VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

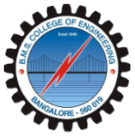
LAB REPORT
on

Object Oriented Modelling & Design Lab

Submitted by

CHIRAG MAHAVEER CHIVATE
(1BM20CS200)

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
Mar-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 "LAB COURSE **Object Oriented Modelling & Design**" carried out by **CHIRAG CHIVATE (1BM20CS200),** 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 Modelling & Design - (20CS6PCOMD)** work prescribed for the said degree.

Dr. Latha N.R,
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

INDEX

Sl. No.	Experiment Title	Page No.
01	Hotel Management System	1
02	Credit Card Processing System	8
03	Library Management System	15
04	Online Shopping System	22
05	Passport Automation System	29
06	Railway Reservation System	36
07	Stock Maintenance System	43

1. HOTEL MANAGEMENT SYSTEM

1.1 PROBLEM STATEMENT:

To develop a comprehensive hotel management software system that can automate and streamline these processes is essential. Such a system can help hotels enhance operational efficiency, reduce costs, boost revenue, and provide personalized recommendations and experiences to guests, enabling them to manage their inventory effectively, reduce the risk of overbooking, and improve staff productivity.

1.2 SYSTEM REQUIREMENTS & SPECIFICATIONS

1) **Introduction:**

1.1) Purpose of this Document: The purpose of this document is to define the requirements for a Hotel Management software system. It outlines the system's goals and features, including its functional and non-functional requirements, design constraints, and preliminary schedule and budget.

1.2) Scope of this Document: This document describes the requirements for a Hotel Management software system that will enable hotels to manage their reservations, guest information, inventory, billing and payments, reporting and analytics, integration with other systems, staff management, and loyalty program management. The system should be user-friendly, efficient, and reliable.

1.3) Overview: The Hotel Management software system is designed to provide hotels with a comprehensive solution for managing their day-to-day operations. The system will be developed with modern technology and will be easily integrated with other systems used by the hotel.

2) General Description: The Hotel Management software system will provide hotels with a user-friendly interface to manage their reservations, guest information, inventory, billing and payments, reporting and analytics, staff management, and loyalty program management. The system will also provide a centralized database for storing and retrieving information.

3) Functional Requirements: The functional requirements for the Hotel Management software system include:

i. Room reservation management
ii. Guest information management
iii. Inventory management
iv. Billing and payment management
v. Reporting and analytics
vi. Integration with third-party systems
vii. Staff management
viii. Loyalty program management

4) Interface Requirements: The Hotel Management software system should be able to communicate with other systems using standard interfaces such as APIs or web services. The system should also provide a user-friendly interface for hotel staff and customers.

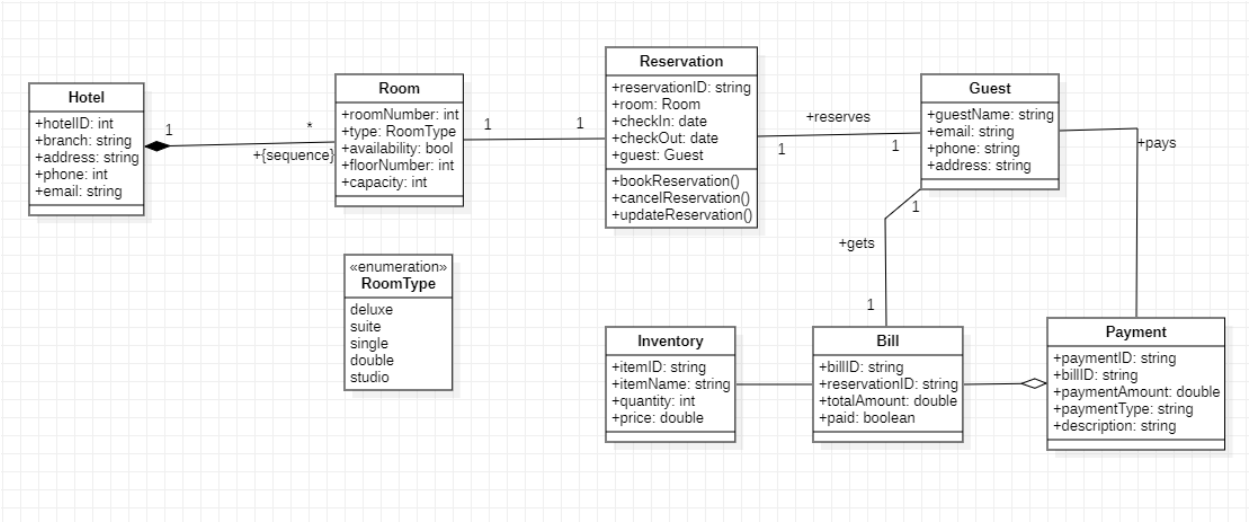
5) Performance Requirements: The Hotel Management software system should be able to handle a large number of simultaneous users and provide fast response times. The system should also be scalable to accommodate future growth.

6) Design Constraints: The Hotel Management software system should be designed using modern technology and should be easily scalable and maintainable. The system should also be designed to ensure data security and privacy.

7) Non-Functional Attributes: The Hotel Management software system should be reliable, secure, and portable. The system should also be easy to use, accessible, and provide good performance.

8) Preliminary Schedule and Budget: The development of the Hotel Management software system will take approximately 12 months and will require a budget of $500,000. The schedule and budget will be reviewed periodically to ensure that the project stays on track.

1.3 CLASS DIAGRAM



i. Hotel: Represents a hotel in the system with its basic information and methods to manage rooms.

ii. Room: Represents a room in the hotel with its details and methods to access and modify room attributes.

iii. Reservation: Represents a guest's reservation for a specific room with reservation details and methods to retrieve information.

iv. Guest: Represents a guest staying at the hotel with their personal information and methods to access guest details.

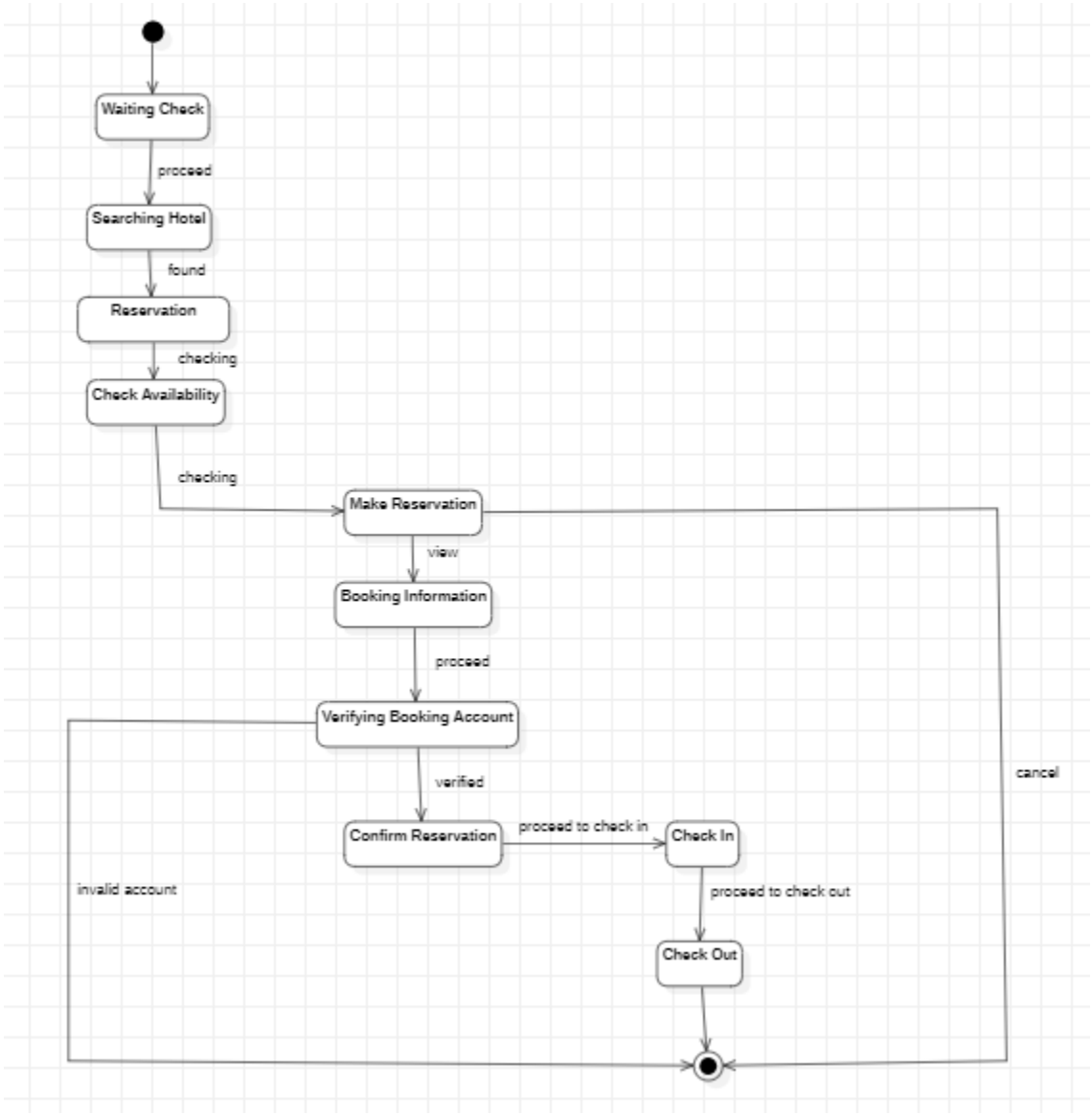
v. Inventory: Represents the hotel's inventory of items with methods to manage the inventory.

vi. Bill: Represents a bill generated for a guest's stay with bill information and methods to retrieve details.

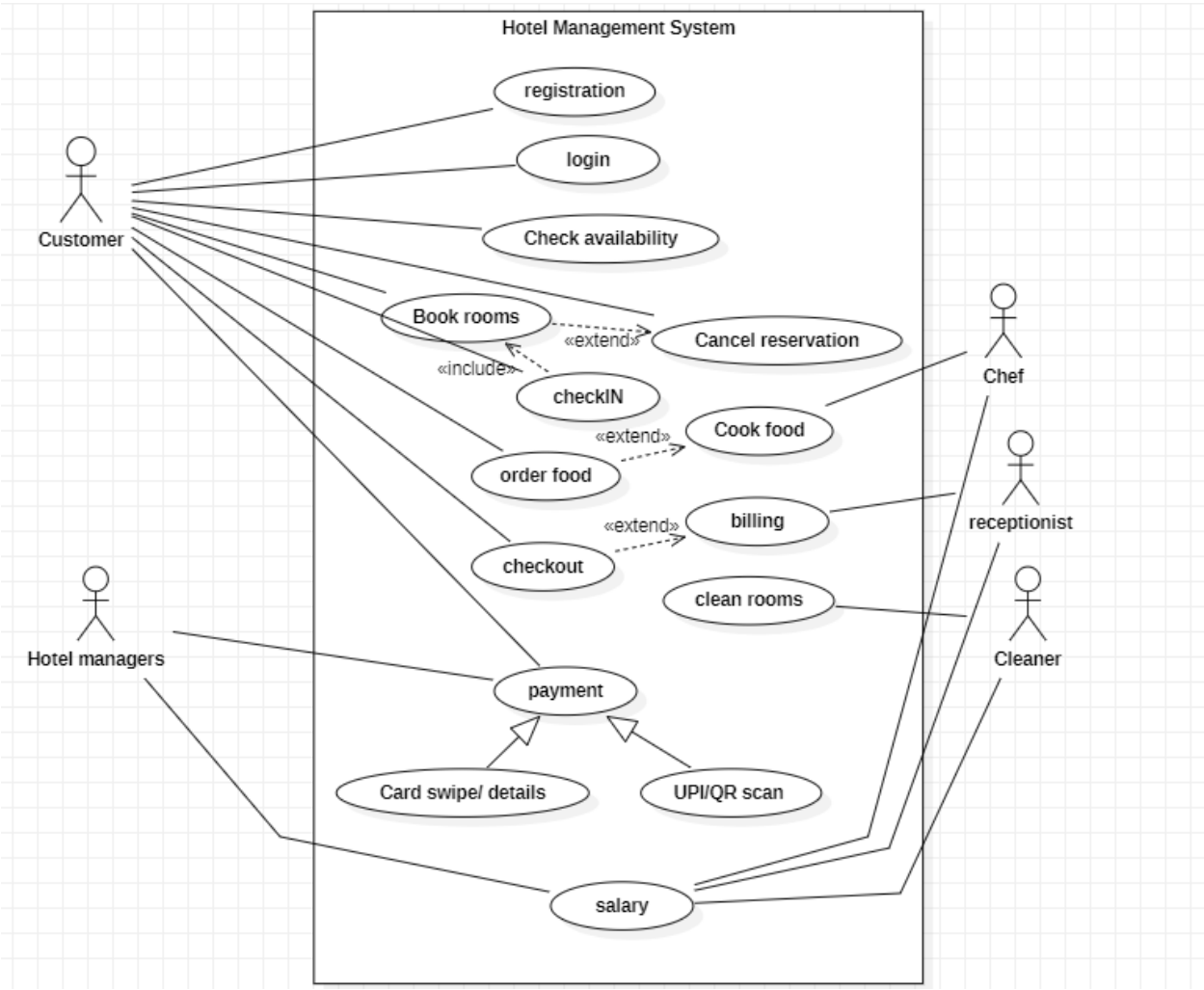
vii. Payment: Represents a payment made by a guest to settle their bill with payment details and methods to access payment information.

viii. RoomType (Enumeration): Represents the types of rooms available in the hotel, providing predefined values for easy classification.

1.4 STATE DIAGRAM



1.5 USE CASE DIAGRAM



Actors:

i. Customer: Represents a person who interacts with the hotel management system as a customer, making reservations, checking in and out, ordering food, and making payments.

ii. Hotel Manager: Represents a role responsible for managing the overall operations of the hotel, overseeing staff, monitoring performance, and making strategic decisions.

iii. Chef: Represents a role responsible for preparing and cooking food in the hotel's kitchen.

iv. Receptionist: Represents a role responsible for handling guest registrations, check-ins, check-outs, and providing assistance to customers.

v. Cleaner: Represents a role responsible for maintaining cleanliness and tidiness of the hotel rooms and common areas.

Use Cases:

i. Registration: Allows a customer to create a new account in the hotel management system.

ii. Login: Allows a user (customer, hotel manager, chef, receptionist, or cleaner) to authenticate and access their account.

iii. Check Availability: Enables a customer to check the availability of rooms for a specified date range.

iv. Book Rooms: Allows a customer to reserve a room for a specific date range. "Book Rooms" extends "Cancel Reservation" as a booked room can be canceled by the customer.

v. Cancel Reservation: Allows a customer to cancel a previously booked room reservation.

vi. Check-in: Facilitates the check-in process for a customer, assigning them a room upon arrival. "Check-in" extends "Book Rooms" since a check-in action can only occur after a room has been booked.

vii. Cook Food: Enables the chef to prepare and cook food based on customer orders.

viii. Order Food: Allows a customer to request food items from the hotel's menu. "Order Food" extends "Cook Food" as food needs to be prepared by the chef before it can be ordered.

ix. Check-out: Handles the check-out process for a customer, finalizing their stay and preparing the bill. "Check-out" extends "Billing" as the check-out process involves generating the final bill for the customer.

x. Billing: Generates the bill for a customer's stay, including room charges and any additional services. (Extends relationship: Check-out)

xi. Clean Rooms: Allows a cleaner to clean and maintain the hotel rooms.

xii. Payment: Represents the process of making a payment for services rendered. (Generalizations: Card Swipe Details, UPI/QR Scan)

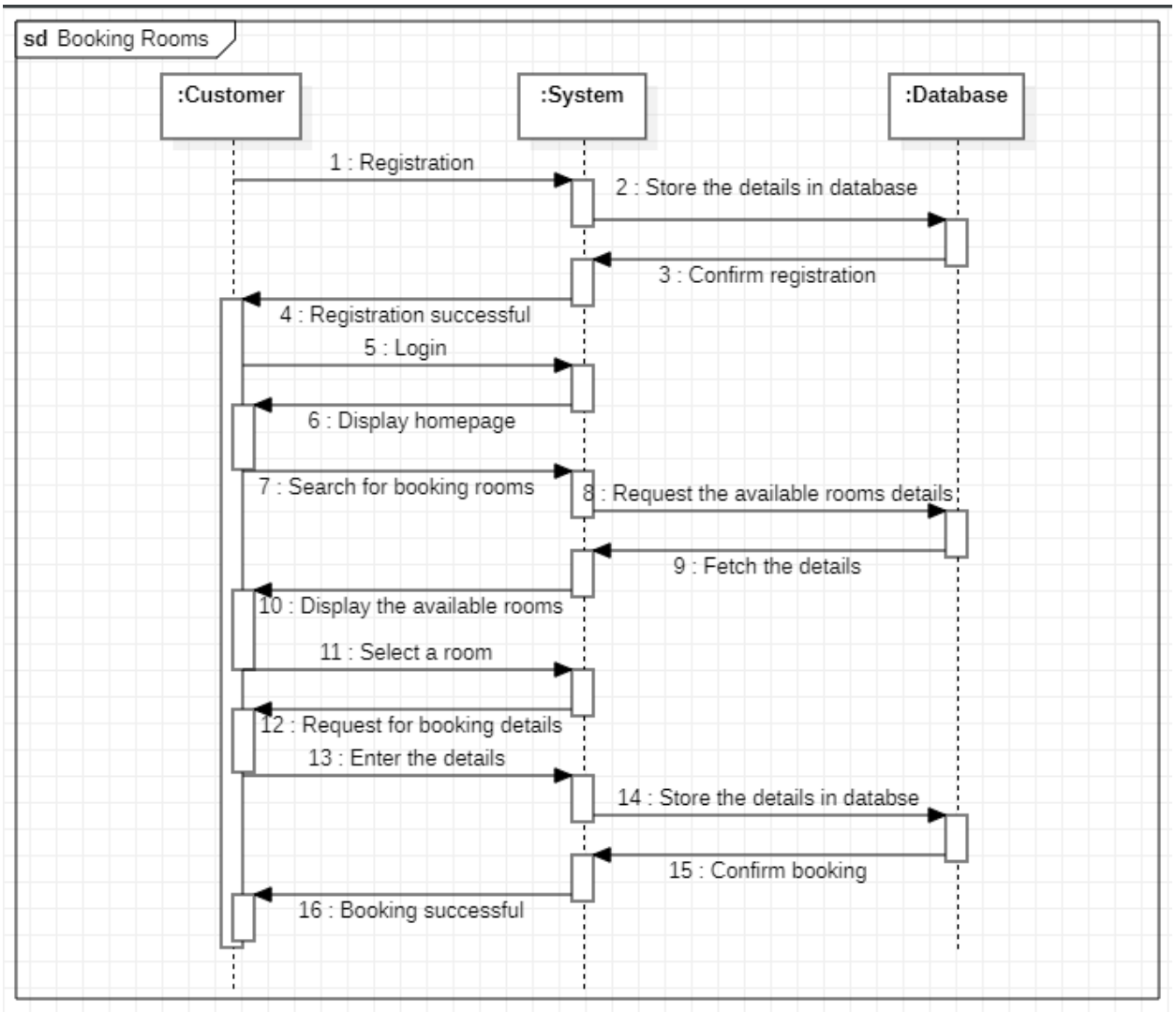
xiii. Card Swipe Details: Represents a specific method of payment involving card swiping.

xiv. UPI/QR Scan: Represents a specific method of payment using UPI or QR code scanning.

