B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

Lab Report On

Object Oriented Modelling Design

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:
Varun Chahal (1BM20CS181)

Department of Computer Science and Engineering
B.M.S College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019
2022-2023

B.M.S COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

This is to certify that the Lab work entitled “Object Oriented Modelling and Design” carried out by Varun Chahal(1BM20CS181), 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 2023. The Lab report has been approved as it satisfies the academic requirements in respect of Object Oriented Modelling And Design - (20CS6PCOMD) work prescribed for the said degree.

Signature of the Candidate

Varun Chahal(1BM20CS181)

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the OOMD Lab report has been carried out by Varun Chahal (1BM20CS181) during the academic year 2022-2023.

Signature of the Faculty in Charge

Dr Seema Patil
Assistant Professor
B.M.S College of Engineering

Signature of the Head of the Department

DR. JYOTHI S NAYAK
Professor and Head
Department of CSE
BMSCE, Bengaluru

Table of Contents

1. Hotel Management System

Problem statement:

The hotel management system is a software solution designed to simplify and streamline the operations of a hotel. The system should provide a user-friendly interface that allows the hotel staff to manage their day-to-day tasks with ease. The system should automate tasks such as room bookings, reservations, check-ins, checkouts, and billing. The system should also provide reporting and analytics features that can help the hotel management to make informed decisions. The goal is to improve the efficiency of the hotel operations, enhance the guest experience, and increase revenue.

Software Requirement Specification (SRS):

1. Introduction:

a) **Purpose:** The purpose of a hotel management system is to automate and simplify the operations of a hotel, with the aim of improving efficiency, enhancing the guest experience, and increasing revenue. The system should be designed to manage various aspects of hotel operations, such as room bookings, reservations, check-ins, checkouts, billing, inventory management, housekeeping, and maintenance.

b) **Scope:** The scope of a hotel management system is to provide a software solution for managing hotel operations such as room reservations, guest check-ins, inventory management, billing, and reporting. The system aims to streamline the hotel management process, increase efficiency, and improve guest satisfaction by providing a centralized platform for managing all hotel-related tasks.

c) **Overview:** A hotel management system is a software solution that enables hotel staff to manage various aspects of hotel operations such as room reservations, check-ins, check-outs, billing, inventory management, and reporting. The system provides a centralized platform for managing all hotel-related tasks, allowing hotel staff to increase efficiency and improve guest satisfaction. The system can be accessed by authorized hotel staff from anywhere with an internet connection, making it convenient and accessible.

2. **General Description:** A hotel management system is a software solution that streamlines and manages various aspects of hotel operations, including reservations, check-ins, check-outs, billing, inventory management, and reporting. It provides a centralized platform for hotel staff to increase efficiency and improve guest satisfaction.

3. **Functional Requirements:**

- User Management
- Room Management
- Reservation Management
- Billing and Payment
- Inventory Management

4. Interface Requirements:

- User Interface
- Multi-Language Support
- Mobile Responsiveness
- Integration with third-party systems
- Security

5. Performance Requirements:

- Response Time
- Scalability
- Reliability
- Data Processing
- Speed

6. Design Constraints:

- Budget: The system should be designed within the available budget, taking into account the cost of hardware, software, development, and maintenance.

- Technology Constraints: The system should be designed based on the available technology infrastructure, including servers, databases, and networking capabilities.

- Legal and Regulatory Requirements: The system should comply with legal and regulatory requirements such as data privacy, security, and accessibility.

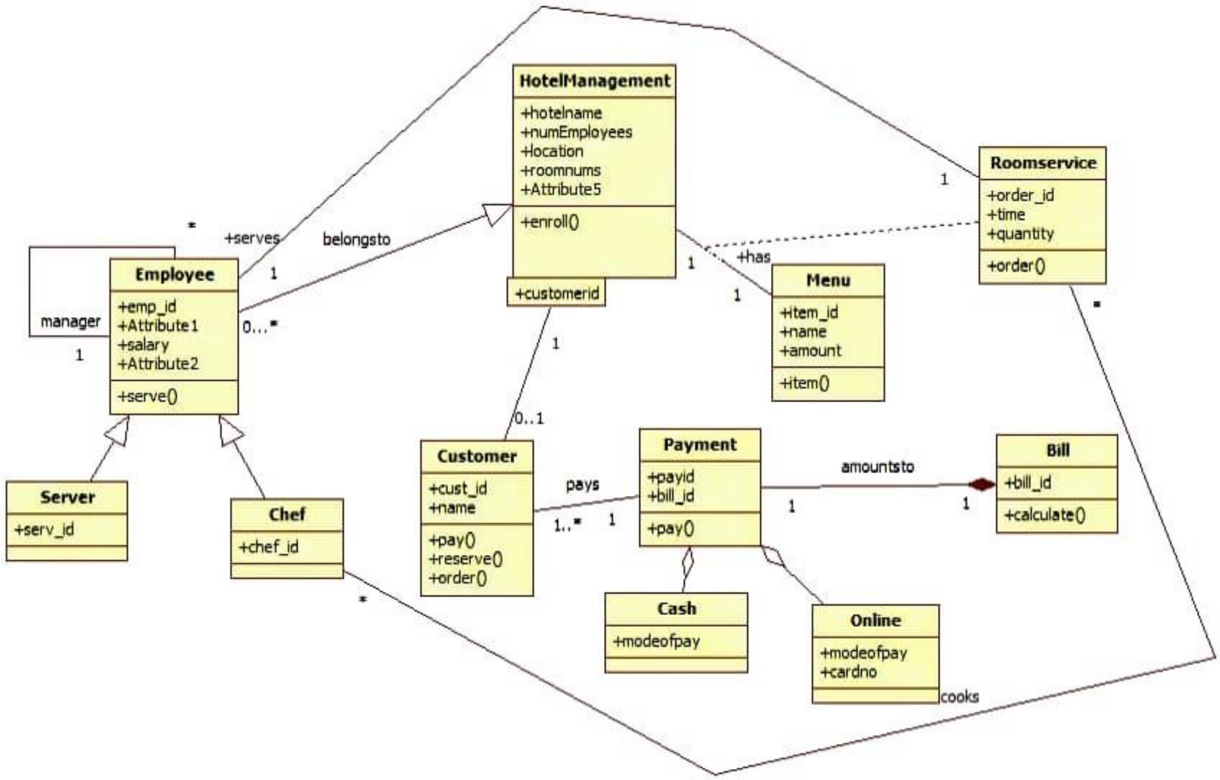
- Integration with existing systems: The system should be designed to integrate with existing systems such as property management systems, accounting software, and online booking engines.

- Time Constraints: The system should be designed and developed within the given timeline and should be ready to launch as per the agreed-upon schedule.

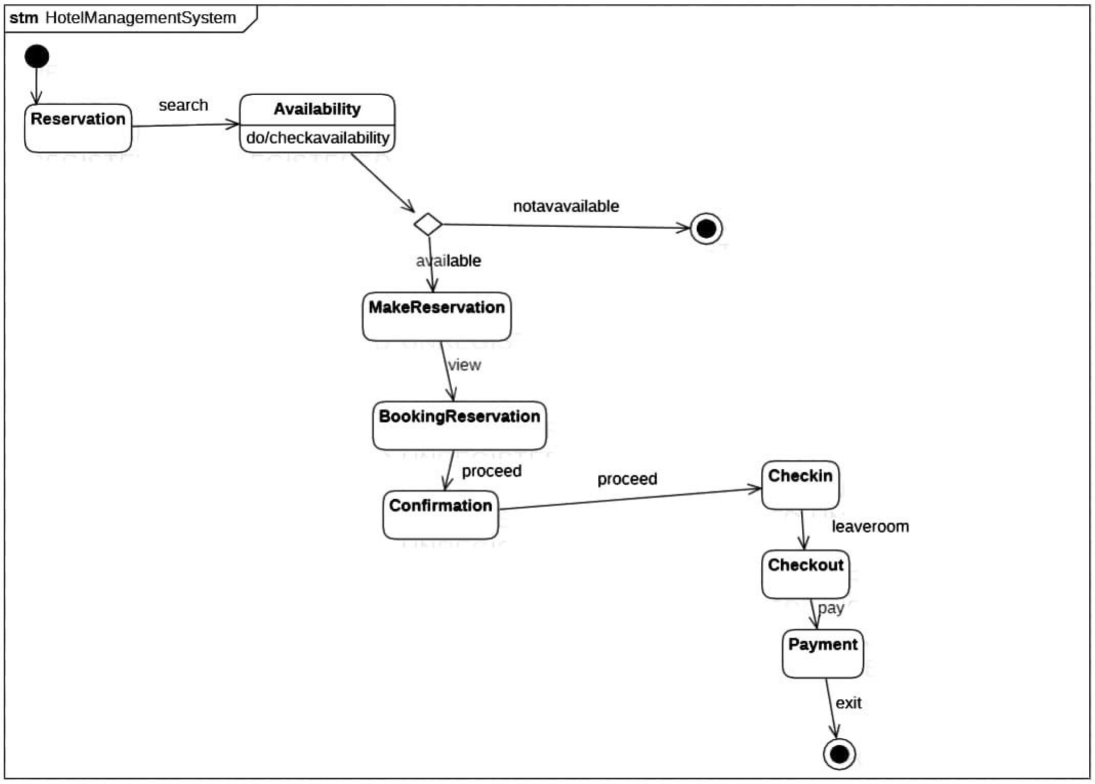
7. Non Functional Requirements:

- Performance
- Security
- Reliability
- Scalability
- Usability

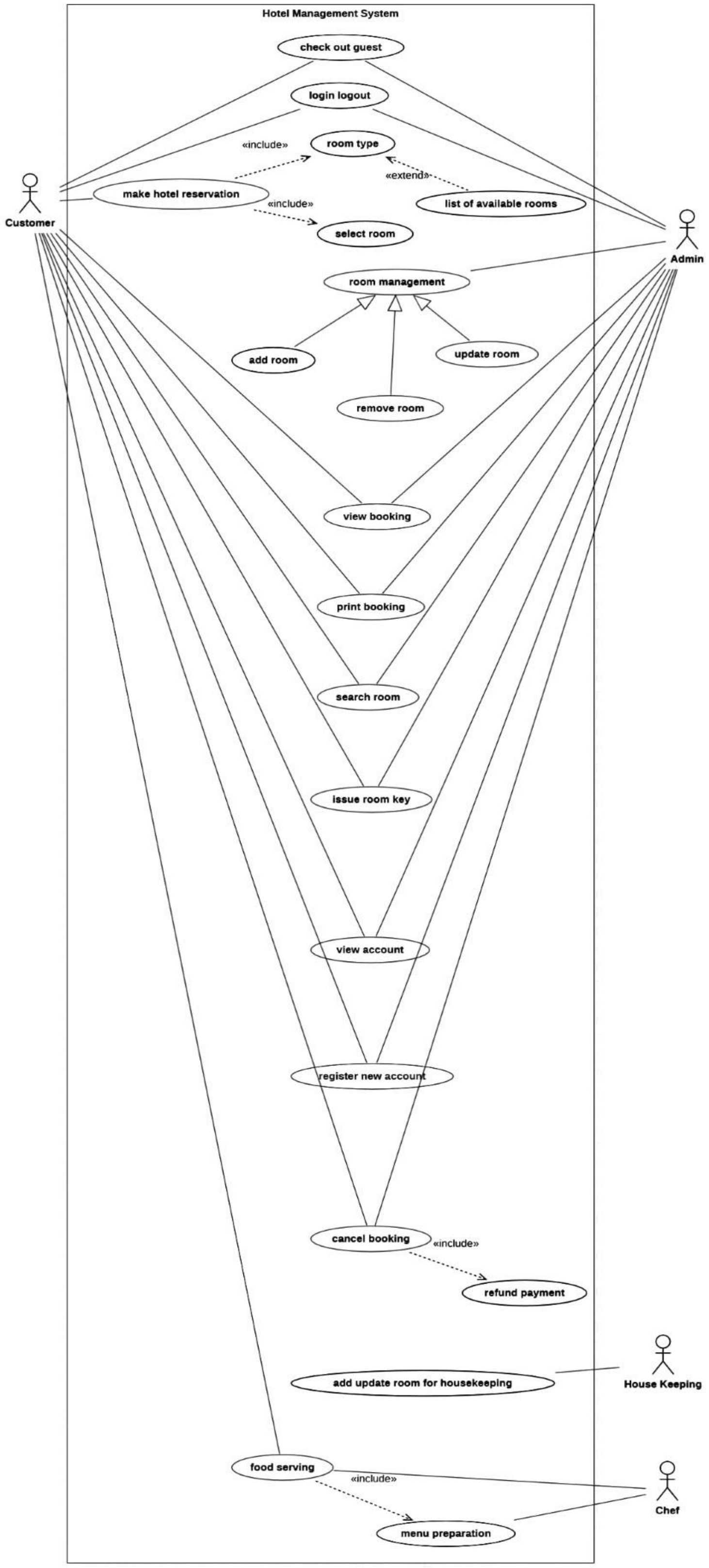
Class Diagrams



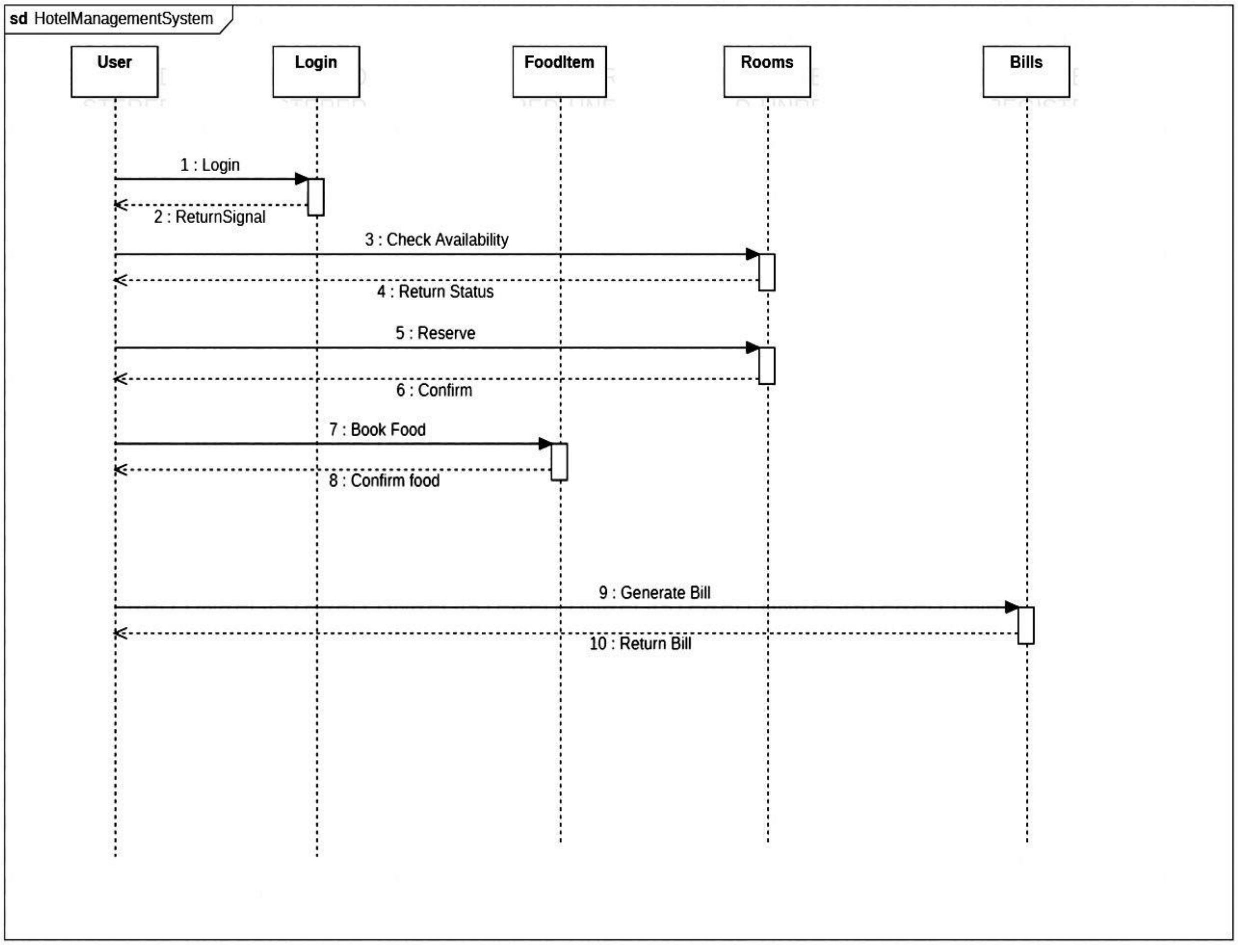
State Diagrams



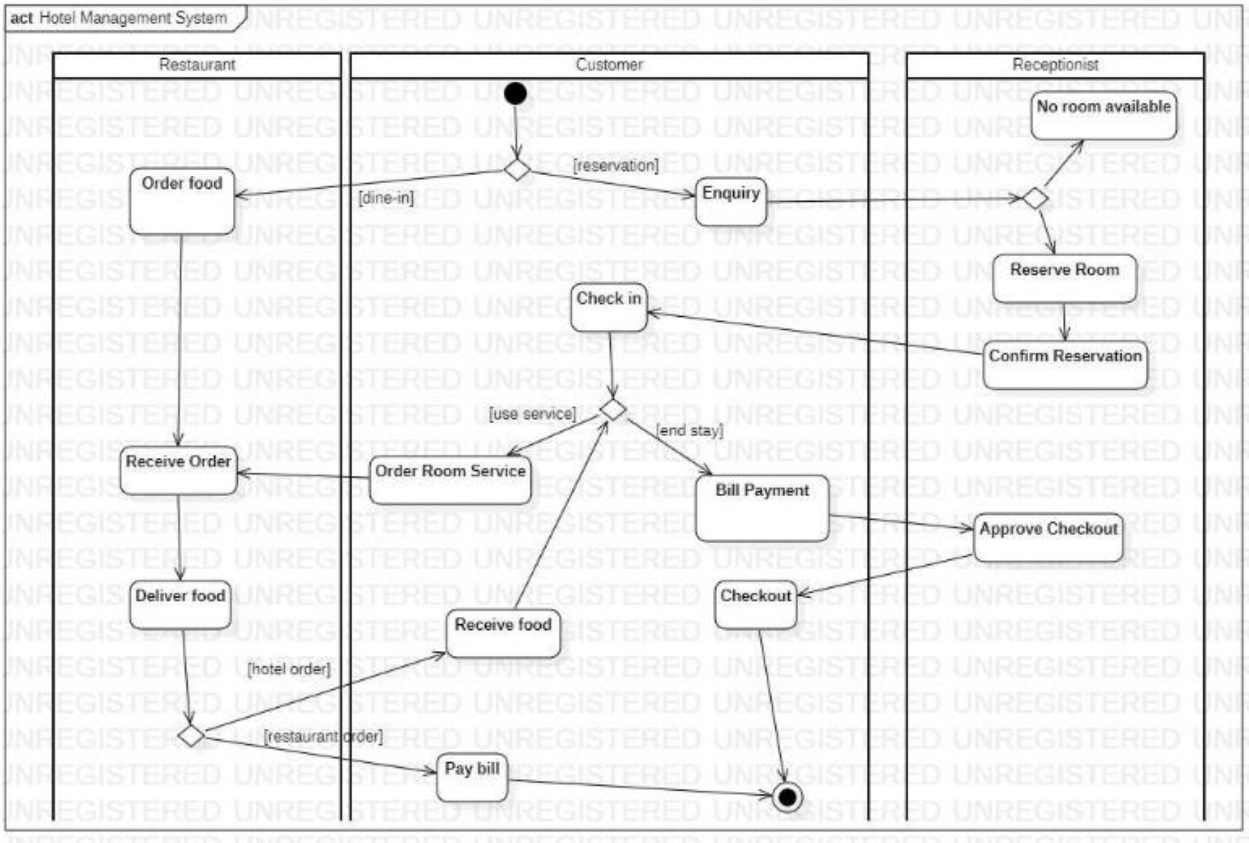
Use Case Diagrams



Sequence Diagrams



Activity Diagrams



2. Credit Card Processing

Problem Statement: Many businesses rely on credit card processing to accept payments from their customers. However, there are various challenges associated with this process, including security concerns, transaction fees, and technical issues. These challenges can impact the customer experience, result in lost sales, and potentially harm the reputation of the business. Therefore, it is essential to find solutions that address these challenges and ensure smooth and secure credit card processing for both businesses and customers.

