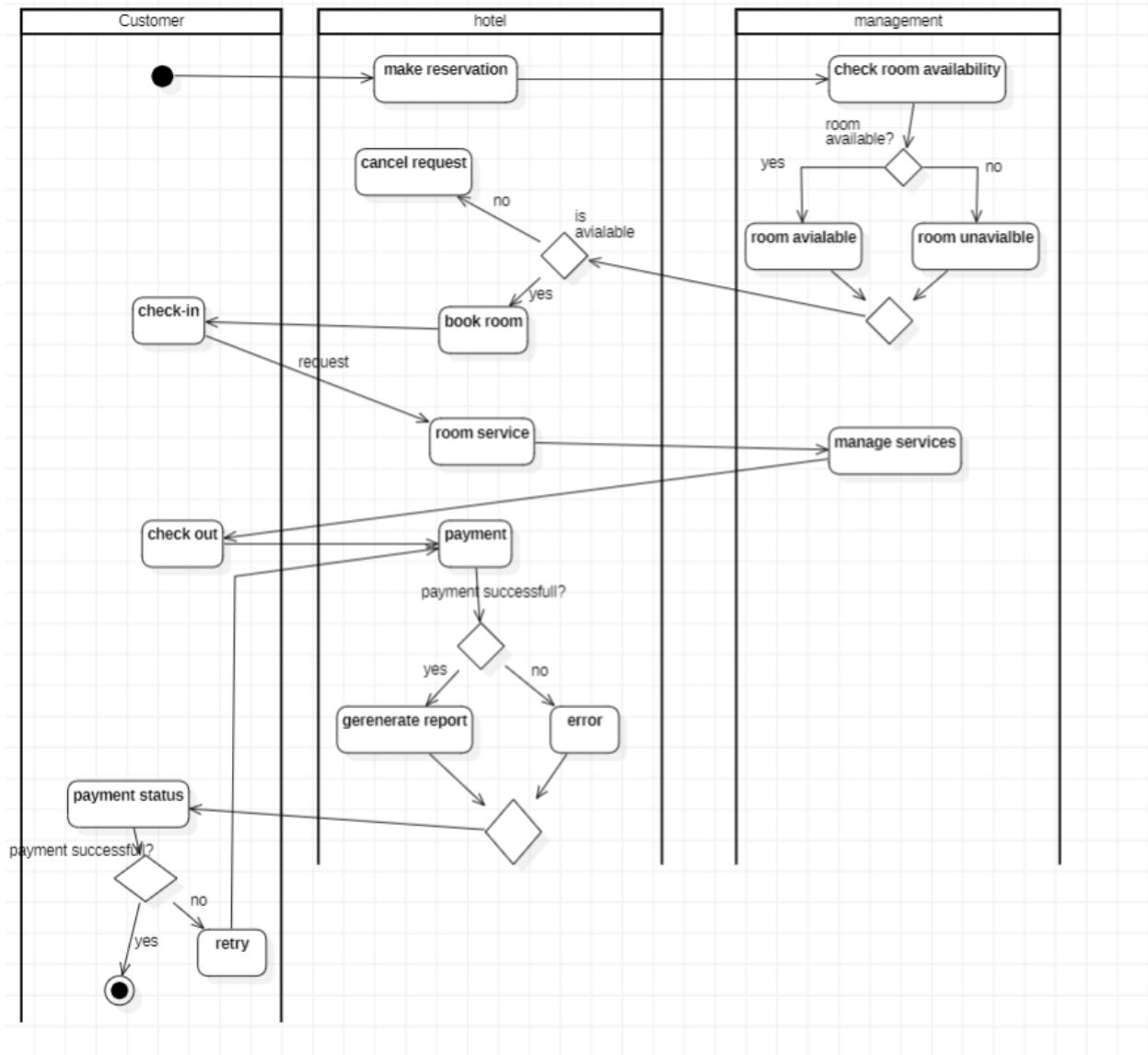
## Use Case Diagram



## Sequence Diagram



## Activity Diagram



## **2. Credit Card Processing**

### **Problem Statement:**

Credit card processing refers to the process of verifying and authorizing credit card transactions made by consumers for goods and services. The process involves several parties, including the cardholder, merchant, acquiring bank, and issuing bank.

The problem with credit card processing arises when transactions are fraudulent, leading to losses for merchants and financial institutions. Fraud can occur in several ways, including stolen credit card information, identity theft, and counterfeit cards.

Additionally, credit card processing can also be challenging for merchants due to high processing fees, chargebacks, and delays in receiving payment. This can result in financial strain and a negative impact on their business.

Therefore, the problem statement for credit card processing can be defined as finding efficient and secure ways to process credit card transactions, reducing fraud and associated losses, and minimizing processing fees and delays for merchants while ensuring a positive customer experience.

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

### **1 Introduction**

#### **1.1 Purpose of this document**

The purpose of this document is to define the software requirements for a system that aims to address the challenges of credit card processing, as described in the problem statement. The document will outline the functional and nonfunctional requirements for the system, providing a clear understanding of the features and capabilities it should have in order to meet

the needs of the users and stakeholders involved in credit card processing. It will serve as a reference for the development team to design, build, test, and deploy the system.

## **1.2 Scope of this document**

The scope of this document is to provide a detailed specification of the requirements for efficient and secure credit card processing. It outlines the functional and non-functional requirements for credit card processing, including but not limited to, the verification and authorization of transactions, fraud prevention measures, processing fees, chargebacks, and payment delays.

## **1.3 Overview**

The document defines the stakeholders, functional and non-functional requirements, and constraints of the system. The purpose of this document is to provide a comprehensive understanding of the system to the development team, stakeholders, and other relevant parties involved in the development, testing, and maintenance of the credit card processing system. This document is intended to serve as a guide and reference for the entire software development life cycle (SDLC) of the system.

# **2 General Description**

The Hotel Management System will provide the following general functions:

## **2.1 Objective of the User:**

The objective of the user is to make secure and efficient credit card transactions for goods and services.

## **2.2 User Characteristics:**

The users of the credit card processing system include cardholders, merchants, acquiring banks, and issuing banks. Cardholders are individuals who possess credit cards and use them to make purchases. Merchants are businesses that accept credit card payments from customers. Acquiring

banks provide payment processing services to merchants, while issuing banks issue credit cards to cardholders.

### **2.3 Features and Benefits:**

The credit card processing system provides several features and benefits, including: Efficient and secure processing of credit card transactions Fraud prevention and detection mechanisms to reduce losses for merchants and financial institutions Fast and reliable payment processing for merchants Seamless integration with existing merchant systems User-friendly interfaces for cardholders and merchants 24/7 customer support for any issues related to credit card processing

### **2.4 User Community:**

The user community for the credit card processing system includes all individuals and businesses that use credit cards for transactions. This includes cardholders, merchants, acquiring banks, and issuing banks. The system aims to provide a positive user experience for all members of the user community and ensure the security and efficiency of credit card transactions.

## **3 Functional Requirements**

The credit card processing system will have the following functional requirements:

- **User Authentication:** The system should provide a secure way for cardholders to authenticate themselves before making a transaction. This can be done through the use of passwords, biometric authentication, or other secure means.
- **Transaction Authorization:** The system should verify the authenticity of the transaction and the cardholder's ability to make the payment. This can be done through the use of card verification codes, address verification, and other fraud prevention measures.
- **Merchant Integration:** The system should be able to integrate with various merchant platforms and point-of-sale systems to enable seamless payment processing. Payment

**Gateway:** The system should provide a secure payment gateway to facilitate the transfer of funds from the cardholder to the merchant's account.

- **Fraud Detection:** The system should be equipped with advanced fraud detection capabilities to prevent unauthorized transactions and identify potential fraudulent activities.
- **Chargeback Management:** The system should be able to manage chargebacks and disputes between cardholders and merchants in a fair and efficient manner.
- **Payment Processing Fees:** The system should be able to calculate and process payment processing fees in a transparent and fair manner

## 4 Interface Requirements

- **User interface:** The system shall provide an intuitive and user-friendly interface for cardholders, merchants, and financial institutions to initiate and manage credit card transactions.
- **Communication protocol:** The system shall use a secure and reliable communication protocol for transmitting credit card information between parties, such as SSL/TLS.
- **Data formats:** The system shall support standard data formats for credit card information, such as ISO 8583, and adhere to the data security standards set by the Payment Card Industry Data Security Standard (PCI DSS).
- **Integration with payment gateways:** The system shall integrate with popular payment gateways, such as PayPal and Stripe, to enable merchants to process credit card transactions online.
- **Integration with point-of-sale (POS) systems:** The system shall integrate with popular POS systems, such as Square and Clover, to enable merchants to process credit card transactions in-person.
- **Error handling:** The system shall provide clear error messages and notifications to users in case of any errors or issues during the credit card processing.
- **Accessibility:** The system shall provide accessibility features, such as support for screen readers and keyboard navigation, to enable users with disabilities to access the system

## **5 Performance Requirements**

- Response Time: The system should provide a fast response time to users for processing credit card transactions. The maximum response time should be less than 5 seconds. Transaction
- Throughput: The system should be able to process a high volume of transactions simultaneously. The minimum transaction throughput should be at least 500 transactions per minute.
- Availability: The system should be available 24/7 and should have a minimum uptime of 99.99%. This means that the system can only be down for a maximum of 5.26 minutes per year.
- Reliability: The system should be reliable and should not fail during a transaction. The system should have a mean time between failures (MTBF) of at least 10,000 hours.
- Security: The system should provide a secure environment for processing credit card transactions. The system should comply with Payment Card Industry Data Security Standards (PCI DSS) and should have appropriate security measures such as encryption, firewalls, and intrusion detection and prevention systems.
- Scalability: The system should be scalable and able to handle an increasing number of transactions as the user base grows. The system should be able to handle at least 20% growth in transaction volume per year.
- Compatibility: The system should be compatible with different operating systems and browsers. The system should be accessible from different devices such as laptops, desktops, and mobile devices. The system should support all major credit card brands such as Visa, Mastercard, American Express, and Discover

## **6 Design Constraints**

- Security: The system must adhere to strict security standards and protocols to ensure that customer data is protected and transactions are secure.
- Compatibility: The system must be compatible with a wide range of credit card types and payment methods to accommodate different customer preferences and needs.

- Regulatory compliance: The system must comply with relevant regulations and standards, such as the Payment Card Industry Data Security Standard (PCI DSS), to ensure that it meets legal and industry requirements.
- Accessibility: The system must be accessible to users with disabilities, such as visually impaired or hearing-impaired users, in compliance with relevant accessibility standards and guidelines.

## 7 Non-Functional Attributes

The following non-functional attributes are required for the Credit card processing:

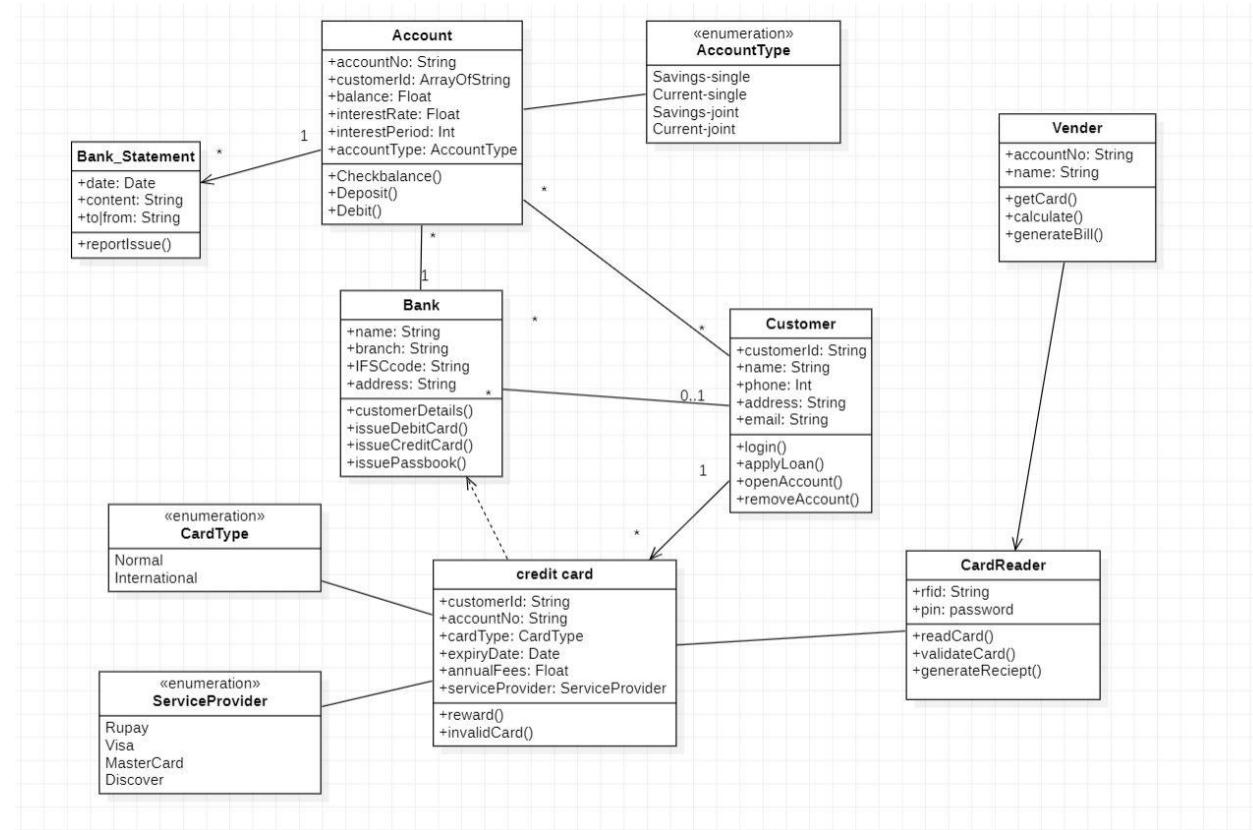
- Security: The system should be secure and protect against unauthorized access, data breaches, and other security threats.
- Scalability: The system should be able to handle an increasing number of transactions as the business grows.
- Maintainability: The system should be easy to maintain and update over time, with minimal disruption to ongoing operations.
- Compliance: The system should comply with relevant regulations and standards, such as PCI DSS.
- Accessibility: The system should be accessible to users with disabilities and comply with relevant accessibility standards

## 8 Preliminary Schedule and Budget

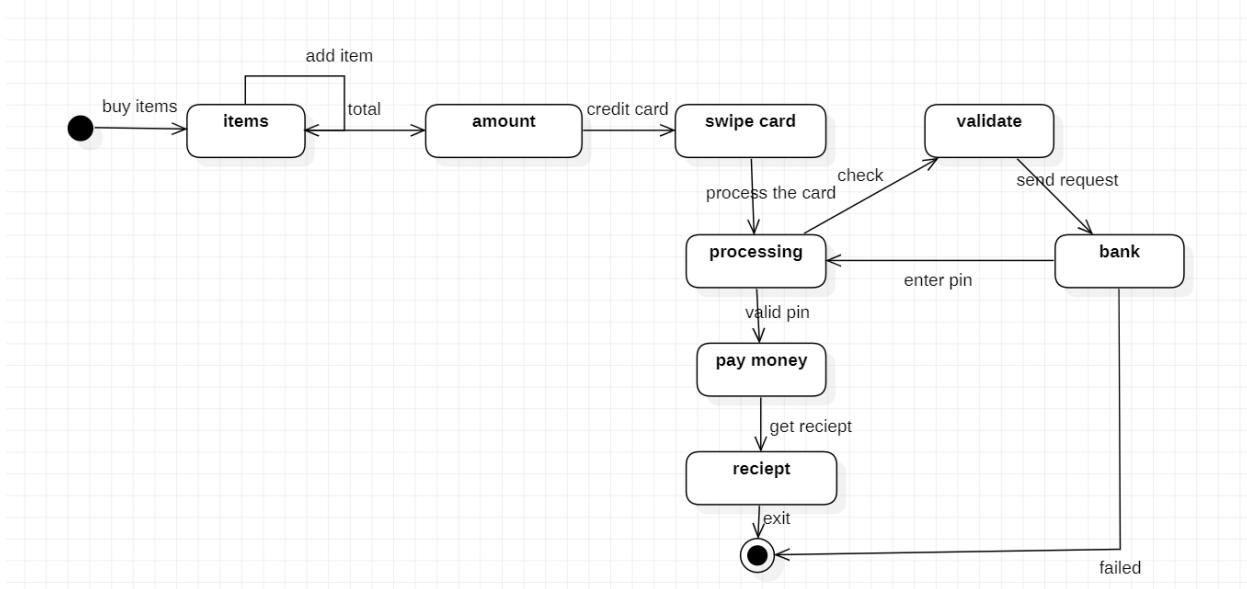
The development of the Credit card processing is expected to take approximately 12 months, with an estimated budget of \$5,000. The development team will work in an agile development

environment, with regular sprints and iterations to ensure the software meets the requirements of stakeholders and users. In conclusion, the Credit card processing will provide a comprehensive and integrated solution for managing Credit card operations, improving efficiency, accuracy, and user experience. The software will be designed to meet the functional and non-functional requirements of stakeholders and users, with a focus on security, reliability, and scalability.

