

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

OBJECT ORIENTED MODELLING AND DESIGN

Submitted by
MD Suraj Kumar
(1BM20CS079)

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 MODELING AND DESIGN**" was carried out by **MD Suraj Kumar (1BM20CS079)**, 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.

Dr.Latha NR
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

Index Sheet

Sl. No.	Experiment Title	Page No.
1	Hotel Management System	1
2	Credit Card Processing	12
3	Library Management System	23
4	Stock Maintenance System	33
5	Passport Automation System	48
6	Railway Reservation System	57
7	Online Shopping System	67

Course Outcome

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

1. Hotel Management System

Problem Statement:

Many hotels struggle to efficiently manage their operations due to the lack of an effective software system and understaffed infrastructure. Manual processes for managing reservations, room assignments, billing and payments, inventory, and reporting can lead to errors, inefficiencies, and reduced customer satisfaction. Therefore, there is a need for a reliable and user-friendly hotel management system that can automate and streamline these processes, improve the accuracy and efficiency of operations, and enhance the overall guest experience.

Software Requirement Specification (SRS)

1 Introduction

1.1 Purpose of this document

The software requirement specification (SRS) for the hotel management system outlines the functional and non-functional requirements for a system that is capable of managing all aspects of a hotel's operations. This document is intended for the developers who will be responsible for designing and implementing the software.

1.2 Scope of this document

The scope of the hotel management system includes the following functionalities:

User Management: The system will manage user accounts for staff and customers.

Reservation Management: The system will manage hotel room reservations, including the ability to check room availability, reserve rooms, modify and cancel reservations, and track payment and deposit information.

Room Management: The system will manage hotel rooms, including the ability to assign rooms to guests, track room occupancy, manage room cleaning and maintenance schedules, and track room availability.

Billing and Payment Management: The system will manage guest billing and payment information, including the ability to generate invoices, track payments, and generate financial reports.

Inventory Management: The system will manage hotel inventory, including the ability to track inventory levels, reorder inventory, and generate reports on inventory usage and availability. Reporting: The system will generate various reports, including financial reports, occupancy reports, inventory reports, and customer feedback reports. The system will be designed to be scalable, secure, and easy to use. The system will be compatible with various hardware and software configurations, including different operating systems, browsers, and devices. The system will be able to handle a large number of users and transactions, with fast response times and minimal downtime.

1.3 Overview

The hotel management system is a software system that will be used to manage all aspects of a hotel's operations, including reservations, room management, billing and payments, inventory, and reporting. The system will provide a user-friendly interface that will allow hotel staff to manage their operations efficiently and effectively.

2 General Description

The hotel management system is a software system designed to manage all aspects of a hotel's operations. The system will provide a user-friendly interface for managing reservations, room assignments, billing and payments, inventory, and reporting. The system will be used by hotel staff to efficiently and effectively manage their operations. The system will be designed to manage user accounts for staff and customers, with staff accounts having access to the

management functions of the system and customer accounts having limited access to view and manage their own reservations.

3 Functional Requirements

- **User Management:** The system must be able to manage user accounts for staff and customers. Staff accounts must be able to access the management functions of the system, while customer accounts will have limited access to view and manage their own reservations.
- **Reservation Management:** The system must be able to manage hotel room reservations, including the ability to check room availability, reserve rooms, modify and cancel reservations, and track payment and deposit information. Room Management: The system must be able to manage hotel rooms, including the ability to assign rooms to guests, track room occupancy, manage room cleaning and maintenance schedules, and track room availability. Billing and Payment Management: The system must be able to manage guest billing and payment information, including the ability to generate invoices, track payments, and generate financial reports. Inventory Management: The system must be able to manage hotel inventory, including the ability to track inventory levels, reorder inventory, and generate reports on inventory usage and availability. Reporting: The system must be able to generate various reports, including financial reports, occupancy reports, inventory reports, and customer feedback reports.

4 Interface Requirements

The system shall provide a user-friendly interface for hotel staff and customers to access and manage their information. Hardware Interface: The system shall be compatible with different hardware configurations, including desktops, laptops, tablets, and smartphones. Software Interface: The system shall be compatible with different operating systems, including Windows, Mac OS, and Linux, and with different web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. Communication Interface: The system

shall use standard communication protocols such as HTTP, HTTPS, and TCP/IP for transmitting data over the internet. External Interface: The system shall be able to integrate with third-party systems such as payment gateways and inventory management systems.