SRS (Software Requirement Specification)

1. **Introduction:**
   a) **Purpose:** The purpose of credit card processing is to enable businesses to accept electronic payments from customers using credit or debit cards. Credit card processing involves a series of steps that authorize and verify the transaction, transfer funds from the customer's account to the merchant's account, and provide a record of the transaction for both parties. Credit card processing allows businesses to expand their customer base and reach a broader market by offering a convenient and preferred method of payment.

   b) **Scope:** The scope for credit card processing systems is significant as it enables businesses to accept electronic payments from customers using credit and debit cards. This system has a broad scope that spans across various industries, including retail, e-commerce, healthcare, finance, and more. The scope of credit card processing systems also includes compliance with regulations and security standards, such as PCI DSS, to ensure the protection of sensitive customer data.

c) **Overview:** Credit card processing is a system that enables businesses to accept electronic payments from customers using credit or debit cards. The process involves several steps, including authorization, verification, and settlement, to transfer funds from the customer's account to the merchant's account. Credit card processing provides a secure and convenient method of payment for customers and allows businesses to expand their customer base.

2. **General Description:** Credit card processing is a system that enables businesses to accept electronic payments from customers using credit or debit cards. The process involves several steps, including authorization, verification, and settlement, to transfer funds from the customer's account to the merchant's account. To process a credit card payment, the merchant swipes or dips the card into a payment terminal, which then sends the payment information to the payment processor.

3. **Functional Requirements:**

- Compatibility with various payment methods
- Security and compliance
- Speed and efficiency
- Accessibility and ease of use
- Integration with other systems

4. Interface Requirements:

- The user interface should be intuitive and easy to navigate, allowing users to quickly process transactions.

- The interface should be customizable to meet the unique needs of different merchants and businesses.

- It should provide clear and concise feedback to users, such as success or error messages, to ensure they understand the status of their transactions.

- The interface should have appropriate error handling capabilities to prevent data entry errors.

- The interface should have adequate accessibility features to cater to users with disabilities.

5. Performance Requirements:

- The credit card processing system should have a fast response time to ensure transactions are processed quickly.

- The system should be able to handle a high volume of transactions without slowing down or crashing.

- The system should be designed to minimize latency and reduce the risk of timeouts.

- The system should be optimized to handle peak usage times and sudden spikes in traffic without compromising performance.

- The system should have effective monitoring and logging capabilities to identify and resolve performance issues.

6. **Design Constraints:**

- Compliance with security standards and regulations such as PCI-DSS and GDPR.

- Integration with various third-party services such as payment gateways, card networks, and banks.

- Support for various types of credit cards and payment methods, as per business needs.

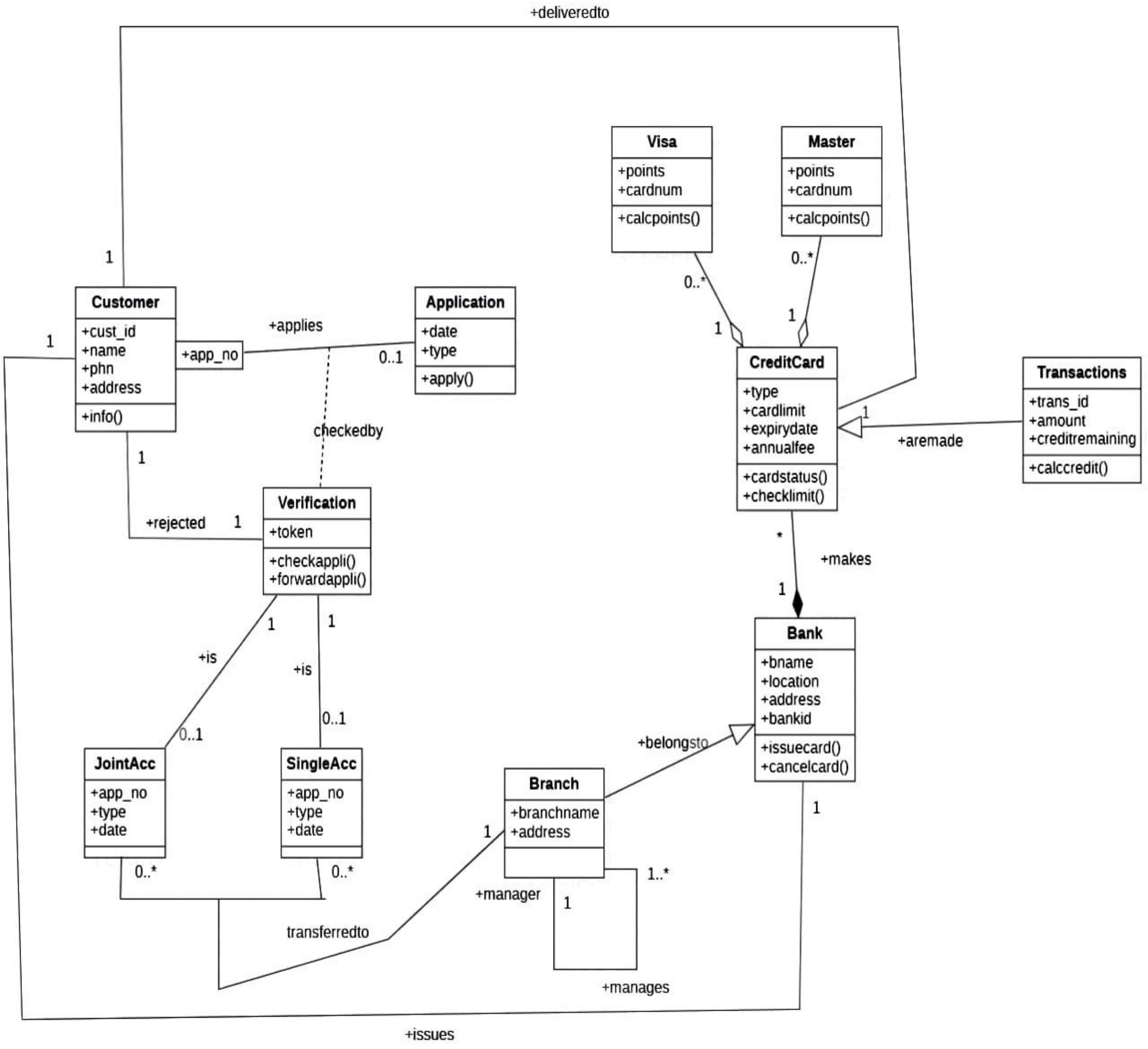
- Compatibility with existing systems, platforms, and technologies used by the business.

- Consideration of hardware and infrastructure requirements for the system, such as server capacity and network bandwidth.