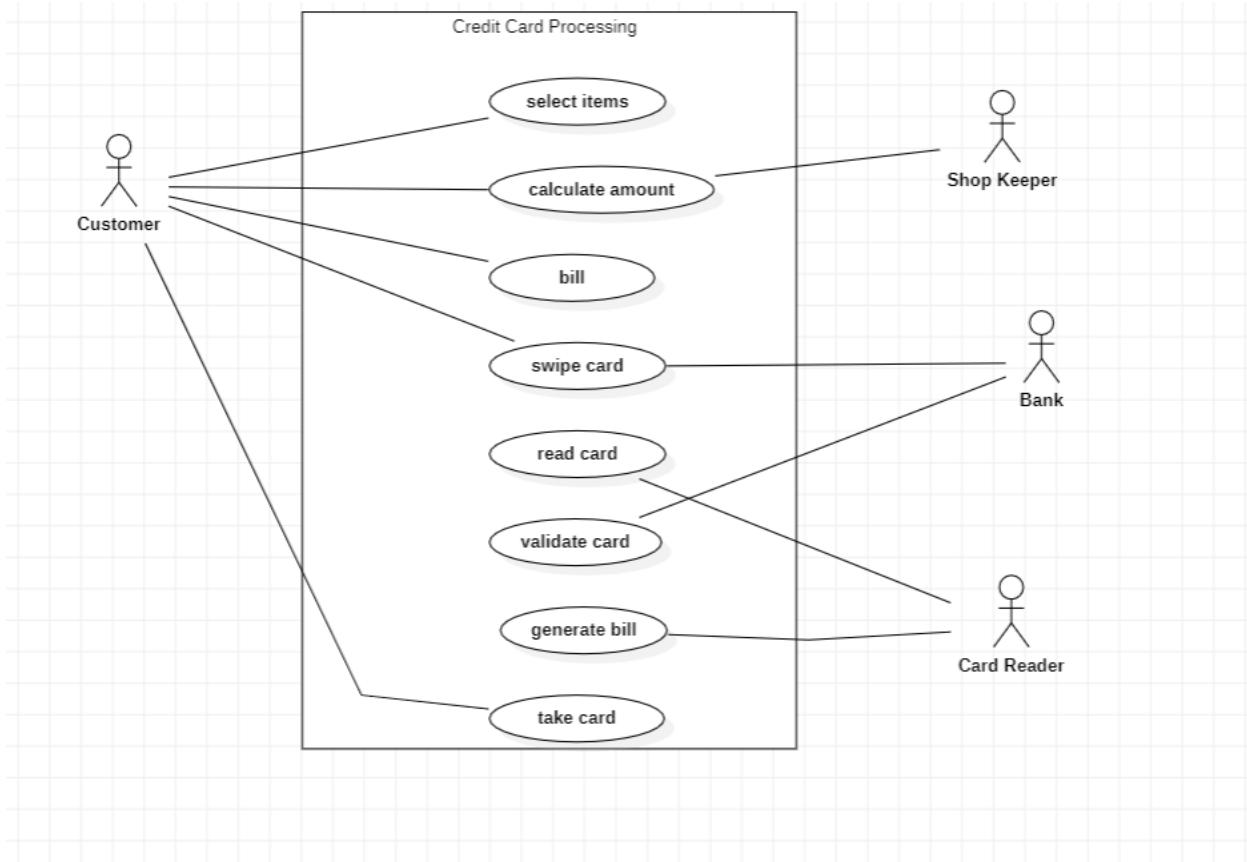
## Class Diagram



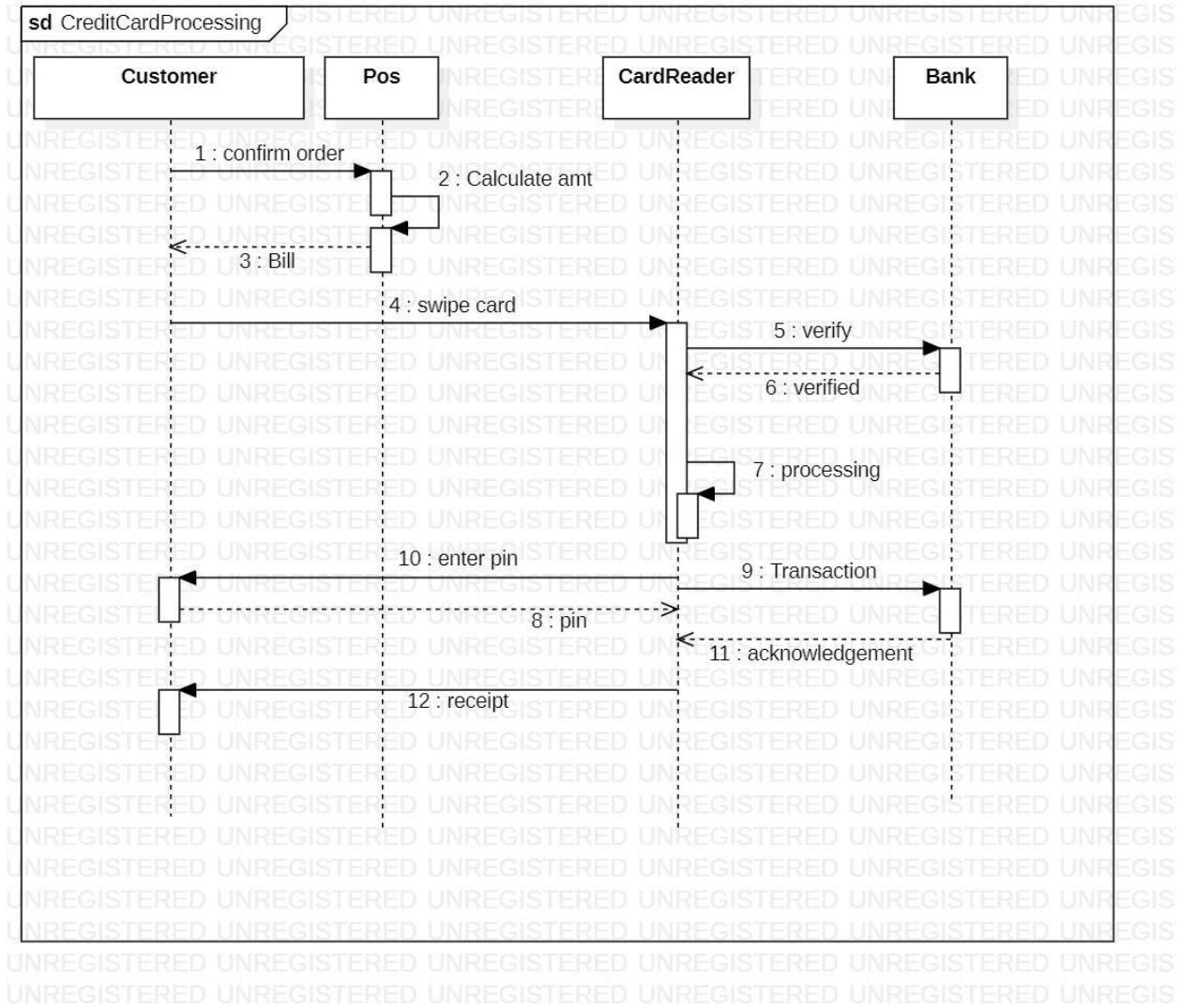
## State Diagram



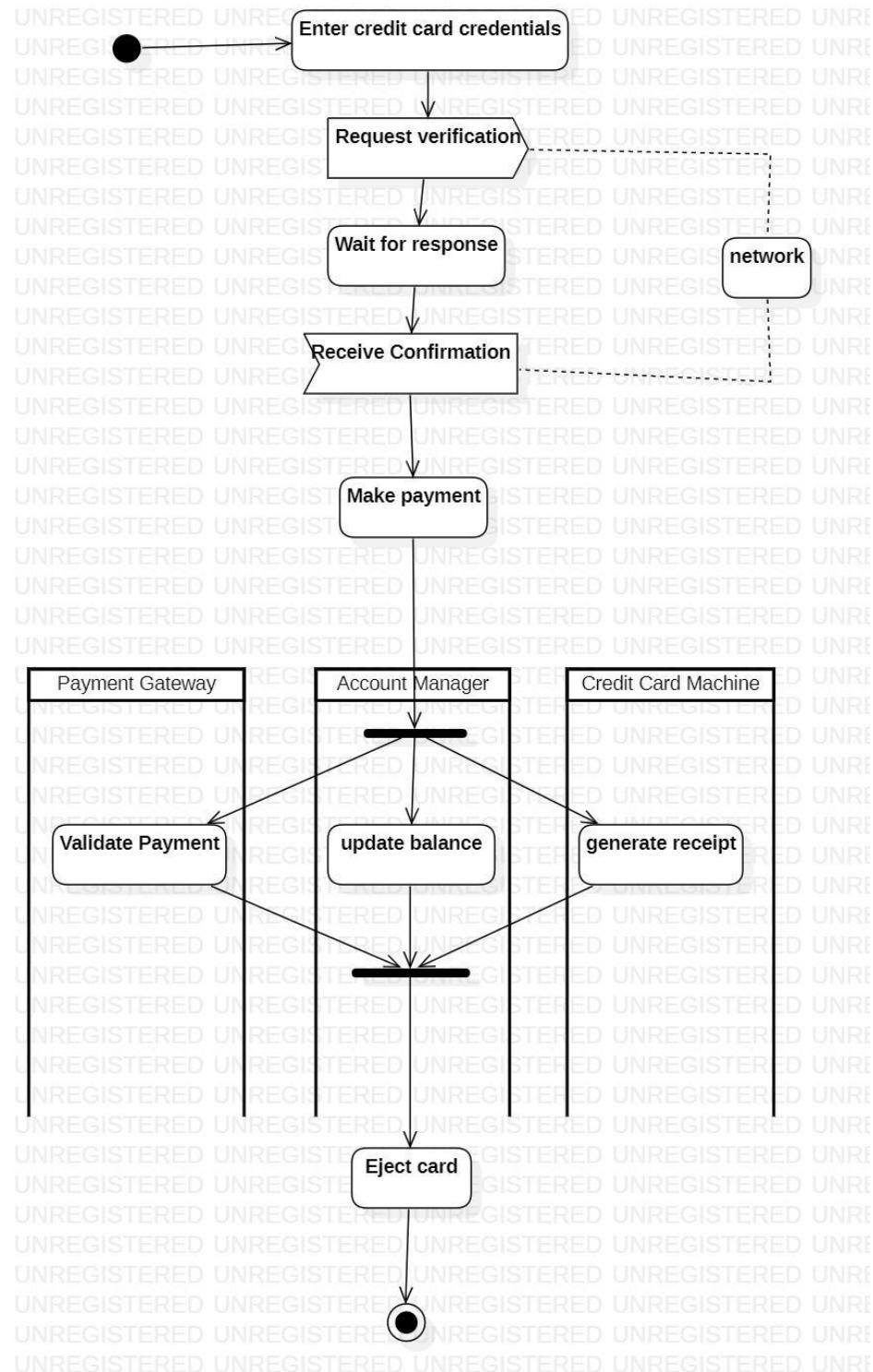
## Use Case Diagram



## Sequence Diagram



## Activity Diagram



### **3. Library Management System**

#### **Problem Statement:**

The current library management system lacks efficiency in tracking the flow of books and providing journals and materials to users. The existing system may suffer from issues such as manual bookkeeping, inaccurate inventory management, delays in book requests and returns, and difficulties in managing journals and other materials. This can lead to confusion, errors, and inconvenience for library staff and users.

Furthermore, the absence of a comprehensive system for managing the library's resources may result in challenges in monitoring overdue books, managing subscriptions and renewals of journals, and keeping track of borrowed materials. This can result in inefficient use of resources, increased workload for library staff, and a subpar experience for library users.

Therefore, the problem statement for the library management system is to develop an efficient and user-friendly solution that can automate and streamline the process of tracking books, managing journals and materials, and providing seamless access to library resources, while ensuring accuracy, efficiency, and convenience for both library staff and users.

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

## **1 Introduction**

### **1.1 Purpose of this document**

The purpose of this document is to define the software requirements for a library management system that aims to address the challenges of tracking books, managing journals and materials, and providing efficient access to library resources. The document will outline the functional and nonfunctional requirements for the system, providing a clear

understanding of the features and capabilities it should have in order to meet the needs of the library staff and users. It will serve as a reference for the development team to design, build, test, and deploy the system.

## **1.2 Scope of this document**

The scope of this document is to provide a detailed specification of the requirements for an efficient and user-friendly library management system. It outlines the functional and nonfunctional requirements for book tracking, journal and material management, and access to library resources, including but not limited to, book borrowing, returns, renewals, overdue management, journal subscriptions, and material requests.

## **1.3 Overview**

The document defines the stakeholders, functional and non-functional requirements, and constraints of the system. The purpose of this document is to provide a comprehensive understanding of the system to the development team, library staff, users, and other relevant parties involved in the development, testing, and maintenance of the library management system. This document is intended to serve as a guide and reference for the entire software development life cycle (SDLC) of the system.

# **2 General Description**

The Library Management System will provide the following general functions:

## **2.1 Objective of the User:**

The objective of the library management system is to automate and streamline the process of tracking books, managing journals and materials, and providing efficient access to library resources for both library staff and users.

## **2.2 User Characteristics:**

The users of the library management system include library staff, who manage the books, journals, and materials, and library users, who borrow and return books, request journals and materials, and renew subscriptions.

### **2.3 Features and Benefits:**

The credit card processing system provides several features and benefits, including:  
Efficient and secure processing of credit card transactions  
Fraud prevention and detection mechanisms to reduce losses for merchants and financial institutions  
Fast and reliable payment processing for merchants  
Seamless integration with existing merchant systems  
User-friendly interfaces for cardholders and merchants  
24/7 customer support for any issues related to credit card processing

### **2.4 User Community:**

The user community for the credit card processing system includes all individuals and businesses that use credit cards for transactions. This includes cardholders, merchants, acquiring banks, and issuing banks. The system aims to provide a positive user experience for all members of the user community and ensure the security and efficiency of credit card transactions.

## **3 Functional Requirements**

- **User Management:** The system should support the registration and management of librarian and member accounts, including login authentication and password management.
- **Book Cataloging:** The system should allow librarians to add, edit, and delete books in the library catalog, including book details such as title, author, ISBN, genre, and availability status.

- Member Registration: The system should allow librarians to register new members, including capturing member details such as name, contact information, and membership type.
- Book Borrowing and Returning: The system should allow members to borrow and return books, with features such as due date calculation, fine calculation for late returns, and automatic book availability status update.
- Book Reservation: The system should allow members to reserve books that are currently unavailable, with features such as reservation queue management and automatic notification when reserved books become available.
- Fine Calculation: The system should automatically calculate fines for late book returns, based on predefined rules such as fine per day, maximum fine amount, and grace period.
- Reporting: The system should generate reports on book inventory, member transactions, and overdue books, as well as support custom report generation based on user-defined criteria.

## 4 Interface Requirements

- User Interface: The system should provide a user-friendly and intuitive interface for librarians and members to perform their tasks efficiently, with features such as easy navigation, search and filtering capabilities, and error handling.
- System Interface: The system should integrate with external systems such as databases, authentication services, and email services, as required for the smooth operation of the LMS.

## 5 Performance Requirements

- Response Time: The system should provide a fast response time to users for processing credit card transactions. The maximum response time should be less than 5 seconds.

- Throughput: The system should be able to process a high volume of transactions simultaneously. The minimum transaction throughput should be at least 500 transactions per minute.
- Availability: The system should be available 24/7 and should have a minimum uptime of 99.99%. This means that the system can only be down for a maximum of 5.26 minutes per year.
- Reliability: The system should be reliable and should not fail during a transaction. The system should have a mean time between failures (MTBF) of at least 10,000 hours.
- Security: The system should provide a secure environment for processing credit card transactions. The system should comply with Payment Card Industry Data Security Standards (PCI DSS) and should have appropriate security measures such as encryption, firewalls, and intrusion detection and prevention systems.
- Compatibility: The system should be compatible with different operating systems and browsers. The system should be accessible from different devices such as laptops, desktops, and mobile devices. The system should support all major credit card brands such as Visa, Mastercard, American Express, and Discover.