5 Performance Requirements

Response Time: The system shall provide a response time of less than 2 seconds for any user action.

Availability: The system shall be available 24/7 with a maximum of 1 hour of downtime per month for maintenance purposes.

Capacity: The system shall be able to handle a minimum of 1000 concurrent users and a minimum of 500 room reservations per day.

Security: The system shall be designed to prevent unauthorized access to data and provide protection against hacking attempts.

6 Design Constraints

Compatibility: The hotel management system must be compatible with various hardware and software configurations, including different operating systems, browsers, and devices.

Security: The system must be designed with security in mind, including measures to prevent unauthorized access, protect against hacking attempts, and ensure the privacy of customer data.

Scalability: The system must be designed to handle a large number of users and transactions, with the ability to scale up or down as needed.

Usability: The system must be easy to use, with a user-friendly interface that can be easily navigated by hotel staff and guests.

Performance: The system must be able to respond quickly to user actions, with fast response times and minimal downtime.

Accessibility: The system must be accessible to people with disabilities, including support for screen readers and keyboard navigation.

Legal Compliance: The system must comply with relevant legal regulations and industry standards, such as the General Data Protection Regulation (GDPR) and Payment Card Industry Data Security Standard (PCI DSS).

Cost: The system must be designed within a reasonable budget, with cost-effective solutions that do not compromise on functionality, security, or performance.

7 Non-Functional Attributes

Usability: The system must be easy to use and navigate, with an intuitive user interface that allows staff to quickly access the information they need.

Performance: The system must be able to handle a large number of users and transactions, with fast response times and minimal downtime.

Security: The system must be secure, with measures in place to prevent unauthorized access and protect sensitive information such as guest billing and payment data.

Scalability: The system must be able to scale to meet the needs of a growing hotel, with the ability to add new rooms, staff members, and inventory as needed.

Compatibility: The system must be compatible with various hardware and software configurations, including different operating systems, browsers, and devices.

8 Preliminary Schedule and Budget

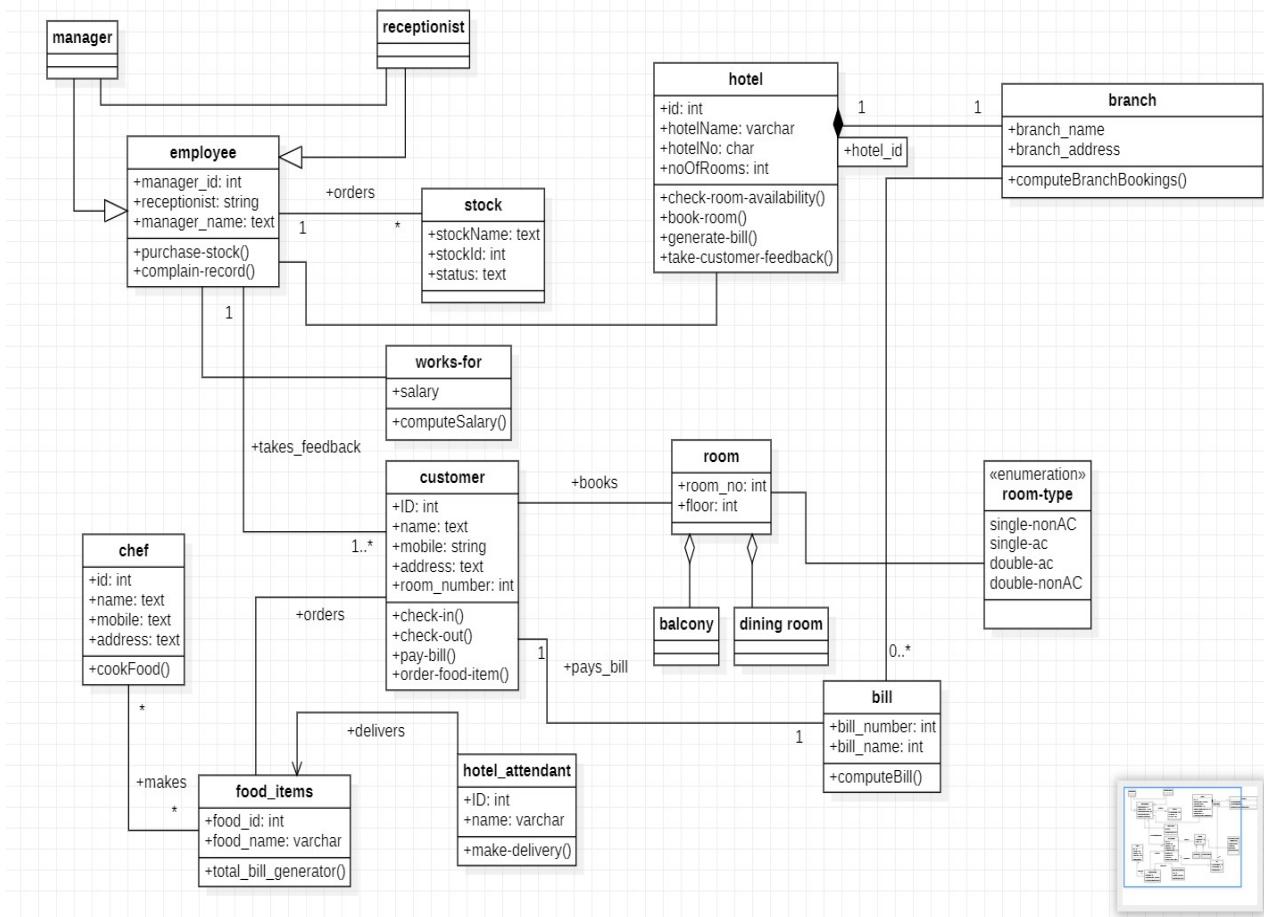
Preliminary Schedule: The development of a hotel management system can take several months to a year, depending on the complexity of the project and the size of the development team. The following is a tentative schedule for the project: Requirements Gathering: 2-4 weeks System Design: 4-6 weeks Development: 16-20 weeks Testing and Quality Assurance: 4-6 weeks Deployment and User Training: 2-4 weeks Total Estimated Time: 28-40 weeks Note that this is a preliminary schedule and can vary depending on the specific requirements and scope of the project.