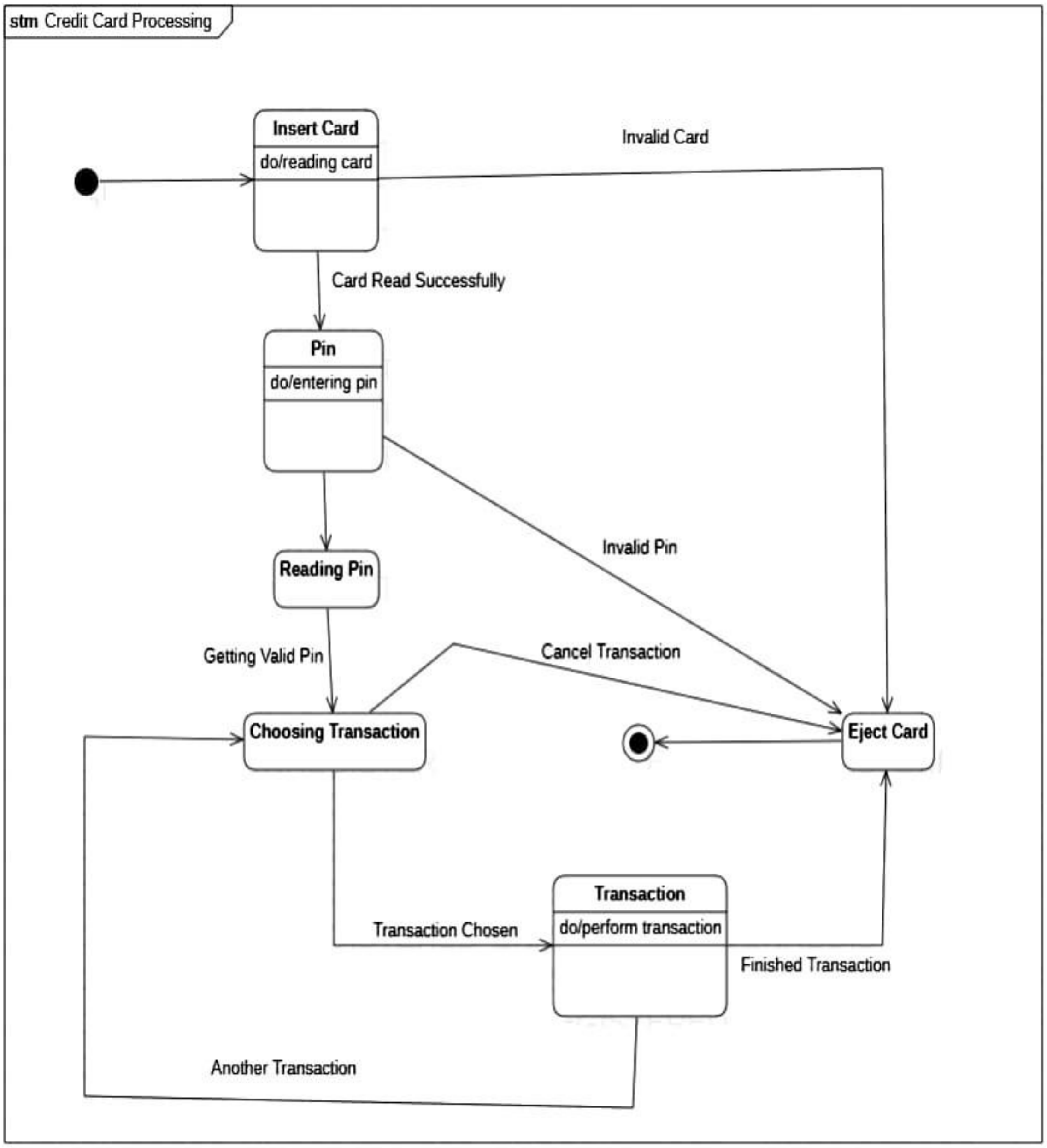
7. **Non Functional Requirements:**

- Security
- Reliability
- Scalability
- Usability

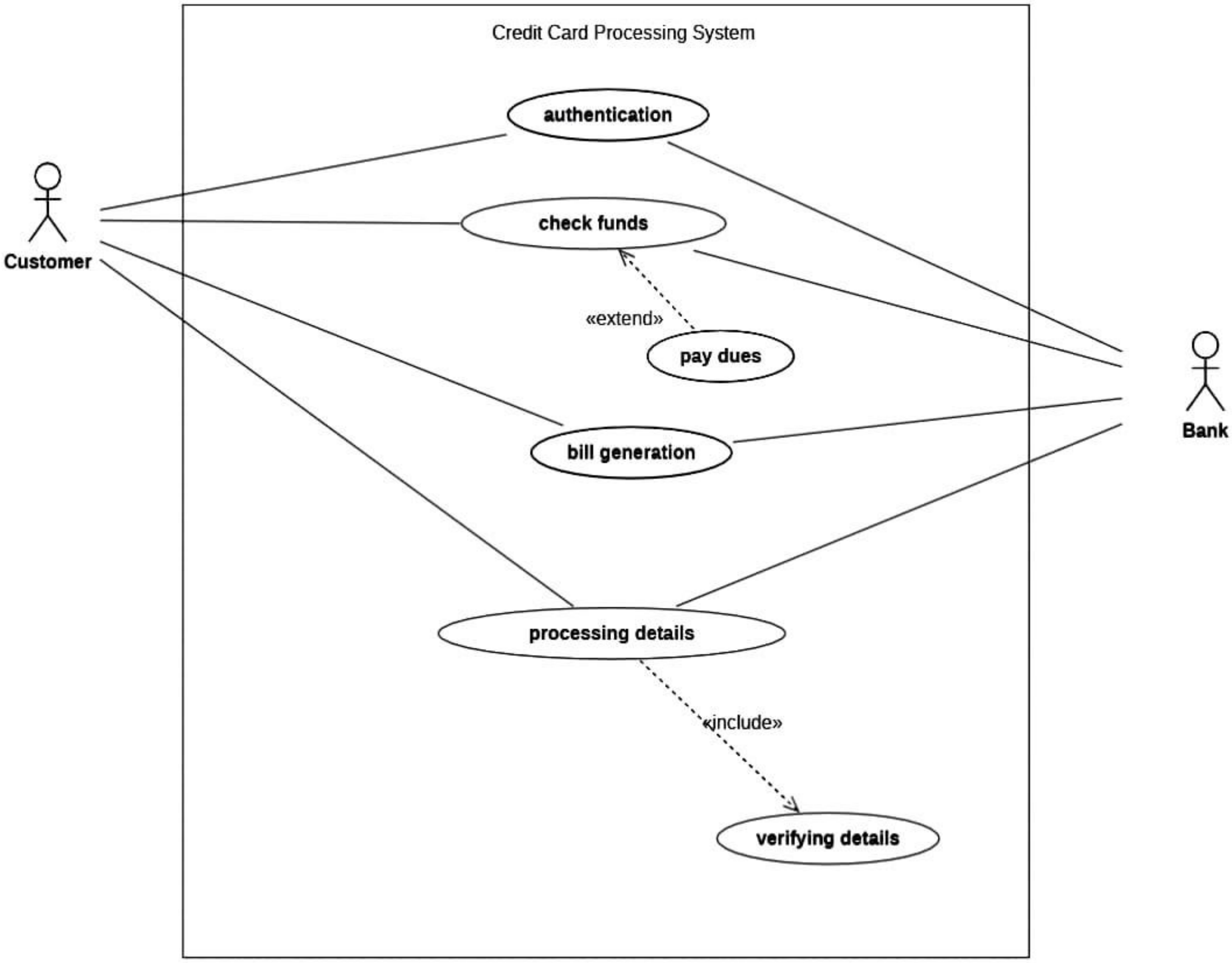
Class Diagrams



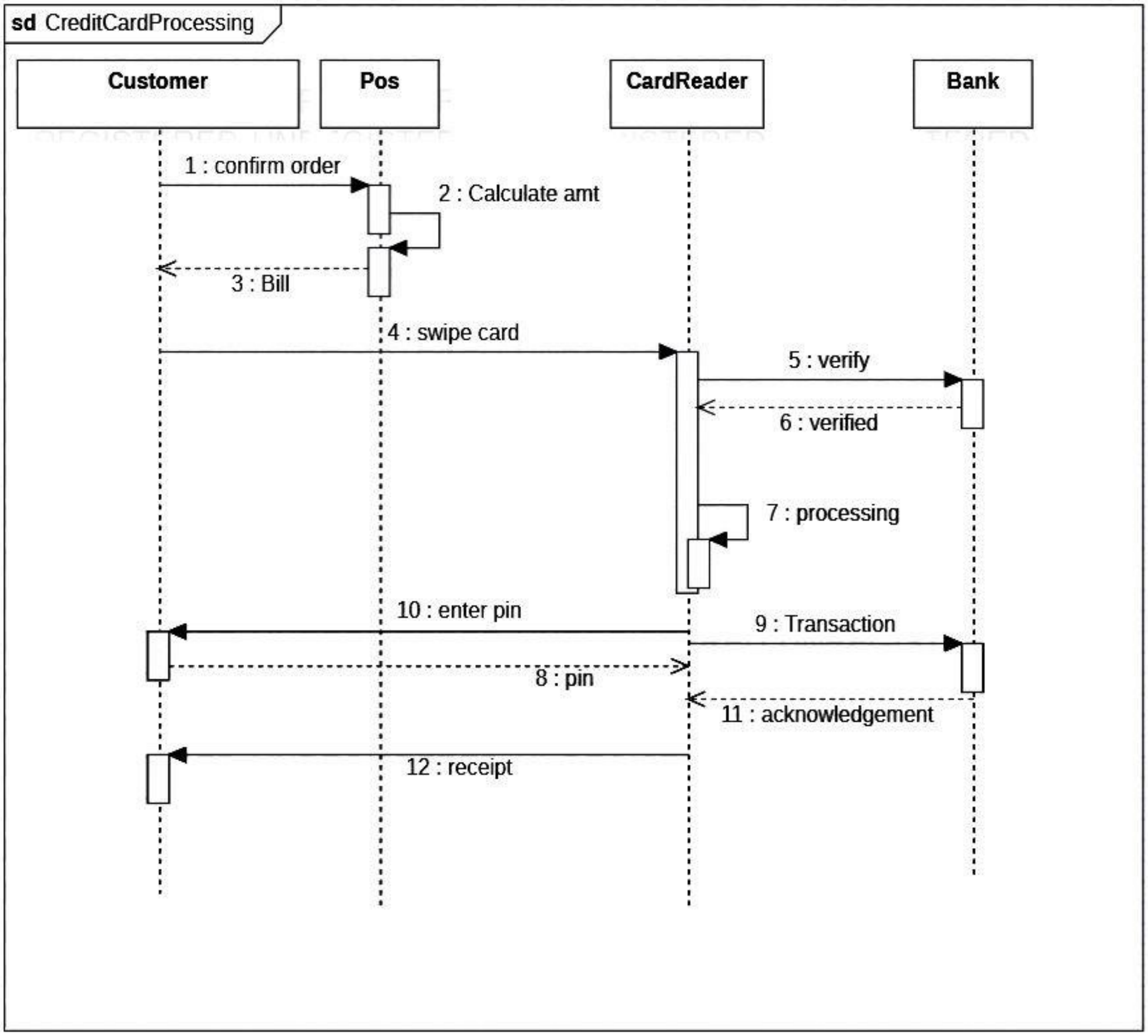
State Diagrams



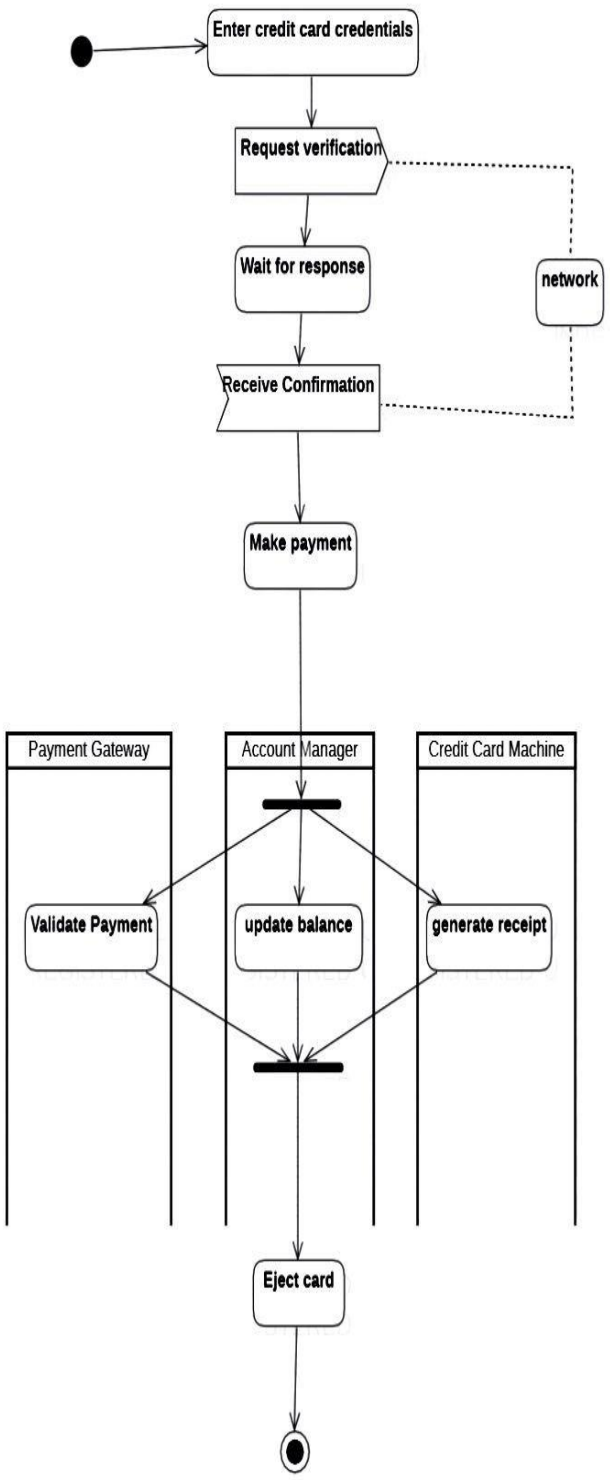
Use Case Diagrams



Sequence Diagrams



Activity Diagrams



3. Library Management System

Problem statement: A library management system is a software application designed to help librarians efficiently manage various library activities such as cataloging, circulation, acquisition, and patron management. The system should allow librarians to track books, manage user accounts, process check-ins and check-outs, and provide access to library resources.

SRS (Software Requirement Specification)

1. **Introduction:**

   a) **Purpose:** The purpose of a library management system is to provide an efficient and effective way of managing the various operations of a library. The system aims to streamline the process of cataloging books, managing user accounts, and tracking circulation. It also provides patrons with access to the library's resources and services.

   b) **Scope:** The scope of a library management system is broad and encompasses various activities related to library operations. It includes the management of the library's physical and digital resources, circulation, and user management. The system should be able to provide access to a vast collection of books, journals, and other library resources.

   c) **Overview:** A library management system is a software application designed to help librarians manage various aspects of a library's operations. The system includes features such as cataloging, circulation, acquisition, and patron management.

2. **General Description:** A library management system is a software application designed to help librarians efficiently manage various library activities such as cataloging, circulation, acquisition, and patron management. The system provides users with access to the library's resources and services and helps librarians track the usage of library materials.

3. **Functional Requirements:**

- Cataloging and classifying library materials
- Managing user accounts, including creating, updating, and deleting accounts
- Processing check-ins and check-outs of library materials
- Tracking overdue items and fines, including sending notifications to patrons
- Generating reports on library usage, financial transactions, and overdue items

4. **Interface Requirements:**

- Intuitive and user-friendly interfaces for both librarians and patrons, with clear navigation and easy-to-use features.
- Responsive and adaptable interfaces that can adjust to different screen sizes and resolutions, providing a consistent user experience across different devices and platforms.
- Consistent interfaces that follow a clear and consistent design language, making it easy for users to navigate and understand the system.
- Support for multiple languages and user preferences, allowing users to customize their experience and access content in their preferred language.
- Accessible interfaces that follow accessibility guidelines such as WCAG 2.1, making the system usable for users with disabilities.

5. Performance Requirements:

- Fast response time for user interactions such as searches, checkouts, and returns.
- Ability to handle large volumes of data, including bibliographic records, patron data, and circulation data.
- Efficient data processing and storage, minimizing system load times and ensuring smooth performance even during peak usage periods.
- Minimal latency for network communications between the system and users, ensuring a responsive and seamless user experience.
- Support for concurrent user connections, allowing multiple users to access the system simultaneously without experiencing performance degradation.

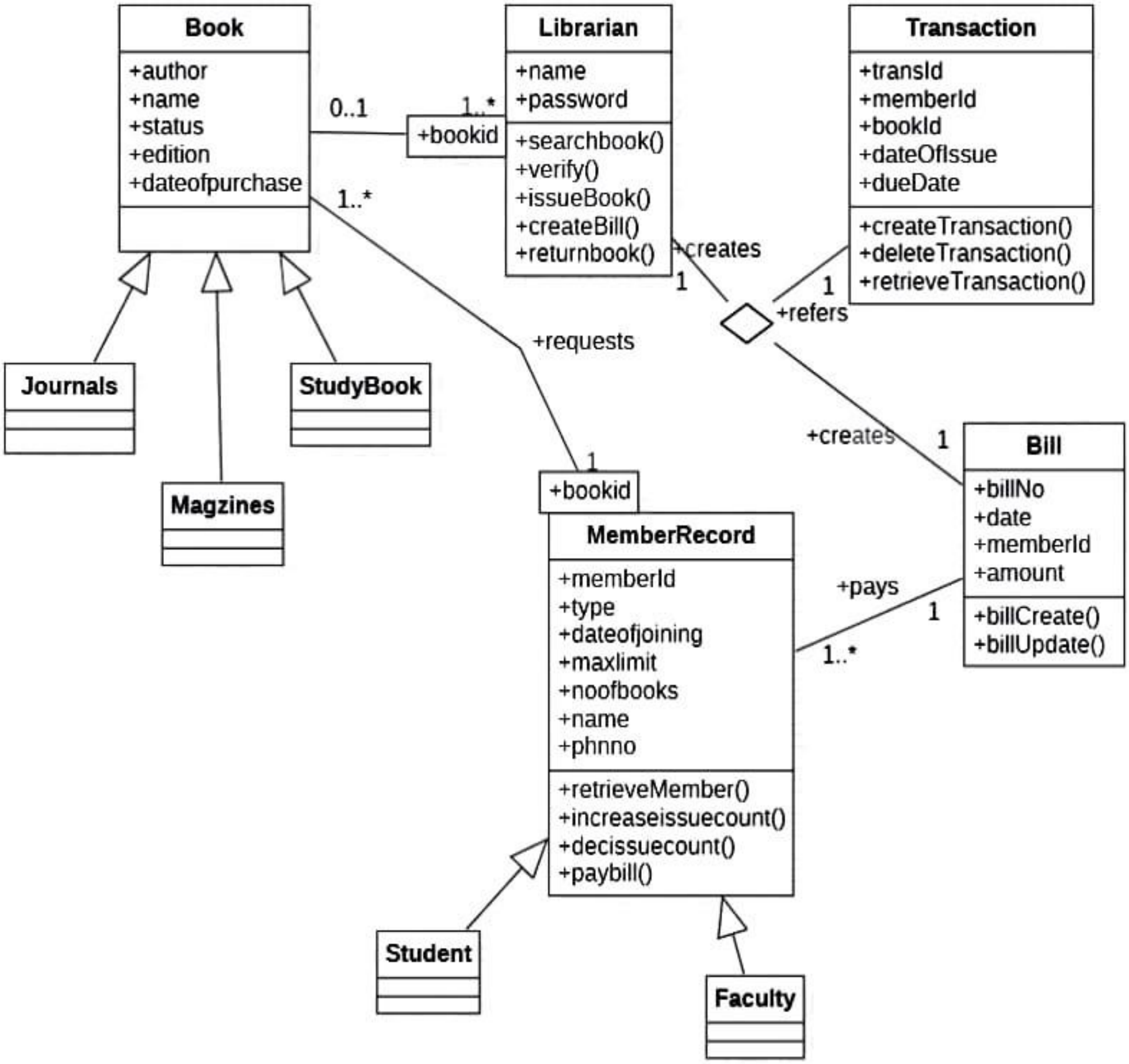
6. Design Constraints:

- Compliance with industry standards and regulations such as MARC and AACR2 to ensure data consistency and interoperability with other library systems.
- Adherence to privacy and data protection laws to protect user and library data.
- Use of open standards and protocols to ensure interoperability with other systems and applications.
- Use of industry-standard database management systems such as MySQL or PostgreSQL to ensure reliability and scalability.
- Use of industry-standard programming languages and frameworks to ensure maintainability and ease of integration with other systems.

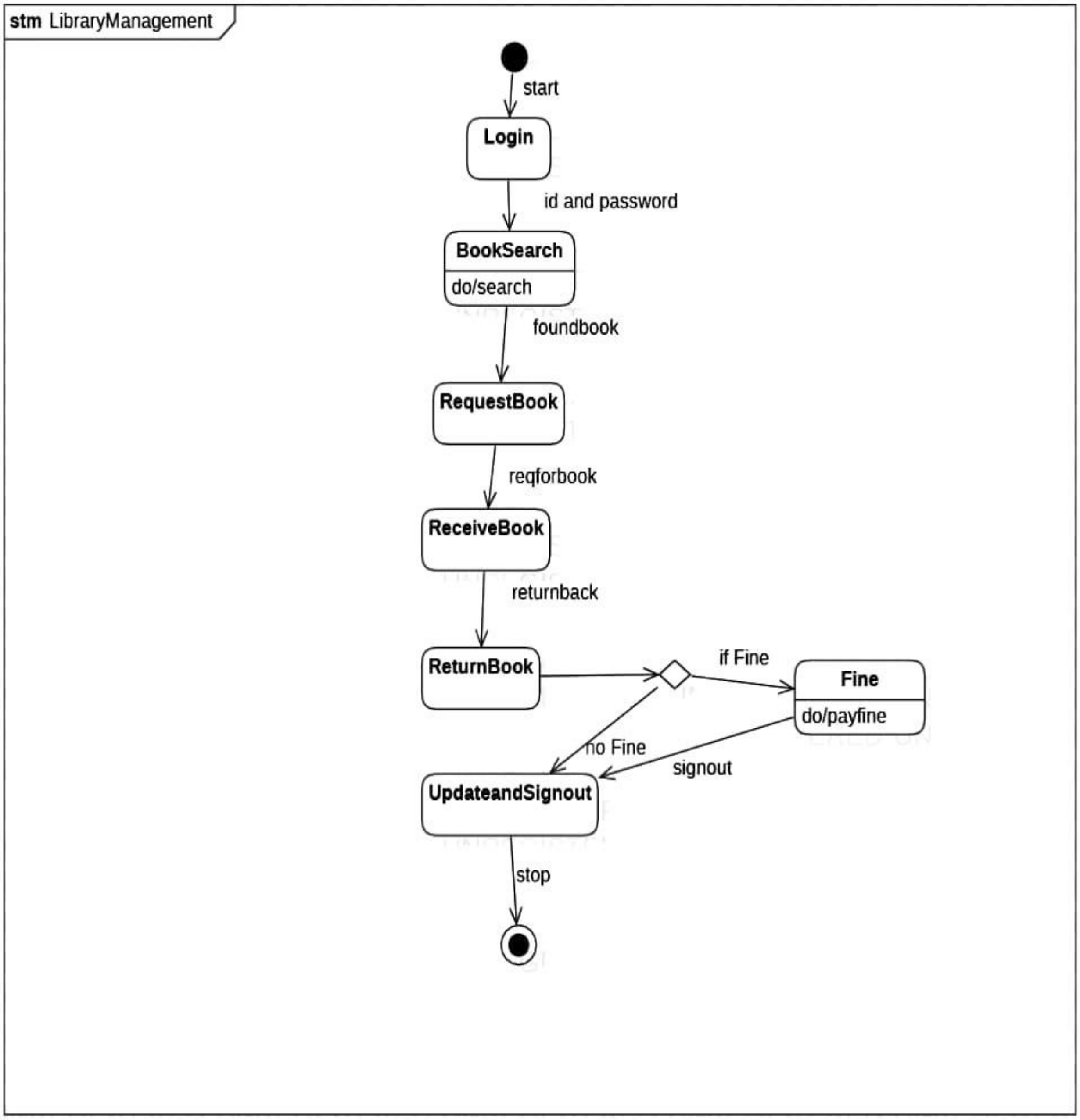
7. Non Functional Requirements:

- Performance
- Security
- Reliability
- Scalability
- Usability

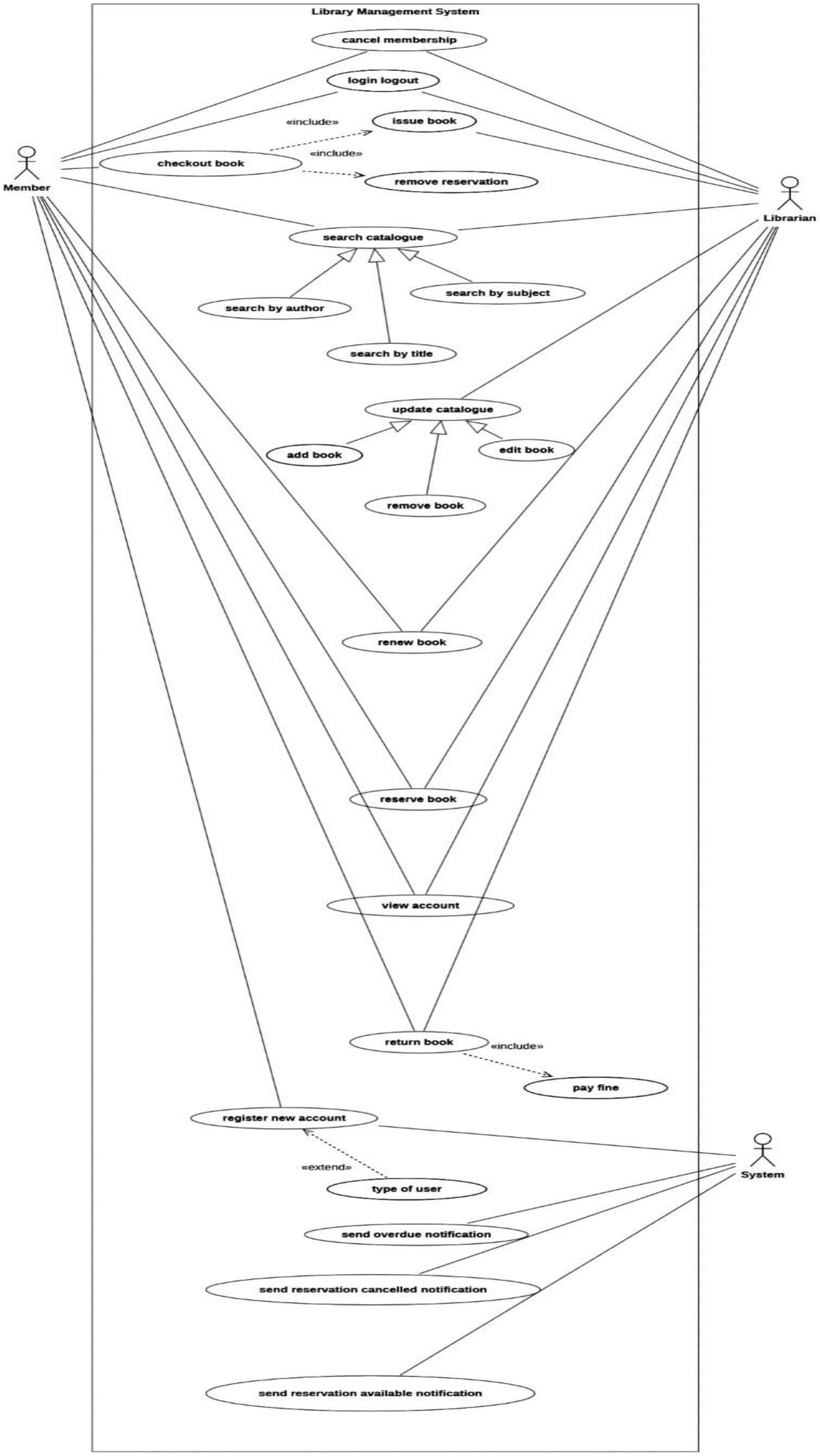
Class Diagrams



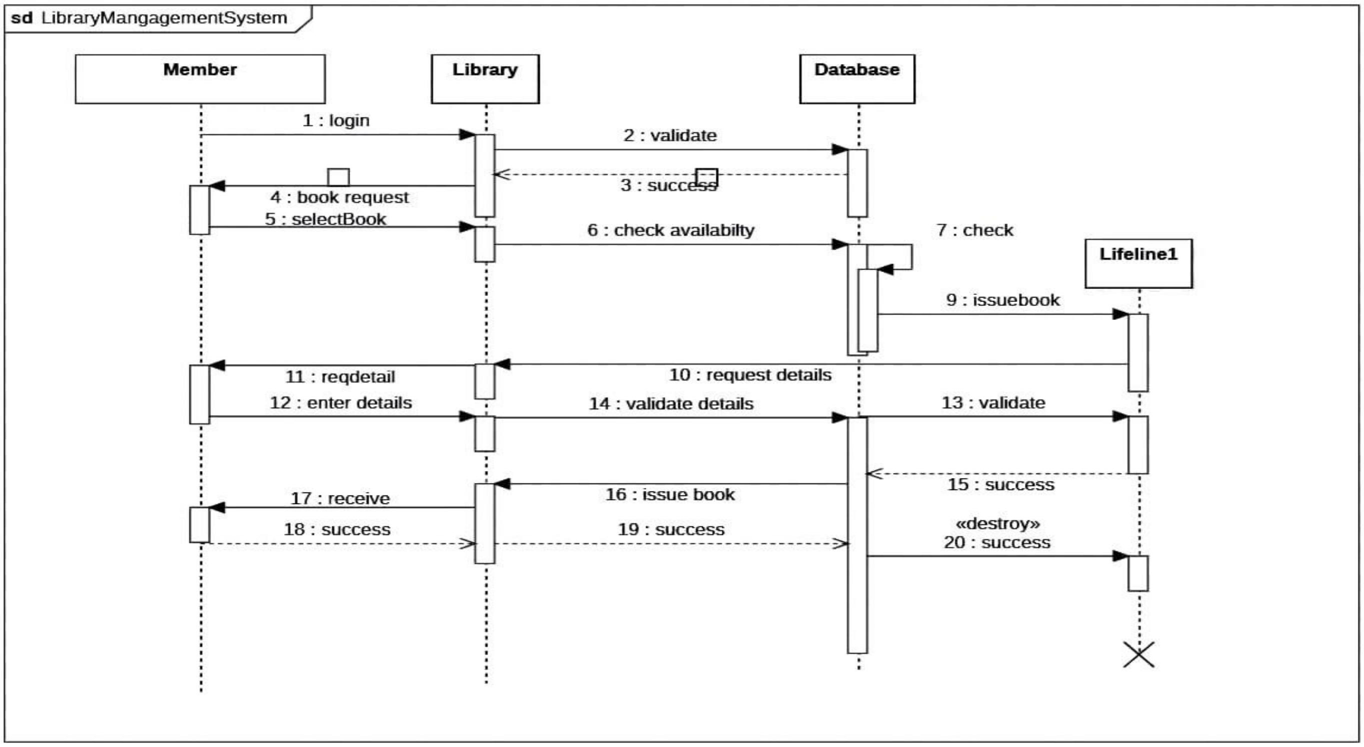
State Diagrams



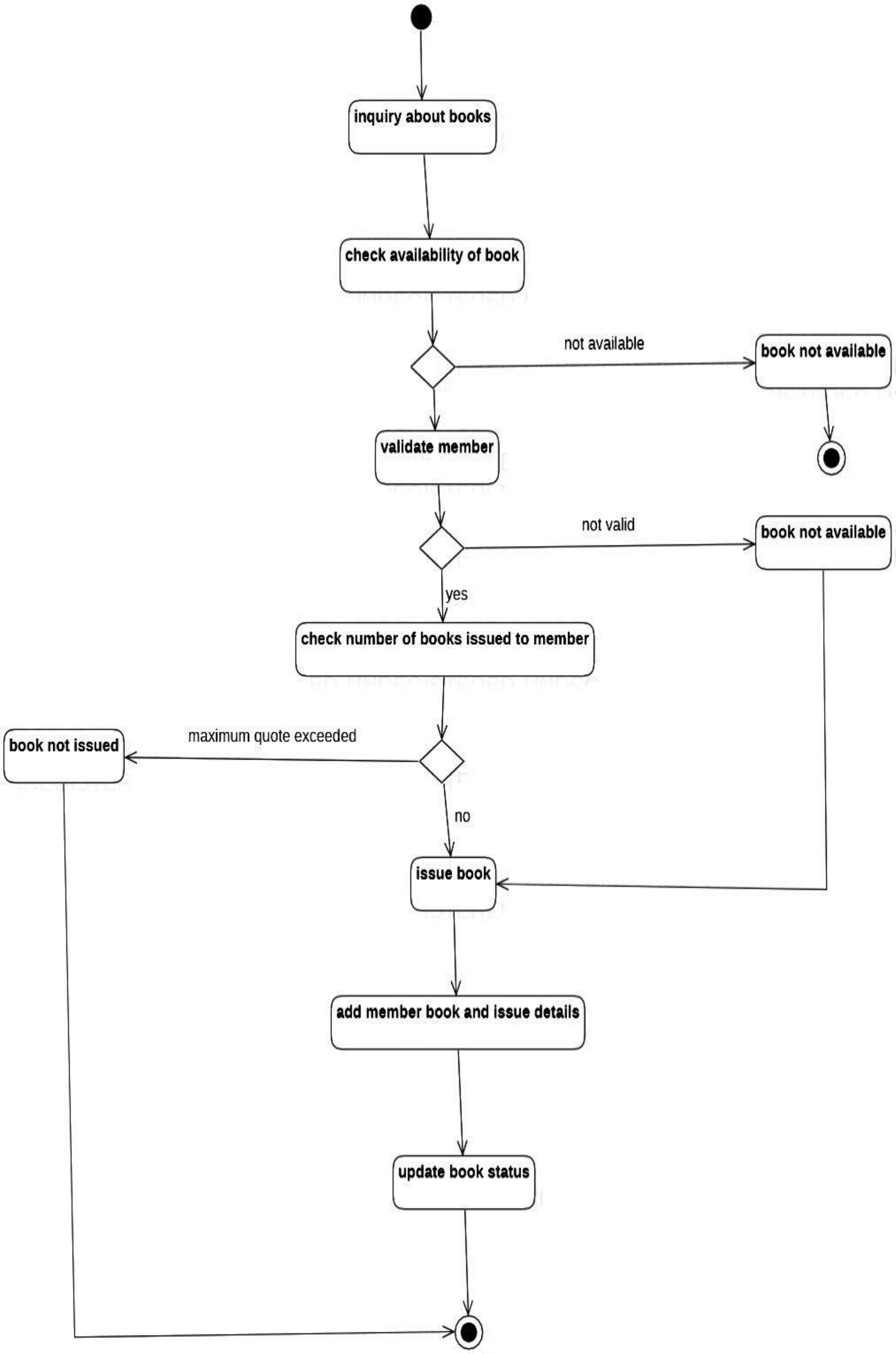
Use Case Diagrams



Sequence Diagrams



Activity Diagrams



4. Stock Maintenance System

Problem statement: The stock maintenance system is a software solution that enables businesses to efficiently manage their inventory levels by monitoring stock levels, tracking orders, and managing shipments. This system allows businesses to maintain a real-time view of their inventory, ensuring that they never run out of stock

SRS (Software Requirement Specification):

1. **Introduction:**
   a) **Purpose:** The purpose of a stock maintenance system is to streamline inventory management and improve operational efficiency by providing real-time visibility into inventory levels, tracking stock movements, and automating inventory-related tasks. By using a stock maintenance system, businesses can ensure that they always have the right amount of stock on hand to meet customer demand, reducing the risk of stockouts or overstocking.

   b) **Scope:** The scope of a stock maintenance system includes managing inventory levels, tracking stock movements, and automating inventory-related tasks. The system can be used in various industries, including retail, manufacturing, healthcare, and logistics. It can manage inventory at different stages of the supply chain, including raw materials, work-in-progress, and finished goods.

   c) **Overview:** A stock maintenance system is a software solution that enables businesses to efficiently manage their inventory levels by tracking stock movements, automating inventory-related tasks, and providing real-time visibility into inventory levels.

2. **General Description:** A stock maintenance system is a software solution that helps businesses manage their inventory levels by providing real-time visibility into stock levels, tracking stock movements, and automating inventory-related tasks. The system can handle different types of inventory, including raw materials, work-in-progress, and finished goods, and can manage inventory across multiple warehouses.

3. **Functional Requirements:**
   - Track inventory levels in real-time
   - Monitor stock movements and update inventory records
   - Generate alerts when stock levels reach a certain threshold
   - Manage different types of inventory, including raw materials, work-in-progress, and finished goods
   - Handle inventory across multiple warehouses

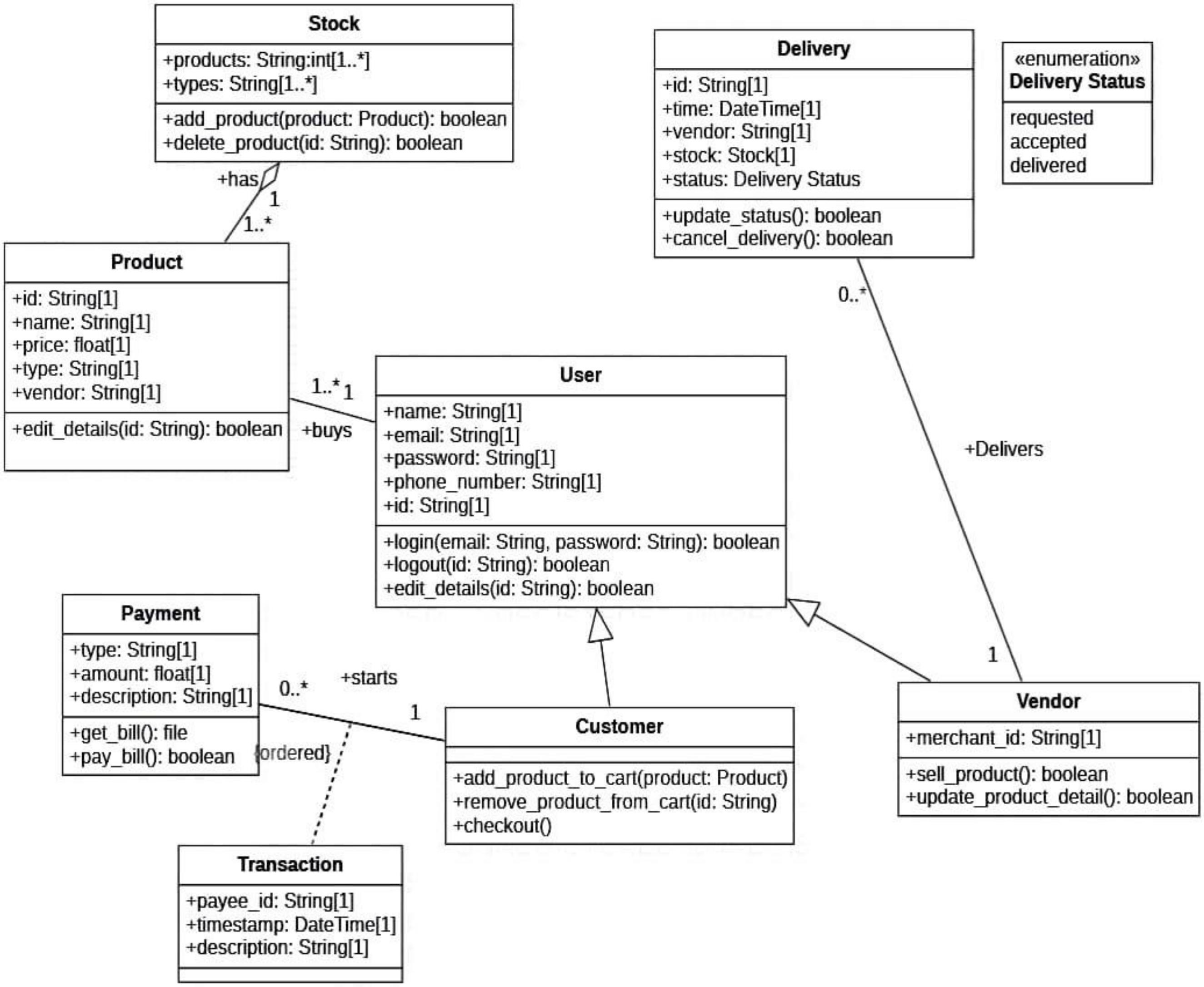
4. **Interface Requirements:**
   - User-friendly interface that is easy to navigate and understand
   - Responsive design that supports different devices and screen sizes
   - Customizable dashboard that provides an overview of inventory levels and stock movements
   - Search and filter functionality to quickly find specific inventory items or transactions
   - Drag-and-drop functionality to move inventory between locations or update stock levels

5. **Performance Requirements:**
   - High availability to ensure the system is always accessible and operational
   - Fast response times to user requests and transactions
   - Scalability to handle increasing inventory volumes and transactions over time
   - Reliable data storage and backup to prevent data loss and ensure data integrity

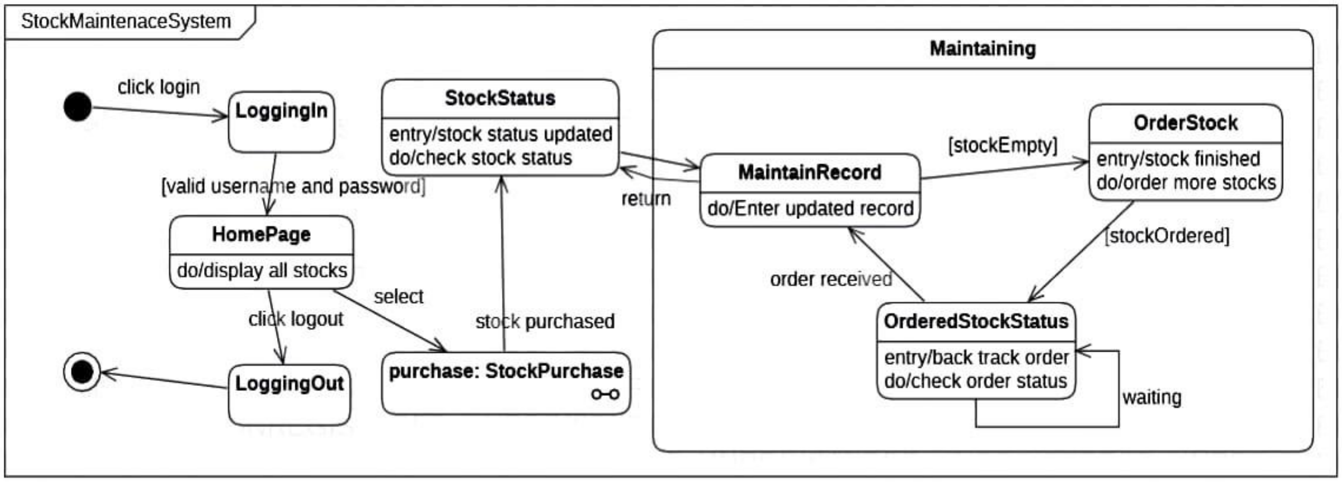
6. **Design Constraints:**
   - Compatibility with existing hardware and software systems in the organization
   - Adherence to industry standards and regulations, such as ISO or FDA guidelines for inventory management
   - Budget constraints and cost-effectiveness of the solution
   - Scalability to accommodate future growth and changing business needs
   - Usability and accessibility for users with different skill levels and abilities

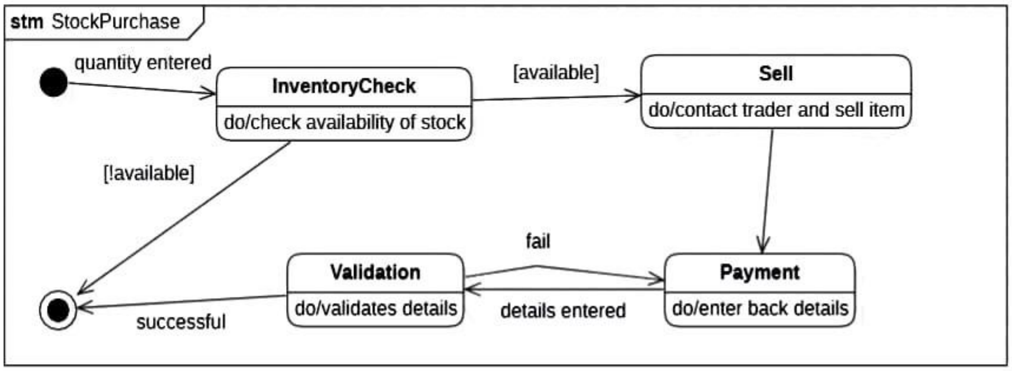
7. **Non Functional Requirements:**
   - Reliability: The system should be highly reliable, with minimal downtime or system failures to ensure business continuity.
   - Security: The system should be designed with strong security controls to ensure data confidentiality, integrity, and availability.
   - Usability: The system should be easy to use and navigate, with a user-friendly interface that requires minimal training.
   - Performance: The system should provide fast response times, high throughput, and low latency to support high transaction volumes and user concurrency.
   - Scalability: The system should be able to handle increasing inventory volumes and user concurrency over time.