## 6 Design Constraints

- Technology Stack: The system should be developed using technologies that are suitable for the intended environment, such as web-based technologies for easy accessibility and cross-platform compatibility.
- Hardware and Software Limitations: The system should be designed to work within the hardware and software limitations of the target environment, such as memory constraints, processing power, and operating system compatibility.
- Security: The system should be designed with appropriate security measures, such as authentication and authorization mechanisms, data encryption, and protection against common security threats such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

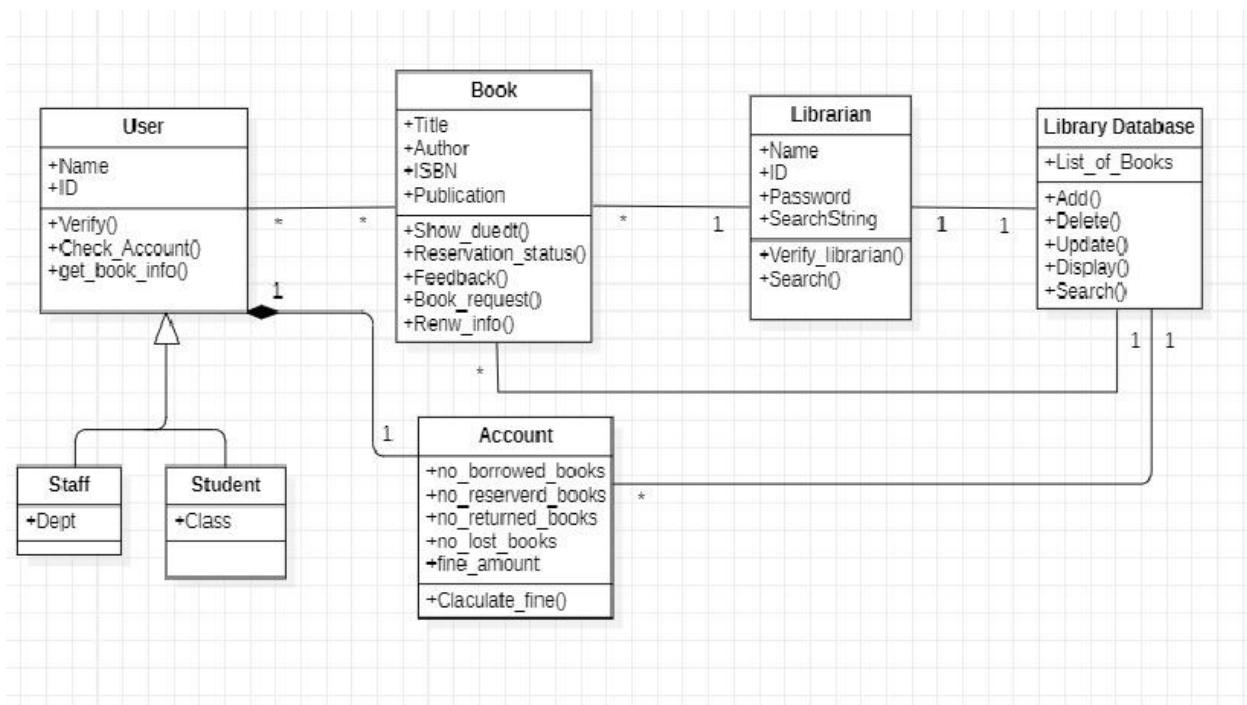
## **7 Non-Functional Attributes**

- Security: The system shall implement robust security measures, including data encryption, user authentication, and role-based access control, to ensure the confidentiality, integrity, and availability of library data and resources.
- Reliability: The system shall be reliable and available 24/7, with minimal downtime and data loss, to ensure uninterrupted access to library resources and services.
- Usability: The system shall be user-friendly and intuitive, with clear navigation, search functionality, and easy-to-use interfaces for both library staff and users, to ensure ease of use and adoption of the system.
- Performance: The system shall have fast response times, minimal latency, and efficient processing of requests, to ensure smooth and efficient performance of the system even during peak usage periods.
- Compatibility: The system shall be compatible with modern web browsers, mobile devices, and operating systems, to ensure accessibility and convenience for library staff and users across different platforms.

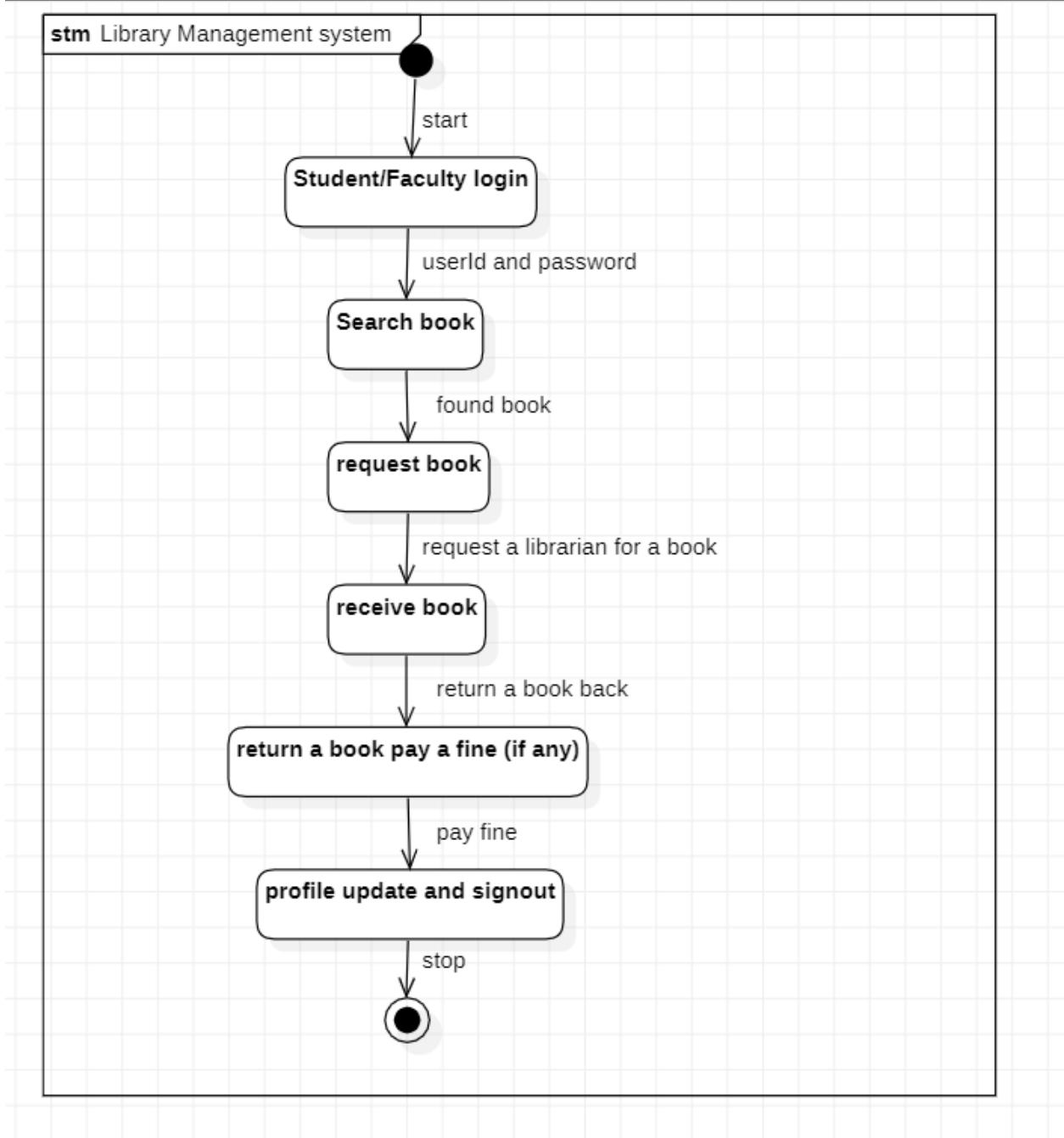
## **8 Preliminary Schedule and Budget**

The development of the Library Management System is estimated to take three months. The project will include design, development, testing, and deployment phases. The project will be managed using agile development methodologies.

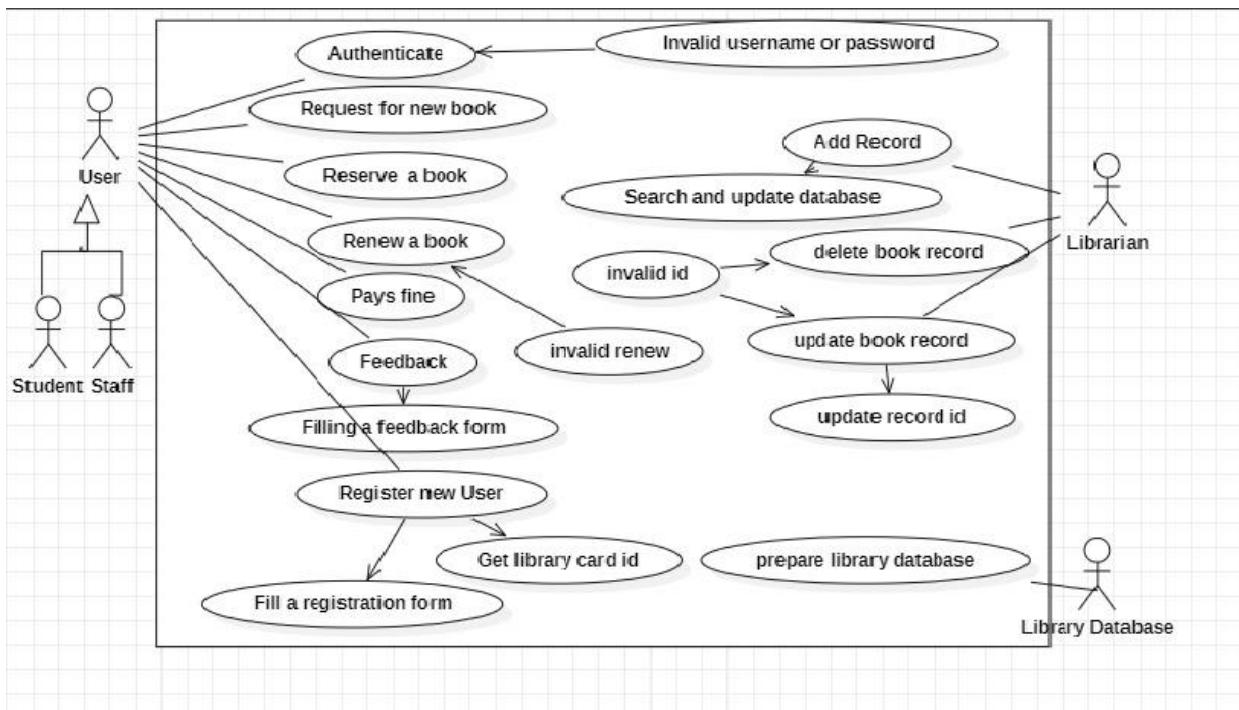
## Class Diagram



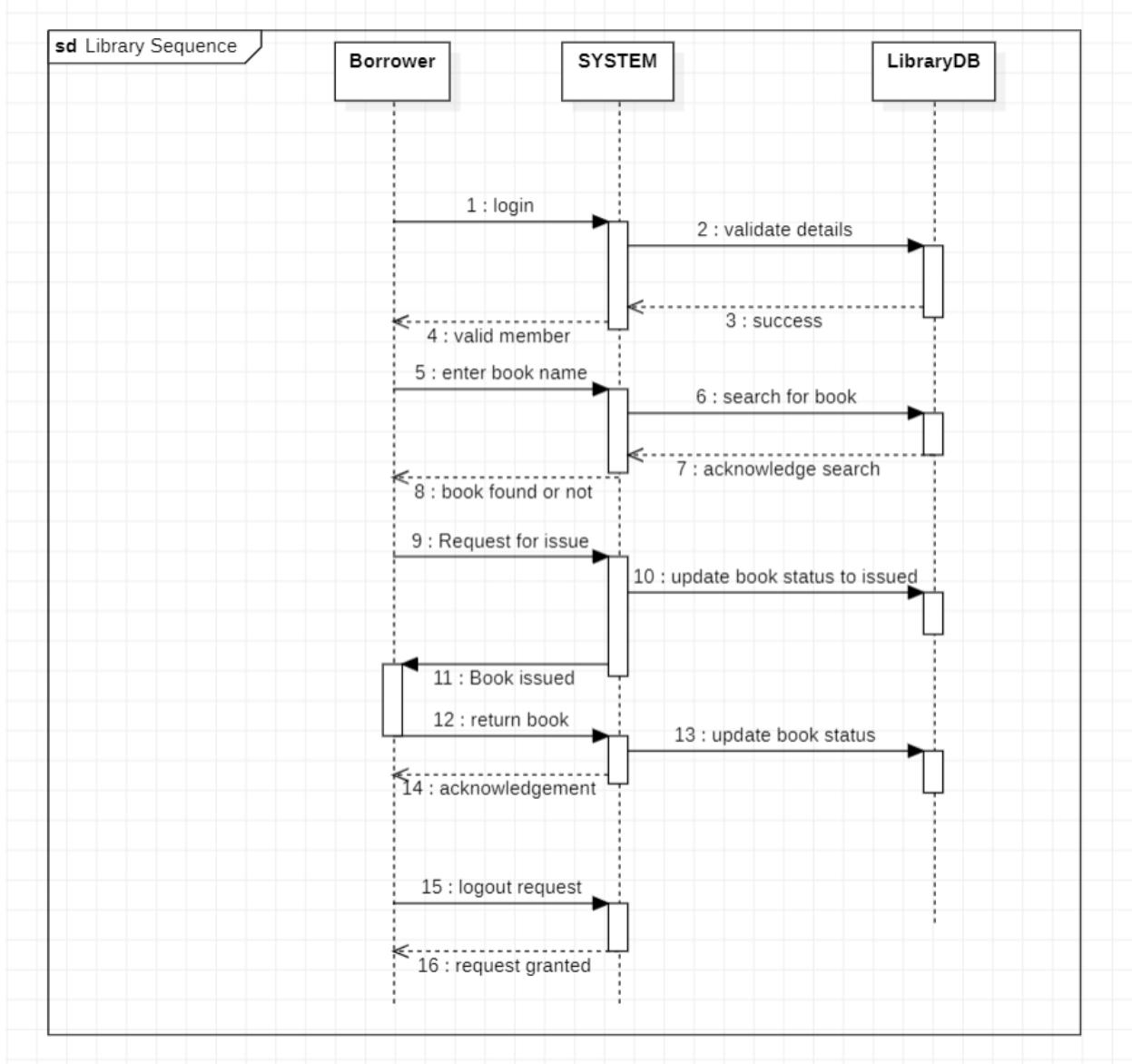
## State Diagram



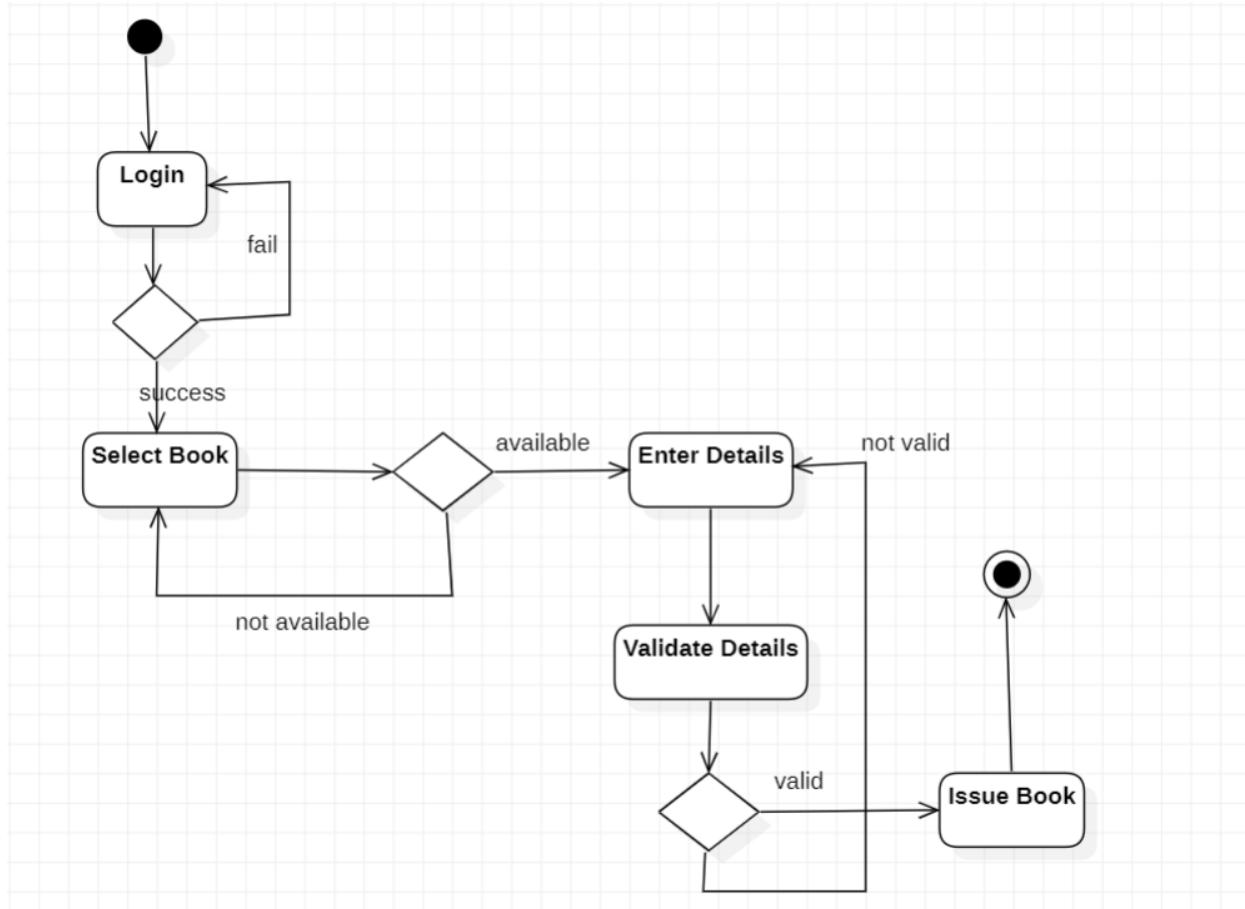
## Use Case Diagram



## Sequence Diagram



## Activity Diagram



## 4. Stock Maintenance System

**Problem Statement:** A company that deals in the distribution of stocks requires an efficient stock maintenance system to keep track of its inventory. The company needs to be able to monitor the quantity and movement of stocks across multiple locations in real time.