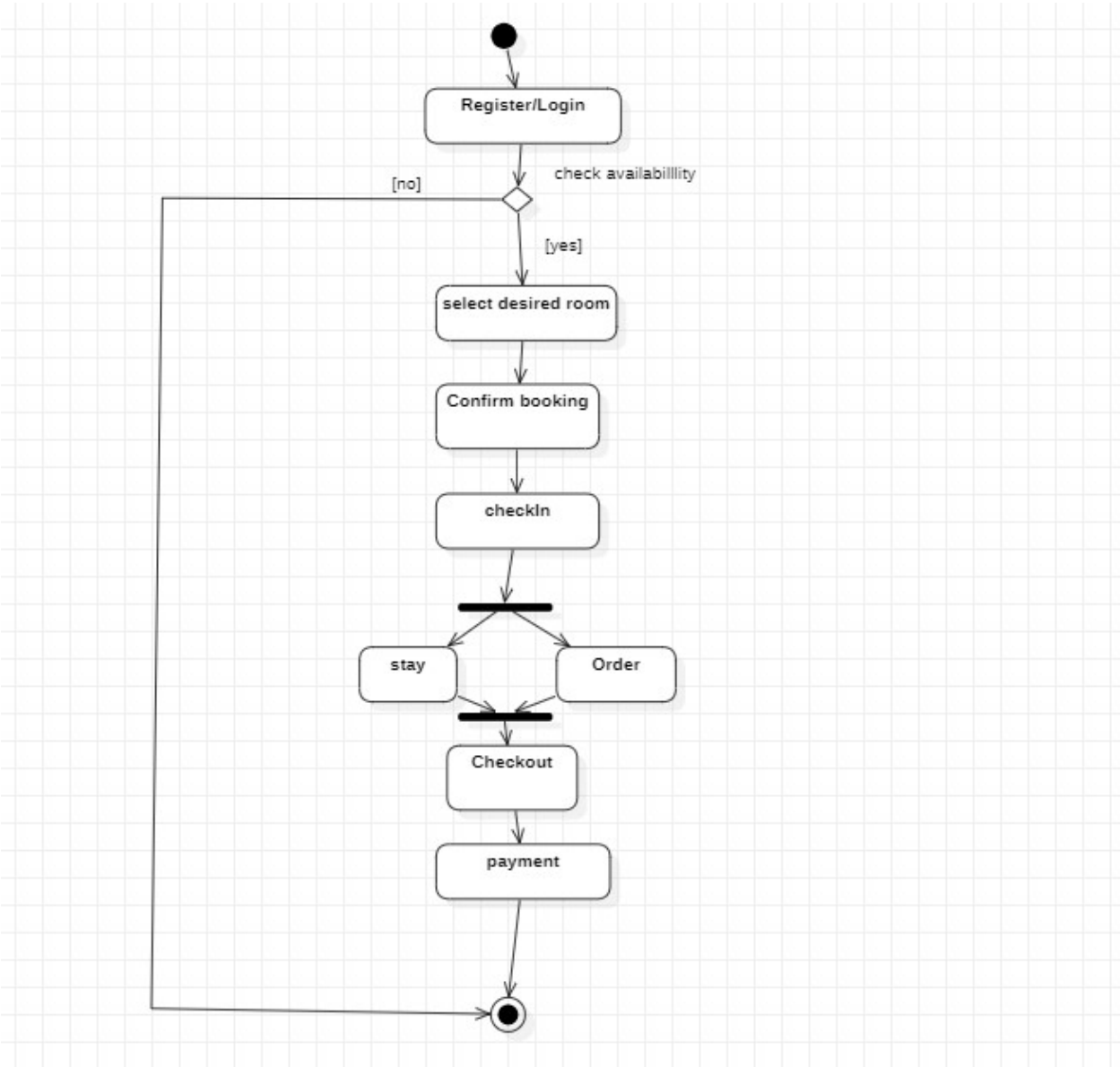
iv. Salary: Represents the process of managing and distributing salary to hotel staff, including chefs, receptionists, and cleaners.

"Card Swipe Details" and "UPI/QR Scan" are generalizations of the "Payment" use case, representing different payment methods available to the customer.

1.6 SEQUENCE DIAGRAM



1.7 ACTIVITY DIAGRAM



2. CREDIT CARD PROCESSING SYSTEM

2.1 PROBLEM STATEMENT:

To create a credit card software system that is needed to provide an efficient and secure platform for managing transactions and customer information. This system should be user-friendly, easily accessible, and capable of processing transactions in real-time. The goal is to create a seamless experience for customers while also ensuring the security and integrity of their personal and financial data. This software system should be scalable, customizable, and adaptable to changing industry standards and regulations.

2.2 SYSTEM REQUIREMENTS & SPECIFICATIONS

1) Introduction:

1.1) Purpose of this Document: This document outlines the system requirements for the development of a Credit Card software system.

1.2) Scope of this document: The document describes the overall objectives of the project and provides a detailed overview of the system requirements. It also outlines the estimated development costs and time required.

1.3) Overview: The Credit Card software system will be a secure, user-friendly, and reliable platform for managing credit card transactions and customer data.

2) General description: The Credit Card software system will be designed to meet the needs of financial institutions, merchants, and customers. Its features will include credit card processing, fraud detection, transaction history tracking, and reporting. The system will also provide customers with access to their account information and the ability to make payments online.

3) Functional Requirements: The Credit Card software system must be able to process credit card transactions securely and accurately. It should also be able to detect and prevent fraudulent transactions. Other functional requirements include the ability to generate transaction reports, manage customer data, and provide customers with a user-friendly interface for managing their accounts.

4) Interface Requirements: The Credit Card software system should be able to integrate with other software applications, such as financial management software and e-commerce platforms. It should also provide a user-friendly interface for customers to manage their accounts and make payments.

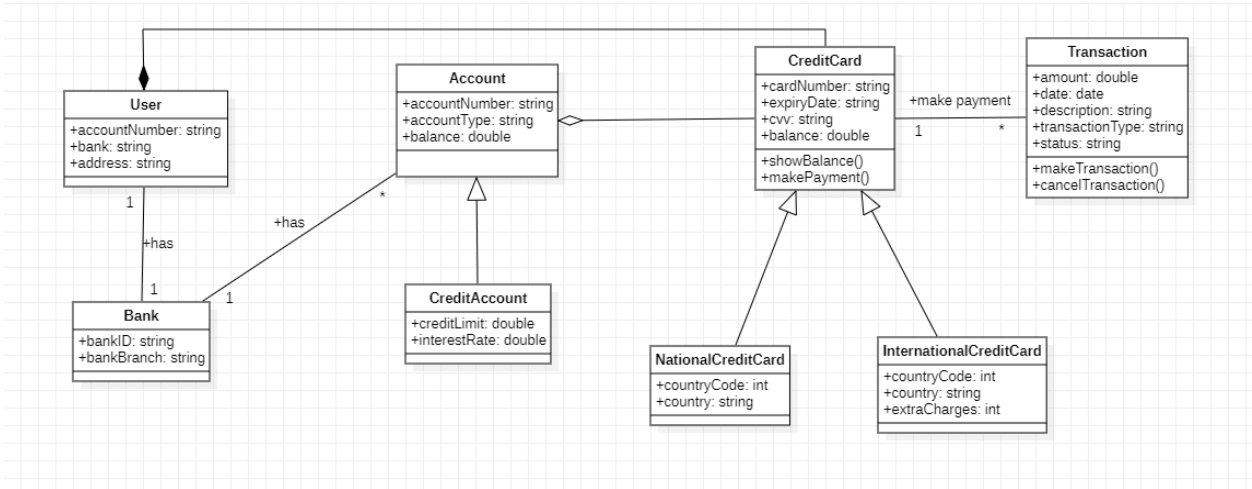
5) Performance Requirements: The Credit Card software system should be able to handle a large number of transactions simultaneously without any delays or errors. It should also be able to process transactions securely and efficiently, with minimal downtime.

6) Design Constraints: The Credit Card software system must comply with industry standards for security and data privacy. It should also be designed with scalability and flexibility in mind, to allow for future updates and enhancements.

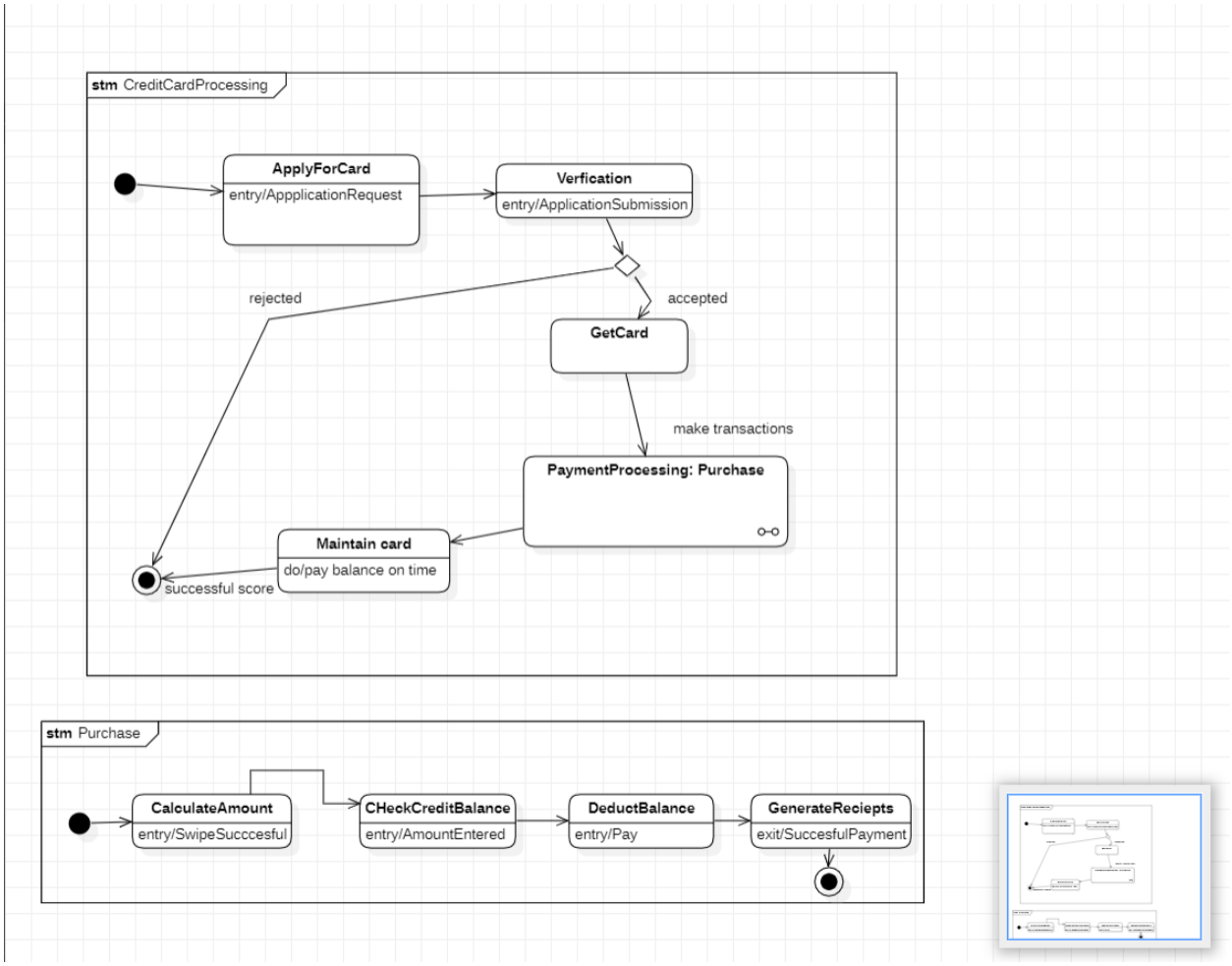
7) Non-Functional Attributes: The Credit Card software system should be secure, reliable, and easy to use. It should also be portable, with the ability to run on multiple platforms and devices. Other non-functional requirements include data integrity, scalability, and performance optimization.

8) Preliminary Schedule and Budget: The development of the Credit Card software system is estimated to take approximately 12 months and will require a budget of $500,000. The development team will consist of six software engineers, two quality assurance analysts, and one project manager.