Class Diagrams

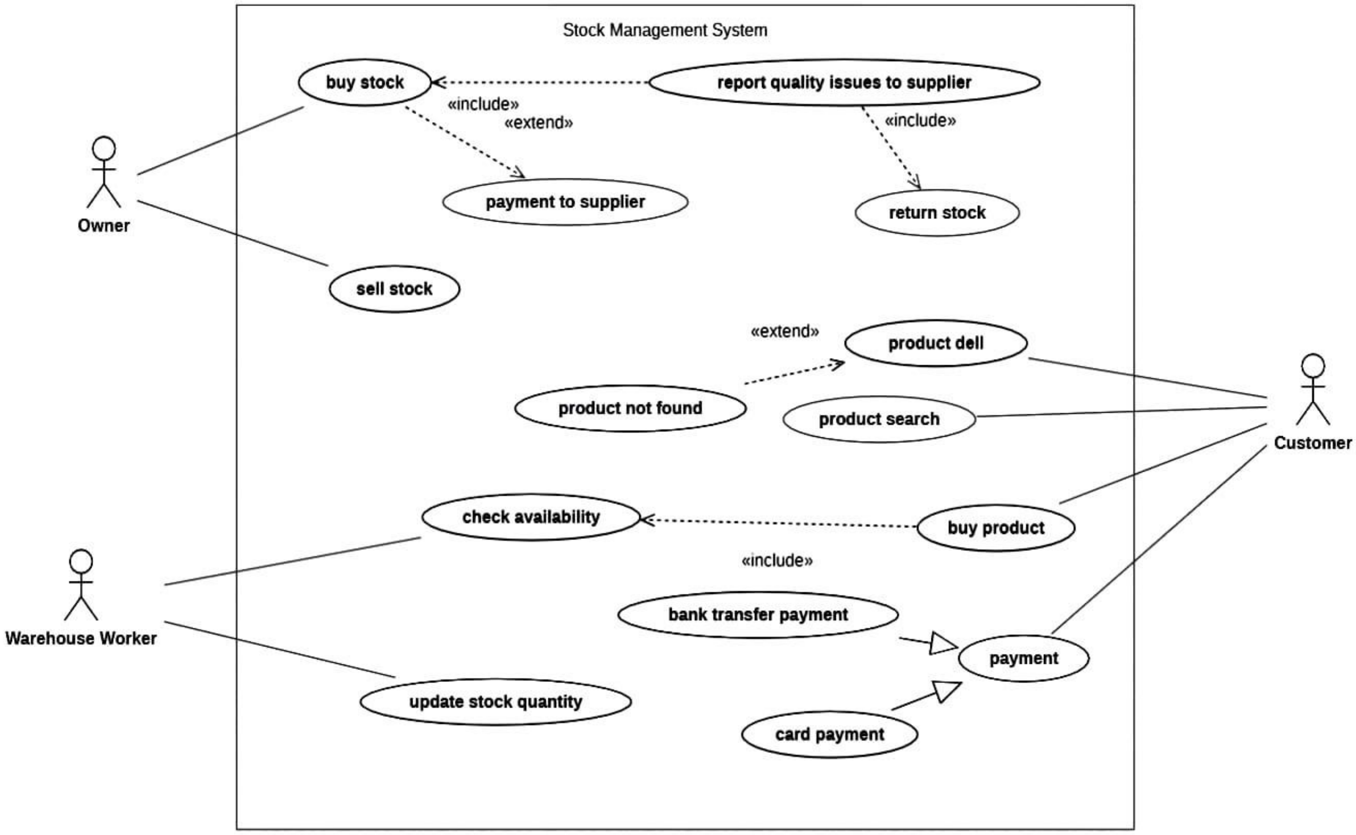


State Diagrams

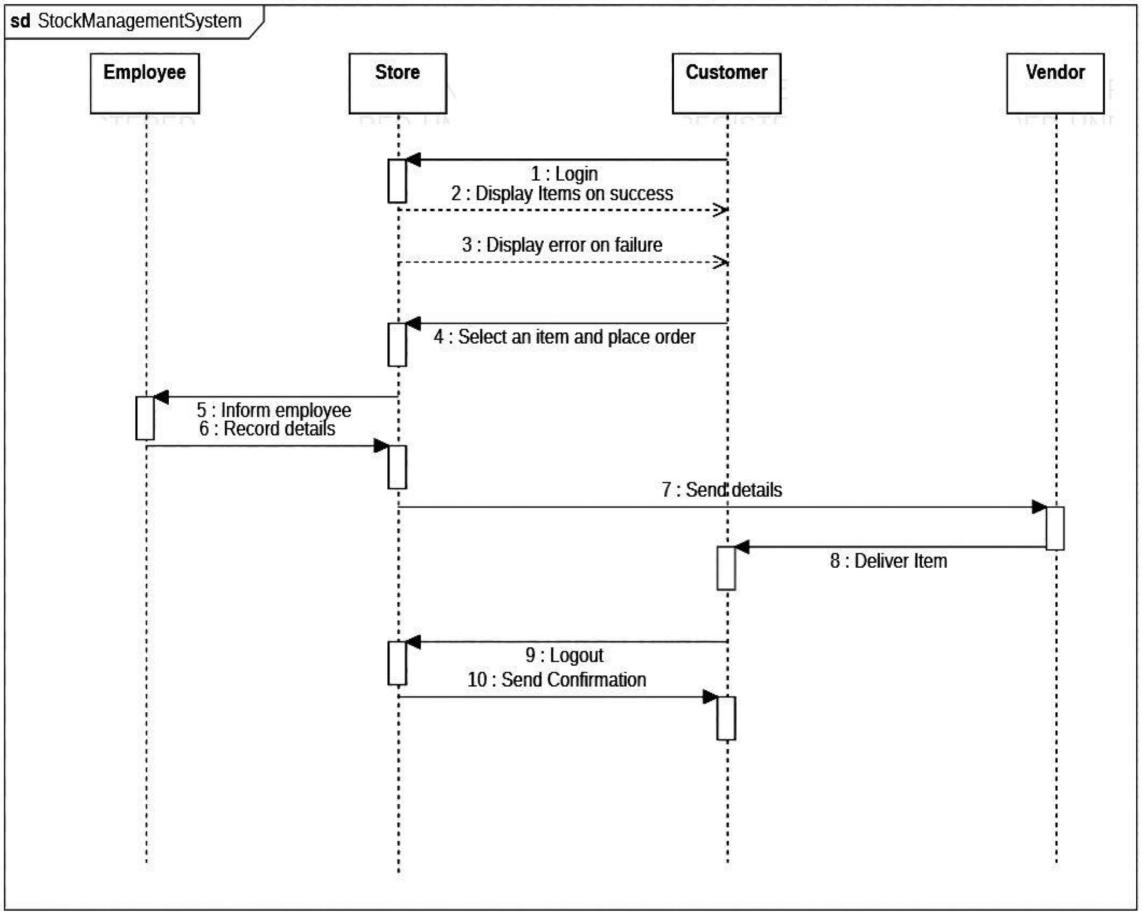




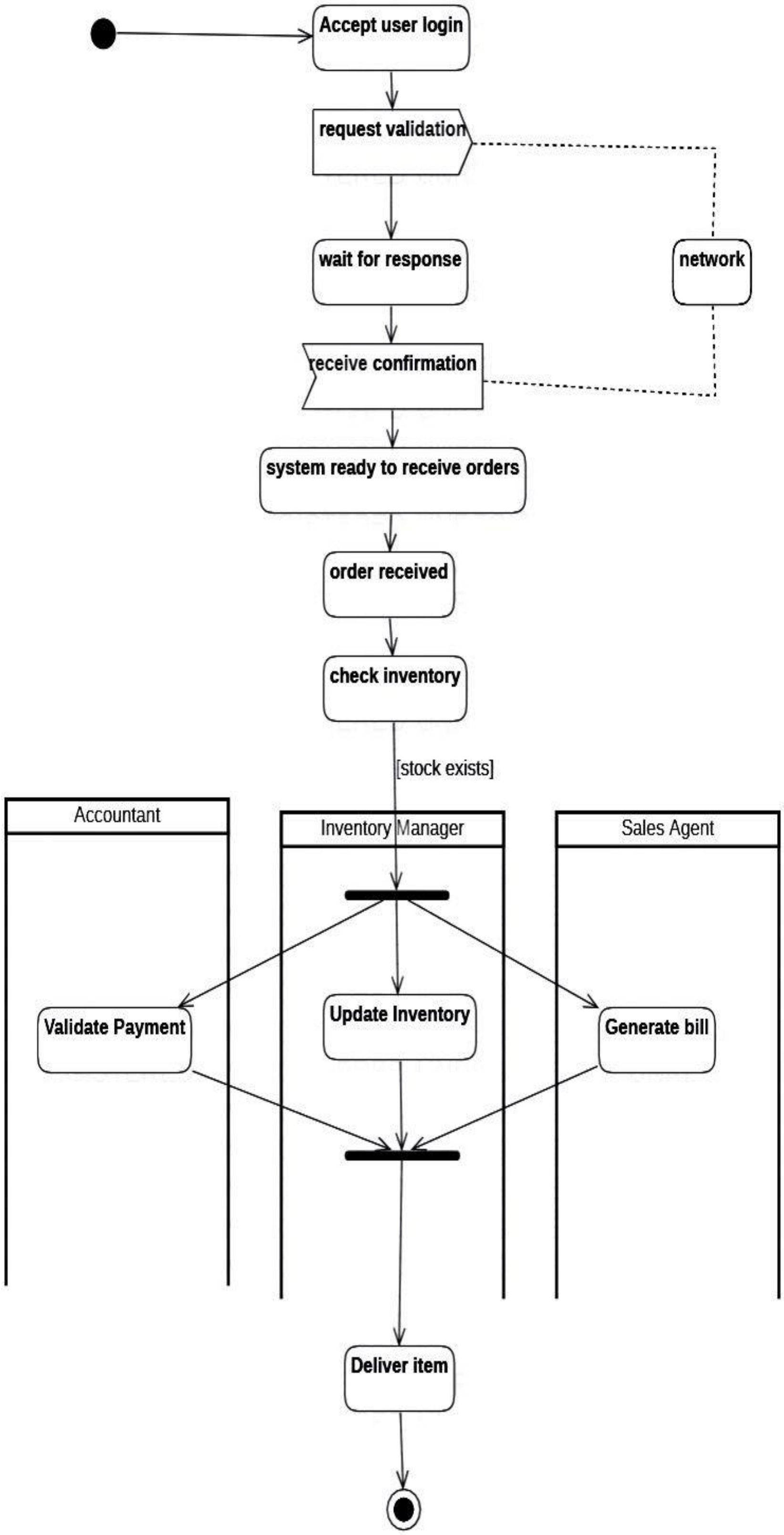
Use Case Diagrams



Sequence Diagrams



Activity Diagrams



5. Passport Automation System

Problem statement: Passport automation system is designed to simplify the process of applying for and issuing passports. The current manual process is time-consuming, prone to errors, and lacks transparency. The objective of the passport automation system is to create a streamlined and efficient process for issuing passports to citizens. The system will allow applicants to apply for a passport online, schedule an appointment for biometric verification, and track the status of their application. It will also allow officials to manage and process applications more efficiently, reducing processing times and improving overall service delivery.

Software Requirement Specification (SRS)

1. Introduction:

1.1 Purpose of this document: The purpose of the passport automation system is to improve the process of issuing passports to citizens. By automating the application process, the system aims to reduce processing times, eliminate errors, increase transparency, and improve overall service delivery. The system also aims to provide citizens with a convenient and user-friendly platform to apply for passports and track the status of their applications.

1.2 Scope of this document: The scope of the passport automation system includes designing and implementing a user-friendly online platform for passport application, appointment scheduling, and status tracking. The system will integrate biometric verification, data validation, and background checks to ensure the accuracy and reliability of information provided by applicants.

1.3 **Overview:** The passport automation system is an online platform designed to streamline the process of applying for and issuing passports. The system will allow citizens to apply for a passport online, schedule an appointment for biometric verification, and track the status of their application. The system will also support online payment and facilitate online communication between applicants and officials.

2. **General Description:** The passport automation system is an automated platform that simplifies the process of passport issuance. The system enables citizens to apply for a passport online, schedule an appointment for biometric verification, and track the status of their application. The system includes biometric verification, data validation, and background checks to ensure the accuracy and reliability of information provided by applicants.

3. **Functional Requirements:**

- User registration and authentication for citizens, officials and admins.
- Online passport application form submission with required information.
- Biometric verification of the applicant's identity.
- Data validation and background checks to ensure the accuracy and reliability of information provided by applicants.
- Scheduling appointments for biometric verification and document submission.

4. Interface Requirments:

- User-friendly interface for citizens, officials and admins.
- Intuitive and responsive design for easy navigation and accessibility on different devices.
- Clear and concise instructions for each step of the passport application process.
- Easy-to-use form fields for applicants to enter their information.
- Multi lingual support to accommodate citizens from different language backgrounds.

5. Performance Requirments:

- High availability and reliability of the system to prevent downtime.
- Fast response time for loading pages and submitting forms.
- Scalability to accommodate a large volume of passport applications and appointments.
- Capacity to handle simultaneous user requests without performance degradation.
- Efficient use of system resources to minimize system load and optimize performance.

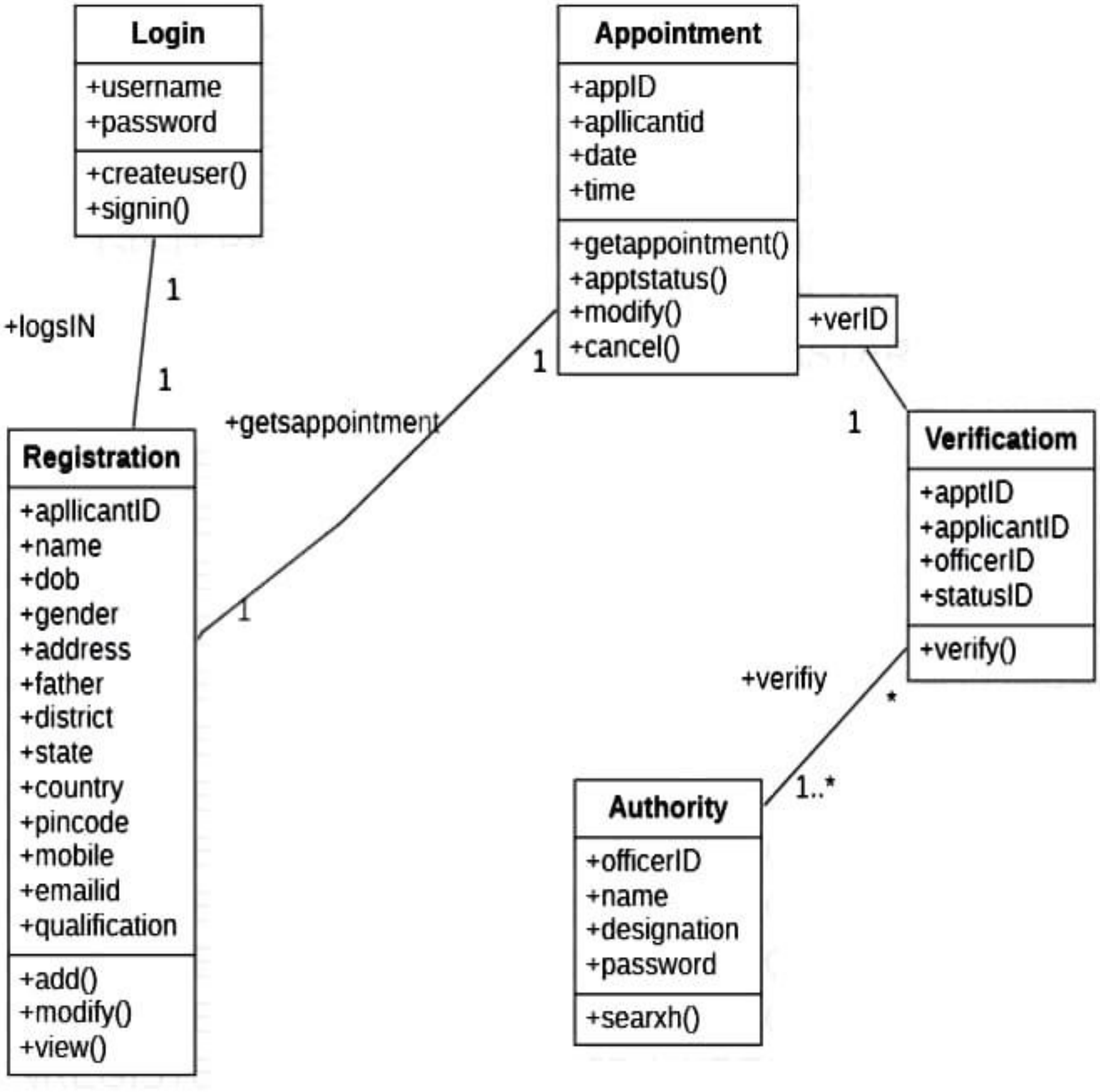
6. Design constraints:

- Compliance with government regulations and standards for passport issuance.
- Compatibility with existing passport systems and infrastructure.
- Integration with biometric verification systems and border security systems.
- Integration with payment gateway and online payment systems.
- Data privacy and security requirements to protect personal data of applicants.