The current manual system for tracking inventory is prone to errors and can be time-consuming. This leads to inaccuracies in inventory levels, delayed restocking, and ultimately, lost revenue due to stockouts or overstocking.

The company needs a robust, automated stock maintenance system that can track inventory levels, generate alerts for low stock levels, forecast future inventory needs, and streamline the process of restocking. The system should be user-friendly and accessible to authorized personnel from multiple locations while ensuring data security and accuracy.

The objective of this project is to design and develop a Stock Maintenance System that will automate the process of inventory management, improve accuracy, reduce stock-outs, and ultimately increase revenue for the company.

# **Software Requirement Specification (SRS)**

## **1 Introduction**

### **1.1 Purpose of this document**

The purpose of this document is to define the software requirement specification for the development of a Stock Maintenance System application that will be used to automate the day-to-day transactions in the stock market. This document will outline the functional and non-functional requirements, design constraints, and user interface specifications for the system. This document serves as the reference for the development team to ensure that the system meets the requirements of the clients as well as for the stakeholders.

### **1.2 Scope of this document**

The document outlines the requirements for the Stock Maintenance System, which is designed to automate inventory management for a company that deals in the distribution of stocks. The system will enable the company to track inventory levels, generate alerts for low stock levels, forecast future inventory needs, and streamline the process of restocking. The system will be accessible to authorized personnel from multiple locations, ensuring data security and accuracy.

### **1.3 Overview**

The Stock Maintenance System is an automated inventory management system designed to help the company maintain accurate inventory levels, reduce stockouts, and increase revenue. The system will include features such as inventory tracking, restocking alerts, and inventory forecasting to ensure that the company always has the right amount of stock on hand. The system will also provide a user-friendly interface for authorized personnel to access and manage inventory data from multiple locations. The system will be developed using modern technologies and programming languages and will adhere to industry best practices for security and performance.

## 2 General Description

The Stock Maintenance System is an inventory management software that will be developed for a company that deals in the distribution of stocks. The system will automate the process of inventory tracking, restocking, and forecasting to help the company maintain optimal inventory levels and reduce stockouts. The system will also be designed to provide a user-friendly interface for authorized personnel to access and manage inventory data from multiple locations.

## 3 Functional Requirements

- **Inventory Tracking:** The system shall be able to track the quantity and movement of stocks across multiple locations in real time.
- **Restocking Alerts:** The system shall generate alerts when inventory levels fall below a pre-defined threshold.
- **Inventory Forecasting:** The system shall be able to forecast the future inventory needs based on the historical data and current trends.
- **User Authentication:** The system shall require authorized personnel to log in with their username and password to access inventory data.
- **User Roles and Permissions:** The system shall provide different levels of access and permissions for different users roles, such as inventory managers and administrators.
- **Reporting and Analytics:** The system shall be able to generate reports and provide analytics on inventory levels, restocking, and other relevant metrics.
- **Integration:** The system shall be able to integrate with existing systems, such as the company's accounting and purchasing systems.

## 4 Interface Requirements

- **User Interface:** The system shall provide a user-friendly interface for authorized personnel to access and manage inventory data.

- **Navigation:** The user interface shall provide intuitive navigation for accessing different features and functionalities.
- **Input and Output:** The user interface shall allow users to input and output inventory data in various formats, such as spreadsheets or CSV files.
- **Compatibility:** The user interface shall be compatible with multiple web browsers and mobile devices.
- **Accessibility:** The user interface shall adhere to the accessibility guidelines, such as to ensure that it is usable by people with disabilities.

## **5 Performance Requirements**

- The system should have a response time of less than 2 seconds for most functions.
- The system should have a high level of availability and reliability, with a minimum uptime of 99.9%.
- The system should be scalable and able to handle increases in traffic and demand.
- The system should have adequate security measures to protect customer data and prevent unauthorized access.
- The system should have a backup and recovery mechanism to ensure business continuity in case of system failure or disaster.
- The system should comply with relevant industry standards and regulations for data privacy, security, and accessibility.
- The system should be able to handle a large volume of requests and transactions without performance degradation.
- The system must be able to respond quickly to user requests, especially those related to stock updates and inventory queries.
- The system should be available for use as much as possible, with minimal downtime or outages.
- The system should be designed to use resources efficiently, including CPU, memory, and network bandwidth.

## **6 Design Constraints**

- The system should have a user-friendly and intuitive interface.
- The system should follow industry-standard design principles and guidelines.
- The system should be responsive and adaptable to different screen sizes and resolutions.
- The system should follow a three-tier architecture with a presentation layer, business logic layer, and data access layer.
- The system should be modular and extensible to allow for future enhancements and modifications.
- The system should be designed to handle high traffic loads and large amounts of data.
- The system should use caching mechanisms to improve performance.

- The system should have a secure architecture that prevents unauthorized access and data breaches.
- The system should use encryption mechanisms to protect sensitive data.
- The system should have a backup and recovery plan in place to prevent data loss in case of security breaches.

## 7 Non-Functional Attributes

- **Reliability:** The system should be reliable, meaning it can consistently perform its functions without failure or error and can recover from any failures that do occur in a timely manner.
- **Maintainability:** The system should be designed for ease of maintenance, such as through modular code, well-defined interfaces, and clear documentation, to reduce the time and effort required to maintain and enhance the system.
- **Scalability:** The system should be scalable, meaning it can accommodate growth in stock volume or changes in the number of users or transactions, without requiring significant changes or performance degradation.
- **Availability:** The system should be available, meaning it can be accessed by users when needed, with minimal downtime or interruptions, and can recover from any disruptions that do occur.
- **Performance:** The system should have acceptable performance, meaning it can respond to user requests in a timely manner, without significant delays or slowdowns.
- **Security:** The system should be secure, meaning it can protect sensitive stock data and transactions from unauthorized access, manipulation, or disclosure, and comply with relevant security standards.
- **Usability:** The system should be usable, meaning it can be used easily and effectively by users with different levels of experience or technical proficiency, with a clear and consistent interface that enables efficient and error-free stock maintenance tasks.
- **Compatibility:** The system should be compatible with other systems, tools, or platforms that may be used by the organization, to ensure seamless integration and interoperability across different systems.

## **8 Preliminary Schedule and Budget**

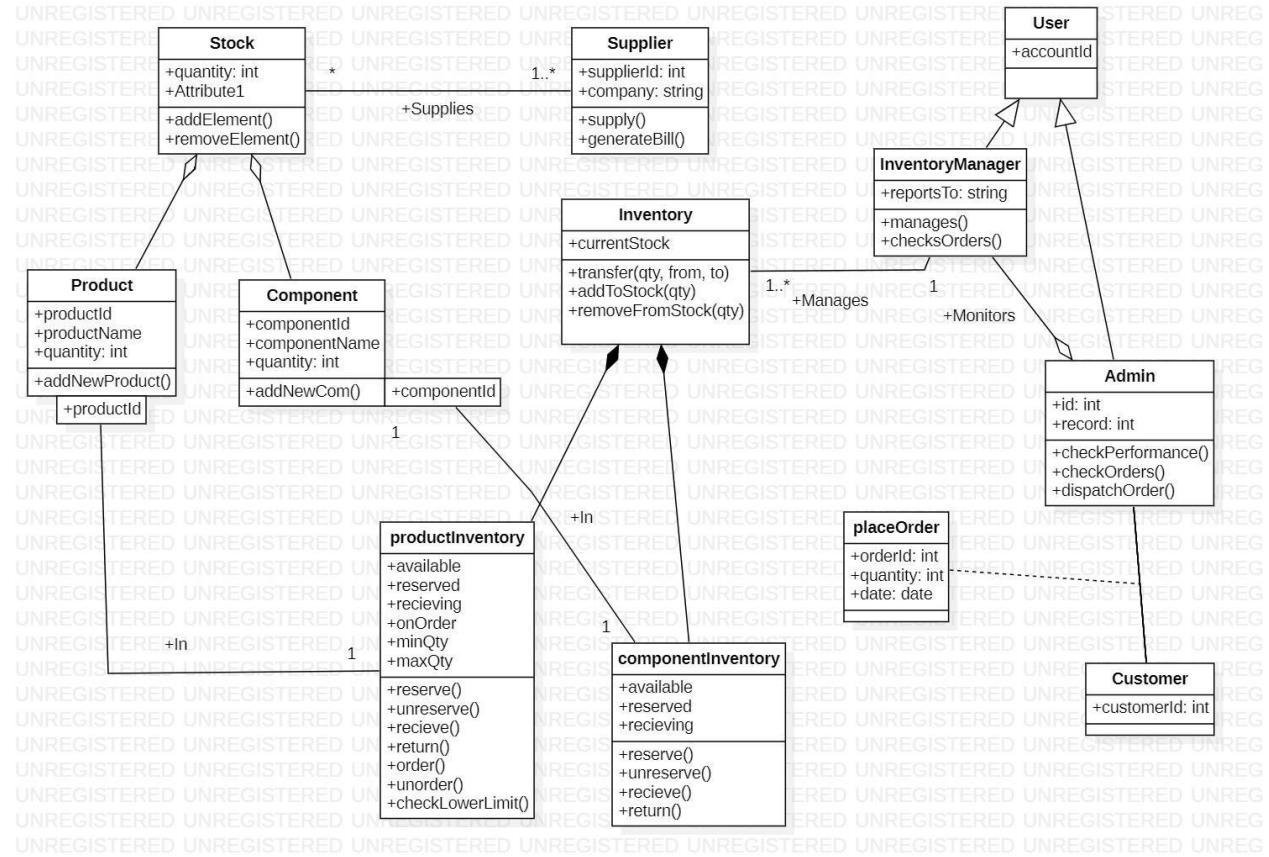
The development of the Stock Maintenance System is estimated to take eight months. The project will include design, development, testing, and deployment phases. The project will be managed using agile development methodologies. Assuming a team of 5 developers, a project duration of 6 months, and an hourly rate of Rs. 1000 per developer, the cost for the development team would be Rs. 4,800,000. Other costs that may need to be factored into the budget include:

**Testing and QA costs:** The cost of testing and quality assurance activities may vary based on the complexity of the system and the testing approach. Assuming a testing budget of 10% of the total development cost, this would be an additional Rs. 480,000.

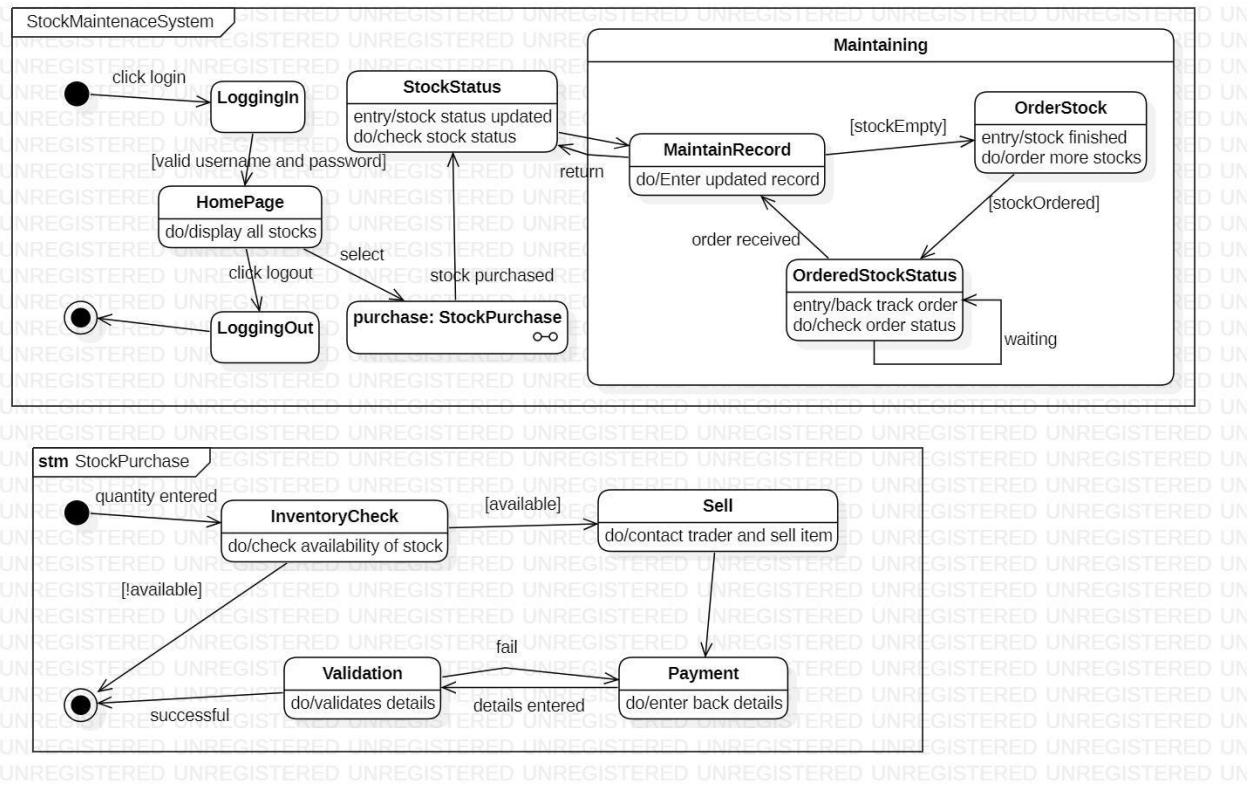
**Project management and documentation costs:** This could include the cost of hiring project manager, technical writers, or other personnel involved in project management or documentation. Assuming a project management and documentation budget of 15% of the total development cost, this would be an additional Rs. 720,000.

Adding all these costs together, the total preliminary budget for the Stock Maintenance System would be approximately Rs. 6,000,000.