Preliminary Budget: The budget for the development of a hotel management system can vary significantly depending on various factors, such as the complexity of the system, the size of the development team, and the location of the development team. The following is a rough estimate of the budget required for the project:

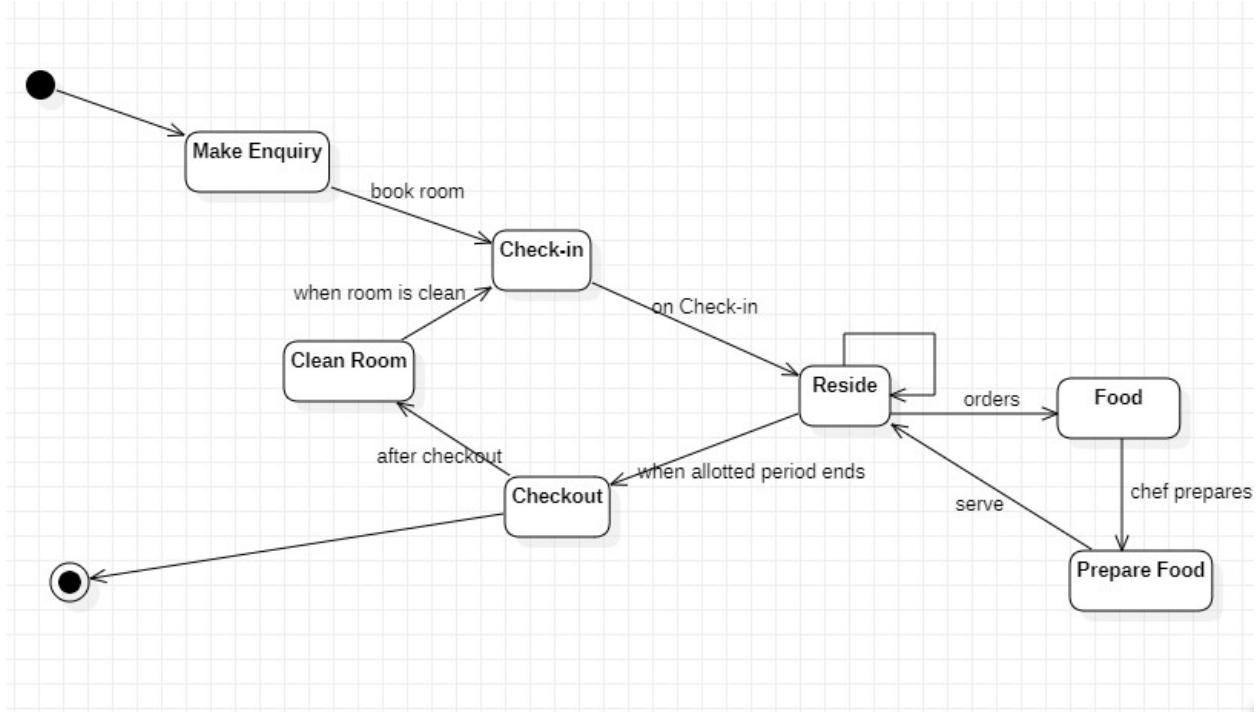
- Requirements Gathering: \$10,000 - \$20,000
- System Design: \$20,000 - \$40,000
- Development: \$100,000 - \$200,000
- Testing and Quality Assurance: \$20,000 - \$40,000
- Deployment and User Training: \$10,000 - \$20,000