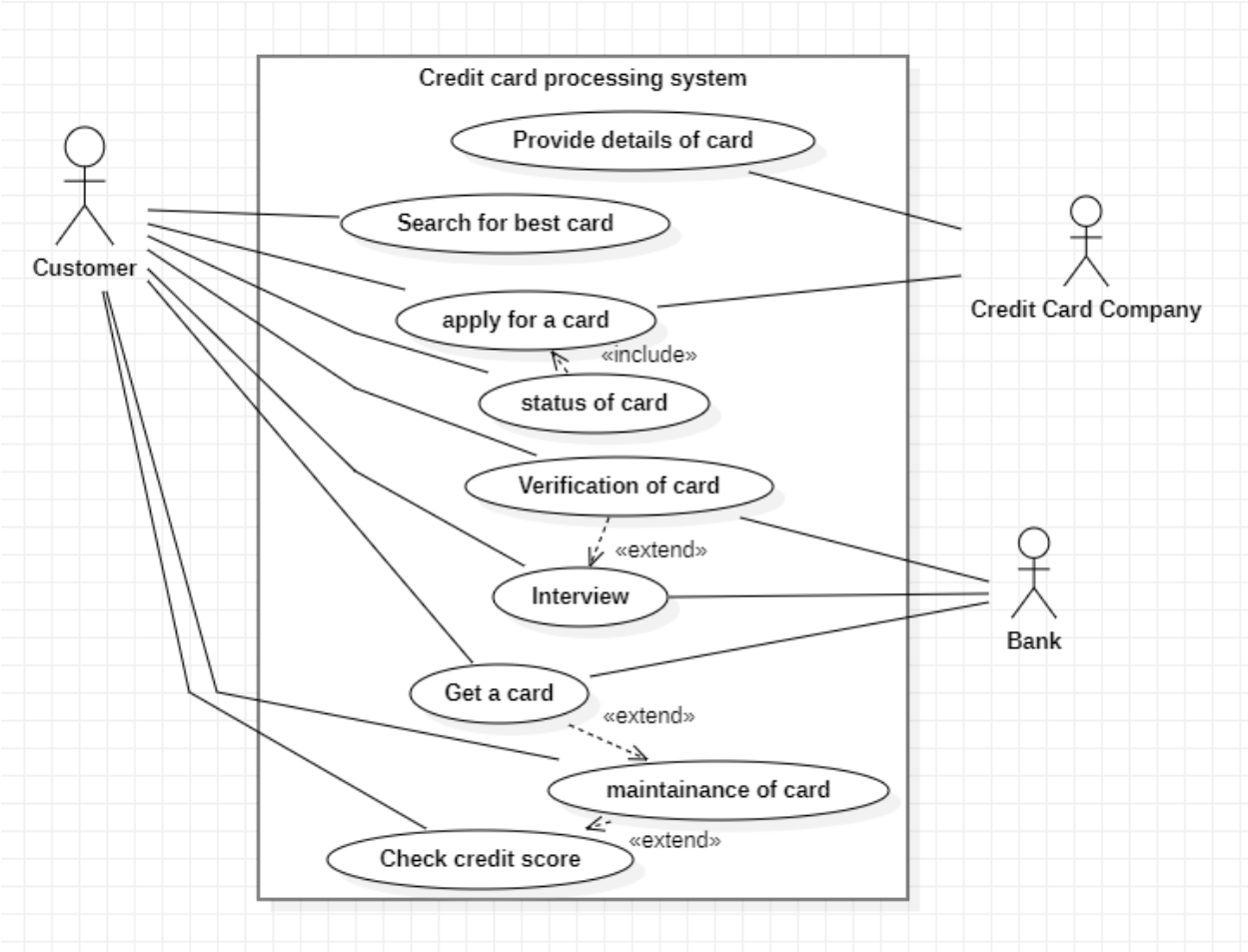
2.3 CLASS DIAGRAM



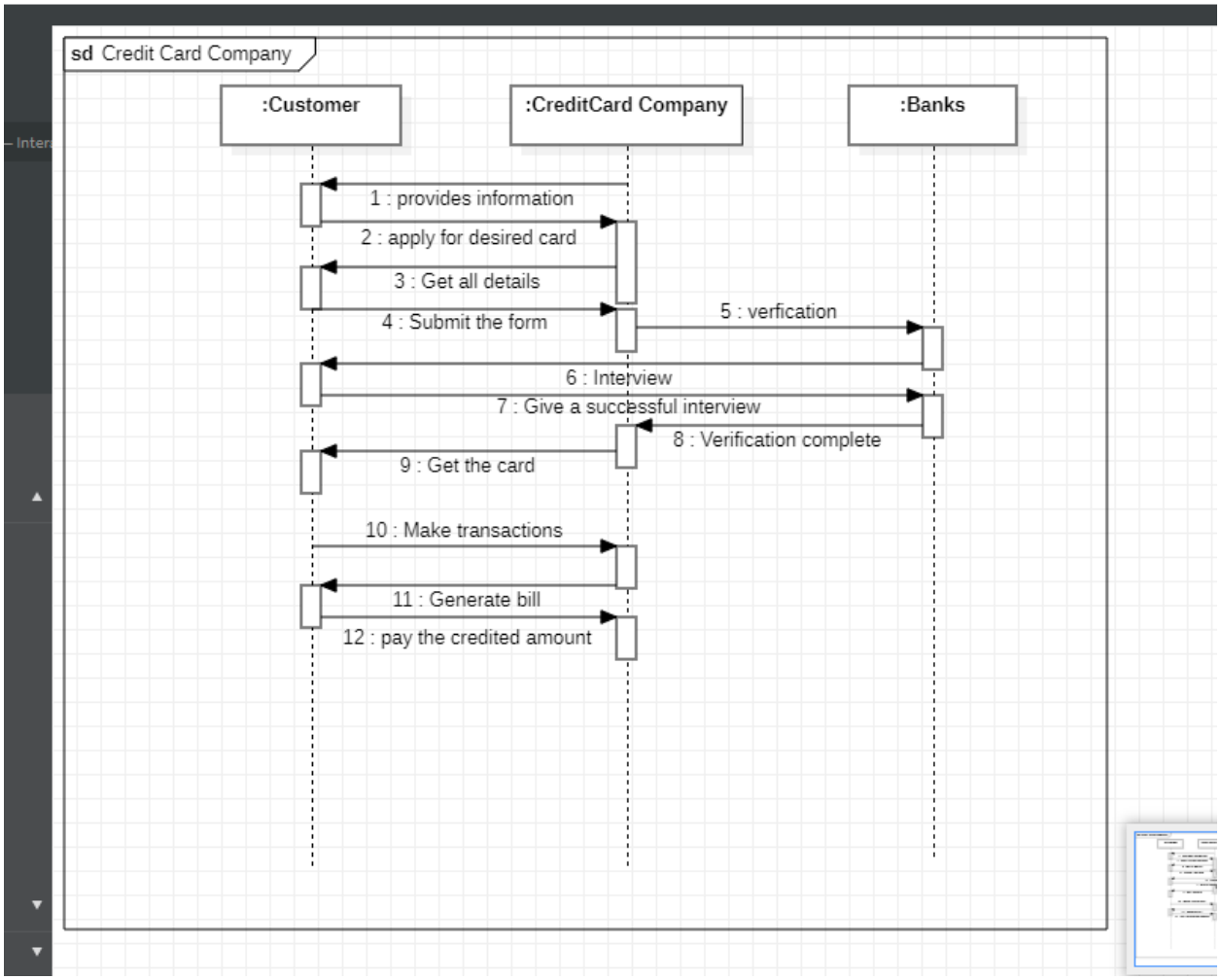
2.4 STATE DIAGRAM



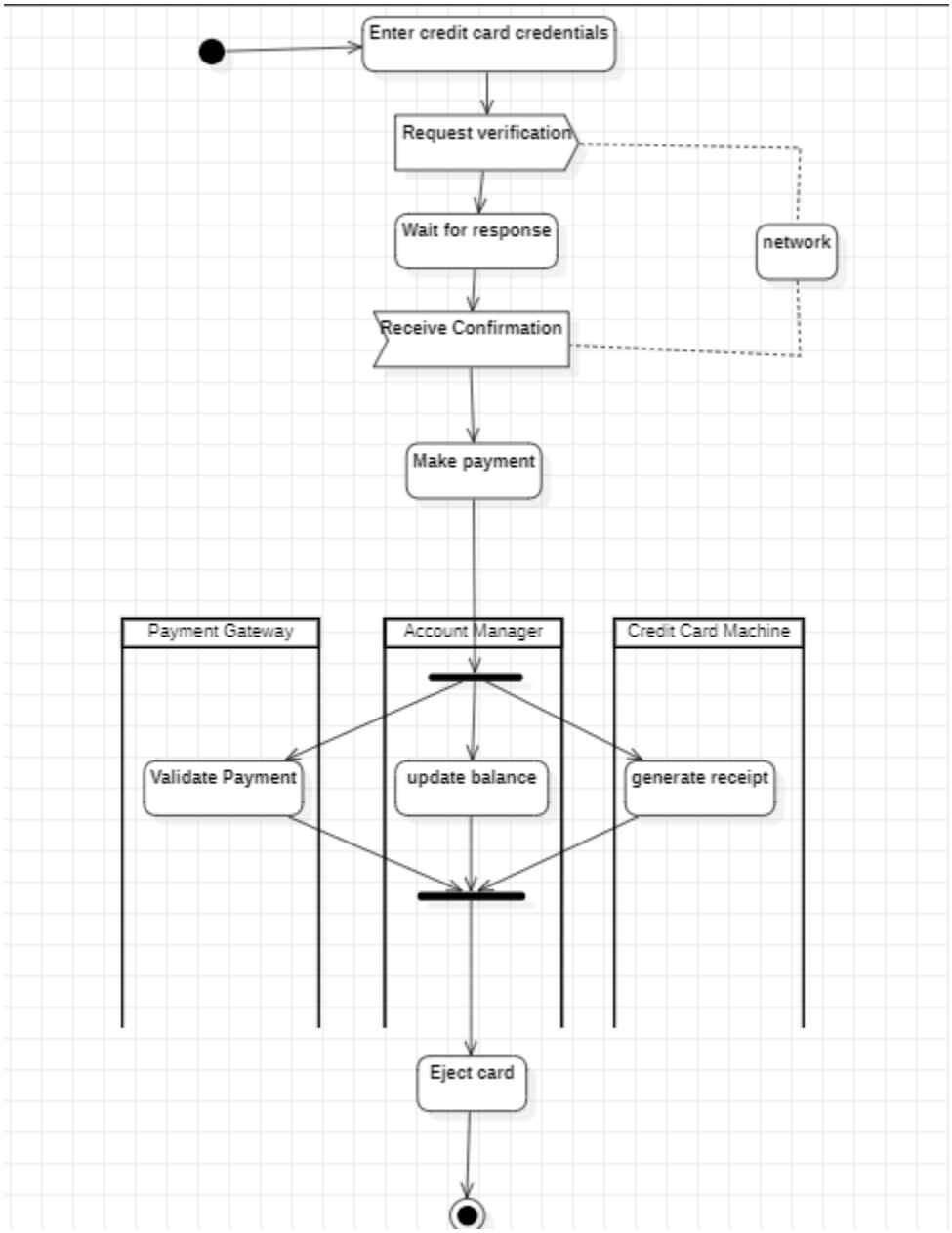
2.5 USE CASE DIAGRAM



2.6 SEQUENCE DIAGRAM



2.7 ACTIVITY DIAGRAM



3. LIBRARY MANAGEMENT SYSTEM

3.1 PROBLEM STATEMENT: The library management system is currently operating manually, leading to issues such as slow retrieval of books, difficulty in tracking book availability and library activity, and delayed book returns. To address these issues, a software system is required that can automate the library's operations and provide a more efficient and streamlined experience for both librarians and patrons.

3.2 SYSTEM REQUIREMENTS & SPECIFICATIONS

1) Introduction:

1.1) Purpose of this Document: The purpose of this document is to provide a comprehensive outline of the requirements for developing a library management software system. The document will describe the system's scope, functions, and non-functional attributes, as well as any design constraints, performance requirements, and interface requirements.

1.2) Scope of this Document: This document will provide a clear understanding of the software system's objectives and features, including how it will benefit users and the library community as a whole. It will also describe the development cost and time required for the project.

1.3) Overview: The library management software system is a digital platform designed to streamline and improve library operations. It will allow librarians to manage the library's resources, including books, journals, and multimedia materials, and help library patrons to find and access library resources easily.

2) General Description: The software system aims to provide librarians with a user-friendly, efficient platform that can help them manage library resources efficiently, including tracking library materials, processing loan requests, and managing the library's digital resources. The system will also provide patrons with an easy-to-use platform to find library resources, reserve materials, and renew items. The system is designed for library staff, librarians, and patrons who have basic computer knowledge. The library management software system will offer several features such as managing library resources, tracking library materials, processing loan requests, managing the library's

digital resources, reserving materials, renewing items, managing patron accounts, and generating reports.

3) Functional Requirements: The functional requirements of the library management software system are:

i. manage library resources, including books, journals, and multimedia materials
ii. track library materials and their location within the library
iii. process loan requests from patrons
iv. manage the library's digital resources, including eBooks and digital journals
v. allow patrons to reserve materials and renew items
vi. manage patron accounts
vii. generate reports

4) Interface Requirements: The library management software system must have the following interfaces:

i.)A user interface for library staff and librarians to manage library resources, including tracking and processing loan requests
ii.) A user interface for patrons to search and access library resources and manage their accounts
iii.) A database to store library resources, patron information, and loan requests

5.) Performance Requirements: The library management software system must perform the following functions:

i. The system must be fast and responsive when searching for library resources and processing loan requests
ii. The system must be able to handle multiple users concurrently
iii. The system must be able to maintain data integrity and security

6) Design Constraints: The library management software system must be developed using the following design constraints:

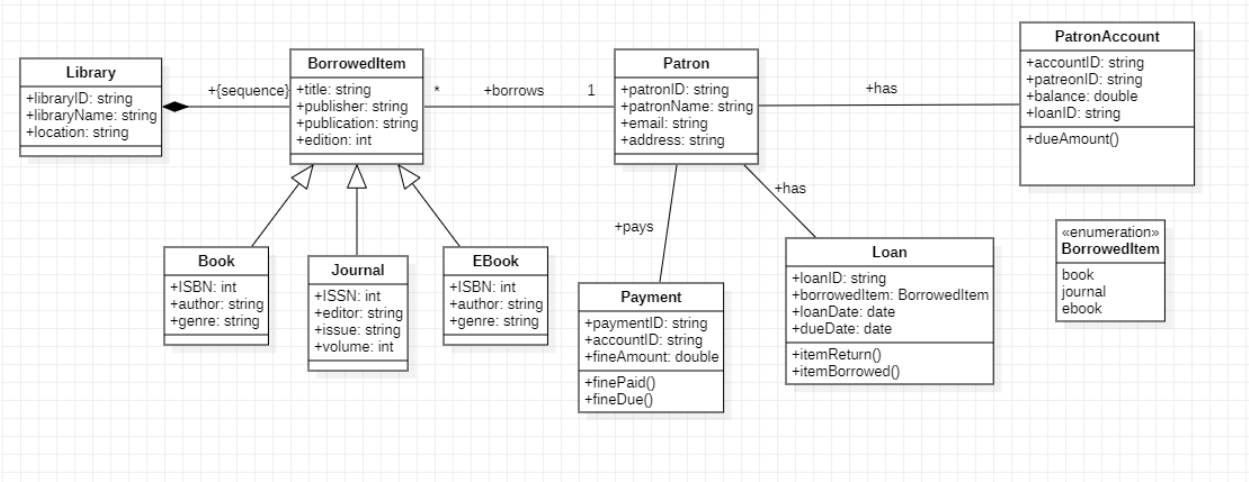
i. The system must use a web-based architecture
ii. The system must be developed using any programming language
iii. The system must be designed to run on Windows and macOS platforms

7) Non-Functional Attributes: The library management software system must meet the following non-functional attributes:

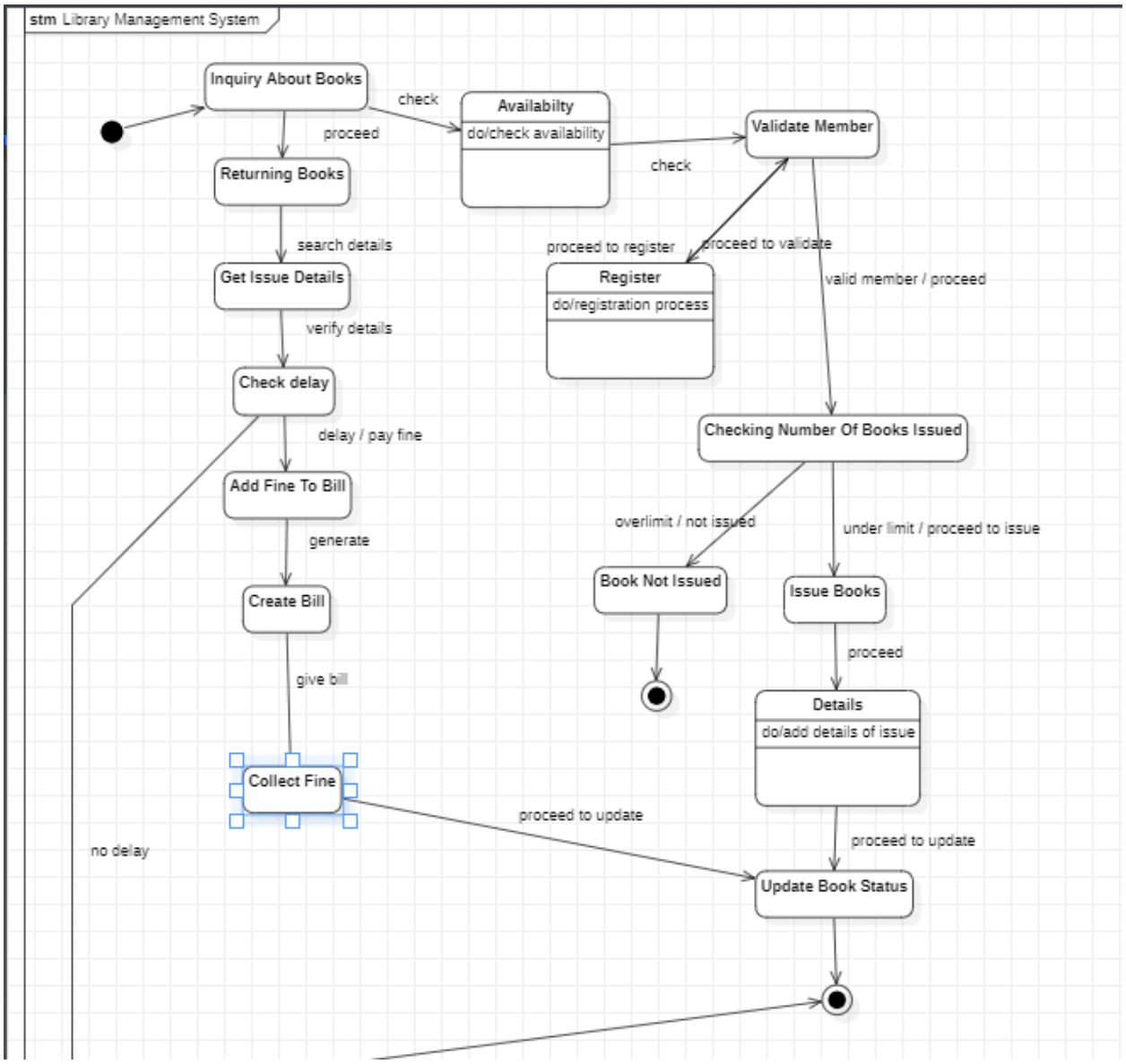
i. Security: The system must be secure, protecting patron data and library resources
ii. Reliability: The system must be reliable and function without errors
iii. Scalability: The system must be scalable to meet the needs of growing libraries

8) Preliminary Schedule and Budget: The preliminary schedule and budget for the library management software system will depend on various factors, such as the complexity of the system, the number of features required, and the size of the development team. The schedule will include the estimated time required for each phase of the development process, such as requirements gathering, design, coding, testing, and deployment. The budget will cover the costs associated with software development, such as salaries, hardware and software expenses, and any other relevant expenses.