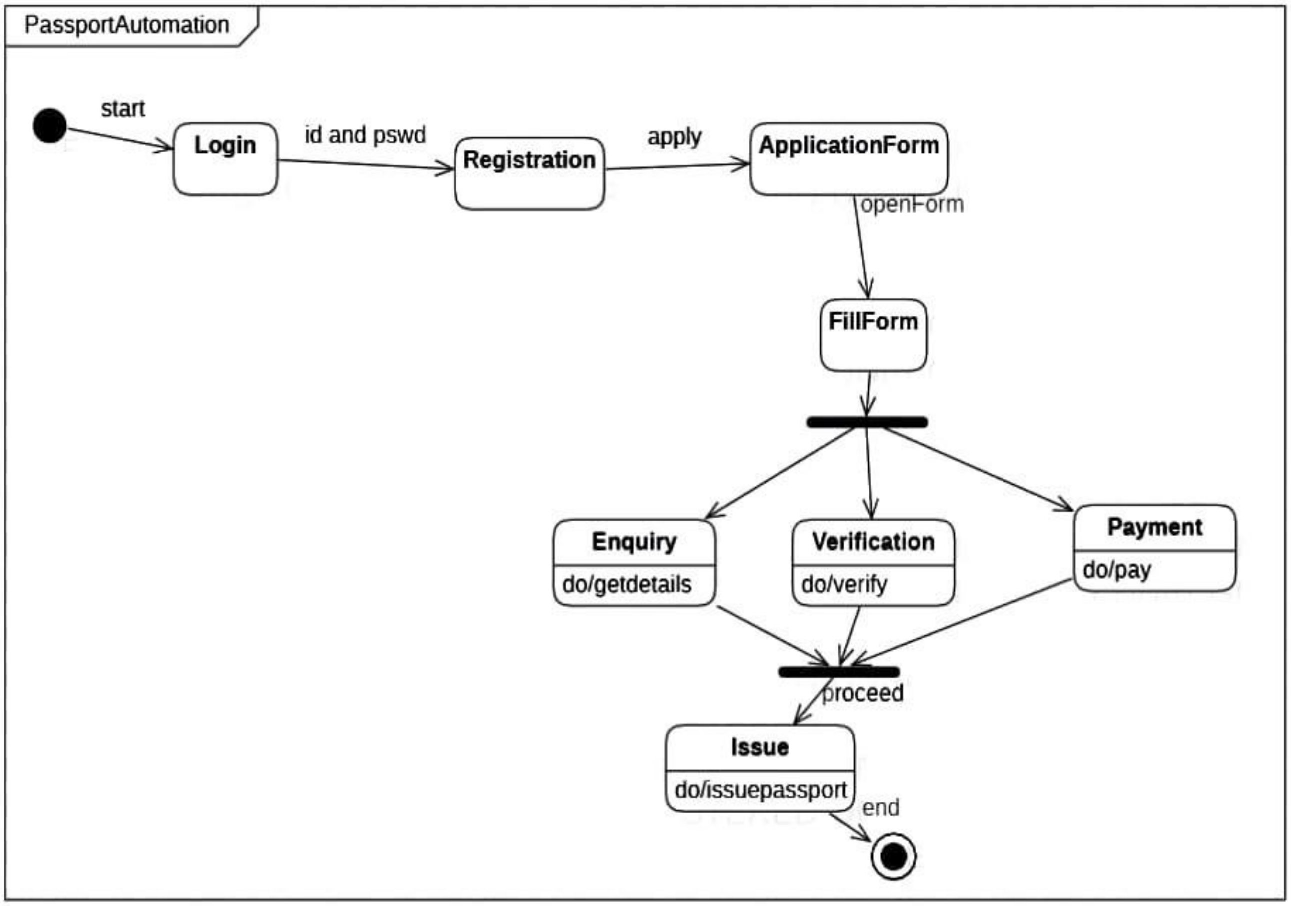
7. Non Functional Requirments:

- Usability: The system should be user-friendly and easy to use for citizens, officials and admins.
- Reliability: The system should be reliable and available 24/7 to ensure smooth passport issuance and appointment scheduling.
- Security: The system should have robust security measures to prevent data breaches or cyber-attacks and ensure the privacy of personal data.
- Performance: The system should have fast response times and be able to handle a large volume of passport applications and appointments without degradation of performance.
- Compatibility: The system should be compatible with different devices, web browsers and operating systems.

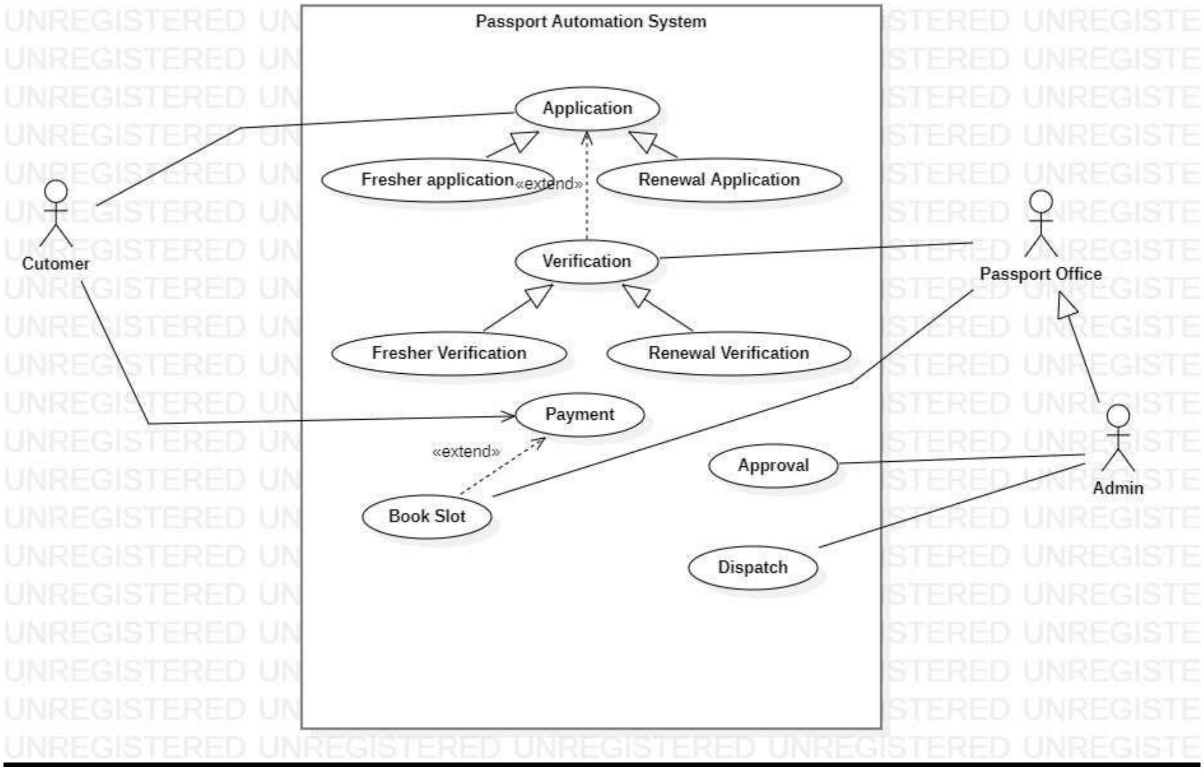
Class Diagrams



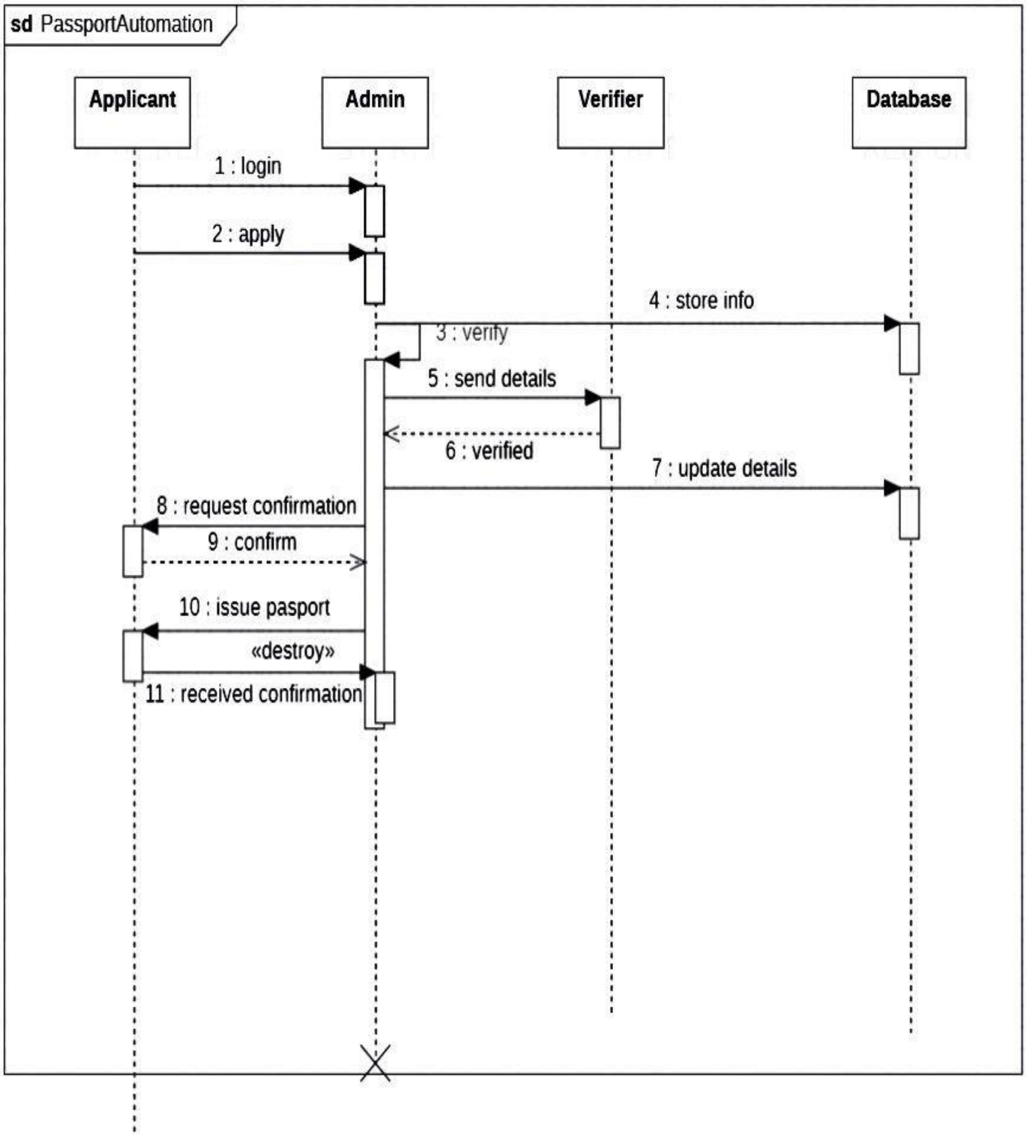
State Diagrams



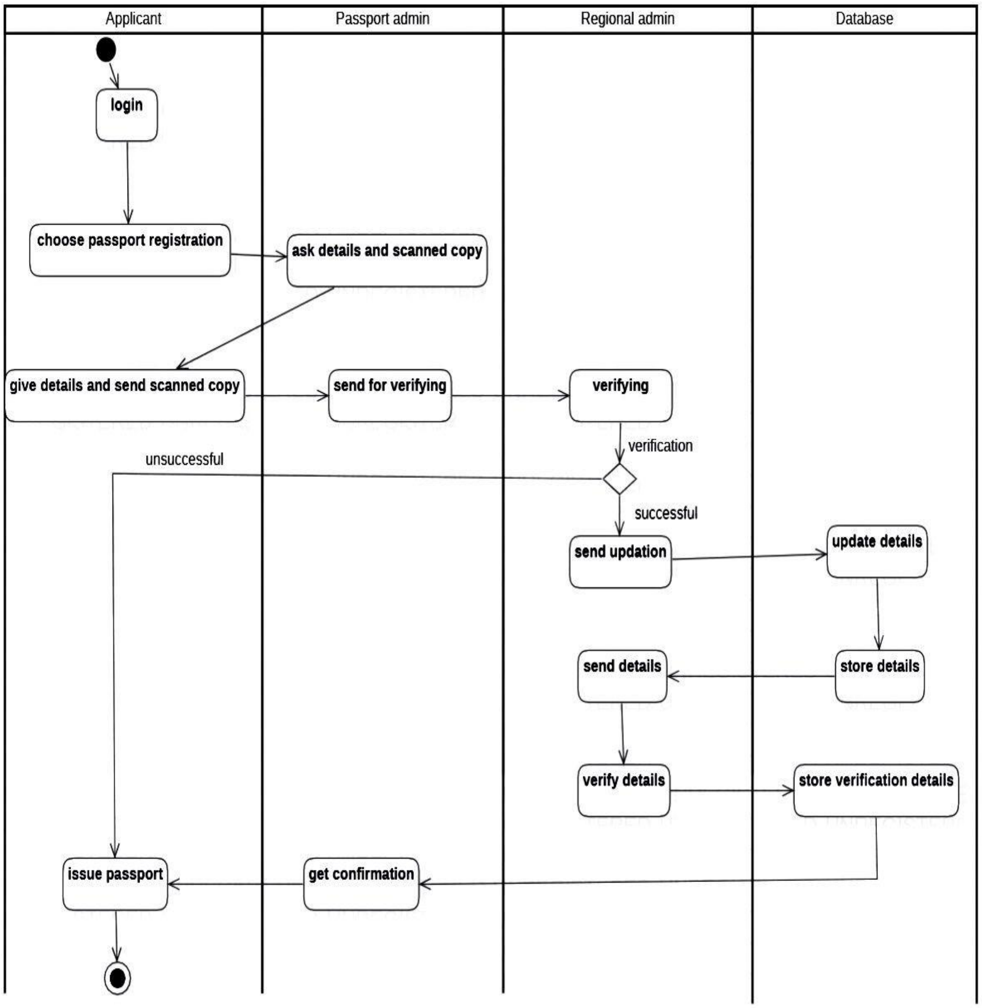
Use Case Diagrams



Sequence Diagrams



Activity Diagrams



6. Railway Reservation System

Problem statement: The railway reservation system aims to provide an efficient and user-friendly platform for customers to book train tickets, check seat availability, and make cancellations or modifications to their bookings. However, the system often faces challenges such as technical glitches, slow processing times, and limited availability of seats, which can result in customer dissatisfaction and revenue loss.

Software Requirement Specification (SRS):

1. Introduction:

1.1 Purpose of this document: The purpose of the railway reservation system is to provide a convenient and reliable platform for customers to book train tickets from anywhere at any time. It aims to streamline the ticket booking process and eliminate the need for customers to visit the booking counters physically. The system also enables railway authorities to manage seat availability, track passenger information, and generate revenue reports efficiently.

1.2 Scope of this document: The scope of the railway reservation system is vast and includes a range of functions such as ticket booking, seat availability management, passenger information management, and revenue reporting. The system can be accessed by customers from anywhere through online platforms or mobile applications, making it accessible and convenient for a large number of users. The system also provides opportunities for integrating additional features such as e-wallets, online payment gateways, and travel insurance.

1.3 **Overview:** The railway reservation system is an automated platform designed to streamline the ticket booking process for train passengers. It provides a convenient and accessible platform for customers to book train tickets, check seat availability, and manage their bookings. The system also enables railway authorities to manage seat allocation, track passenger information, and generate revenue reports.

2. **General Description:** The railway reservation system is a computerized platform that allows passengers to book train tickets, check seat availability, and manage their bookings. The system is accessible through online platforms and mobile applications, providing a convenient and user-friendly experience for customers. It enables railway authorities to manage seat allocation, track passenger information, and generate revenue reports.