Total Estimated Budget: \$160,000 - \$320,000 Note that this is a rough estimate and the actual budget can vary depending on the specific requirements and scope of the project. It is recommended to consult with a software development company or a project manager to get a more accurate estimate of the budget required for the project

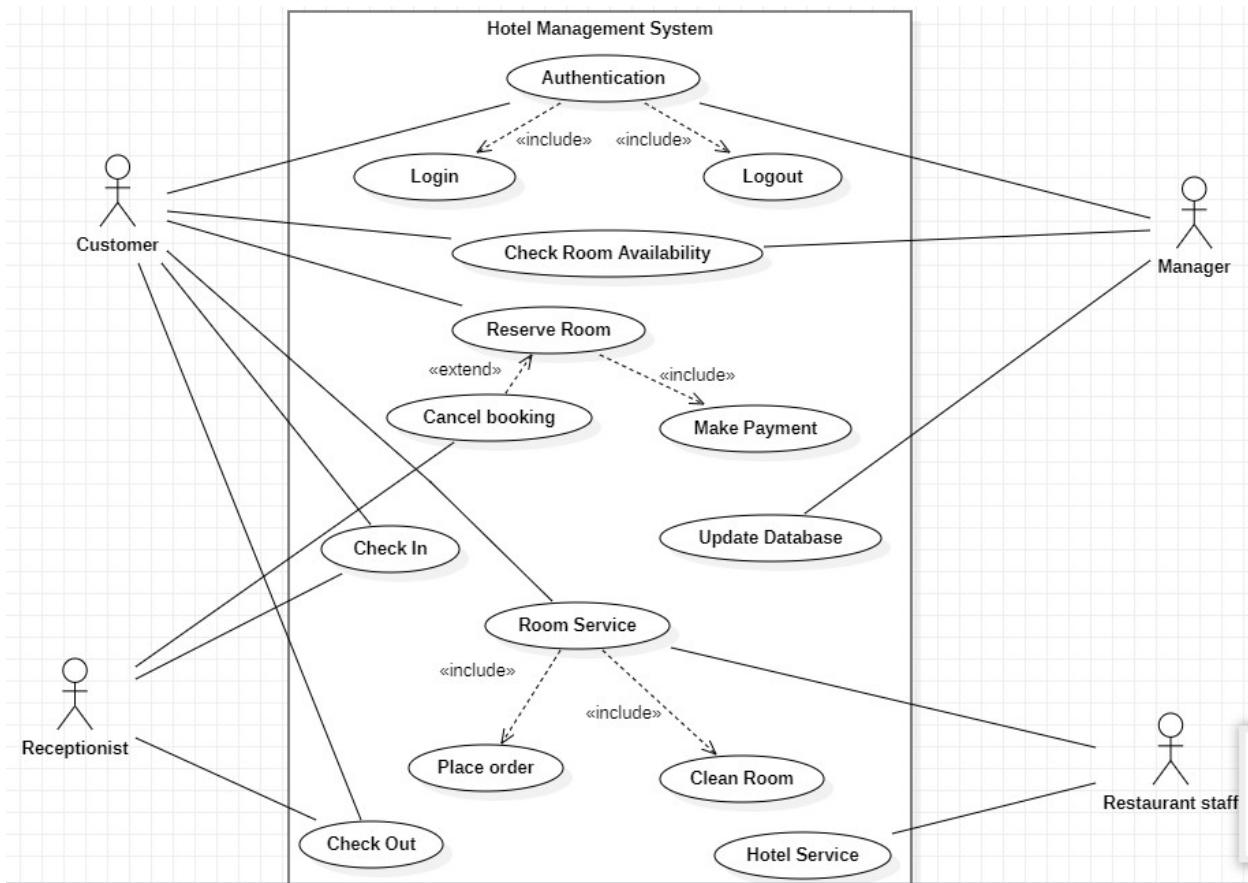
Class Diagram



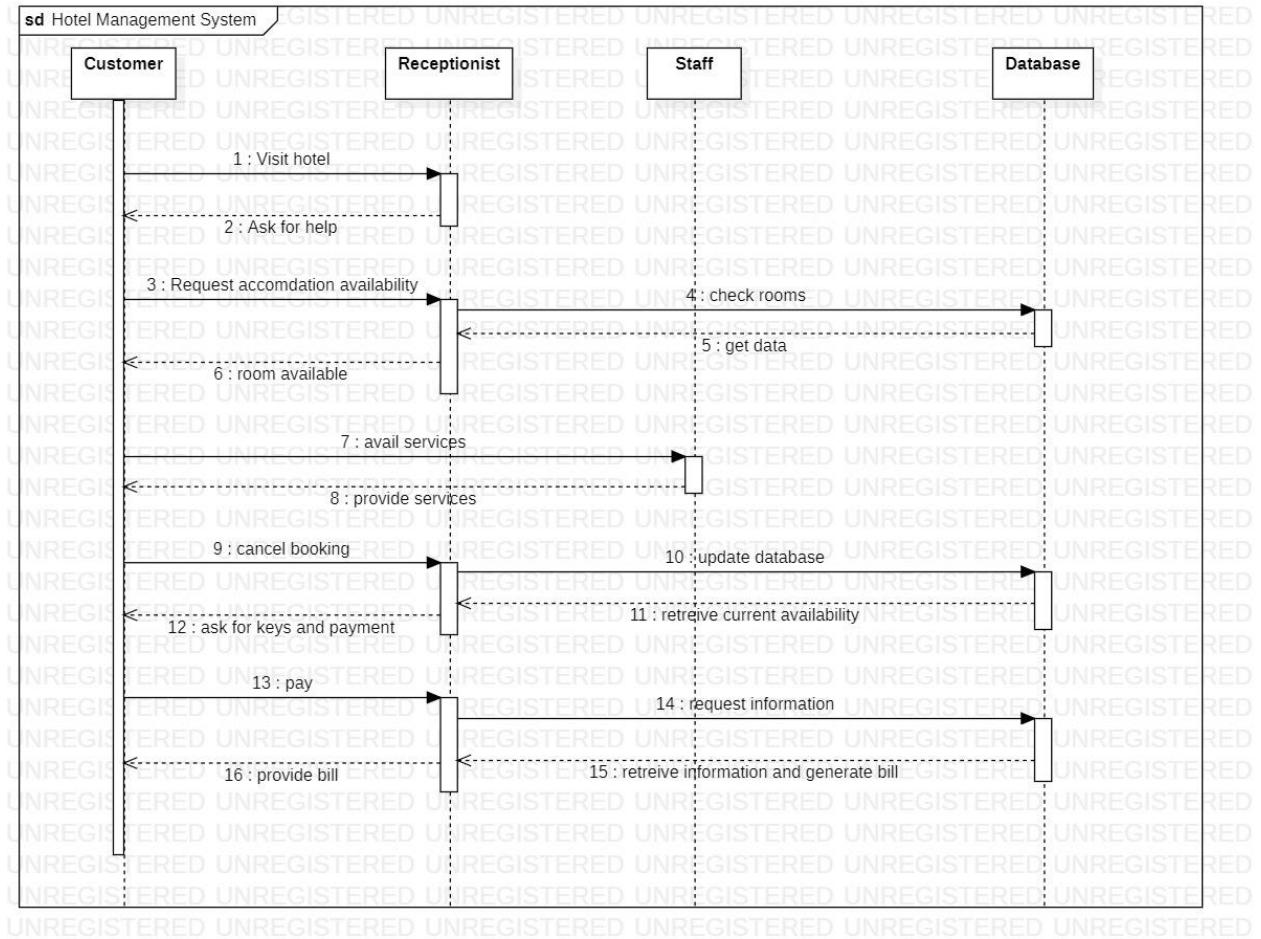
State Diagram



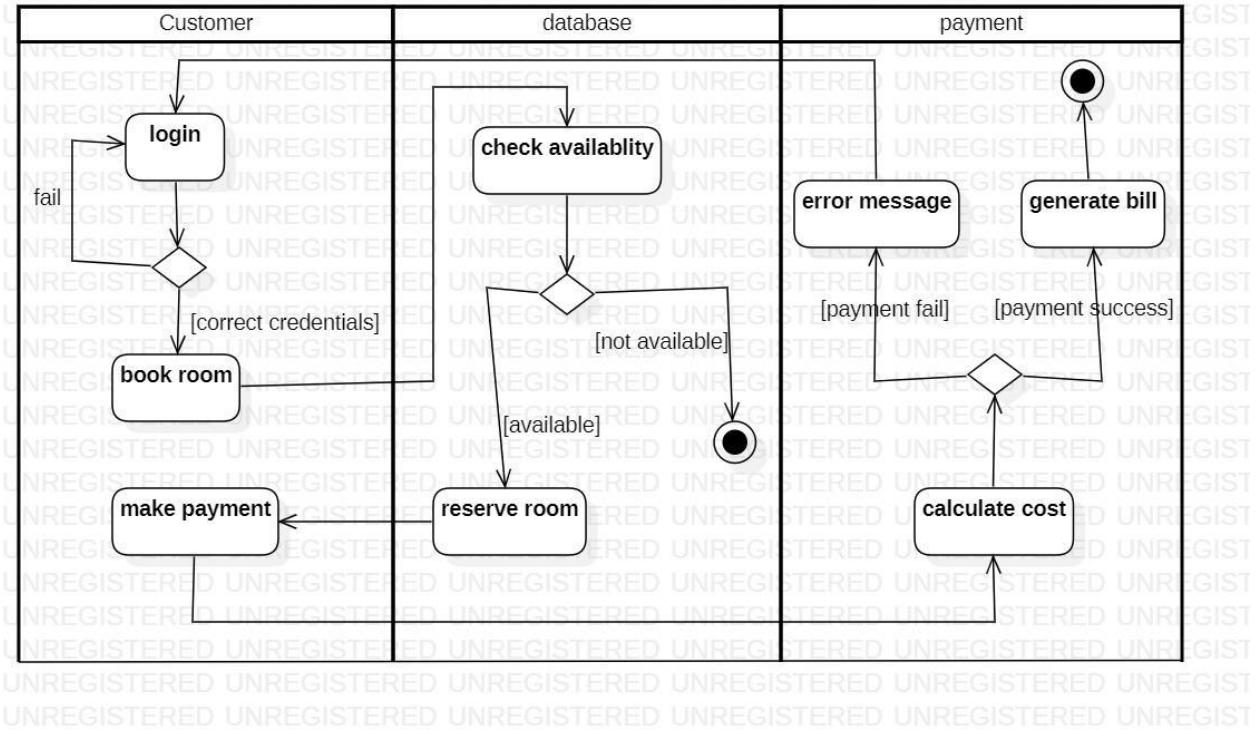
Use Case Diagram



Sequence Diagram



Activity Diagram



2. Credit Card Processing

Problem Statement:

Many small and medium-sized businesses struggle to process credit card payments efficiently due to the lack of a reliable and secure credit card processing system. This often results in lost sales and dissatisfied customers. The problem is exacerbated by the ever-increasing complexity of payment regulations and the need for businesses to comply with security standards such as PCI-DSS. Therefore, there is a need for a credit card processing system that is easy to use, secure, and compliant with the latest payment regulations and security standards. The system should be accessible to businesses of all sizes, and it should be able to handle various payment methods, including credit and debit cards, mobile payments, and e-wallets. The system should also provide real-time reporting and analytics to help businesses optimize their payment processing operations.