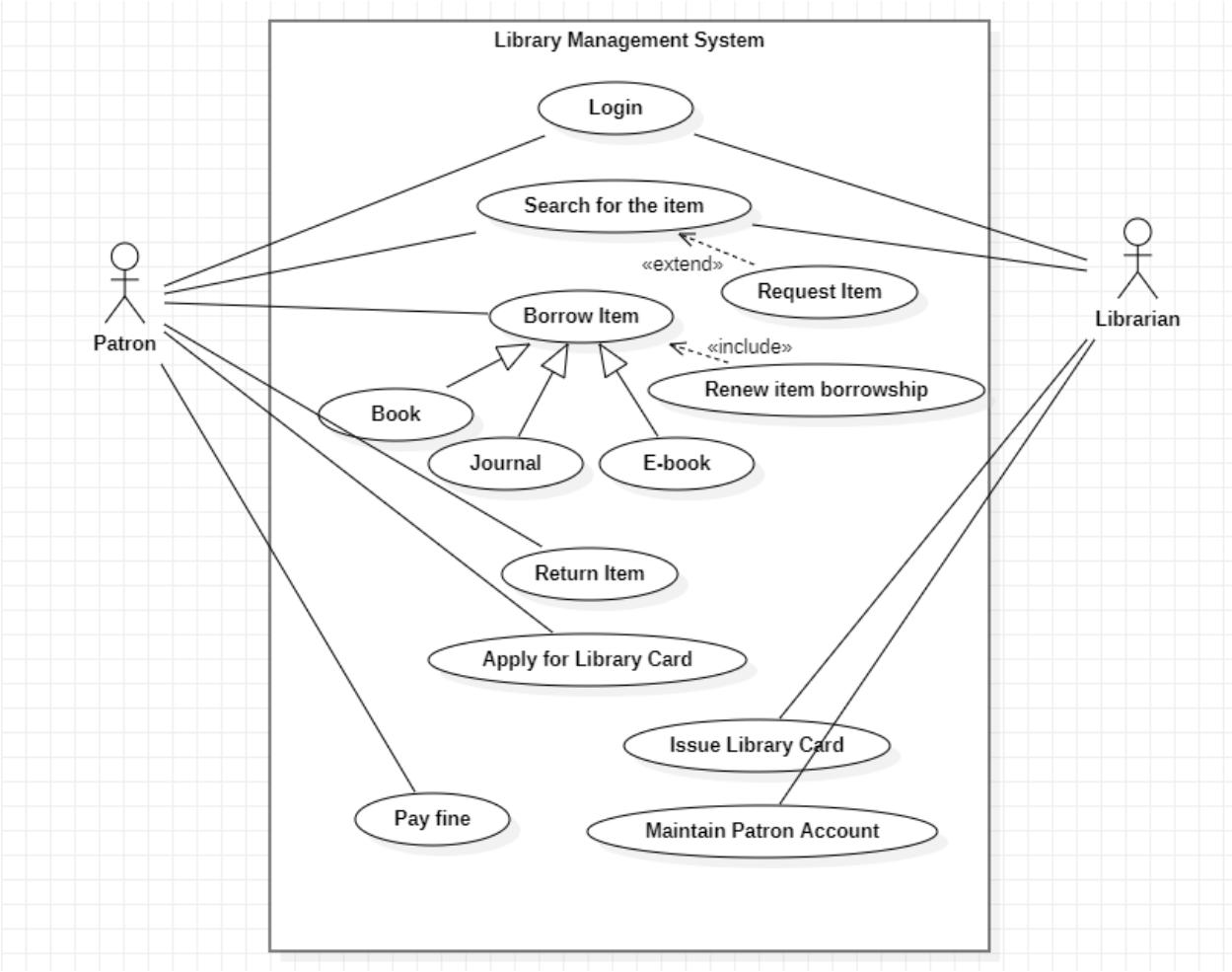
3.3 CLASS DIAGRAM



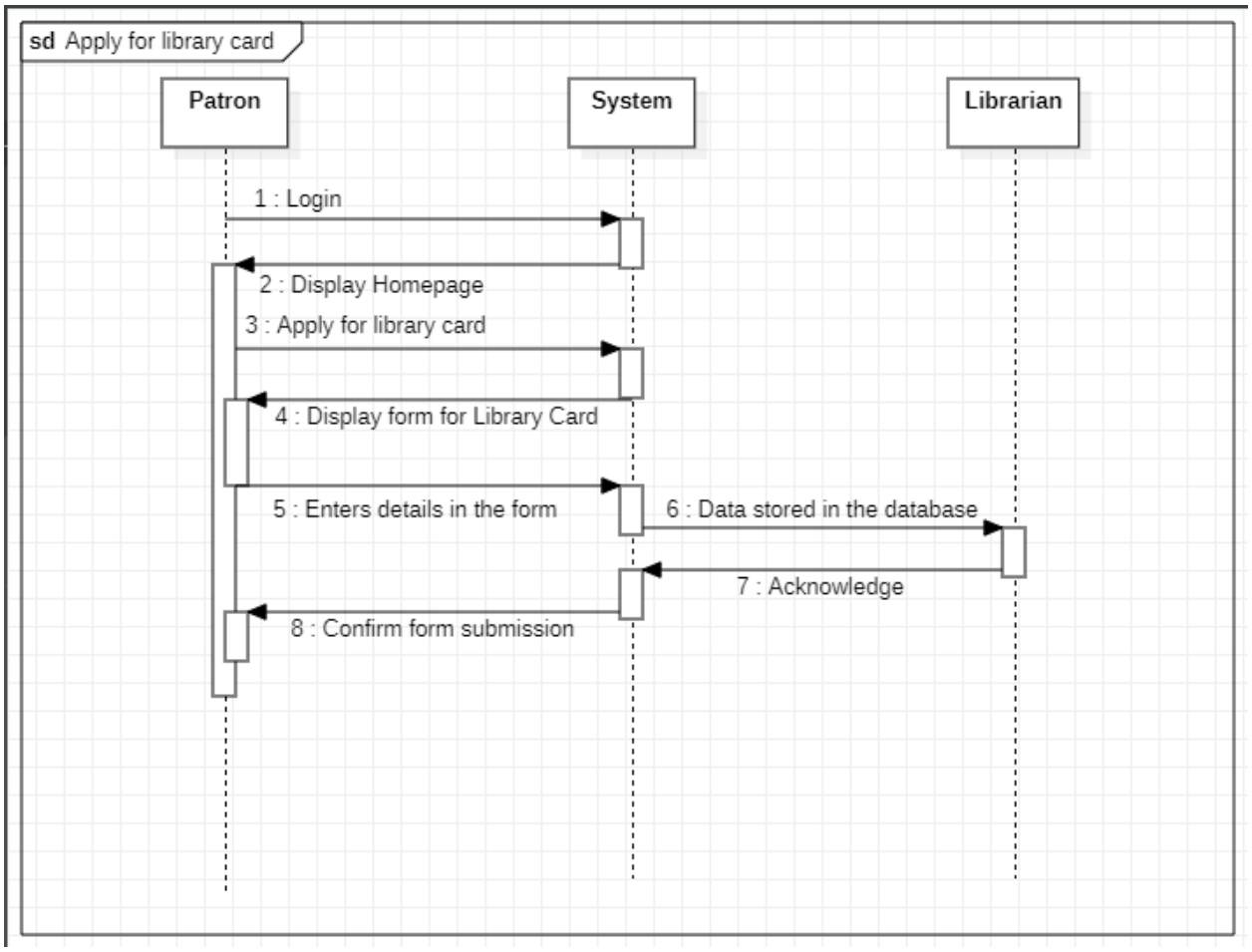
3.4 STATE DIAGRAM



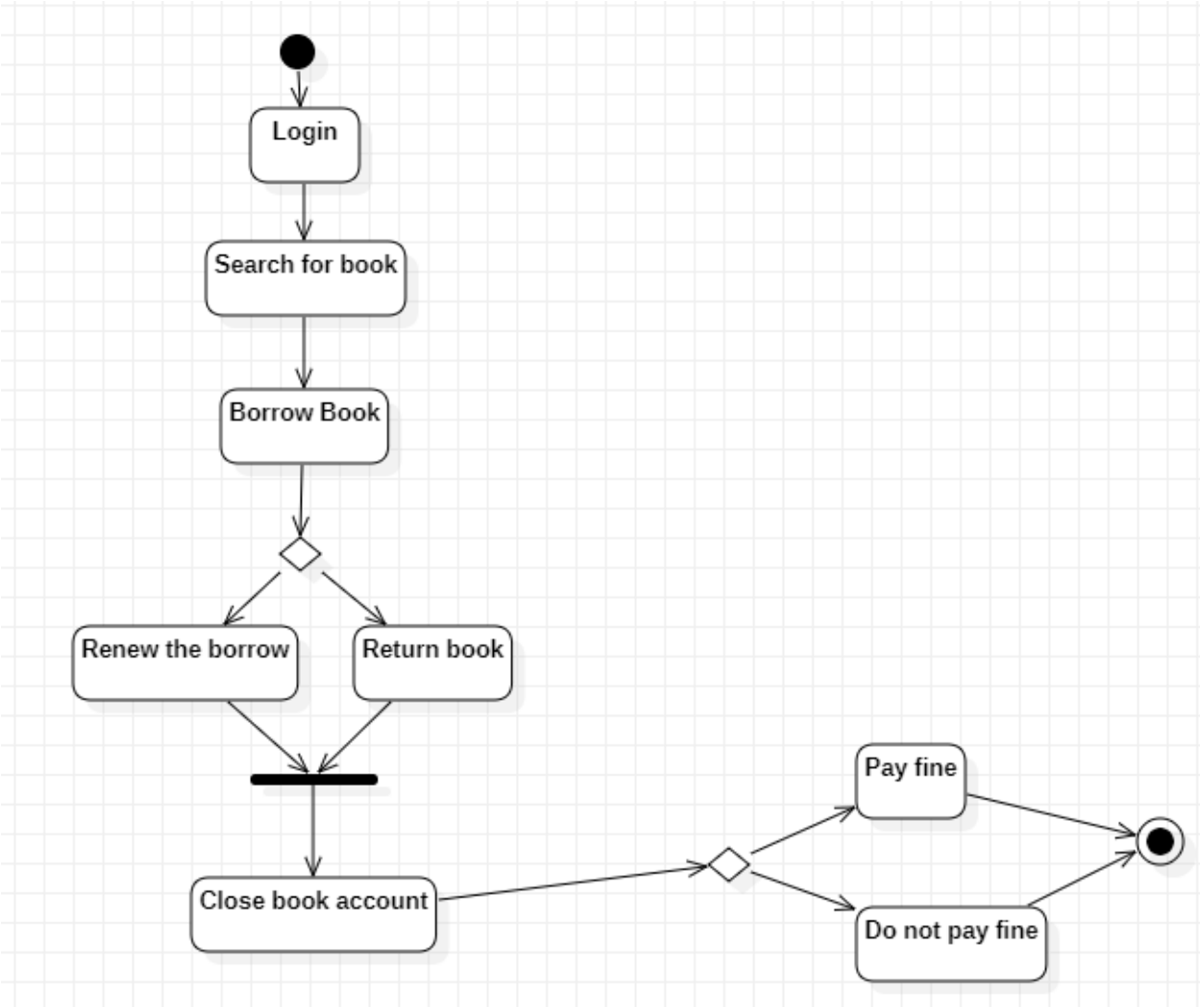
3.5 USE CASE DIAGRAM



3.6 SEQUENCE DIAGRAM



3.7 ACTIVITY DIAGRAM



4. ONLINE SHOPPING SYSTEM

4.1 PROBLEM STATEMENT: To design an online shopping software system that allows customers to browse products, add items to their cart, make payments, and track order statuses. The system should also allow administrators to manage products, view order information, and generate reports.

4.2 SYSTEM REQUIREMENTS & SPECIFICATIONS

1)Introduction:

1.1) Purpose: The purpose of this document is to outline the requirements for an online shopping software system

1.2) Scope: This document describes the objectives and values that the software system will provide to the customers. It also includes the estimated cost and time required for development

1.3) Overview: This section provides a brief summary of the online shopping software system.

2) General Description: The software system should be able to provide a platform for customers to browse and purchase products online. The system should have features such as a shopping cart, order tracking, and a search function. The system should be user-friendly and accessible to people of all ages and technical backgrounds.

3) Functional Requirements: The software system should be able to perform the following functions:

i. Allow customers to browse products by category or keyword search
ii. Provide detailed information about each product
iii. Allow customers to add products to a shopping cart
iv. Allow customers to make payments securely
v. Provide customers with an order confirmation and tracking information

4) Interface Requirements: The software system should have a user-friendly interface that allows customers to easily navigate and use the system. It should have a clear layout, and customers should be able to find the products they are looking for quickly and easily.

5) Performance Requirements: The software system should be able to handle a large number of users and transactions simultaneously without crashing. It should have fast response times and minimal downtime. It should be able to handle a large volume of product data without slowing down.

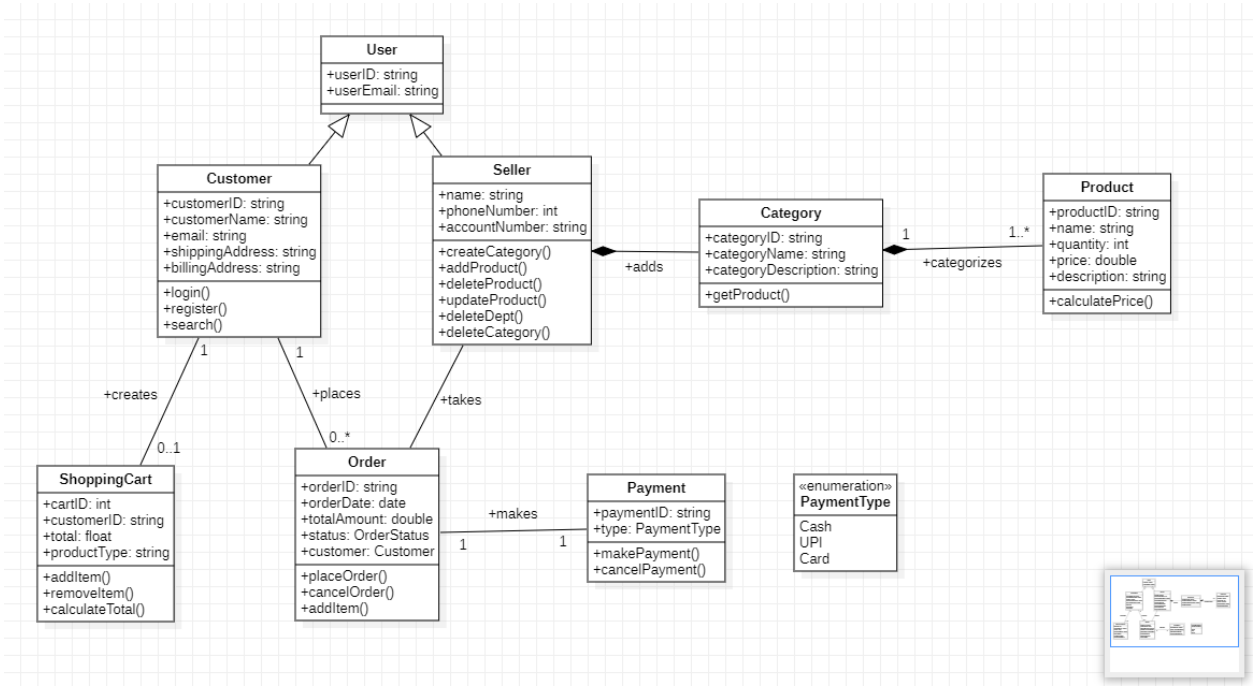
6) Design Constraints: The software system should be designed to be compatible with a variety of platforms and devices. It should be designed to work on both desktop and mobile devices. The system should be designed to be scalable, allowing for future expansion.

7) Non-Functional Attributes: The software system should have the following non-functional attributes:

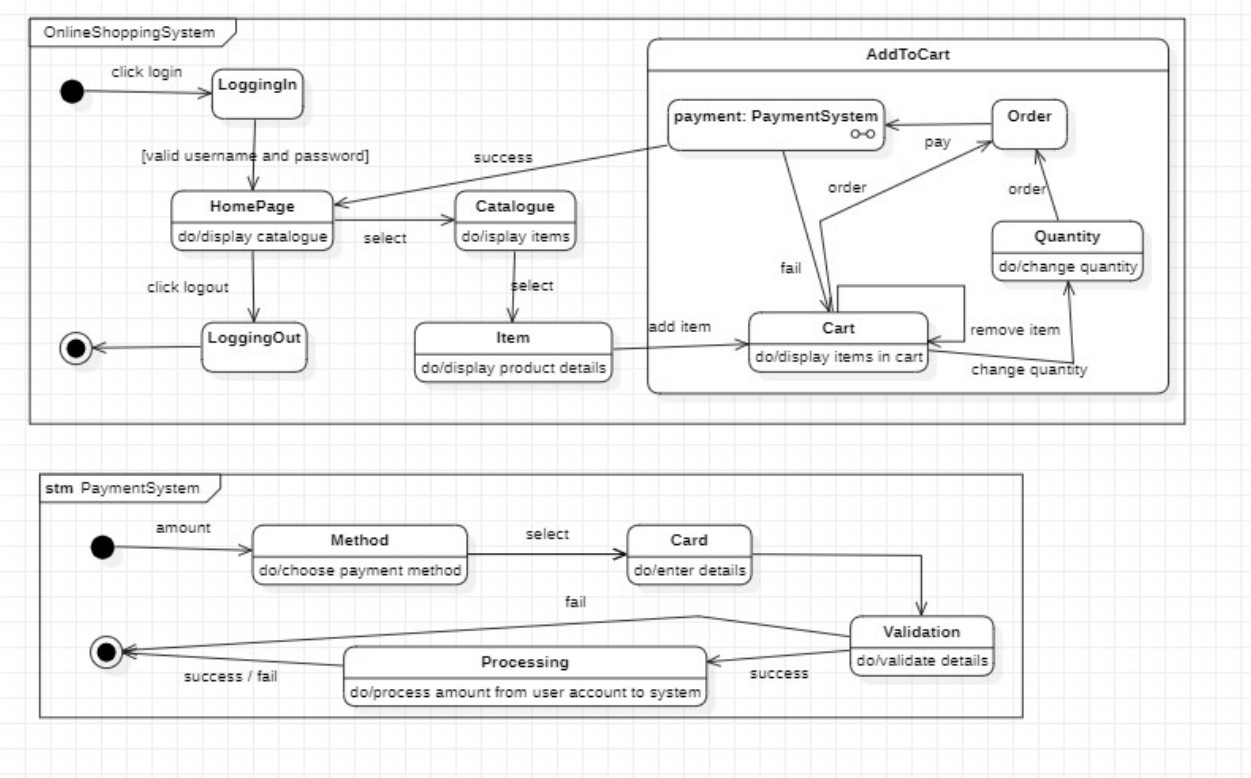
i. Security: The system should be secure and protect customer information from unauthorized access.
ii. Reliability: The system should be reliable and perform as expected.
iii. Portability: The system should be portable and able to be used on multiple platforms.
iv. Usability: The system should be easy to use and navigate.
v. Scalability: The system should be designed to be scalable and able to handle a growing number of users and transactions.