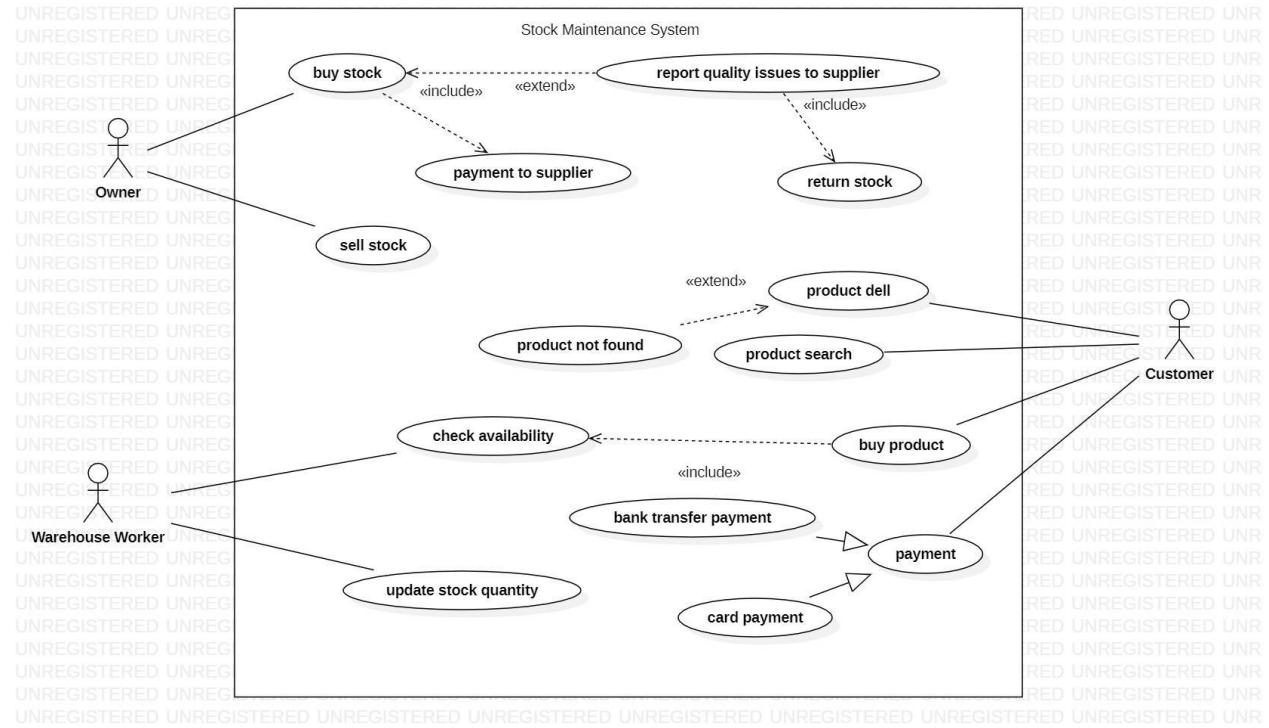
## Class Diagram



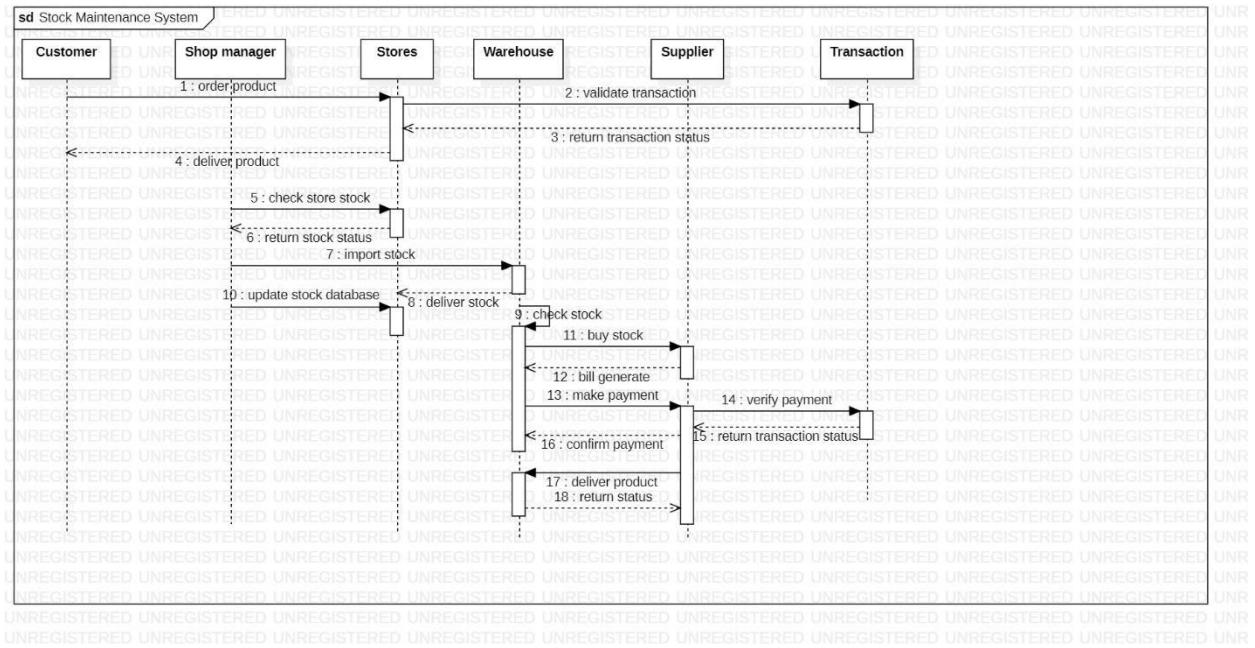
## State Diagram



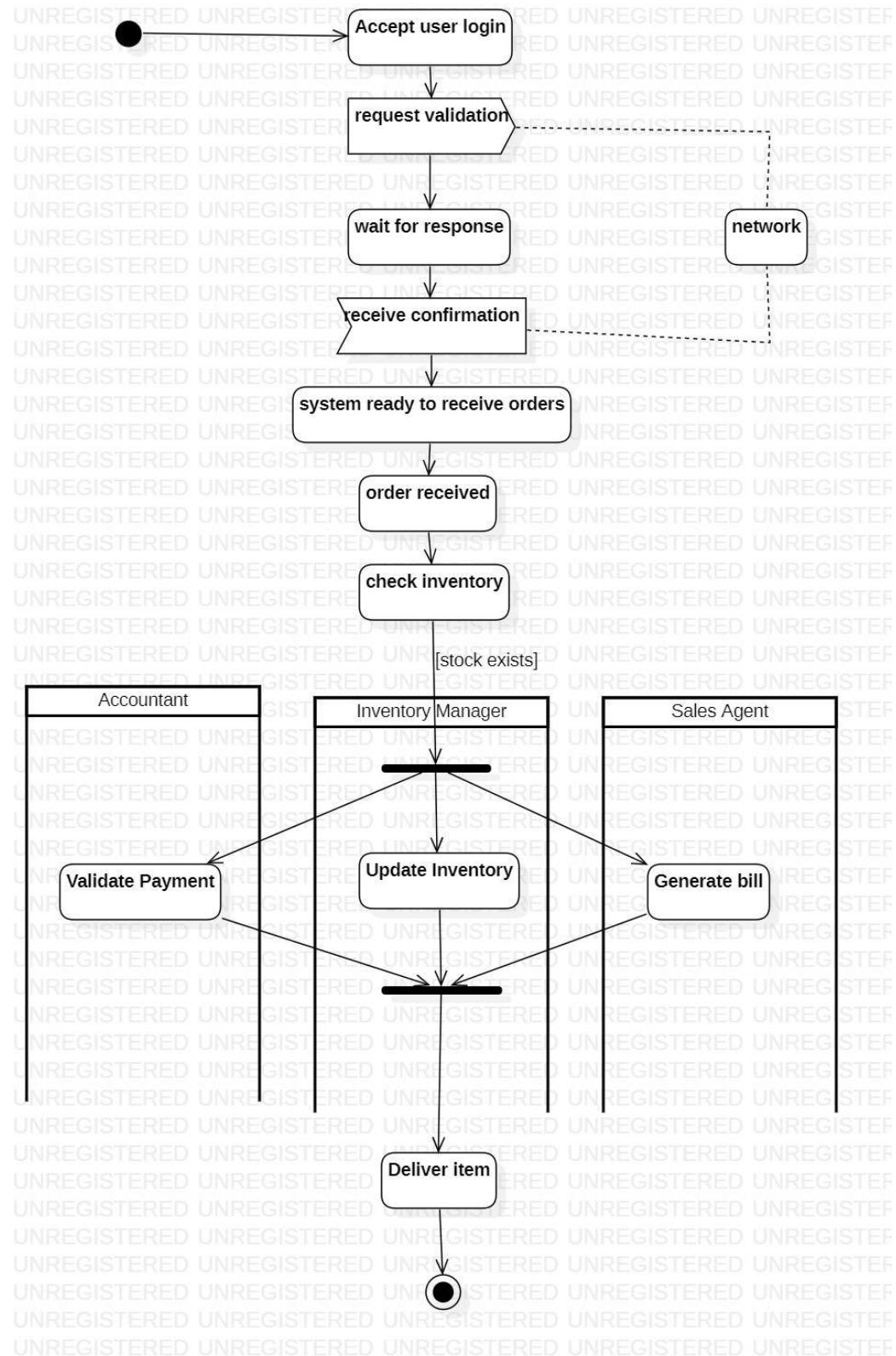
## Use Case Diagram



## Sequence Diagram



## Activity Diagram



## **5. Passport Automation System**

**Problem Statement:** Design and implement a software system that automates the process of issuing passports to citizens. The system should provide a secure and efficient way to handle the entire process of passport issuance, including application submission, verification, approval, printing, and delivery.

The system should be able to integrate with existing government databases, such as the National ID database, to streamline the verification process. It should also allow citizens to track the status of their passport application and receive real-time updates. The system forwards the necessary details to the police for its separate verification whose report is then presented to the administrator. The administrator will be provided with an option to display the status of the application to the applicant, which they can view in their online interface. After all the necessary criteria have been met, the original information is added to the database, and the passport is sent to the applicant.

# **Software Requirement Specification (SRS)**

## **1 Introduction**

### **1.1 Purpose of this document**

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the requirements for the development of a passport Automation system. This document will serve as a reference guide for the development team to design and implement the system.

### **1.2 Scope of this document**

This document outlines the functional and non-functional requirements of the Passport Automation System. It covers the entire process of passport issuance, including application submission, verification, approval, printing, and delivery. The scope of the system is to provide a secure and efficient way to handle the passport issuance process while ensuring the privacy and security of personal information.

### **1.3 Overview**

The Passport Automation System is a software system that automates the process of issuing passports to citizens. It provides a user-friendly interface that allows citizens to submit their passport applications, track the status of their applications, and receive real-time updates. The system will integrate with existing government databases to streamline the verification process.

## **2 General Description**

The Passport Automation System will be a web-based application that can be accessed from anywhere with an internet connection. The system will be built on a secure and scalable architecture that can handle a large volume of passport applications. The Passport Automation System will integrate with other government systems to automate the verification process and ensure the accuracy of the information provided.

## **3 Functional Requirements**

### **3.1 User Management**

- The system should have a user management module that allows citizens to register and create an account.
- The system should have an admin module that allows authorized personnel to manage user accounts.

### **3.2 Application Submission**

- The system should allow citizens to submit their passport applications online.
- The system should provide citizens with a user-friendly interface for completing their application.
- The system should validate the information provided by the citizen to ensure accuracy.

### **3.3 Application Verification**

- The system should integrate with the National ID database to verify the identity of the citizen.
- The system should perform a background check on the citizens to ensure they are eligible for a passport.

### **3.4 Application Approval**

- The system should allow authorized personnel to approve or reject passport applications.
- The system should notify the citizen of the status of their application.

### **3.5 Passport Printing and Delivery**

- The system should generate passport documents based on the approved application.
- The system should deliver the passport to the citizen through a secure and reliable delivery service.

## **4 Interface Requirements**

### **4.1 User Interface**

The system should have a user-friendly interface for citizens to complete their passport applications and also an admin interface for authorized personnel to manage passport applications.

### **4.2 API Integration**

The system should integrate with the National ID database to verify the identity of the citizen.

## **5 Performance Requirements**

- The system should be able to handle a large volume of passport applications.
- The system should have a response time of less than 3 seconds for user requests.
- The system shall be available 24/7, with a minimum uptime of 99.9%.

## **6 Design Constraints**

- The system should be designed to comply with all relevant regulations and standards, such as data protection laws.
- The system shall be designed using a modern, scalable architecture, with a focus on performance and security. The system should use encryption mechanisms to protect sensitive data.

## **7 Non-Functional Attributes**

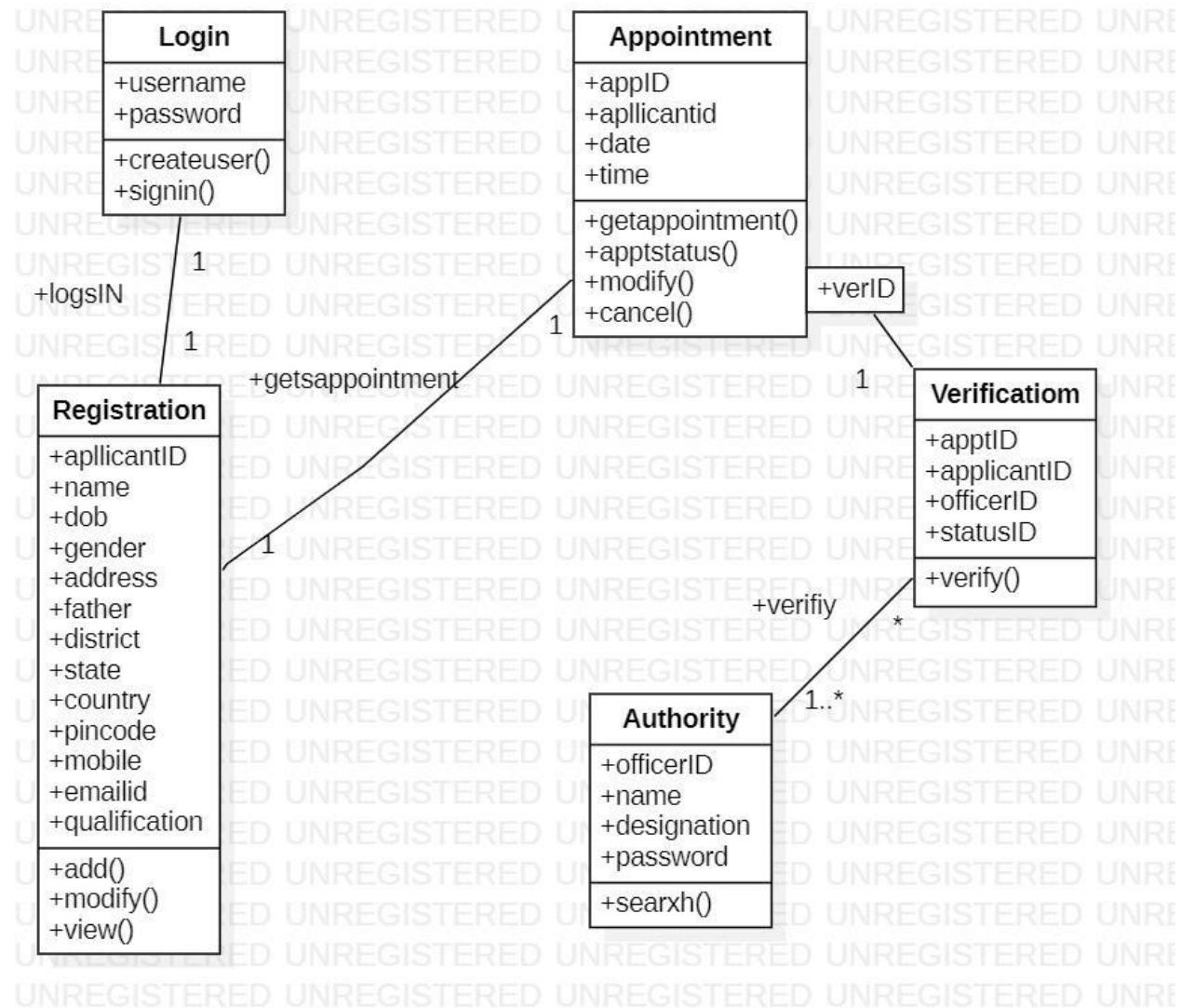
- **Usability:** The system shall be easy to use and intuitive for merchants of all skill levels.
- **Security:** The system shall be secure and compliant with industry standards for handling sensitive financial information.
- **Reliability:** The system shall be reliable and perform consistently under high traffic and load conditions.

- **Performance:** The system should have a response time of less than 3 seconds for most functions and be able to handle a large volume of requests and transactions without performance degradation.
- **Scalability:** The system should be scalable and able to handle increases in traffic and demand.
- **Maintainability:** The system should be easy to maintain and update, with clear documentation and modular architecture.
- **Compatibility:** The system should be compatible with different browsers, operating systems, and hardware configurations.

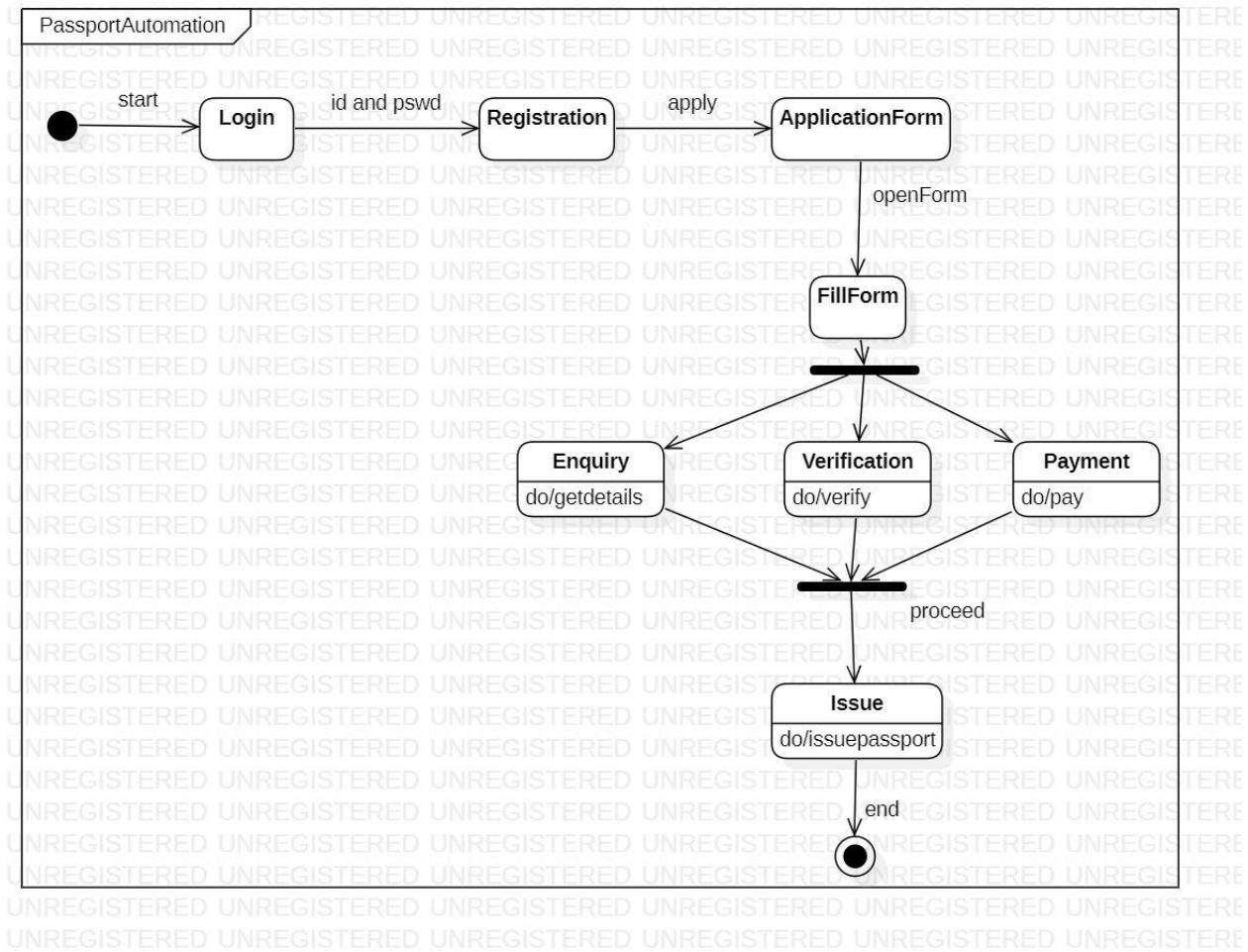
## 8 Preliminary Schedule and Budget

The development of the Passport Automation System is estimated to take approximately 12 months, with a total budget of \$500,000. The schedule and budget may be adjusted based on project requirements and constraints.