3. **Functional Requirements:**

- User Registration and Authentication
- Train and Schedule Management
- Ticket Booking
- Seat Availability
- Booking Modifications and Cancellation

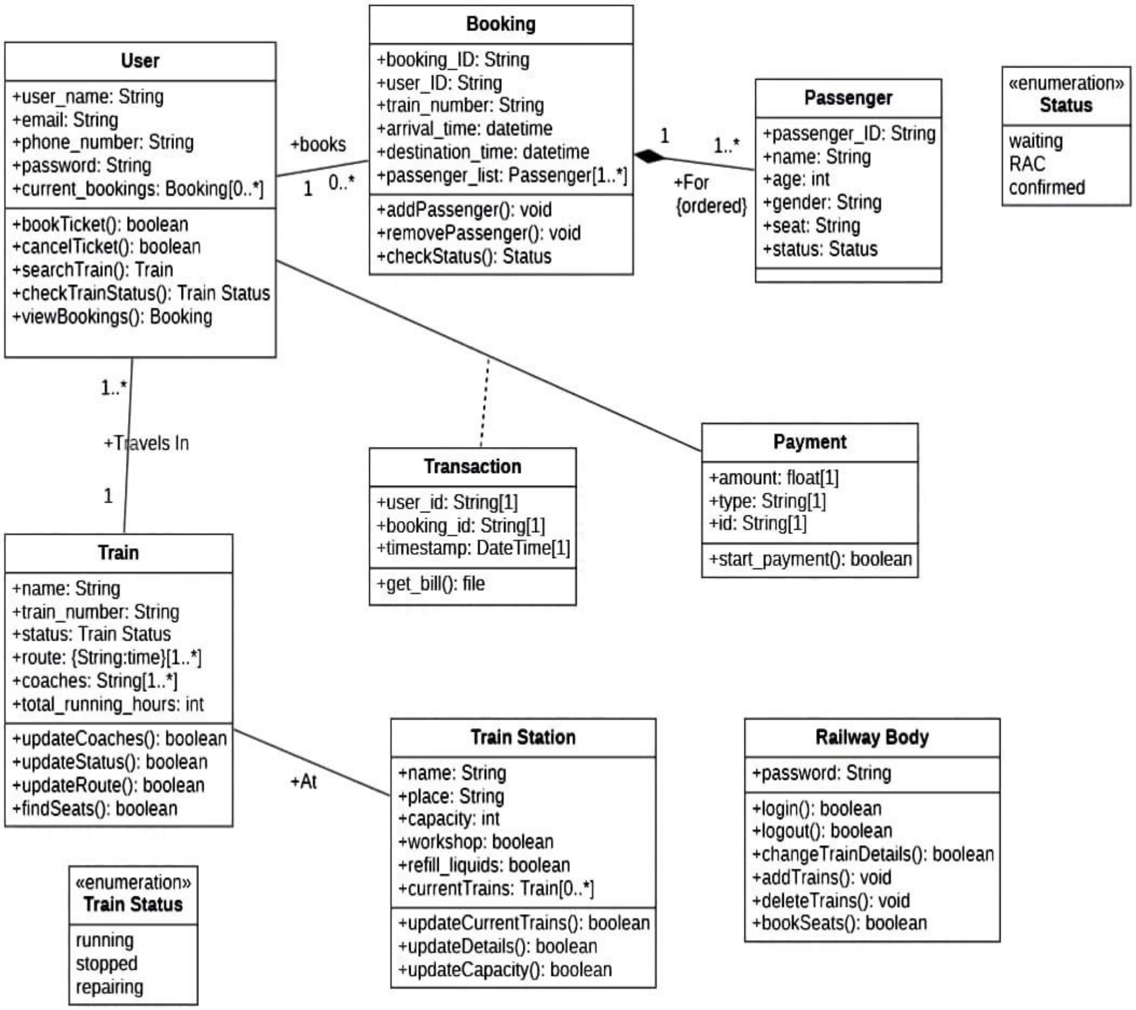
4. **Interface Requirements:**
   - User-friendly Interface
   - Responsive Design
   - Multilingual Support
   - Notifications
   - Personalization

5. **Performance Requirements:**
   - Scalability
   - Responsiveness
   - Reliability
   - Security
   - Availability

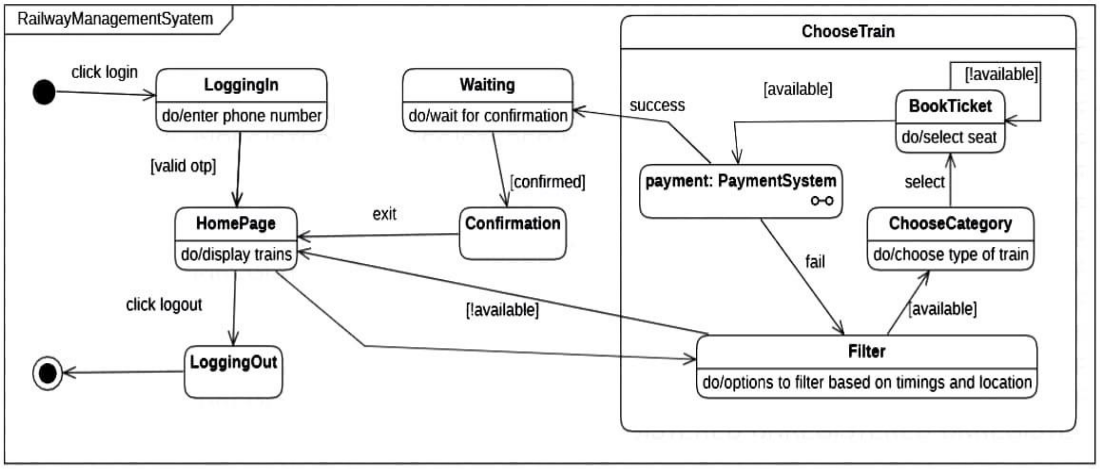
6. **Design constraints:**
   - Compatibility with Legacy Systems
   - Data Privacy Regulations
   - System Integration
   - Technology Constraints
   - User Accessibility

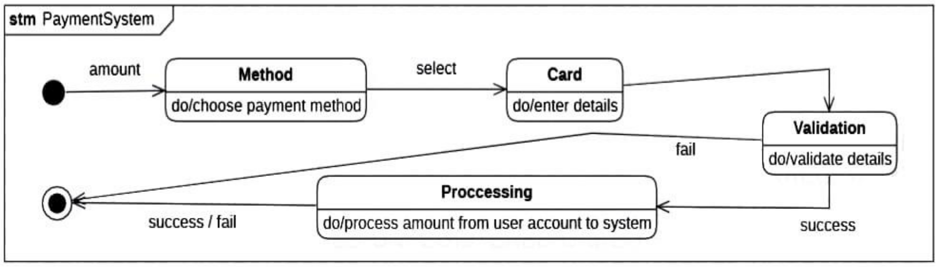
7. **Non Functional Requirements:**
   - Usability: The system should be easy to use and navigate, with clear and concise instructions for users.
   - Performance: The system should provide optimal performance with fast response times and high availability.
   - Reliability: The system should be reliable and available 24/7 with minimal downtime and data loss.
   - Security: The system should be secure, with robust authentication, access controls, and data encryption.
   - Maintainability: The system should be easy to maintain, with clear documentation, modular code, and frequent updates.

Class Diagrams

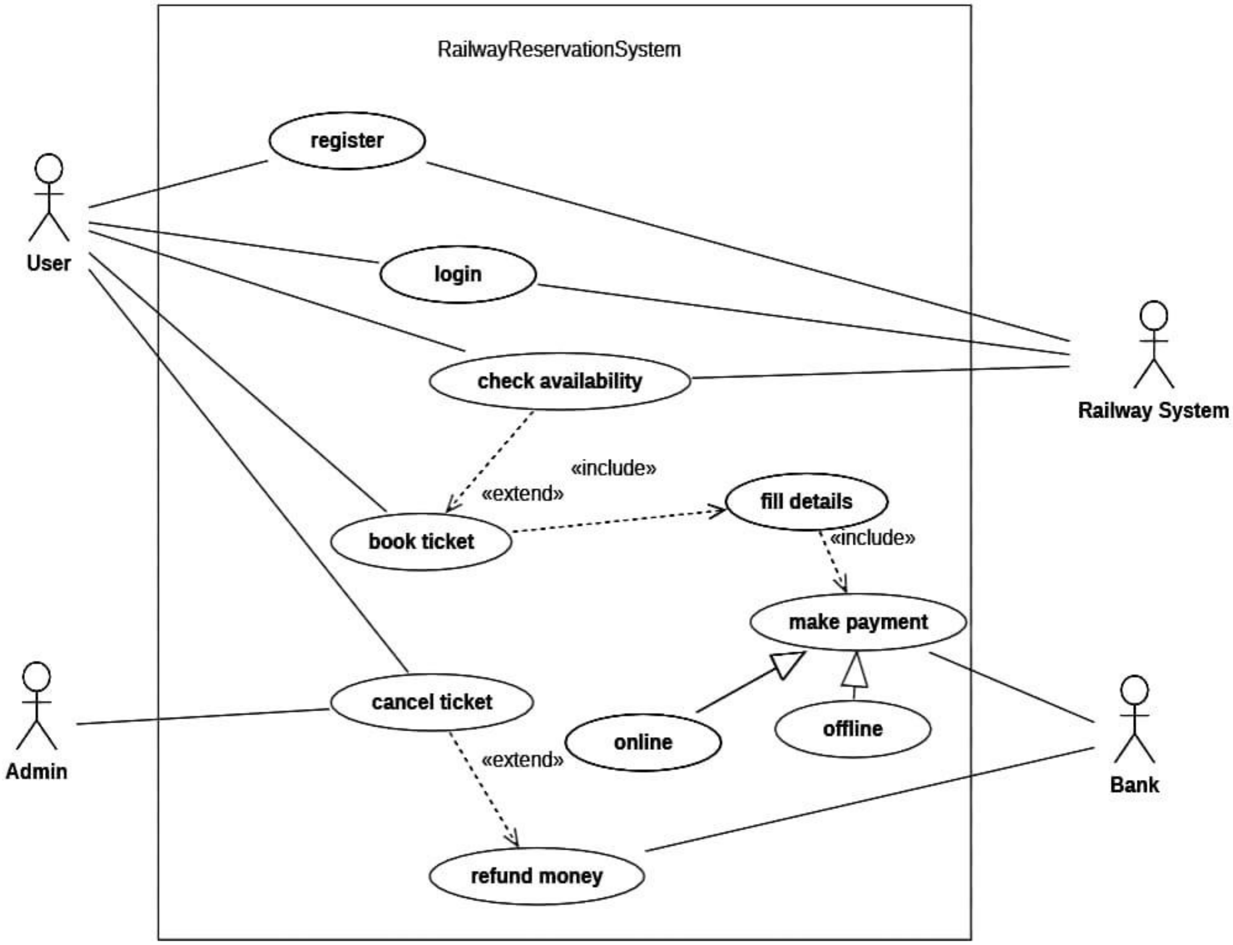


State Diagrams

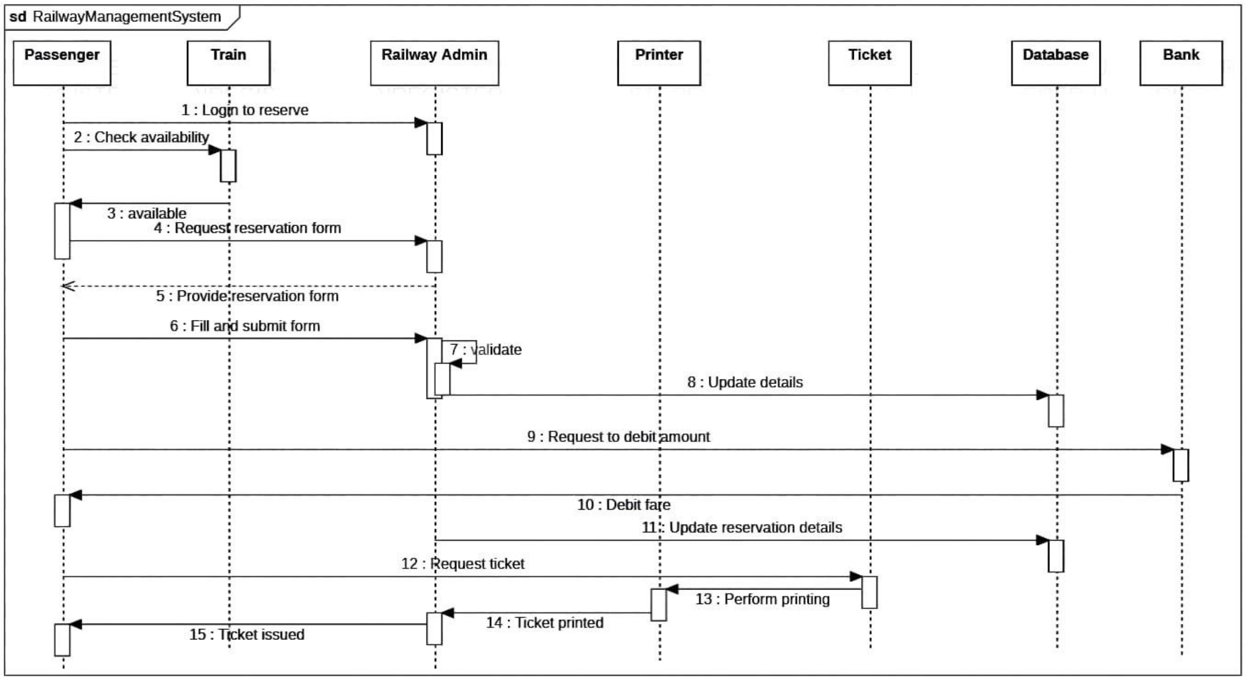




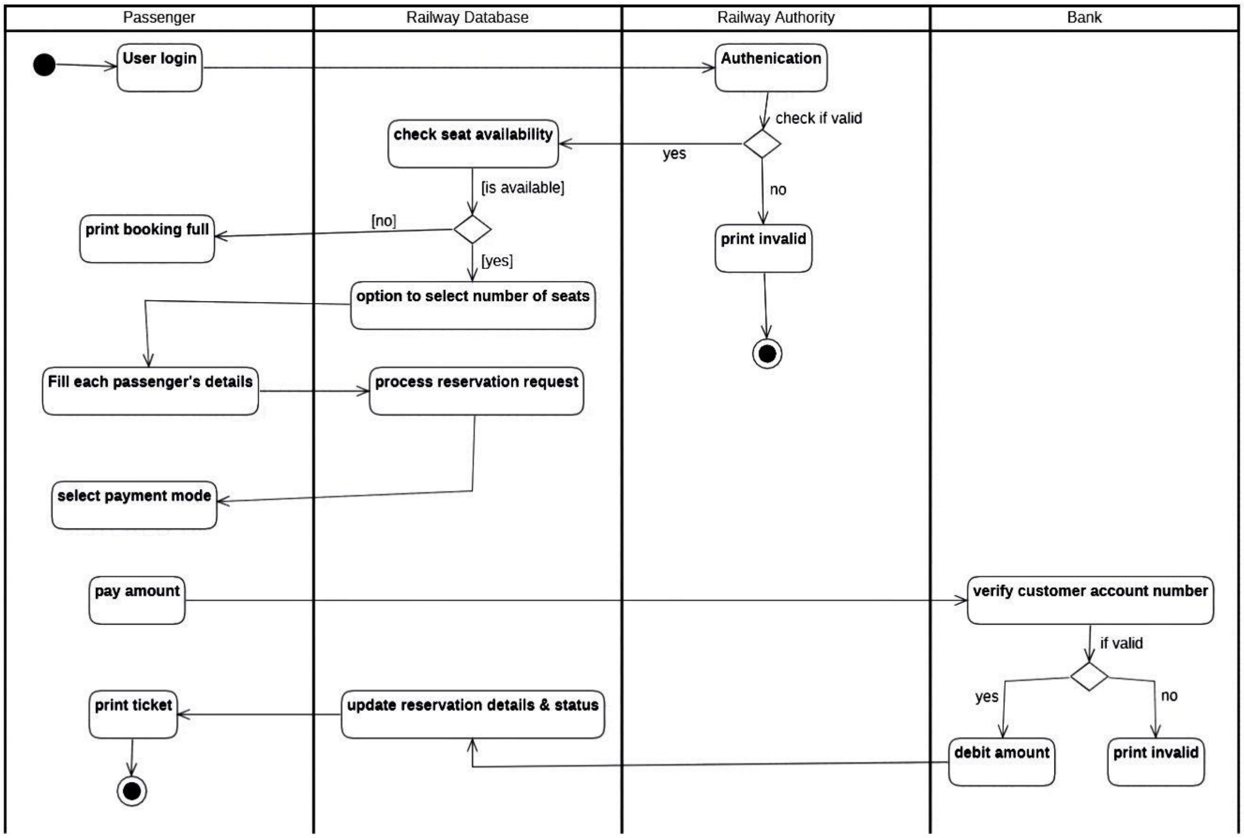
Use Case Diagrams



Sequence Diagrams



Activity Diagrams



7. Online Shopping System

Problem Statement: The online shopping system aims to provide a convenient and seamless shopping experience to customers, enabling them to browse, select, and purchase products online. However, there are several challenges associated with such a system, such as ensuring the security of sensitive customer information, managing large inventories, providing accurate product descriptions, dealing with returns and refunds, and ensuring timely delivery of products.

Software Requirement Specification (SRS):

1. Introduction:

1.1 **Purpose of this document:** The purpose of an online shopping system is to provide a convenient and accessible platform for customers to purchase products from anywhere at any time. It enables businesses to expand their reach beyond geographical boundaries, increase their customer base, and reduce operational costs associated with physical stores.

1.2 **Scope of this document:** The scope of an online shopping system is vast and dynamic. It includes the development and implementation of a robust e-commerce platform, enabling customers to browse and purchase products from a diverse range of categories. It also includes the integration of secure payment gateways, logistics management systems, and customer support mechanisms to ensure a seamless shopping experience.

1.3 **Overview:** An online shopping system is an e-commerce platform that enables customers to browse, select, and purchase products online from anywhere at any time. It provides a convenient and accessible platform for businesses to expand their reach and increase their customer base. The system typically includes a user-friendly interface, secure payment gateways, and efficient logistics management to ensure timely delivery of products.

2. **General Description:** An online shopping system is a web-based platform that enables customers to browse, select, and purchase products online. It typically involves a website or mobile application that offers a user-friendly interface, product catalogs, and secure payment gateways. The system also includes logistics management and customer support mechanisms to ensure the timely delivery of products and address any issues faced by customers.