8) Preliminary Schedule and Budget: The development of the online shopping software system should be completed within a reasonable time frame and budget. The estimated cost and time required for development should be included in the project plan.

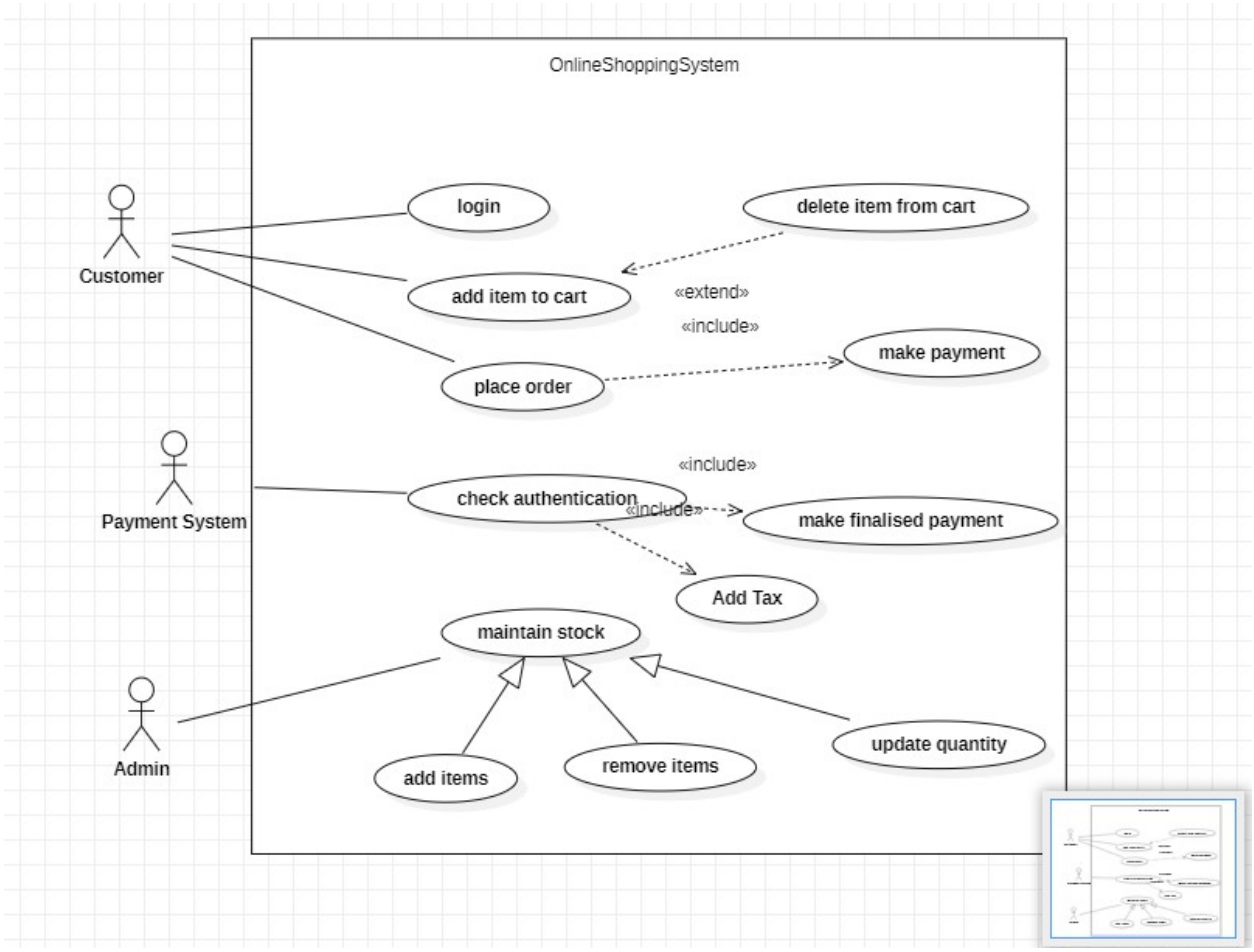
4.3 CLASS DIAGRAM



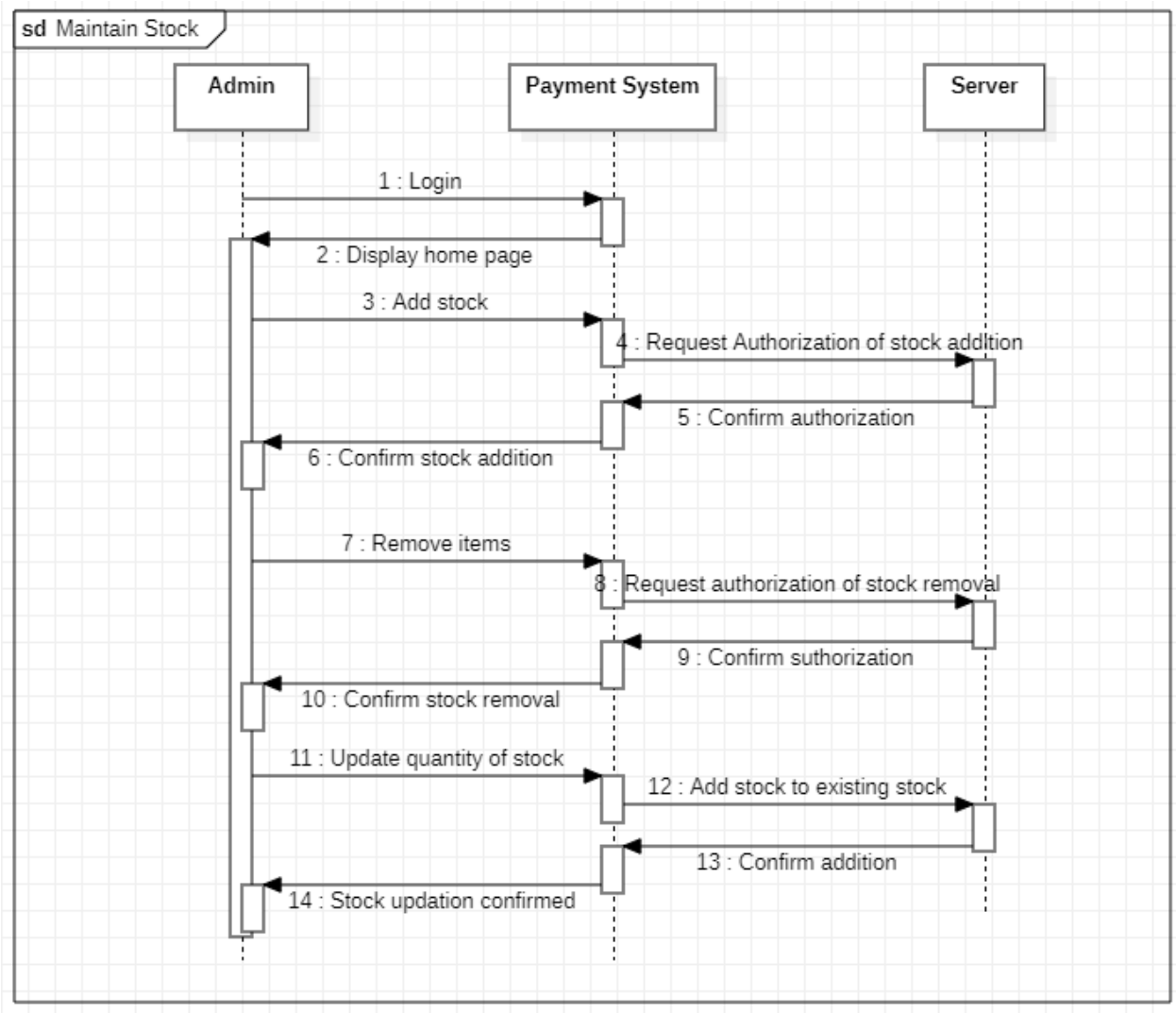
4.4 STATE DIAGRAM



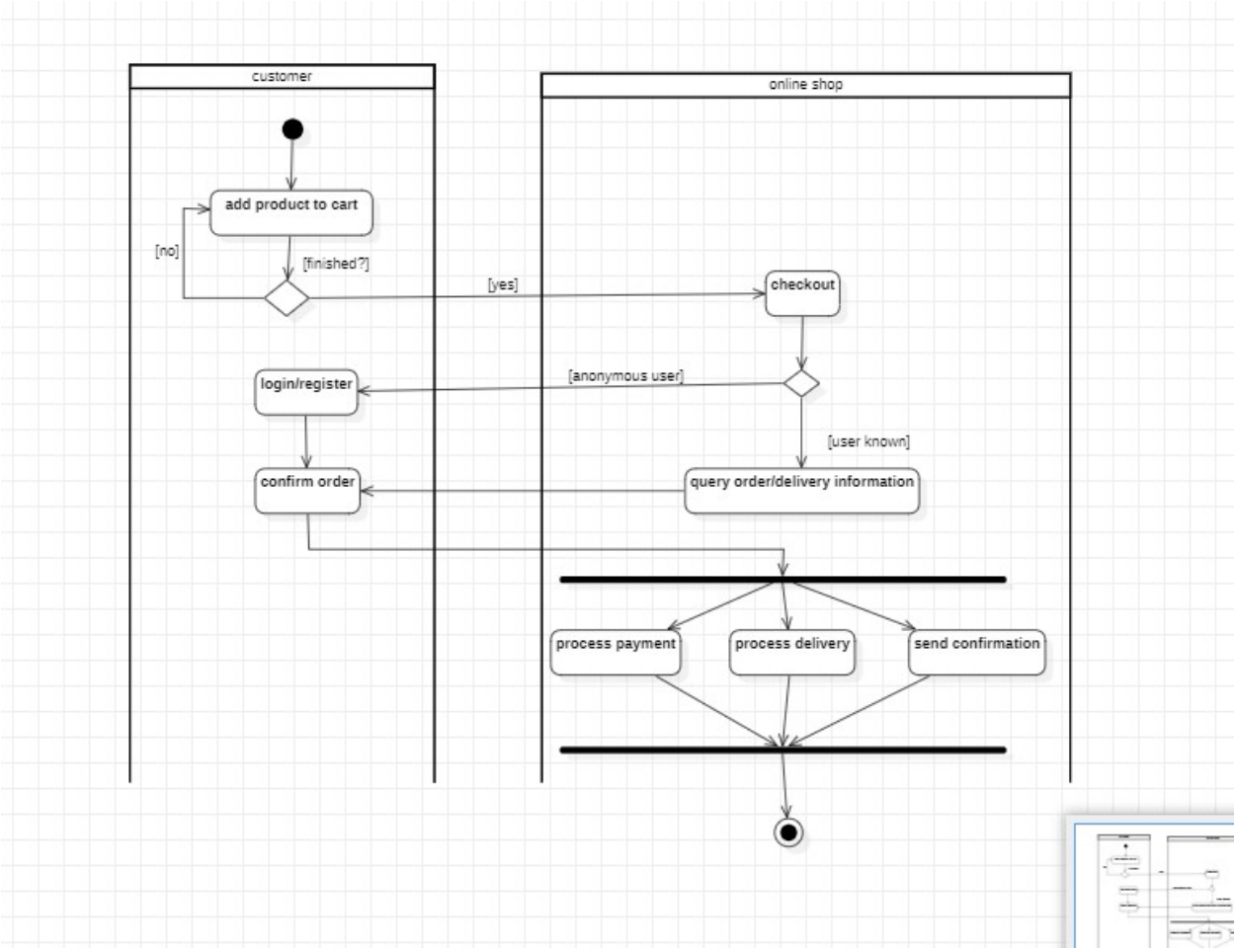
4.5 USE CASE DIAGRAM



4.6 SEQUENCE DIAGRAM



4.7 ACTIVITY DIAGRAM



5. PASSPORT AUTOMATION SYSTEM

5.1 PROBLEM STATEMENT: The current passport application process involves several manual steps, leading to long wait times, inefficiencies, and errors. To address these issues, a passport automation software system is needed to streamline the process and enhance the user experience. The software should allow users to easily apply for passports online, track their application status, and schedule appointments at passport centers. The system should also ensure data security and accuracy, and be scalable to accommodate increasing volumes of passport applications.

5.2 SYSTEM REQUIREMENTS & SPECIFICATIONS

1) Introduction:

1.1) Purpose of this Document: The purpose of this document is to define the system requirements for a passport automation software system

1.2) Scope of this document: The scope of this document is to describe the functionality, objectives, benefits, and value of the passport automation software system, as well as the development cost and time required

1.3) Overview: This section provides an overview of the passport automation software system, including a summary of its features and benefits.

2) General Description: The passport automation software system aims to automate the passport application process, improve the accuracy of data, and provide a user-friendly interface for both applicants and passport office staff. It will feature online application submission, automated form validation, biometric verification, and real-time status updates. The system is designed for use by both citizens and government officials.

3) Functional Requirements: The passport automation software system should be able to process passport applications, validate form data, capture biometric information, and issue passports. The system should also provide real-time status updates and generate reports. All functional requirements should be ranked in order of importance.

4) Interface Requirements: The system should have user-friendly interfaces for both applicants and passport office staff. These interfaces should include data input screens, data validation messages, and real-time status updates.

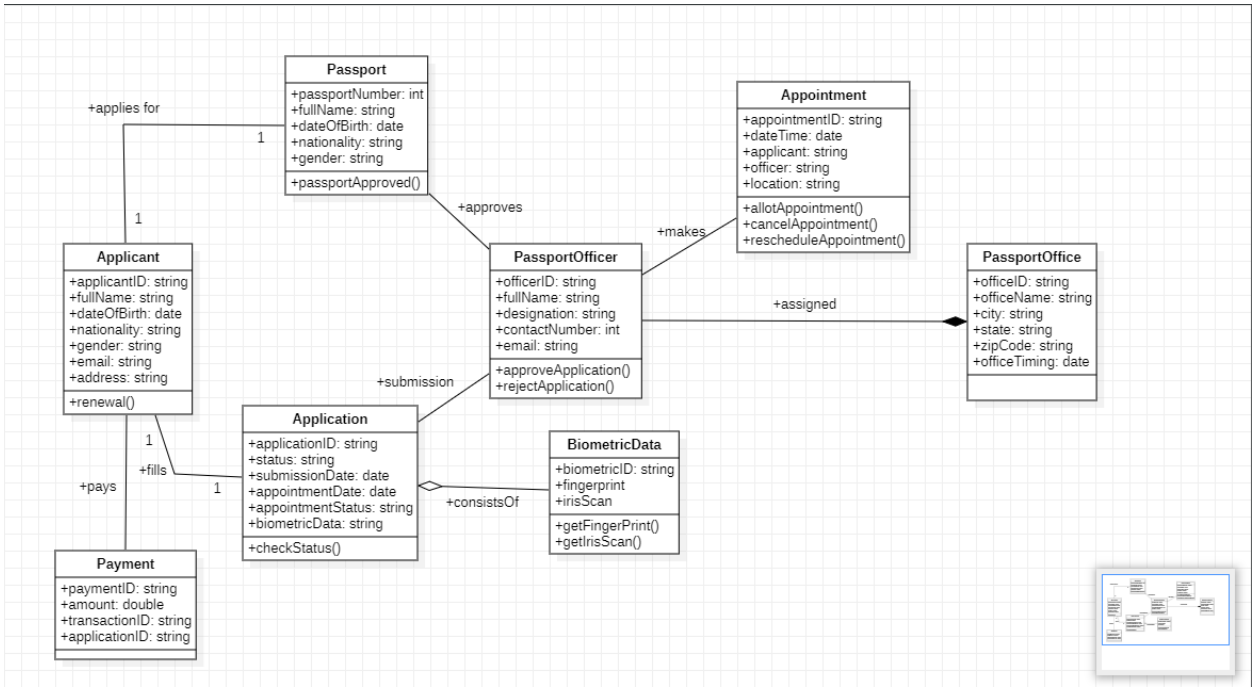
5) Performance Requirements: The system should be able to handle a high volume of passport applications and process them in a timely and accurate manner. The system should also have a maximum error rate of less than 1%.

6) Design Constraints: The passport automation software system should adhere to all relevant government regulations and guidelines. The system should also be designed to work on a variety of hardware and software platforms.