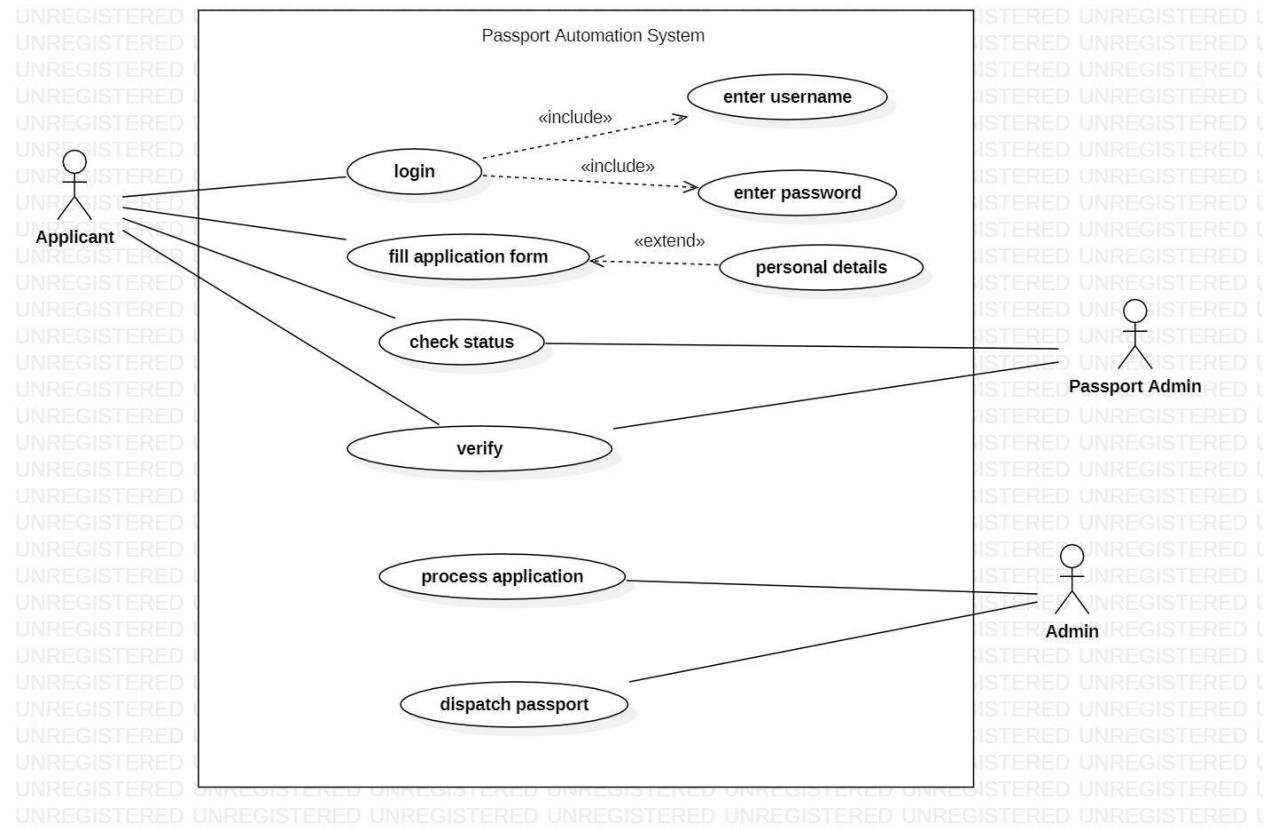
## Class Diagram



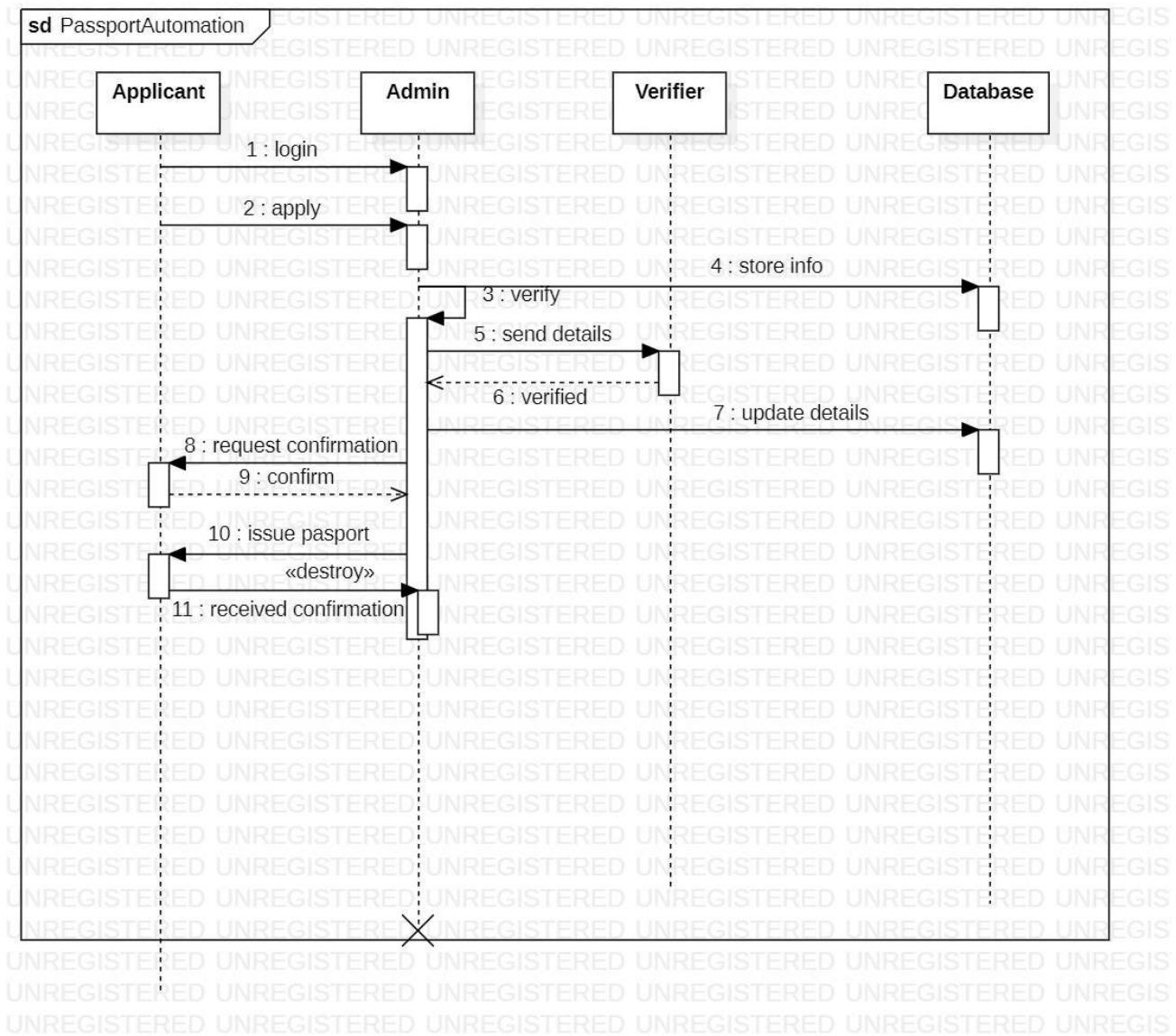
## State Diagram



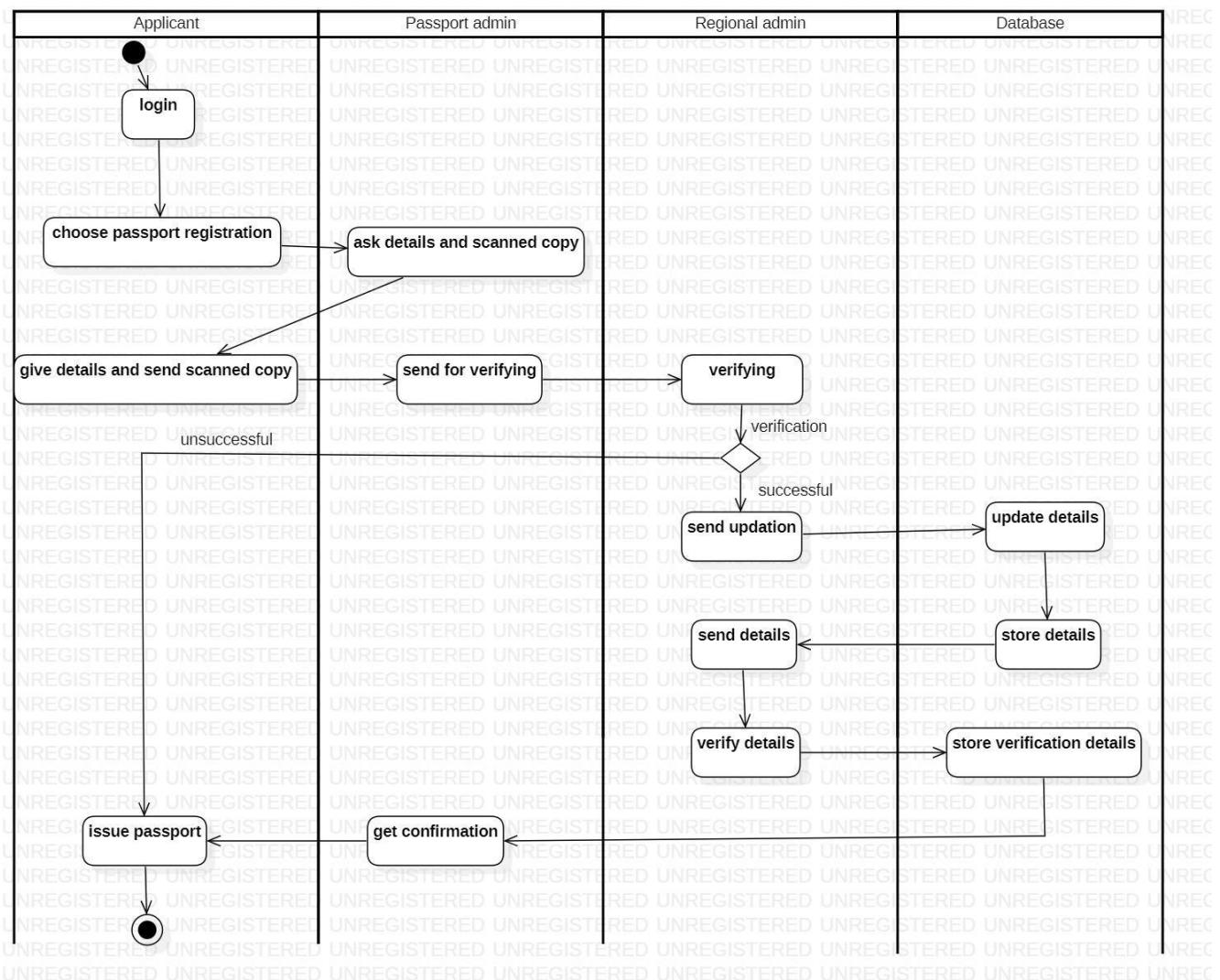
## Use Case Diagram



## Sequence Diagram



## Activity Diagram



## 6. Railway Reservation System

### **Problem Statement:** Developing a User-Friendly Railway Management System

The current railway management system suffers from several issues that limit its user-friendliness and provide an unpleasant experience to passengers. These issues include a lack of accessibility, inconsistent user experience, limited information availability, and a complex ticket booking process.

The lack of accessibility of the current system makes it difficult for people with disabilities and those who are not tech-savvy to use the system effectively. The inconsistent user experience of the system leads to confusion and difficulty in performing necessary tasks. The limited information available on train schedules, delays, or cancellations frustrates passengers. Additionally, the complex ticket booking process requires users to navigate a lengthy and confusing process to book tickets.

The goal of this project is to develop a modern and user-friendly railway management system that provides a seamless and intuitive experience to passengers. The new system should address these issues by incorporating accessibility features, providing consistent user experience, offering real-time information updates, and streamlining the ticket booking process.

The accessibility features of the new system should include audio descriptions, large fonts, or screen readers to make it easier for people with disabilities to use the system. The consistent user experience of the system should be available across all platforms, including desktop and mobile, making it easier for passengers to navigate and perform tasks. The real-time information updates should provide up-to-date information on train schedules, delays, and cancellations, giving passengers more accurate information to plan their travel. The streamlined ticket booking process should simplify the process, allowing passengers to easily select their desired train, seat, and class, and make payments online, reducing the time and effort required to book tickets.

By developing a user-friendly railway management system, passengers will have a better travel experience, and the railway staff will have an efficient tool to manage their operations. This system will also provide valuable data insights that can be used to improve the railway service and make it more efficient.

# **Software Requirement Specification (SRS)**

## **1 Introduction**

### **1.1 Purpose of this document**

The purpose of this Software Requirements Specification (SRS) document is to specify the requirements of a Railway Management System. This document will outline the functional and non-functional requirements of the system, which will be used as a basis for system design, development, and testing.

### **1.2 Scope of this document**

This document is intended for the development team, stakeholders, and users of the Railway Management System. The SRS document outlines the technical requirements, functional requirements, and constraints of the system.

### **1.3 Overview**

The Railway Management System is an online system designed to manage railway operations efficiently. The system provides users with features such as ticket booking, train schedules, seat availability, and fare details. The system will ensure an excellent user experience and improve the efficiency of the railway service.

## **2 General Description**

The Railway Management System will be developed as a web-based application. The system will be built on a scalable architecture that allows for future expansion and enhancement. The system will have the following features:

- Ticket booking
- Train schedules

- Seat availability
- Fare details
- Passenger details management
- Train management
- Staff management

### **3 Functional Requirements**

The functional requirements of the Railway Management System are as follows:

- Users should be able to register and log in to the system.
- Users should be able to search for train schedules by date, time, and destination.
- Users should be able to view available seats for a particular train.
- Users should be able to book tickets for a particular train and class.
- The system should provide real-time information on train schedules, delays, and cancellations.
- The system should allow staff members to manage train schedules, seat availability, and fare details.
- The system should allow staff members to manage passenger details and bookings.
- The system should provide reports on daily bookings and revenue.

### **4 Interface Requirements**

#### **4.1 User Interface**

The user interface of the Railway Management System should be intuitive and easy to use. The system should be responsive and accessible on desktop and mobile devices.

#### **4.2 Integration Interface**

The Railway Management System should be able to integrate with other third-party systems, such as payment gateways, to process payments seamlessly.

## **5 Performance Requirements**

The Railway Management System must meet the following performance requirements:

- Response time: The system should respond to user interactions within three seconds or less.
- Concurrent users: The system should be able to handle at least 1,000 concurrent users without significant performance degradation.
- Scalability: The system should be designed to scale to handle up to 10,000 concurrent users in the future.
- Availability: The system should be available 24/7, with a maximum of one hour of scheduled downtime per week for maintenance.

## **6 Design Constraints**

The Railway Management System must meet the following design constraints:

- Platform: The system must be developed as a web-based application and must be compatible with standard web browsers.
- Technology: The system should be developed using industry-standard technologies, such as HTML, CSS, JavaScript, and a server-side language such as PHP or Java.
- Security: The system must be designed with security in mind, including measures such as encryption, firewalls, and access controls.
- Compatibility: The system must be compatible with third-party payment gateways and other systems as required.