3. **Functional Requirements:**

- User registration and account management
- Product catalogue and search
- Shopping cart and checkout
- Order management
- Inventory management

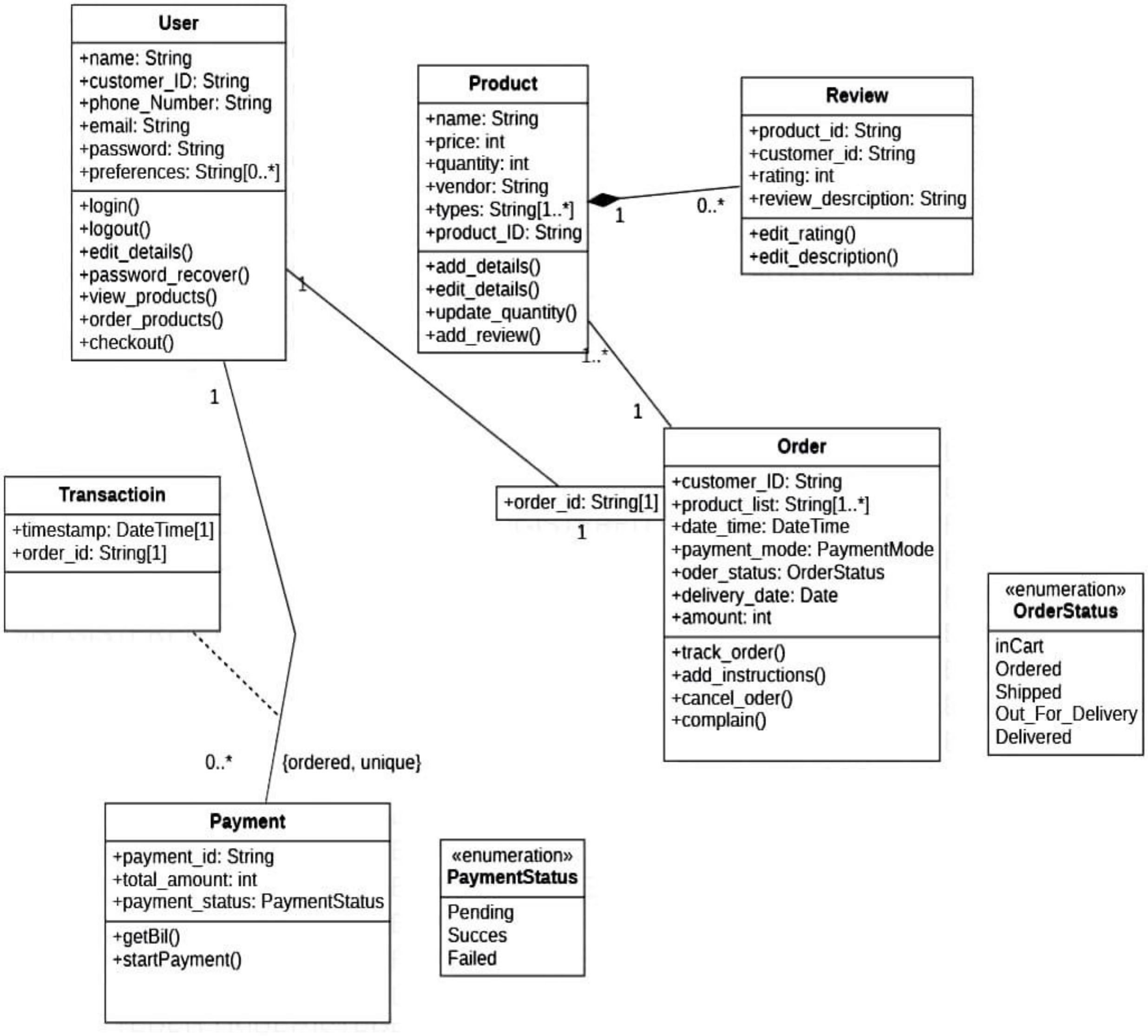
4. **Interface Requirements:**
    - User-friendly interface
    - Responsive design
    - Product display
    - Search and filter
    - Shopping cart

5. **Performance Requirements:**
    - Fast loading times
    - High availability and uptime
    - Scalability
    - Reliable payment processing
    - Efficient order fulfilment

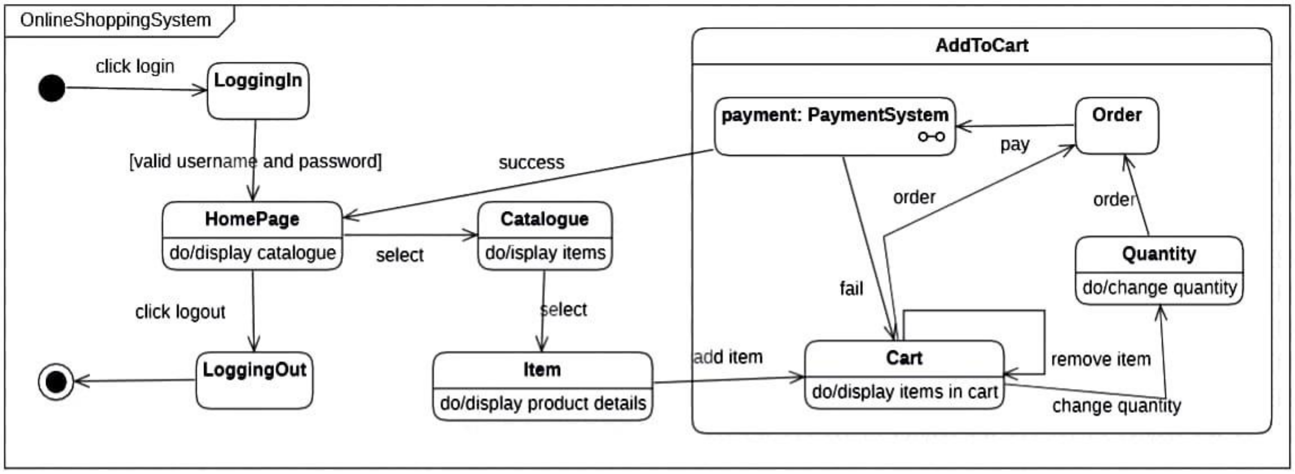
6. **Design constraints:**
    - Technology constraints
    - Compatibility constraints
    - Accessibility constraints
    - Usability constraints
    - Security constraints

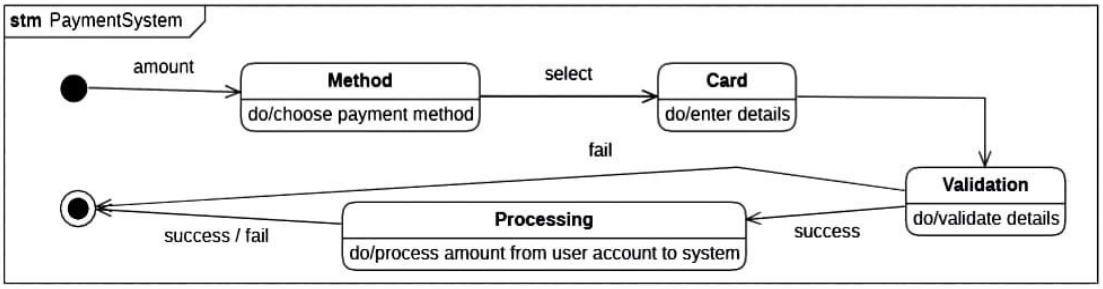
7. **Non Functional Requirements:**
    - Security: The system should be designed to be secure and protect user data and transactions from unauthorized access or theft.
    - Reliability: The system should be designed to be reliable and available to users at all times, with minimal downtime or maintenance.
    - Performance: The system should be designed to perform efficiently and effectively, with fast loading times and minimal lag or delay.
    - Scalability: The system should be designed to scale up or down as needed to accommodate changes in user demand or business growth.
    - Usability: The system should be designed to be user-friendly and easy to use, with minimal confusion and frustration.

Class Diagrams

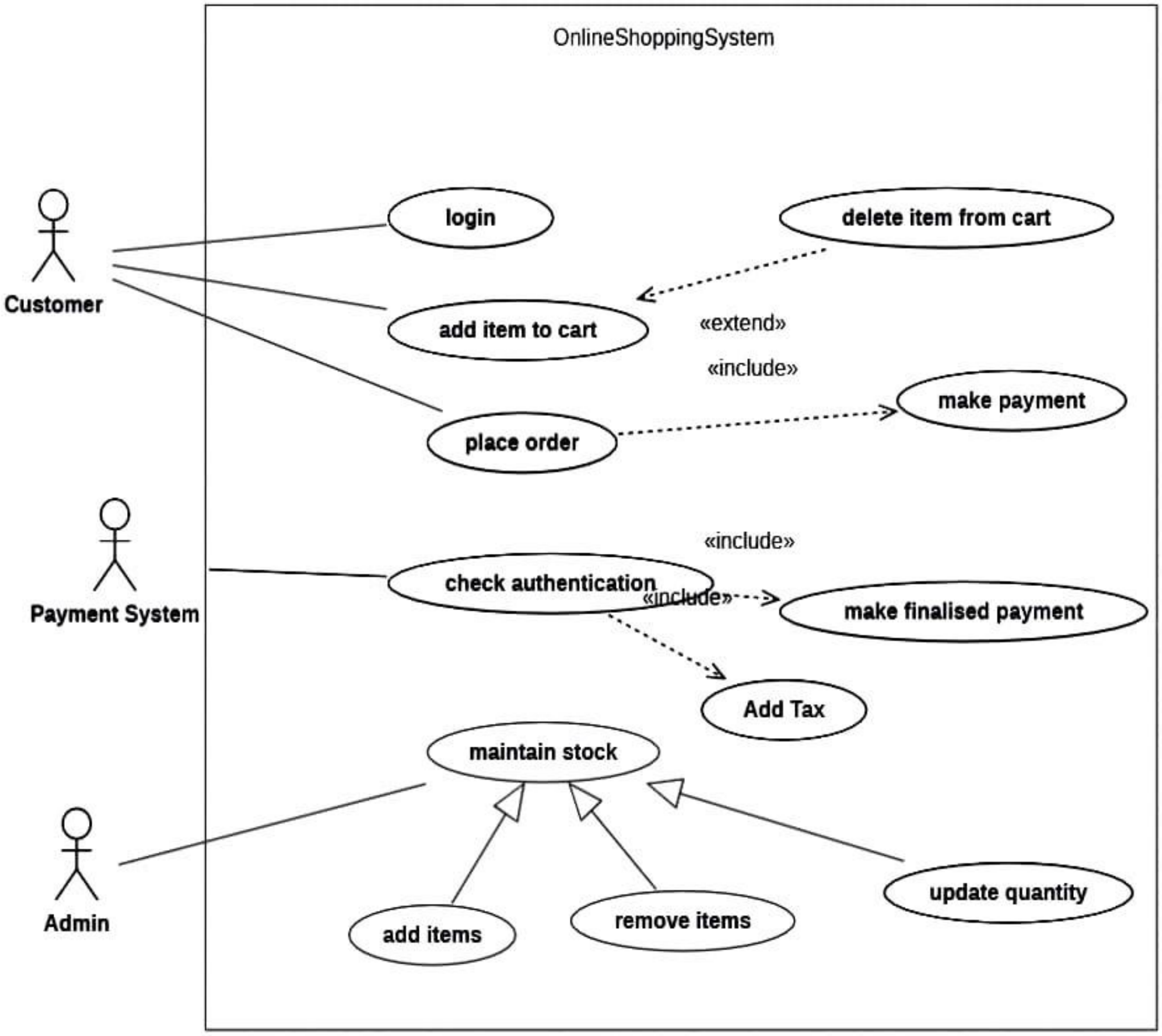


Sequence Diagrams

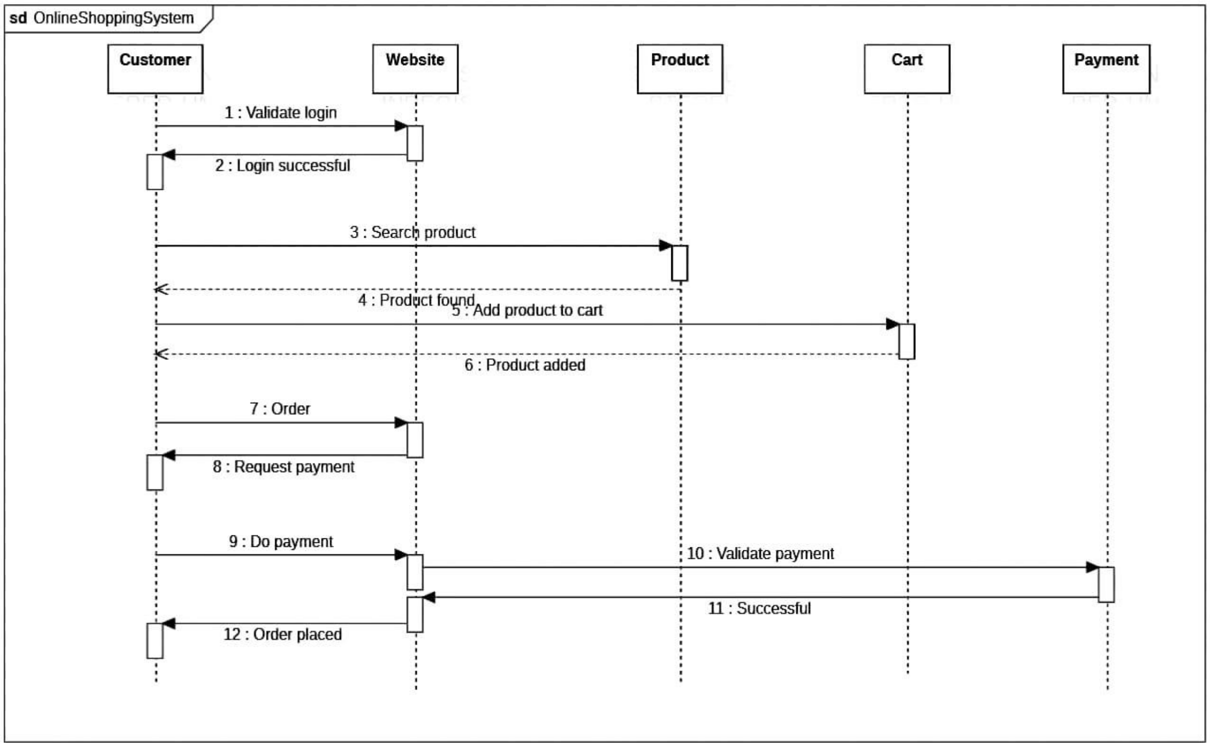




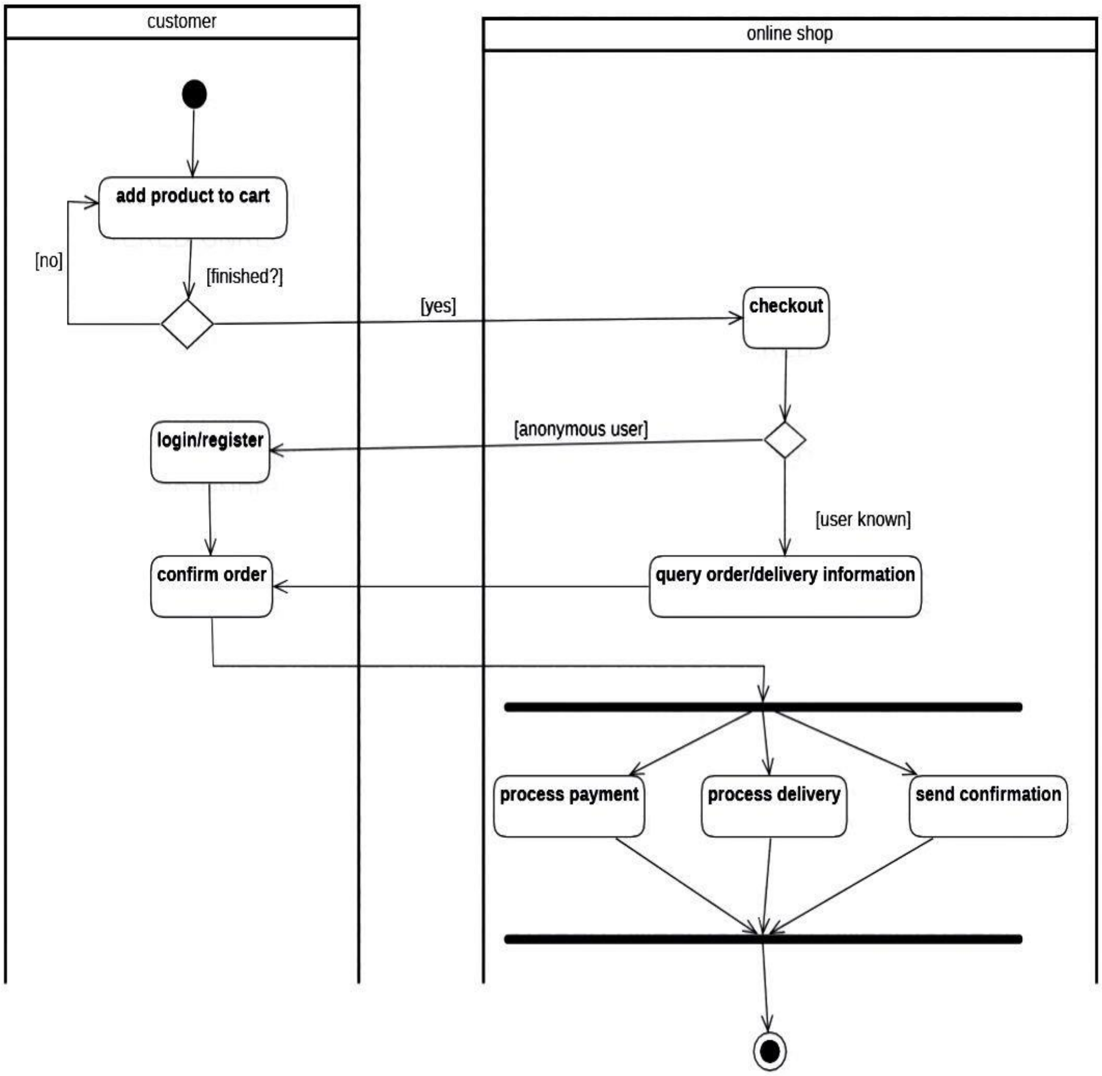
Use Case Diagrams



Sequence Diagrams



Activity Diagrams


customer
online shop
add product to cart
[no]
[finished?]
[yes]
checkout
login/register
[anonymous user]
[user known]
confirm order
query order/delivery information
process payment
process delivery
send confirmation