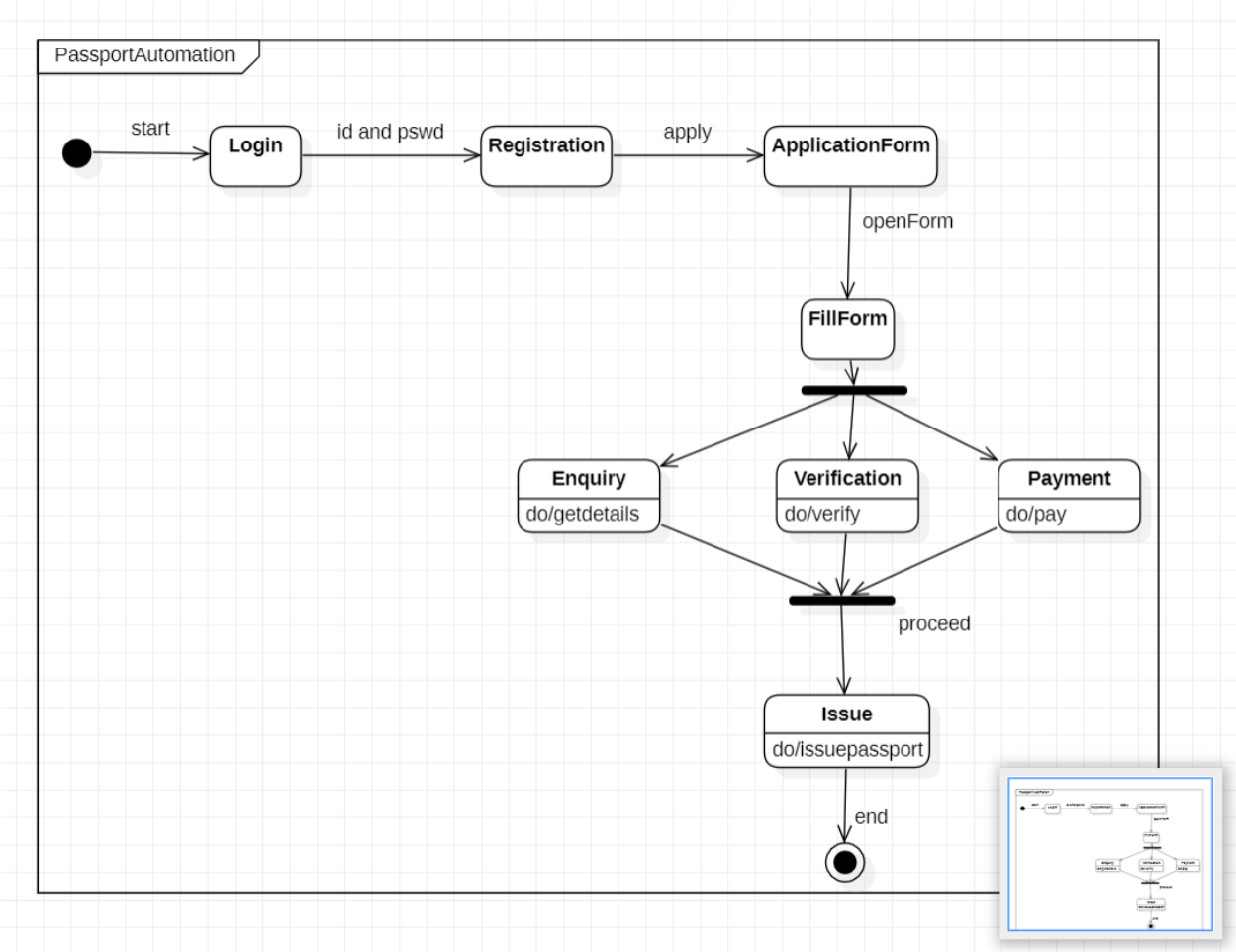
7) Non-Functional Attributes: The passport automation software system should be secure, reliable, portable, and compatible with other systems. The system should also ensure data integrity, scalability, and reusability.

8) Preliminary Schedule and Budget: The development of the passport automation software system is estimated to take approximately six months and will require a budget of $500,000.

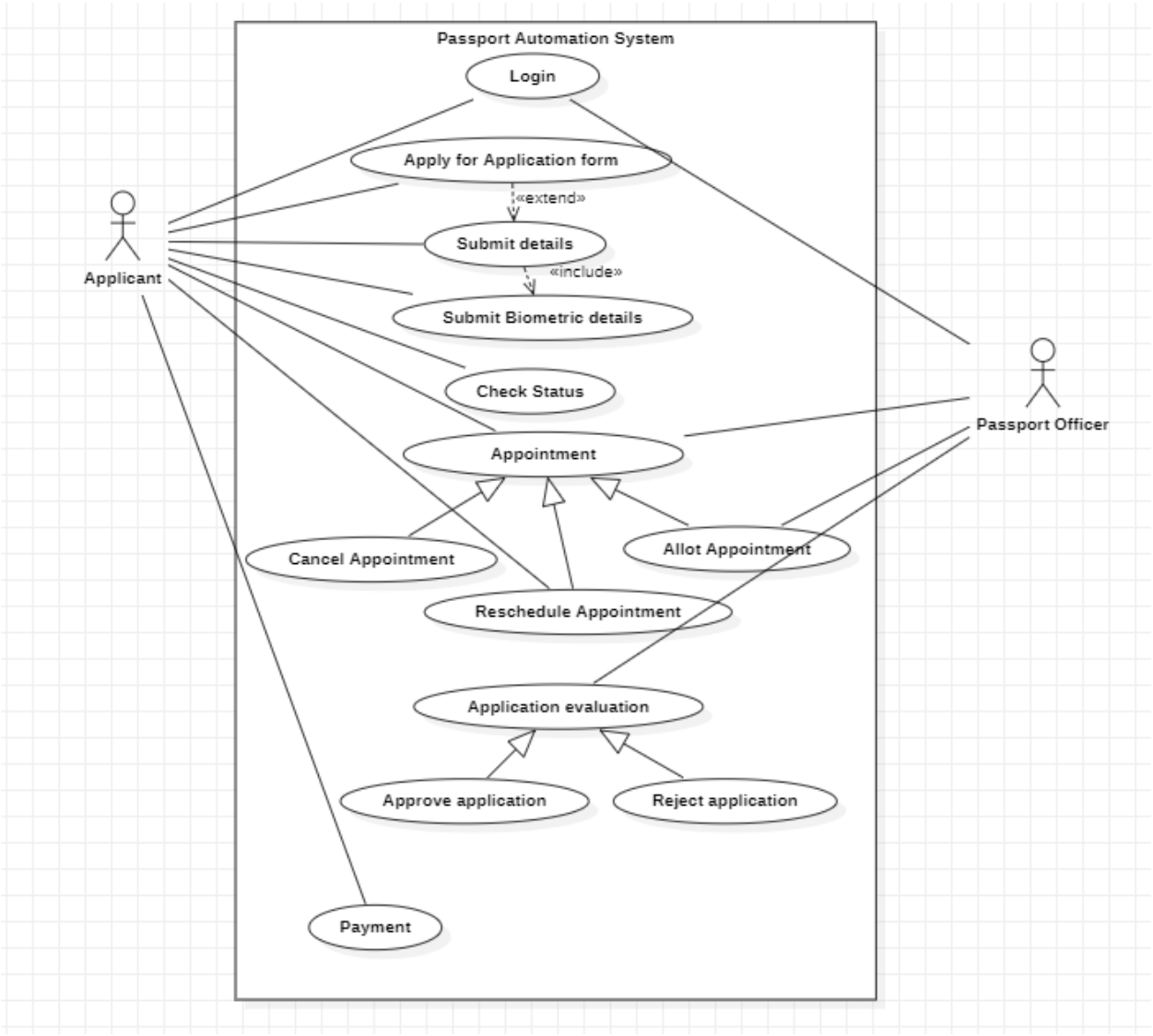
5.3 CLASS DIAGRAM



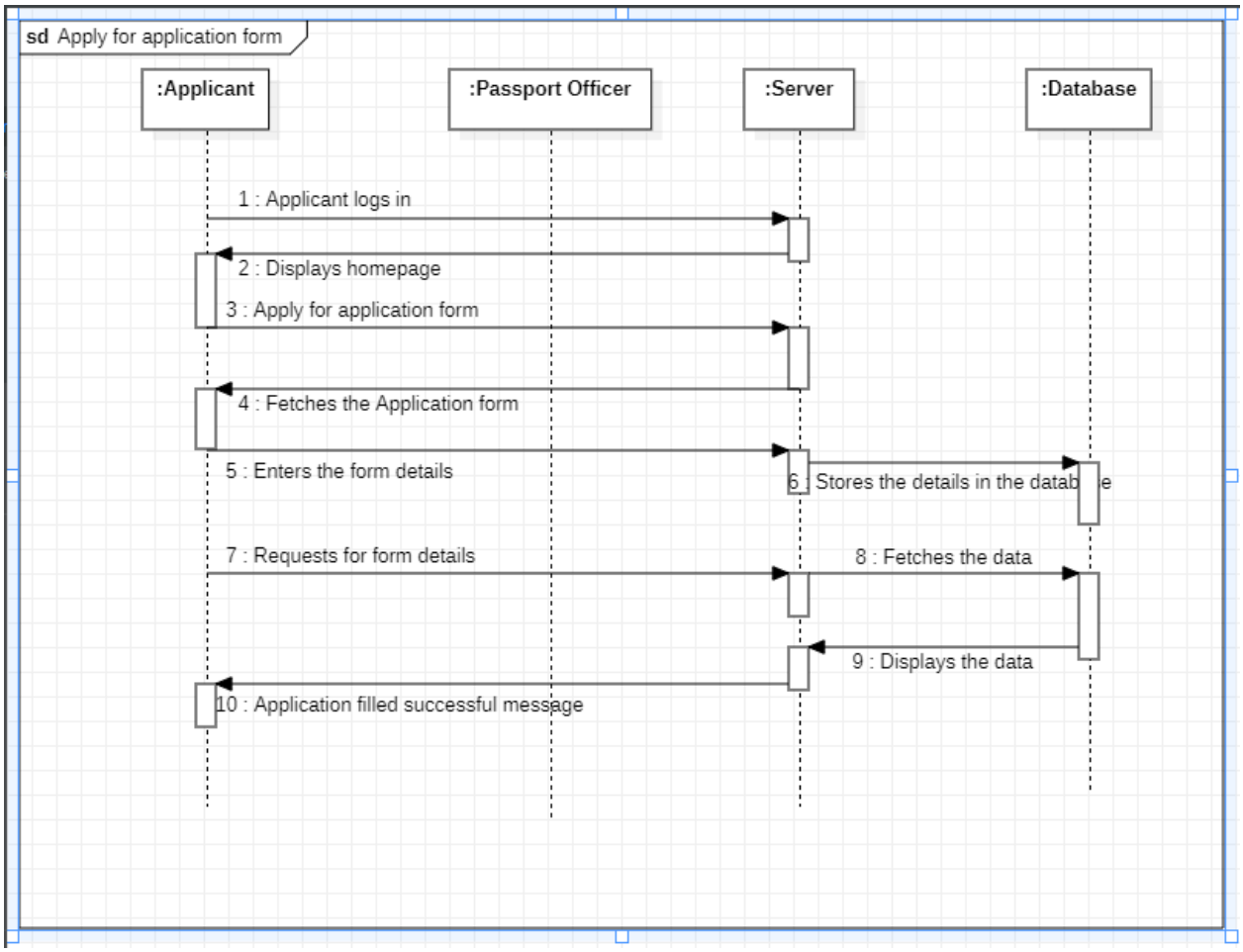
5.4 STATE DIAGRAM



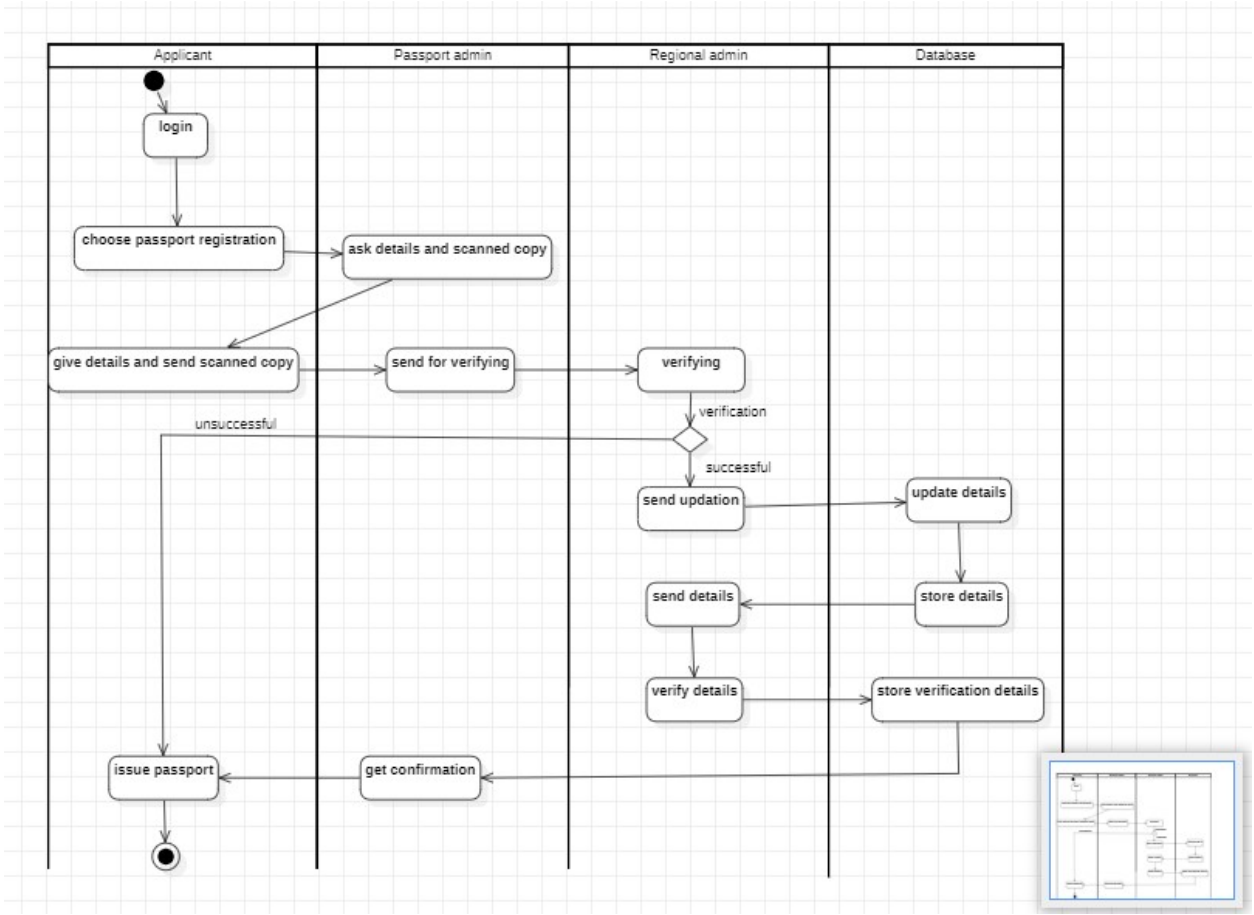
5.5 USE CASE DIAGRAM



5.6 SEQUENCE DIAGRAM



5.7 ACTIVITY DIAGRAM



6. RAILWAY RESERVATION SYSTEM

6.1 PROBLEM STATEMENT: To develop a Railway Reservation Software System that allows passengers to book tickets, cancel bookings, check train schedules, and manage reservations online. The system should also allow the railway authorities to manage train schedules, allocate seats, and manage train routes and fares efficiently. The system should be user-friendly, secure, reliable, and scalable to handle a large number of simultaneous users.

6.2 SYSTEM REQUIREMENTS & SPECIFICATIONS

1) Introduction:

1.1) Purpose of this Document: The purpose of this document is to outline the requirements for the development of a railway reservation software system, which will enable users to book, modify and cancel railway tickets online.

1.2) Scope of this Document: This document outlines the objectives and scope of the railway reservation software system, along with the estimated development cost and timeline

1.3) Overview: The railway reservation software system will provide a user-friendly platform for passengers to book and manage their railway tickets online, reducing the need for manual ticket booking and providing greater convenience to users.

2) General Description: The railway reservation software system will allow passengers to book railway tickets online through a user-friendly interface, with features including real-time train availability, seat selection, and payment processing. The system will cater to a diverse user community, including passengers with different levels of technical expertise, mobility needs, and language preferences.

3) Functional Requirements: The functional requirements of the railway reservation software system include the following:

i. Users should be able to search for available trains by specifying the source and destination stations, date and time of travel, and class of travel.
ii. Users should be able to view the train schedule, fare, and seat availability for their chosen route and date.
iii. Users should be able to select their preferred seat(s) and book their ticket(s) online, with payment processing through a secure payment gateway.
iv. Users should be able to view and manage their booked tickets, including modifying or canceling their booking if necessary.
v. The system should generate e-tickets with all necessary details, including the passenger name, train number, seat number, and journey details.
vi. The system should provide a user-friendly interface with clear instructions and feedback messages, catering to users with different levels of technical expertise and language preferences.

4) Interface Requirements: The railway reservation software system should provide a user interface that is intuitive and easy to use, with clear instructions and feedback messages for users. The system should communicate with users through a variety of interfaces, including web browsers, mobile applications, and email notifications. The system should also communicate with external systems, such as payment gateways and train schedules, through standard interfaces such as APIs.

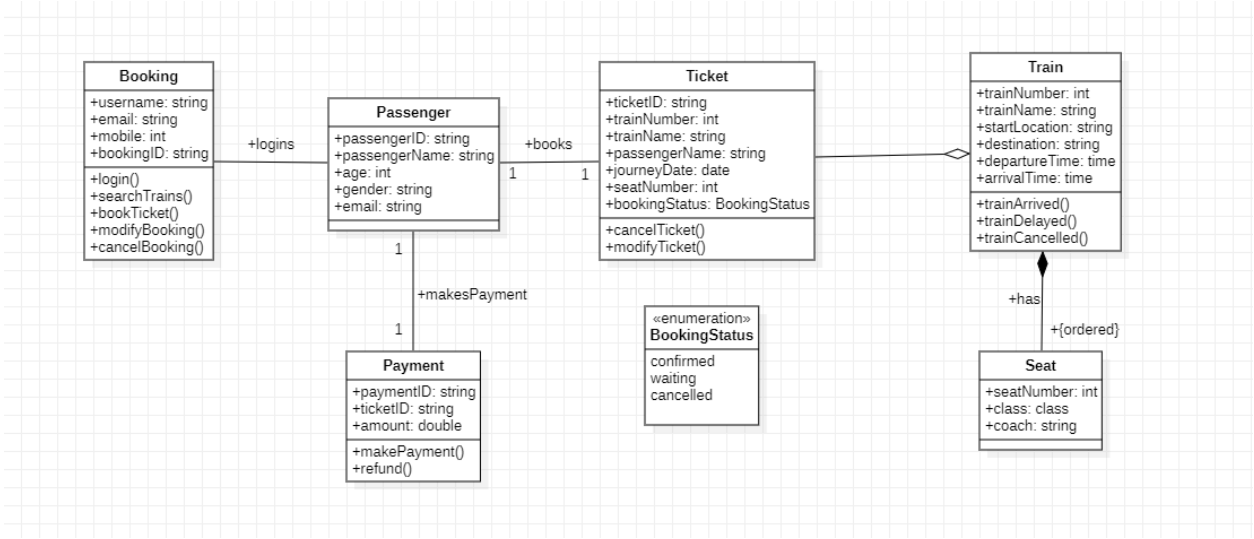
5) Performance Requirements: The railway reservation software system should be able to handle a large number of concurrent users, with fast response times and minimal downtime. The system should be able to handle peak loads during festival seasons and other busy periods, without compromising on performance. The system should also be able to handle unexpected errors and failures, with built-in redundancy and failover mechanisms.