## **7 Non-Functional Attributes**

The non-functional attributes of the Railway Management System are as follows:

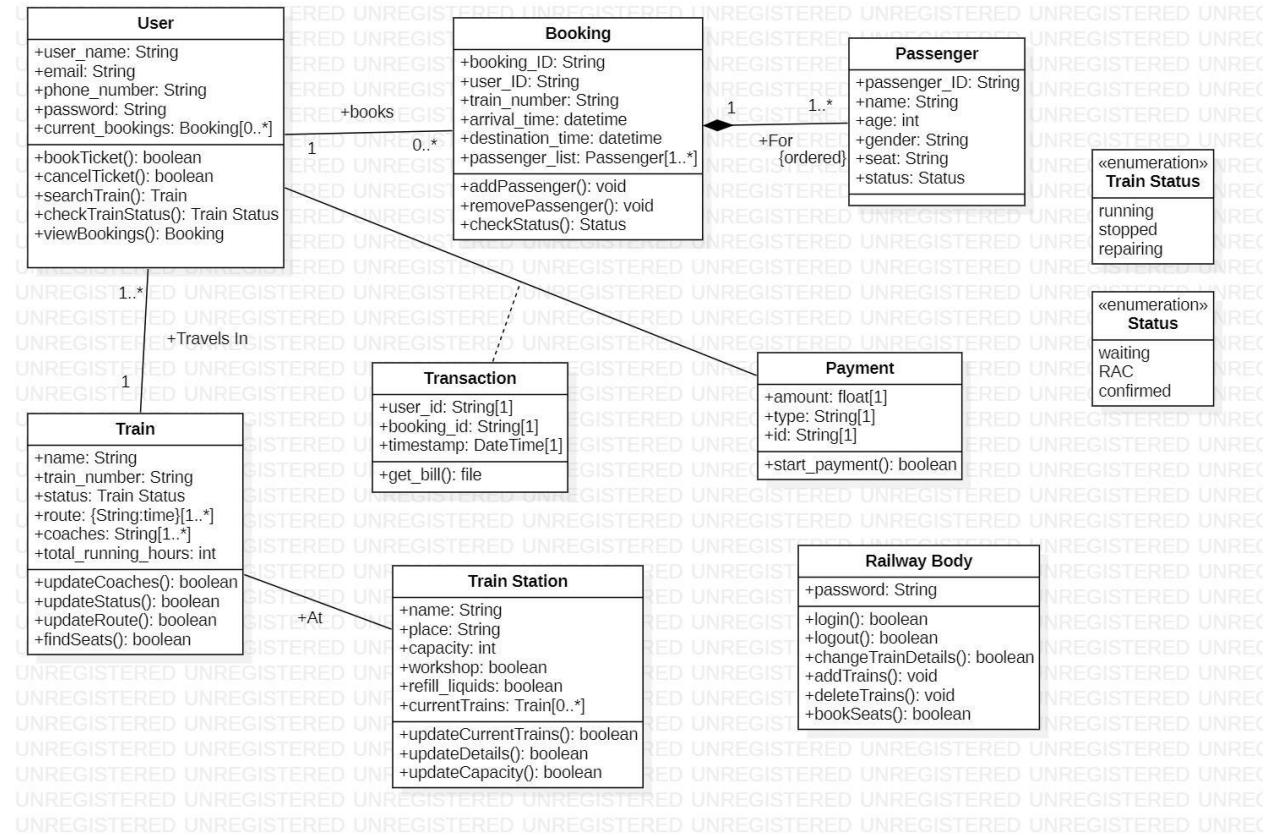
- Usability: The system should be user-friendly and easy to use.
- Reliability: The system should be reliable and available 24/7.
- Security: The system should be secure and protect user data.

- Scalability: The system should be scalable to handle a growing number of users and transactions.
- Maintainability: The system should be easy to maintain and update.

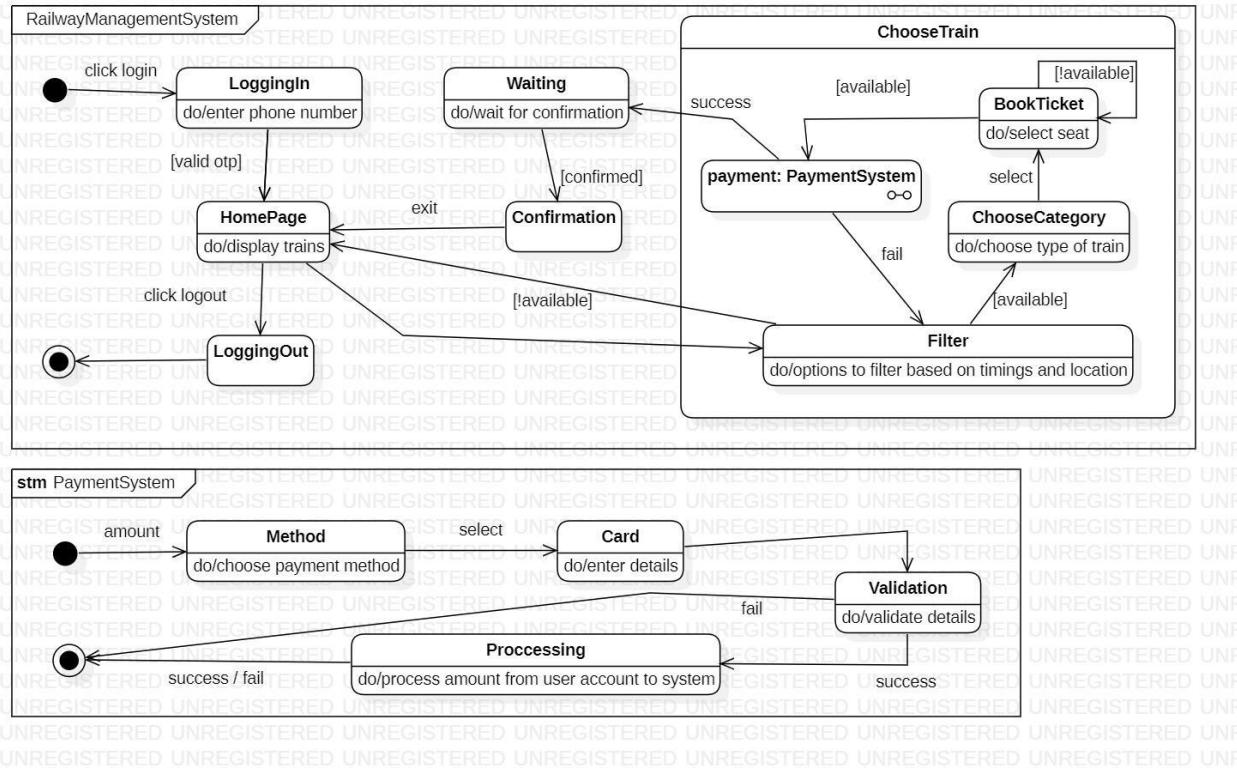
## 8 Preliminary Schedule and Budget

The preliminary schedule and budget for the Railway Management System will be determined during the project planning phase. The development team will work with stakeholders to determine the project timeline and budget.

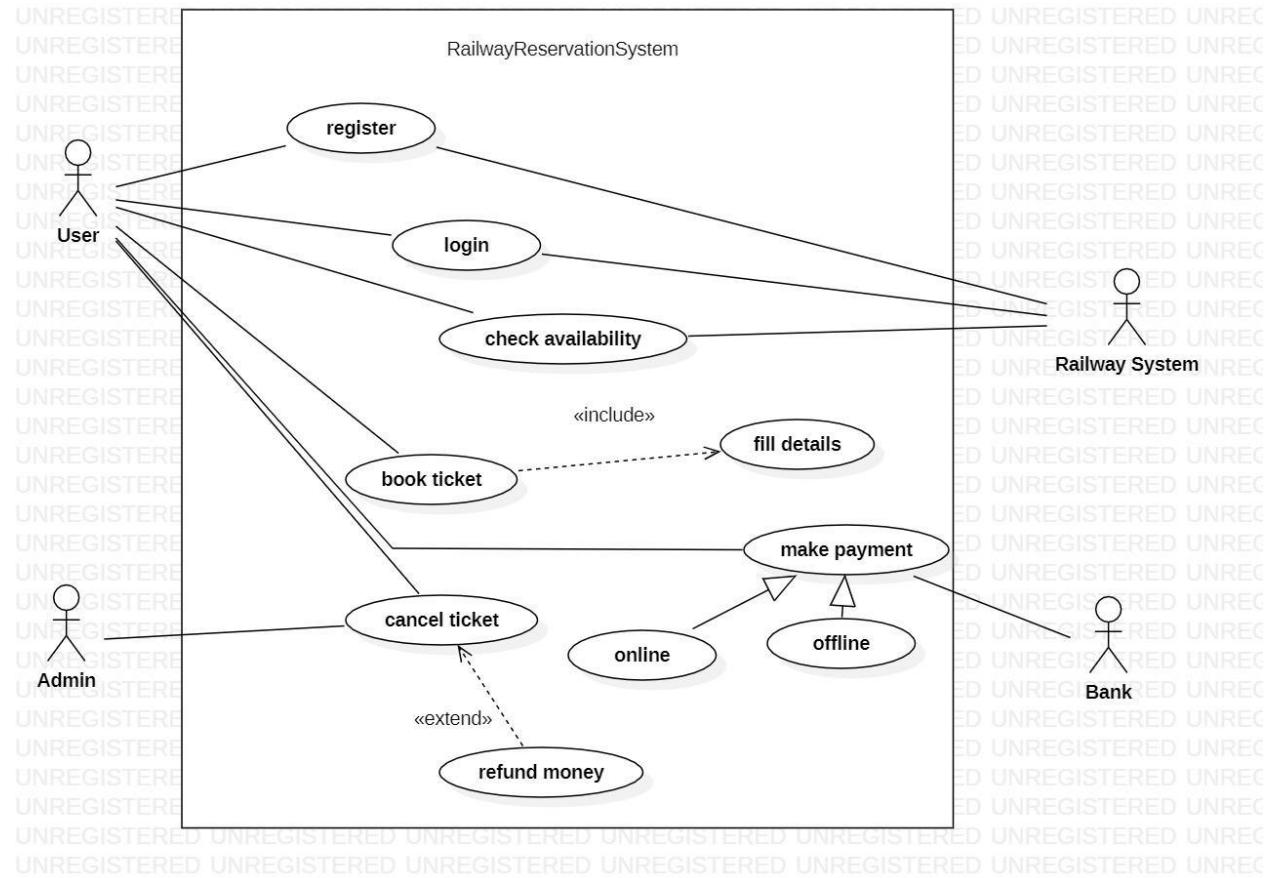
# Class Diagram



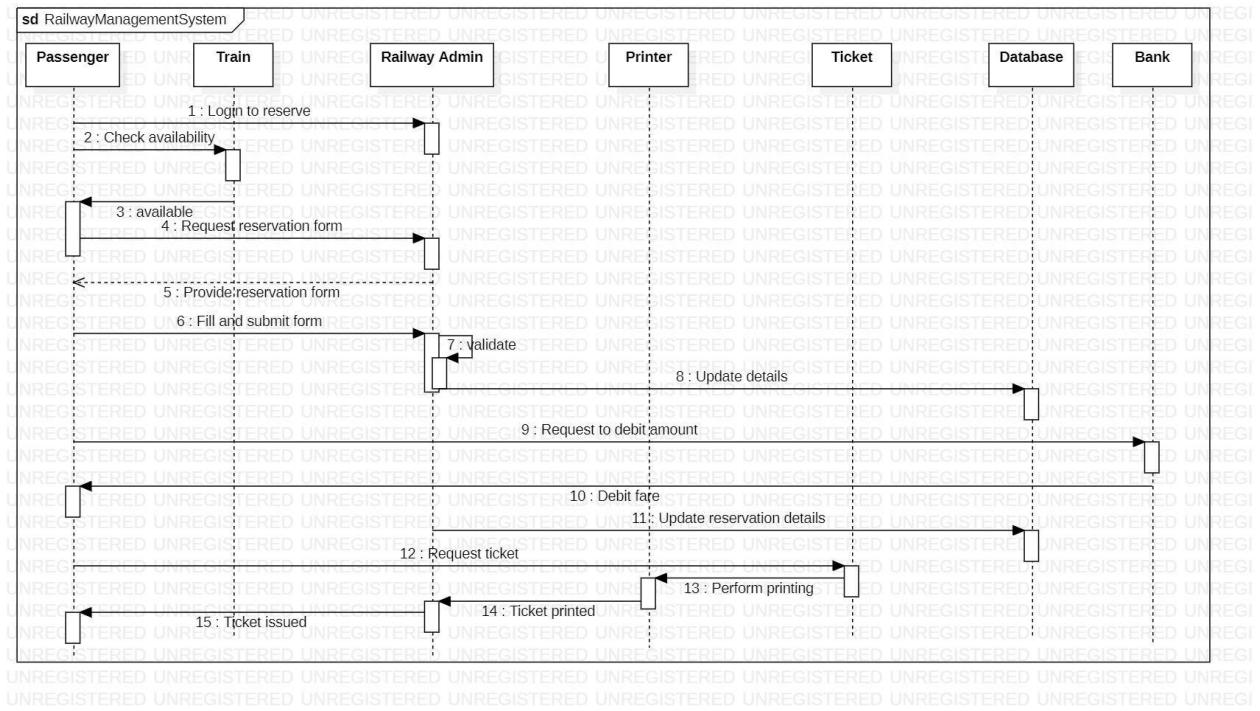
## State Diagram



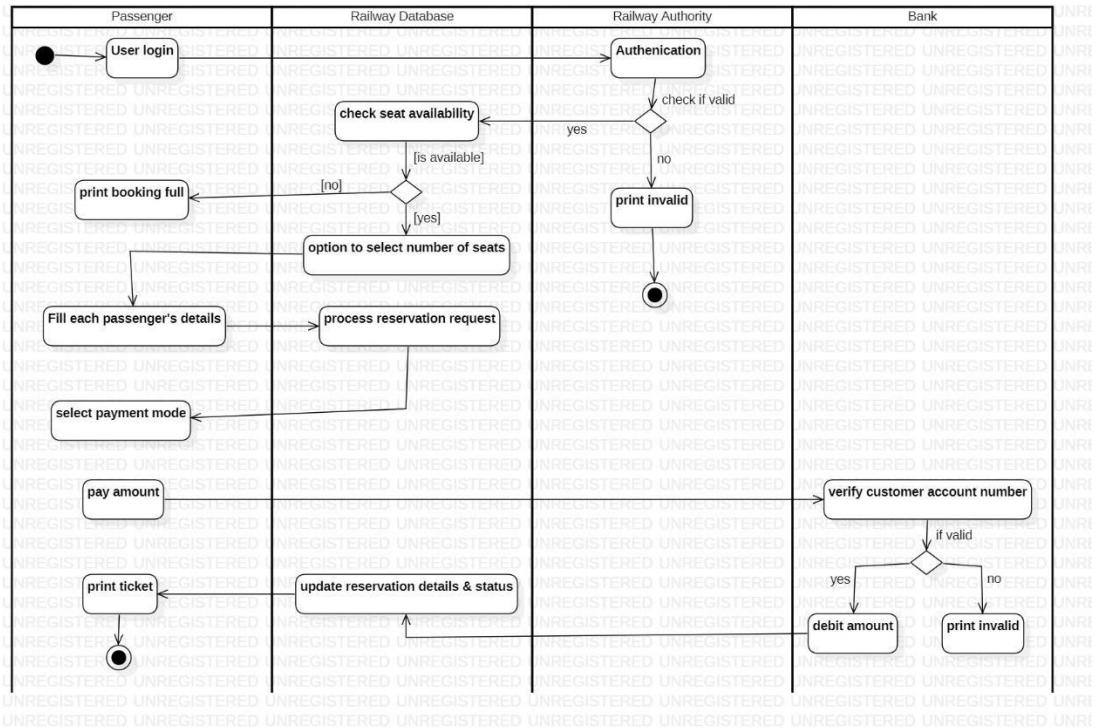
## Use Case Diagram



## Sequence Diagram



## Activity Diagram



## 7. Online Shopping System

**Problem Statement:** The traditional method of shopping involves physically visiting a store, selecting a product, and making a payment. However, with the advancement of technology, online shopping has become a popular way of purchasing products. Online shopping systems provide customers with the convenience of shopping from the comfort of their own homes and offer a wide range of products at competitive prices.

The objective of this project is to design an online shopping system that provides customers with a user-friendly interface to browse and purchase products. The system should be able to handle a large volume of traffic, process transactions securely, and provide customers with reliable delivery of products.

Some of the main challenges that need to be addressed in the online shopping system are:

1. **User authentication and authorization:** The system should ensure that only authorized users are able to access and purchase products.
2. **Product catalog management:** The system should provide an efficient way for the administrator to manage the product catalog, which includes adding, updating, and deleting products.
3. **Shopping cart management:** The system should allow users to add products to their shopping cart and provide them with the ability to edit or remove items before making a payment.
4. **Payment processing:** The system should securely process payments made by customers and provide them with payment confirmation.
5. **Order processing and delivery:** The system should efficiently process orders, track shipments, and provide customers with timely delivery of products.
6. **Customer support:** The system should provide customers with an efficient way to contact support in case of any issues or queries.

The online shopping system should be designed to meet these challenges while providing a seamless and enjoyable shopping experience for customers. The system should also be scalable and customizable to accommodate future growth and changing business requirements.

# **Software Requirement Specification (SRS)**

## **1 Introduction**

### **1.1 Purpose of this document**

The purpose of this document is to provide a comprehensive Software Requirements Specification (SRS) for the development of a user-friendly online shopping system.

### **1.2 Scope of this document**

This document outlines the functional and non-functional requirements, constraints, and specifications of the online shopping system.

### **1.3 Overview**

The online shopping system is a web-based platform that allows customers to purchase products or services from the comfort of their own home. The system includes a catalog of products, a shopping cart feature for customers to add items to their order, a payment gateway for processing transactions, and an order tracking system for customers to monitor their purchases.

## **2 General Description**

The online shopping system should be user-friendly and easy to navigate for customers of all ages and technical skill levels. The system should be able to handle a large volume of traffic and transactions, with a high level of security to protect customer information. The website should be responsive, meaning it should be easily accessible and functional on both desktop and mobile devices.

### **3 Functional Requirements**

- Product Catalog: The online shopping system should have a comprehensive product catalog, including product descriptions, images, and pricing information. The catalog should be easily searchable and filterable, with options to sort by price, popularity, or category.
- Shopping Cart: The system should allow customers to add items to their cart and view their total order cost. The cart should be easily accessible and editable, allowing customers to add or remove items as needed.
- Payment Gateway: The system should have a secure payment gateway that supports multiple payment methods, including credit cards, debit cards, and digital wallets. The payment gateway should be easy to use, with clear instructions for customers to follow.
- Order Tracking: The system should provide customers with real-time updates on the status of their order, including shipping and delivery information. Customers should be able to track their order history and view their previous purchases.
- User Account: The system should allow customers to create and manage their own user accounts. This should include features such as order history, saved payment methods, and shipping addresses.
- Customer Service: The system should have a robust customer service system, including a FAQ section, a support ticket system, and live chat or phone support.
- Security: The system should be designed with strong security measures to protect customer data, including SSL encryption, secure logins, and regular data backups.
- Performance: The system should be optimized for fast load times and minimal downtime. This should include features such as caching, content delivery networks (CDNs), and scalable infrastructure to handle spikes in traffic.

### **4 Interface Requirements**

- The system shall have a user-friendly interface that is easy to navigate.
- The system shall provide a search function that allows users to search for products by keywords or categories.

- The system shall display product information, including images, descriptions, prices, and availability.
- The system shall provide a shopping cart that allows users to add and remove items from their order before checkout.
- The system shall display shipping options and costs, and allow users to select their preferred shipping method.
- The system shall provide a secure payment gateway that accepts multiple payment methods, such as credit cards, debit cards, and online payment services.
- The system shall send confirmation emails to users after they place an order, including order details and shipping information.

## 5 Performance Requirements

- The system shall be available 24/7 with a minimum uptime of 99%.
- The system shall be able to handle a minimum of 1000 concurrent users without significant degradation in performance.
- The system shall be able to process orders within 10 seconds of user submission.
- The system shall be able to handle a minimum of 100,000 products in the database without significant degradation in performance.
- The system shall be able to handle a minimum of 100,000 registered users without significant degradation in performance.
- The system shall be able to handle a minimum of 10,000 transactions per day without significant degradation in performance.
- The system shall have a response time of no more than 3 seconds for all user interactions.

## 6 Design Constraints

- The system must be designed to be scalable, allowing for future growth and expansion.
- The system must be designed to be modular, allowing for easy updates and maintenance.
- The system must be designed to be platform-agnostic, supporting multiple operating systems and web browsers.

- The system must be designed to be secure, with proper measures in place to prevent data breaches and protect user privacy.
- The system must be designed to comply with applicable laws and regulations related to online shopping and e-commerce.

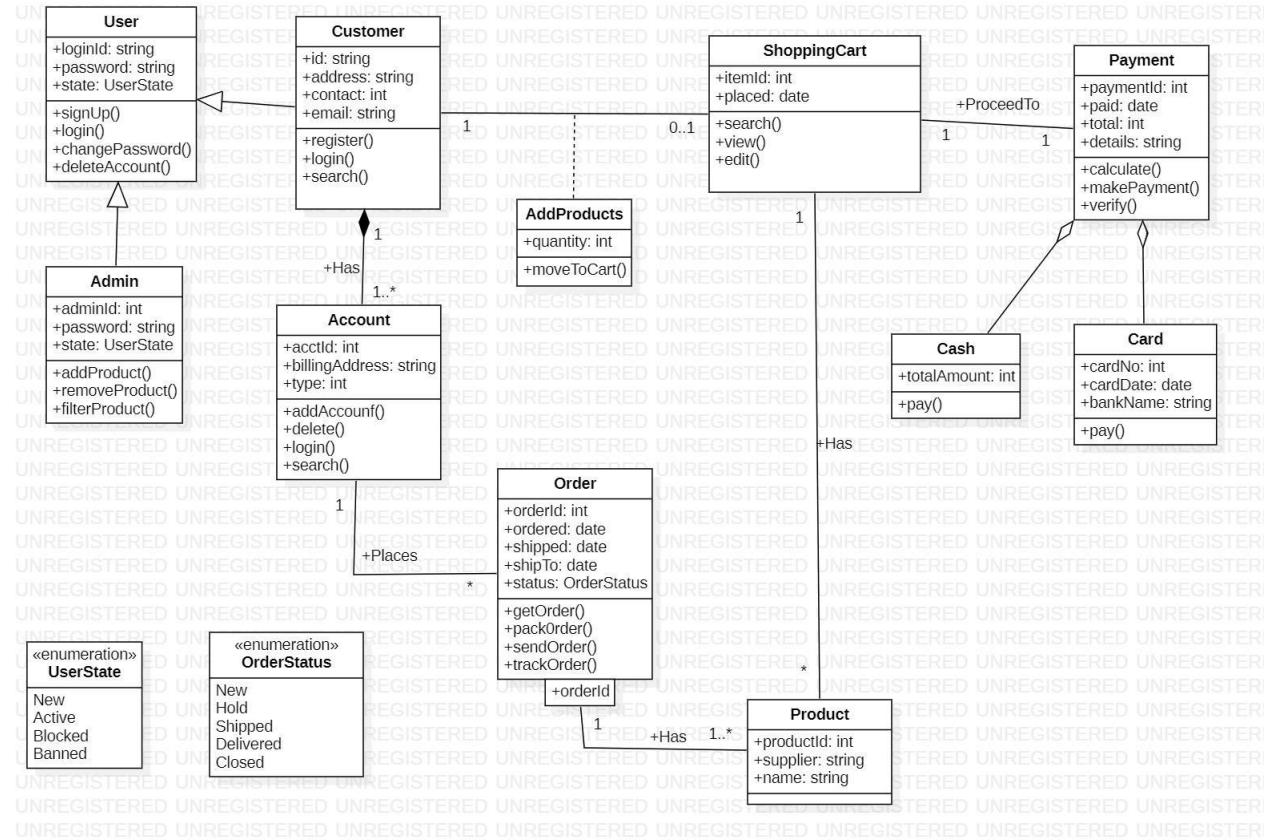
## 7 Non-Functional Attributes

- **Reliability:** The system must be reliable, with minimal downtime and data loss.
- **Availability:** The system must be available 24/7, with a high level of uptime and minimal maintenance windows.
- **Performance:** The system must be able to handle high traffic volumes and user requests without significant latency or slowdowns.
- **Usability:** The system must be user-friendly and easy to navigate, with clear and concise instructions for all user interactions.
- **Security:** The system must be secure, with proper authentication, encryption, and data protection measures in place to prevent unauthorized access and data breaches.
- **Maintainability:** The system must be designed to be easy to maintain and update, with clear and concise documentation and modular code architecture.
- **Portability:** The system must be designed to be portable, allowing for easy deployment to multiple environments and platforms.

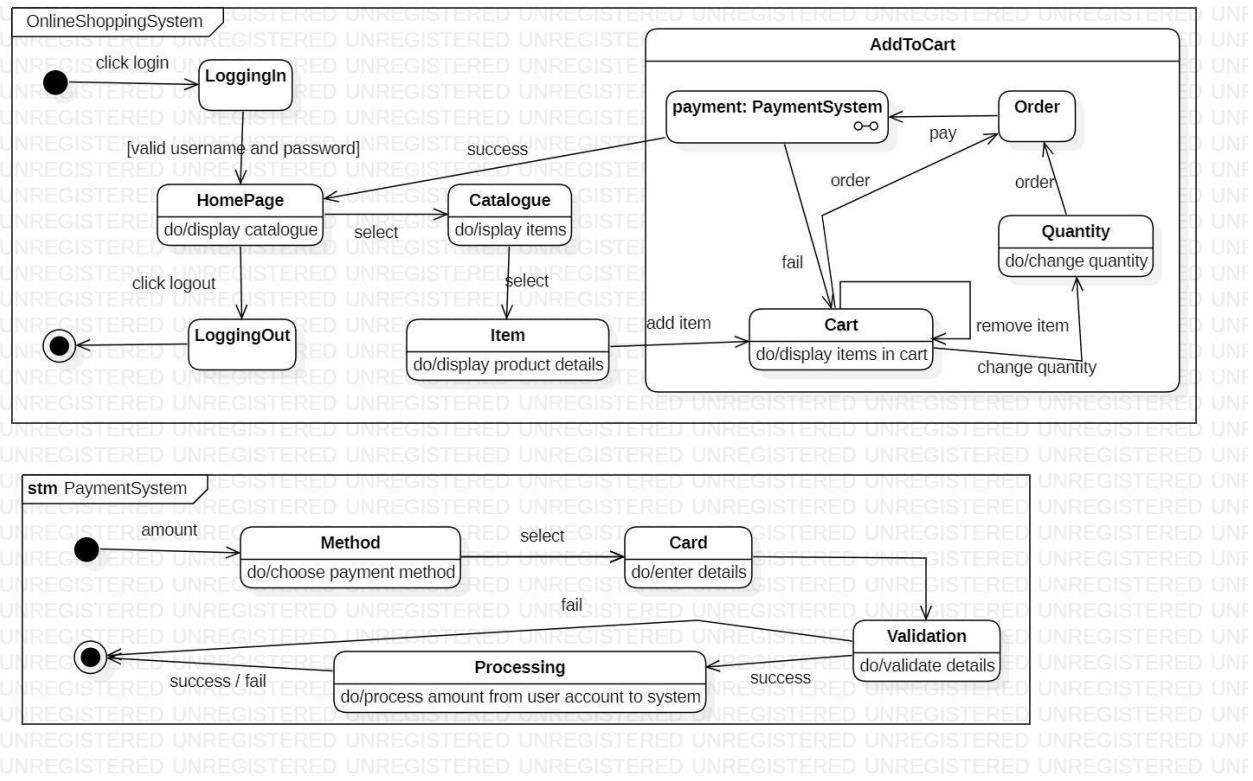
## 8 Preliminary Schedule and Budget

The development of the Online Shopping System is estimated to take two months. The project will include design, development, testing, and deployment phases. The project will be managed using agile development methodologies.

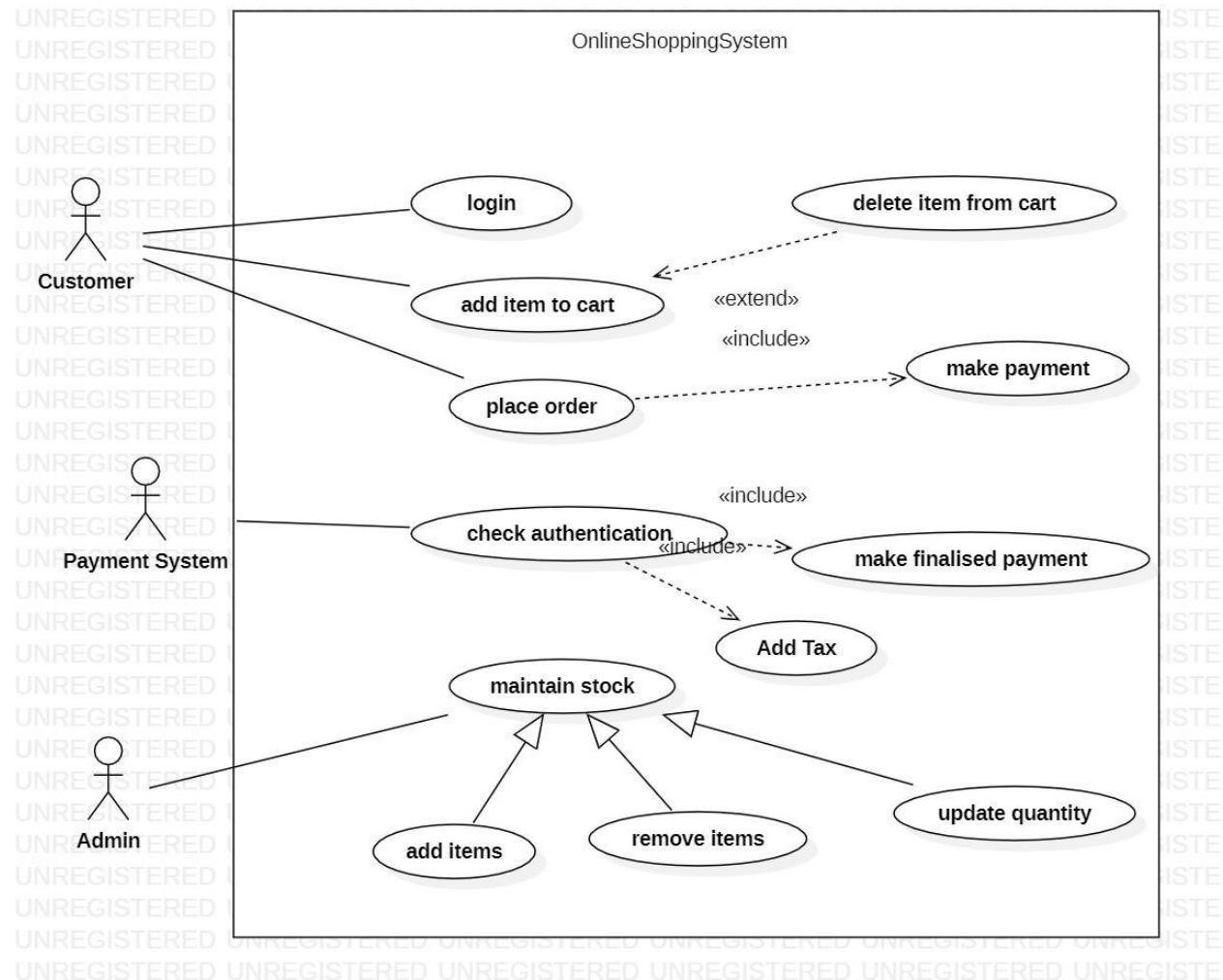
## Class Diagram



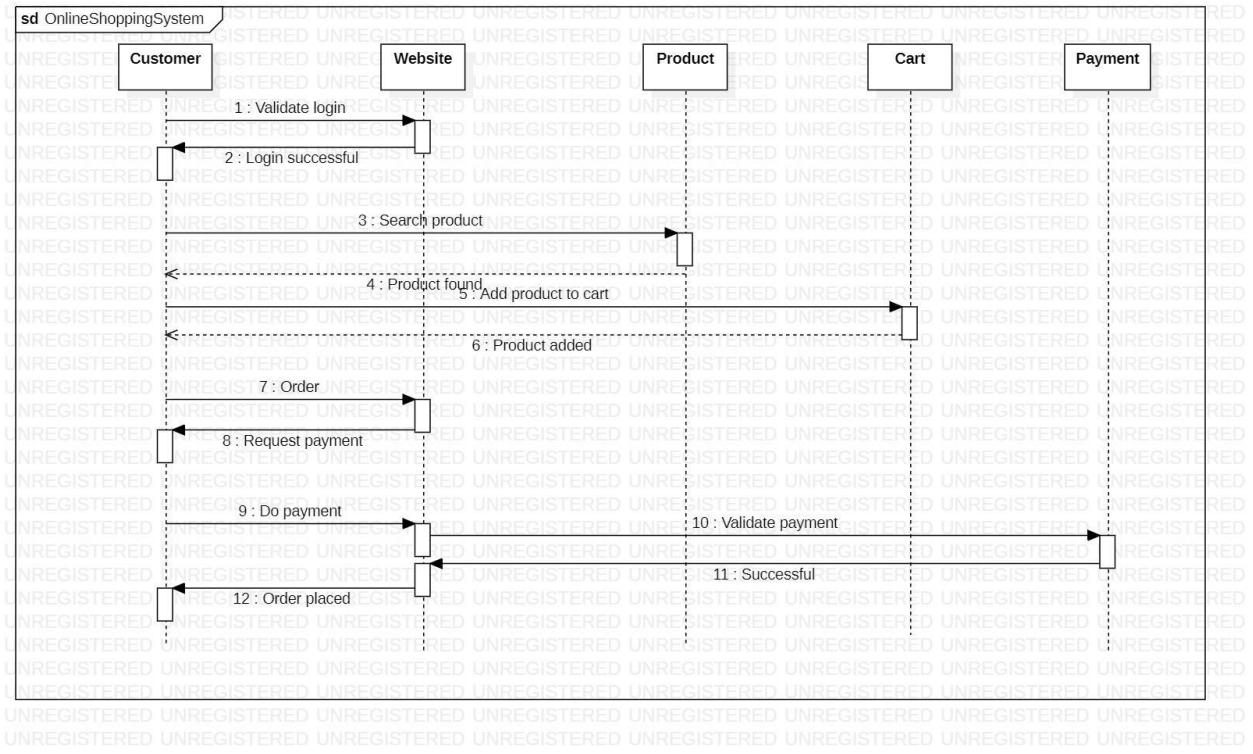
## State Diagram



## Use Case Diagram



## Sequence Diagram



## Activity Diagram