6) Design Constraints: The railway reservation software system should be designed to comply with industry standards and best practices, with a modular and scalable architecture. The system should be designed to be easily maintainable and extensible, with clear separation of concerns and a robust error handling mechanism. The system should also be designed to work with existing infrastructure and systems, such as payment gateways and train schedules.

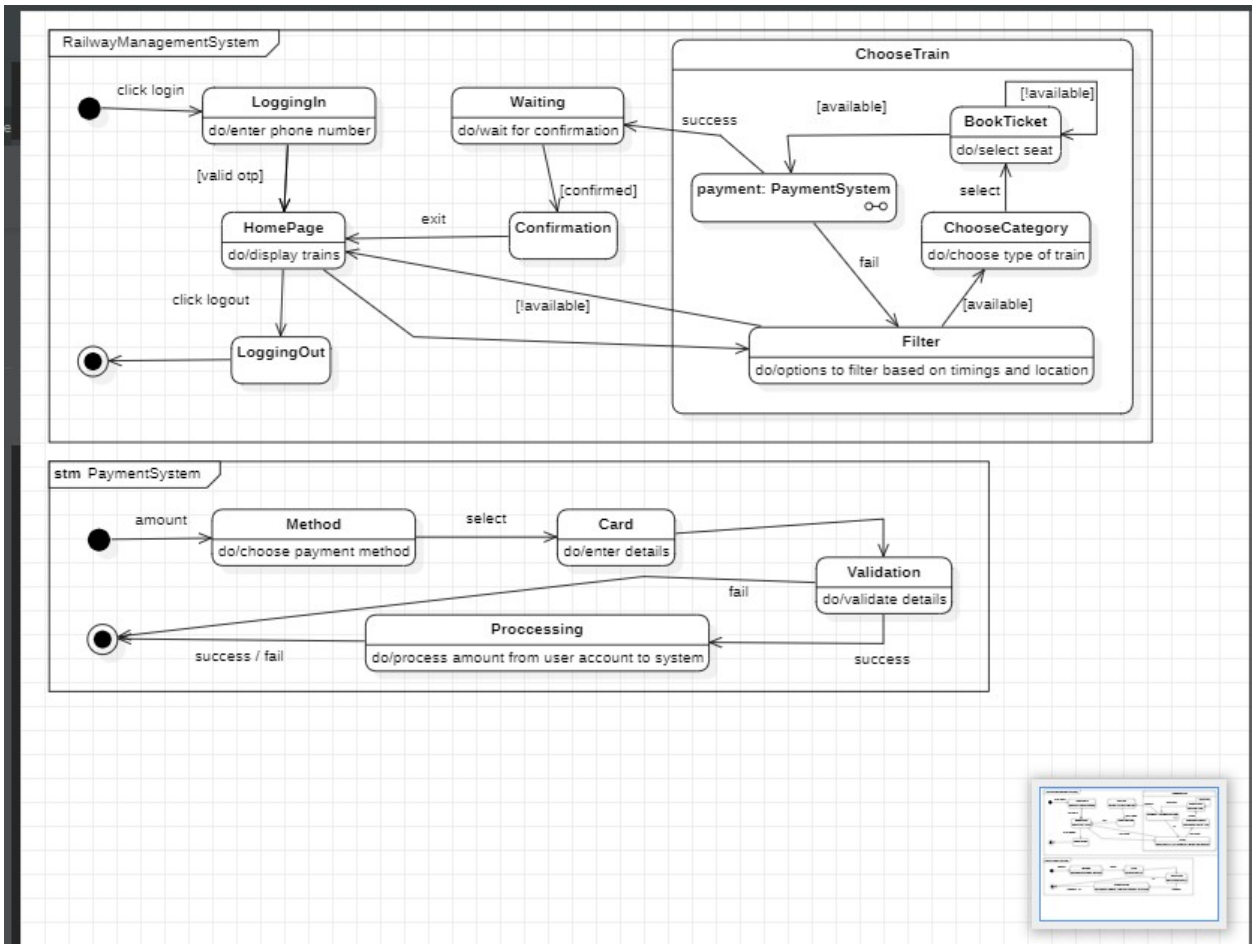
7) Non-Functional Attributes: The railway reservation software system should comply with various non-functional attributes, such as security, portability, reliability, reusability, application compatibility, data integrity, and scalability. The system should provide a secure platform for users to book and manage their tickets, with built-in security mechanisms such as encryption and authentication. The system should also be portable, compatible with different operating systems and devices, and provide reliable and consistent performance across different environments.

8) Preliminary Schedule and Budget: The development of the railway reservation software system is expected to take around 12 months, with an estimated budget of $500,000. The project plan includes various phases, such as requirements gathering, design, development, testing, deployment, and maintenance, with clear milestones

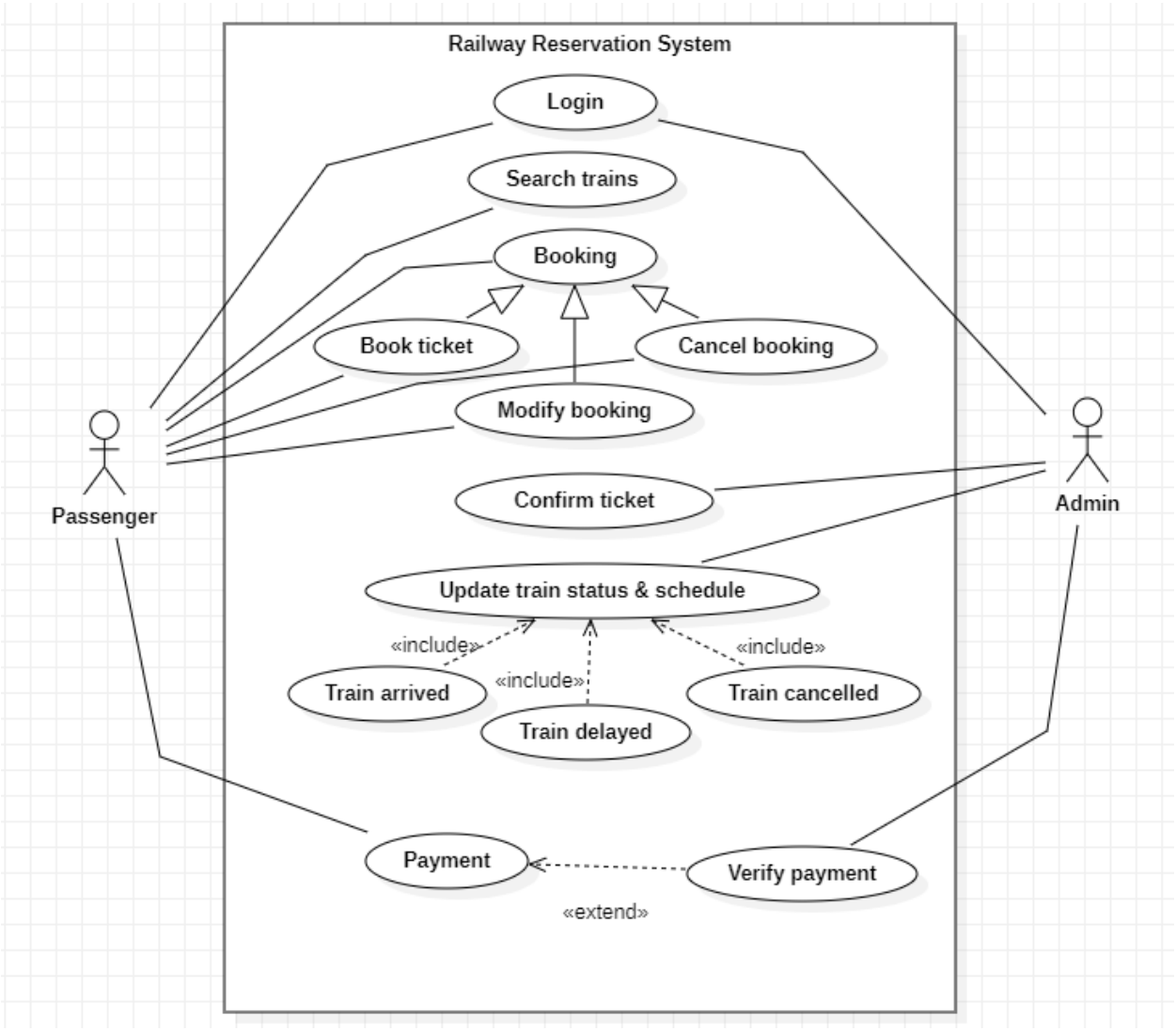
6.3 CLASS DIAGRAM



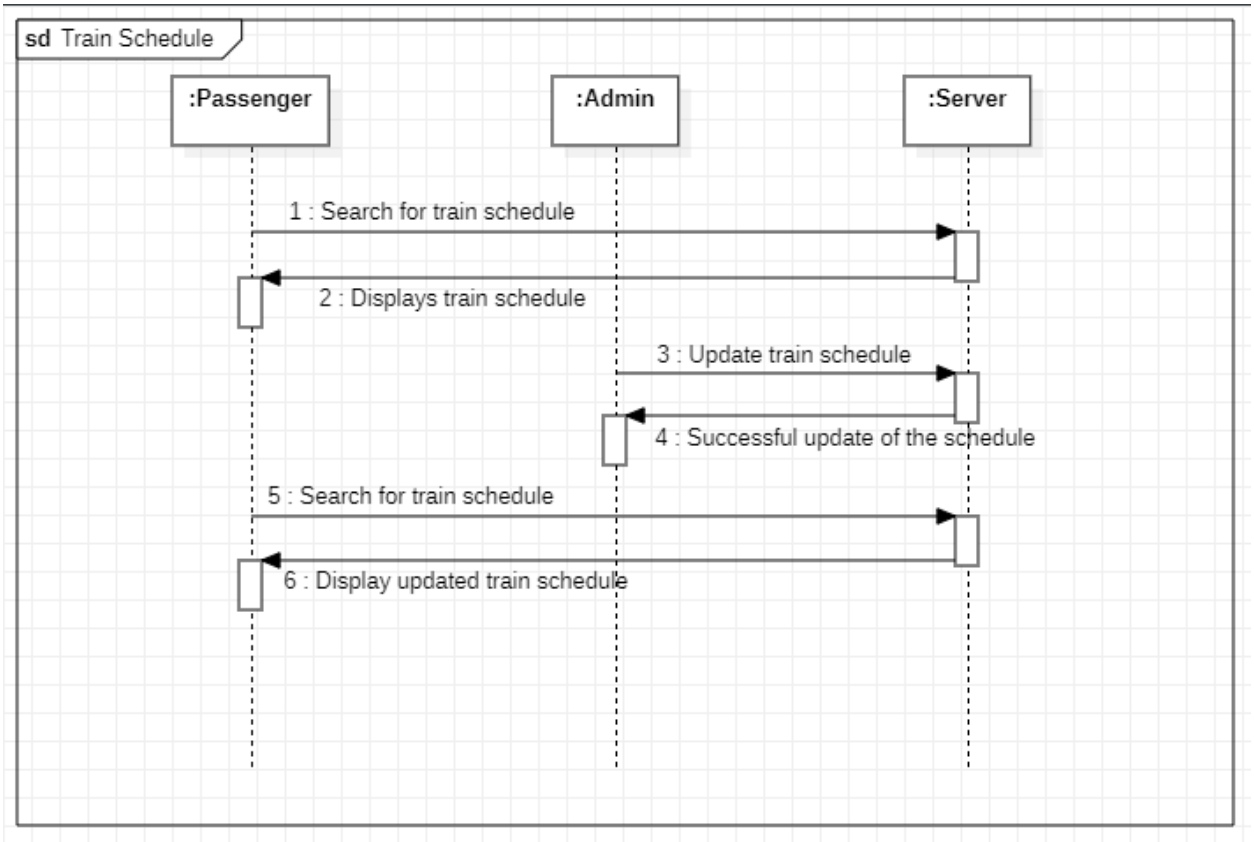
6.4 STATE DIAGRAM



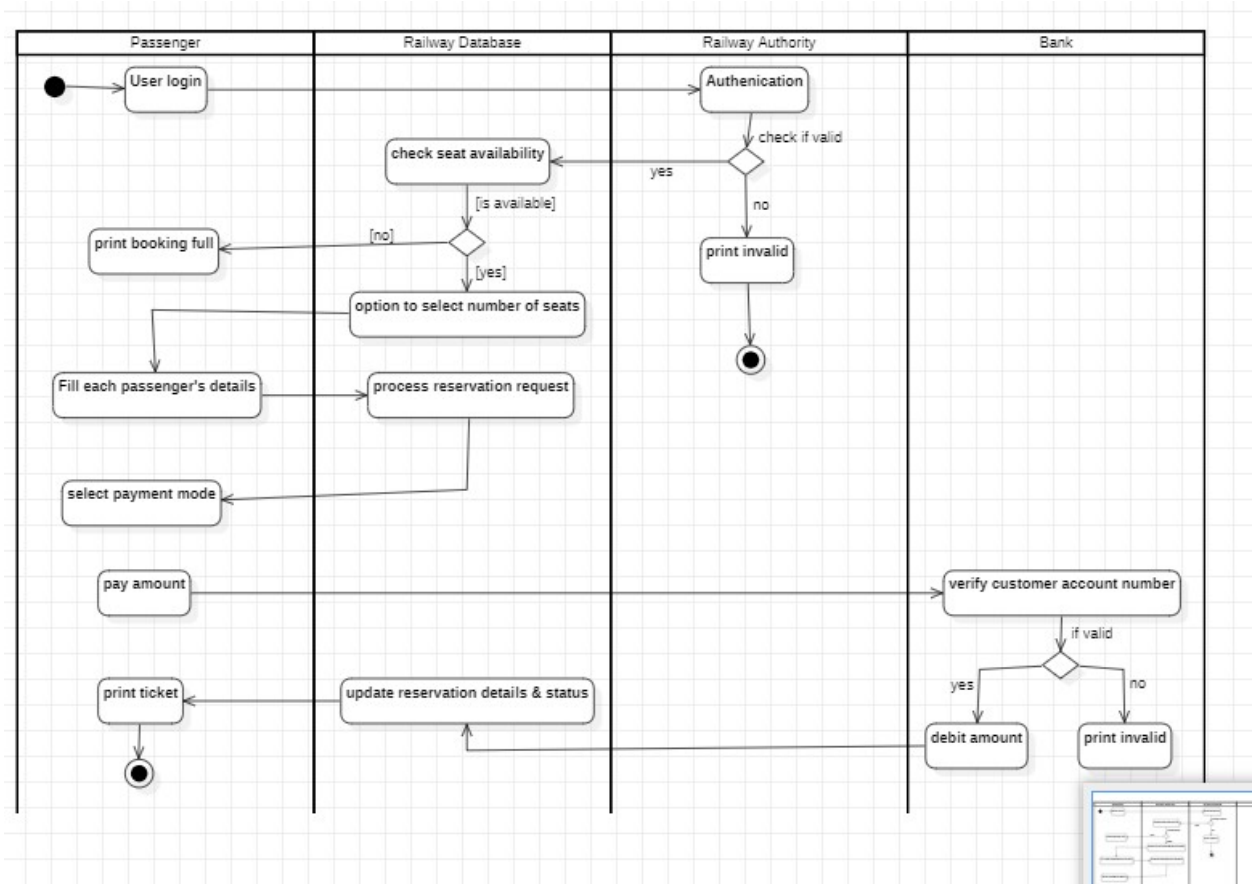
6.5 USE CASE DIAGRAM



6.6 SEQUENCE DIAGRAM



6.7 ACTIVITY DIAGRAM



7. STOCK MAINTENANCE SYSTEM

7.1 PROBLEM STATEMENT: To develop a stock maintenance system that allows businesses to easily track inventory levels, monitor stock movement, and generate real-time reports. This system will help businesses streamline their inventory management processes, reduce errors, and make data-driven decisions to optimize their stock levels.

7.2 SYSTEM REQUIREMENTS & SPECIFICATIONS

1) Introduction:

1.1) Purpose of this Document: The purpose of this document is to provide a detailed description of the requirements for a stock maintenance software system.

1.2) Scope of this document: This document describes the overall working and main objective of the stock maintenance software system, and the value it will provide to the customer. It also includes a description of the development cost and time required.

1.3) Overview: The stock maintenance software system is designed to assist businesses in maintaining accurate records of their inventory, including stock levels, product information, and supplier details.

2) General Description: The stock maintenance software system is intended for use by businesses of all sizes that manage inventory. The system will provide users with the ability to track inventory levels, monitor stock movements, and generate reports on stock availability, pricing, and trends. The software will be easy to use, with an intuitive interface that is suitable for users of all levels of experience.

3) Functional Requirements: The stock maintenance software system must be able to perform the following functions:

i. Track inventory levels and locations
ii. Record product information, including name, description, SKU, and supplier details
iii. Monitor stock movements, including incoming and outgoing shipments, and product transfers between locations
iv. Generate reports on inventory levels, stock movements, pricing, and trends

v. Provide users with the ability to set minimum and maximum stock levels and receive alerts when these levels are reached
vi. Allow for the creation and management of purchase orders
vii. Provide an interface for the management of supplier information, including contact details and payment terms

4) Interface Requirements: The stock maintenance software system will communicate with users through a graphical user interface (GUI). The system will also communicate with other software systems, such as accounting software, through APIs.

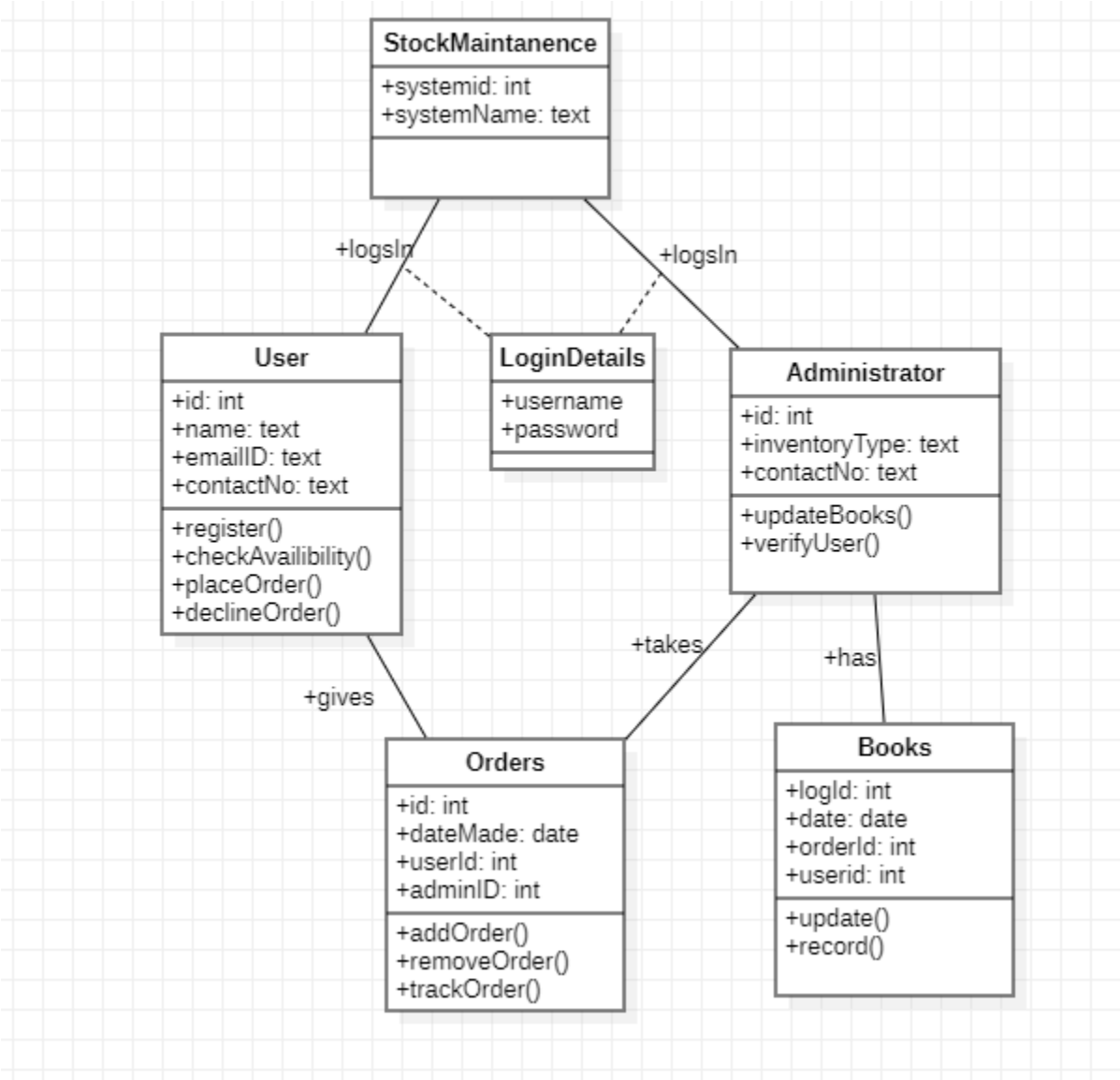
5) Performance Requirements: The stock maintenance software system must be able to perform its functions efficiently and accurately, with a low error rate. The system should be able to handle large volumes of data, and provide quick response times to user queries.

6) Design Constraints: The stock maintenance software system must be designed to operate on a variety of hardware and software platforms. The system must also be designed to be scalable, with the ability to handle increased volumes of data as the business grows.

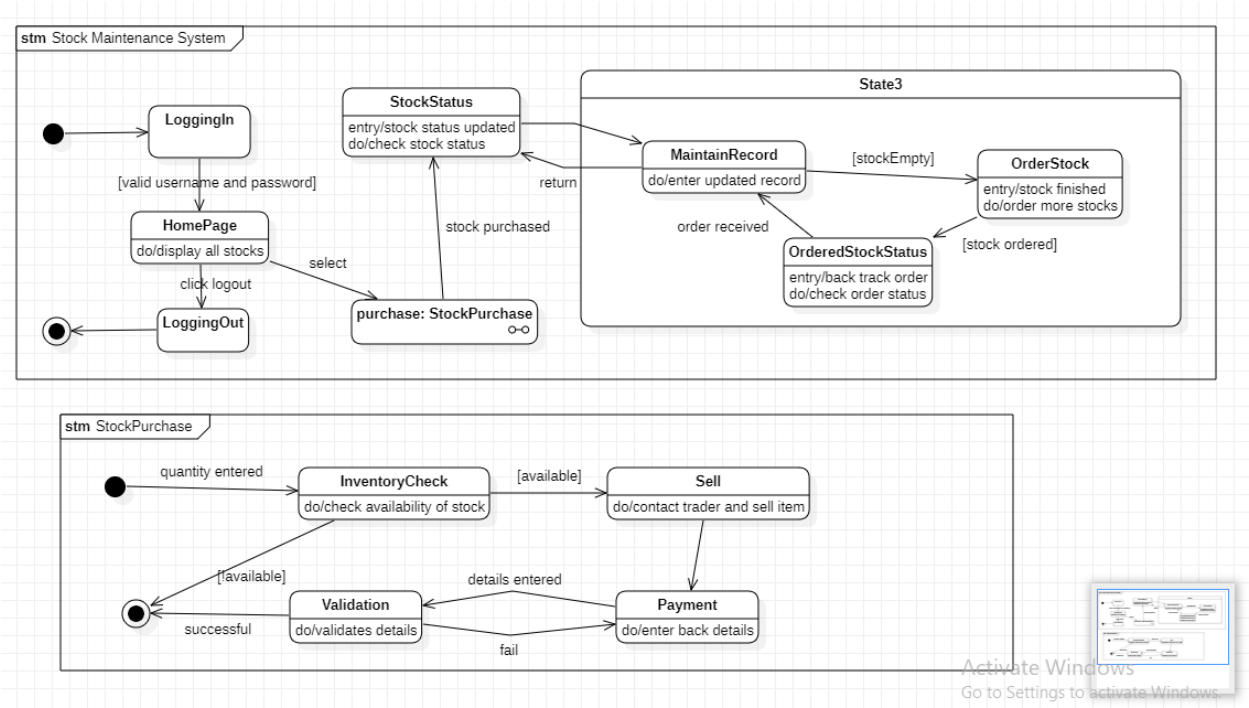
7) Non-Functional Attributes: The stock maintenance software system must be secure, with the ability to protect user data from unauthorized access. The system should also be reliable, with the ability to recover from errors and system failures. The system should be easy to maintain and update, with a modular design that allows for easy addition of new features.

8) Preliminary Schedule and Budget: The development of the stock maintenance software system is expected to take six months, with a budget of $100,000. This budget includes the cost of software development, testing, and implementation, as well as any necessary hardware and software purchases.

7.3 CLASS DIAGRAM

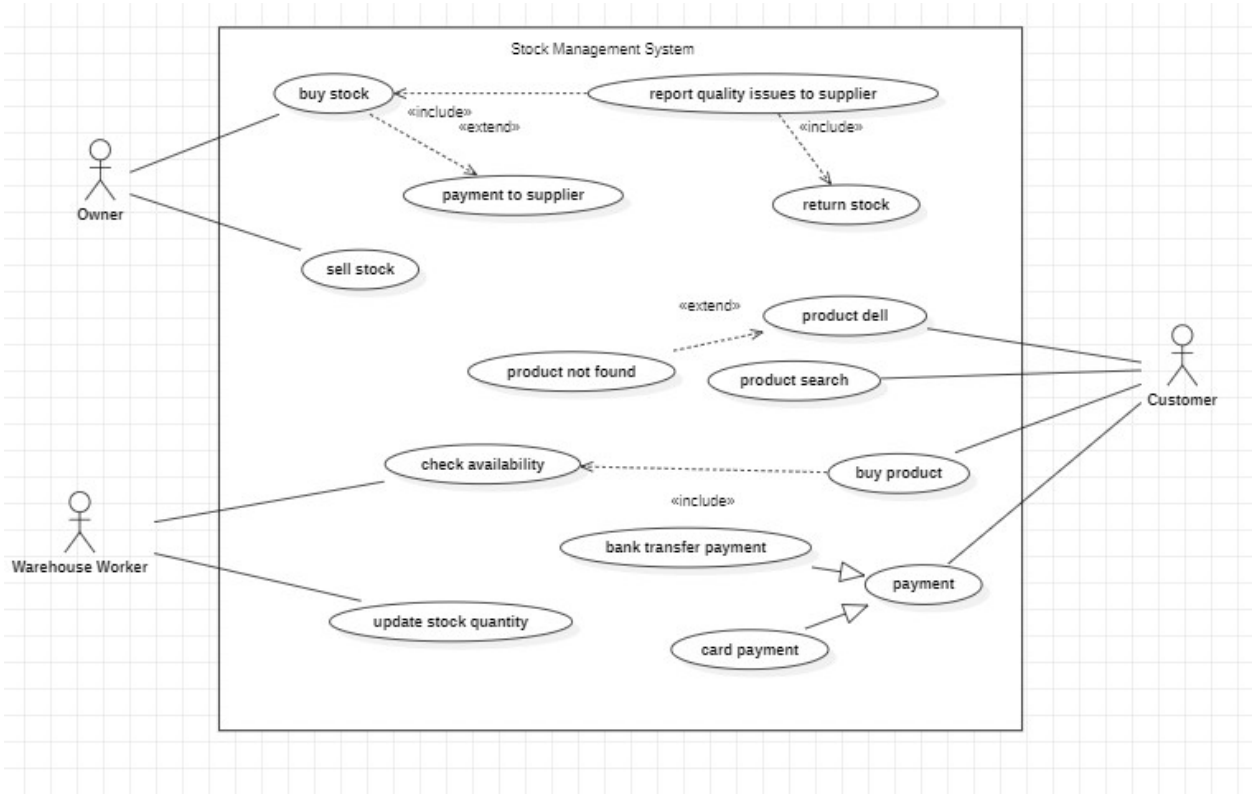


7.4 STATE DIAGRAM

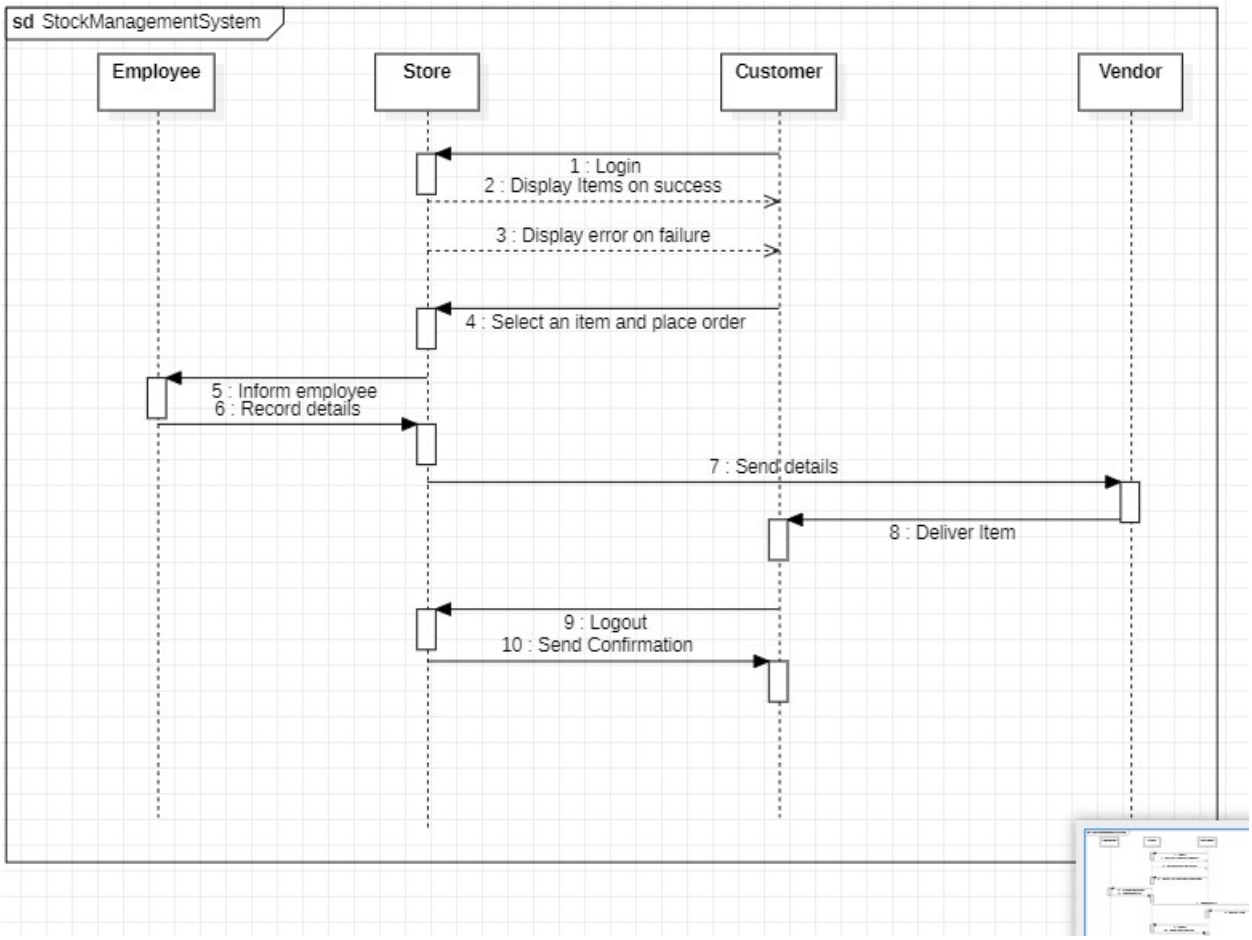
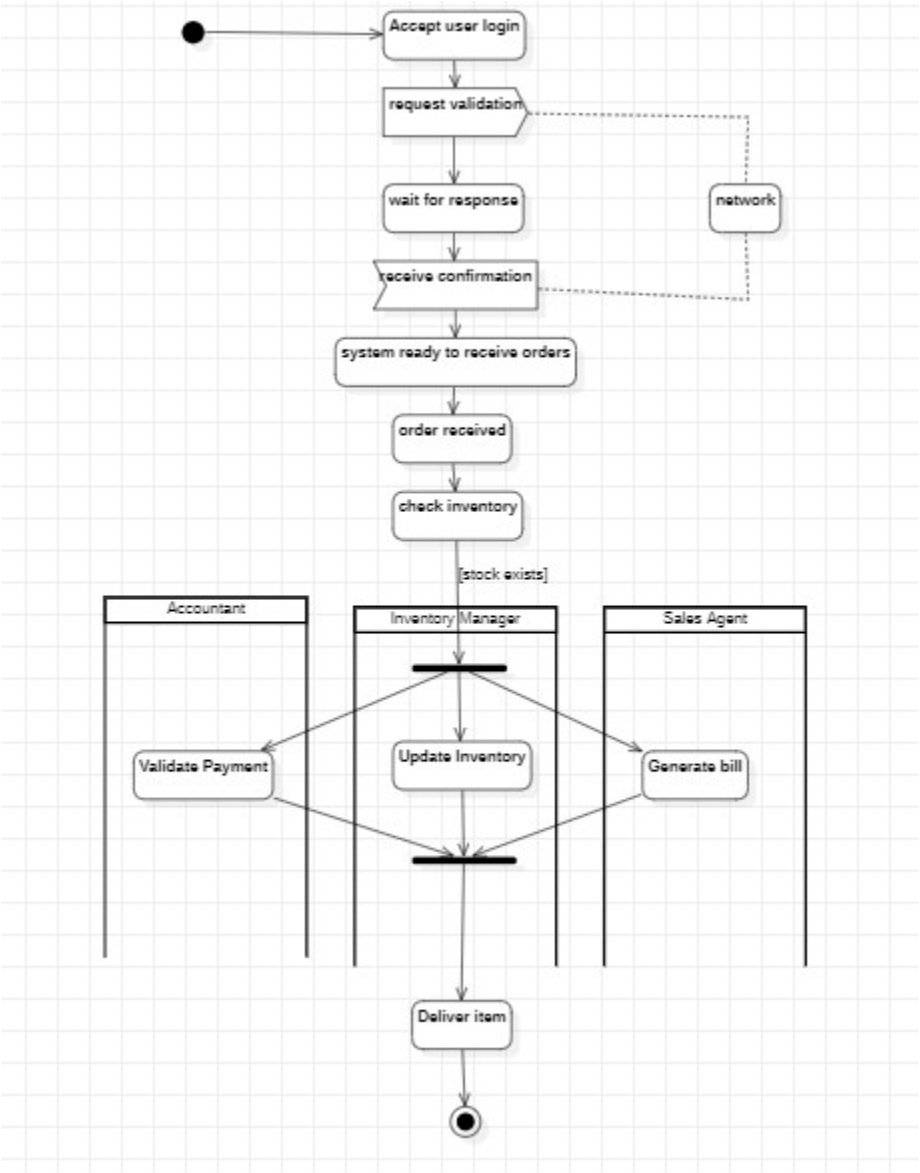


Activate Windows
Go to Settings to activate Windows.

7.5 USE CASE DIAGRAM



7.6 SEQUENCE DIAGRAM



7.7 ACTIVITY DIAGRAM