Software Requirement Specification (SRS)

1 Introduction

1.1 Purpose of this document

The purpose of a credit card processing system is to enable businesses to accept and process credit and debit card payments from their customers in a secure and efficient manner. The system must ensure that transactions are processed accurately and quickly, while also providing the necessary security measures to protect customer data.

1.2 Scope of this document

The credit card processing system will be designed to meet the needs of businesses of all sizes, from small retailers to large corporations. The system will be designed to be user-friendly and customizable to meet the specific requirements of each business.

1.3 Overview

A credit card processing system is a software application that enables businesses to accept and process credit and debit card payments from their customers. The system works by capturing the card information, verifying the transaction, and transferring funds from the customer's account to the merchant's account. The credit card processing system is essential for any business that accepts electronic payments and needs to ensure secure and efficient processing of transactions.

2 General Description

The credit card processing system will be a web-based application that will allow businesses to process credit and debit card transactions securely and efficiently. The system will capture the card information and verify the transaction through a payment gateway that communicates with the

issuing bank. The system will also provide real-time reporting and analytics to help businesses manage their finances and track their sales.

3 Functional Requirements

- **Secure Card Data Storage:** The system shall securely store customer card data in compliance with the Payment Card Industry Data Security Standard (PCI DSS).
- **Payment Processing:** The system shall be able to process credit and debit card payments securely and efficiently through a payment gateway.
- **Refunds and Voiding Transactions:** The system shall allow businesses to issue refunds and void transactions as needed.
- **Transaction Reporting:** The system shall provide real-time reporting and analytics for transactions processed through the system.
- **Customizable Settings:** The system shall allow businesses to customize their settings for card processing, including transaction fees, tax rates, and currency settings.

4 Interface Requirements

- **User Interface:** The system shall provide a user-friendly interface for business owners and employees to access and manage their transactions and settings.
- **Hardware Interface:** The system shall be compatible with different hardware configurations, including desktops, laptops, tablets, and smartphones.
- **Software Interface:** The system shall be compatible with different operating systems, including Windows, Mac OS, and Linux, and with different web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
- **Communication Interface:** The system shall use standard communication protocols such as HTTP, HTTPS, and TCP/IP for transmitting data over the internet.

5 Performance Requirements

- **Response Time:** The system shall provide a response time of less than 2 seconds for any user action.
- **Availability:** The system shall be available 24/7 with a maximum of 1 hour of downtime per month for maintenance purposes. Capacity: The system shall be able to handle a minimum of 1000 concurrent users and a minimum of 500 transactions per day.

6 Design Constraints

- **Compatibility with Different Operating Systems and Devices:** The system must be compatible with various operating systems, devices, and web browsers to ensure that it can be used by businesses of all sizes.
- **Scalability:** The system must be designed to handle a large volume of transactions and users, with the ability to scale up as the business grows.
- **Reliability and Availability:** The system must be reliable and available 24/7, with minimal downtime and quick recovery times in case of system failure.
- **Cost-Effective:** The system must be cost-effective, with affordable pricing models that can be customized to meet the specific needs of each business.

7 Non-Functional Attributes

- **Security:** The system shall be designed with security in mind, including measures to prevent unauthorized access and protect against hacking attempts.
- **Reliability:** The system shall be reliable, with minimal downtime and quick recovery times in case of system failure.
- **Usability:** The system shall be user-friendly, with a simple and intuitive interface that can be easily navigated by business owners and employees.

- **Scalability:** The system shall be scalable, with the ability to handle a large number of transactions and users as the business grows.
- **Accessibility:** The system shall be accessible to people with disabilities, including support for screen readers and keyboard navigation.

8 Preliminary Schedule and Budget

The development of a credit card processing system can take several months, depending on the complexity of the project and the size of the development team.

The following is a tentative schedule for the project:

Requirements Gathering: 2-4 weeks

System Design: 4-6 weeks

Development: 16-20 weeks

Testing and Quality Assurance: 4-6 weeks

Deployment and User

The budget for a credit card processing system can vary depending on several factors such as the complexity of the system, the size of the development team, and the technologies used. Here is a rough estimate of the preliminary budget for the project:

Requirements Gathering: \$5,000 - \$10,000

System Design: \$10,000 - \$20,000

Development: \$50,000 - \$100,000

Testing and Quality Assurance: \$10,000 - \$20,000

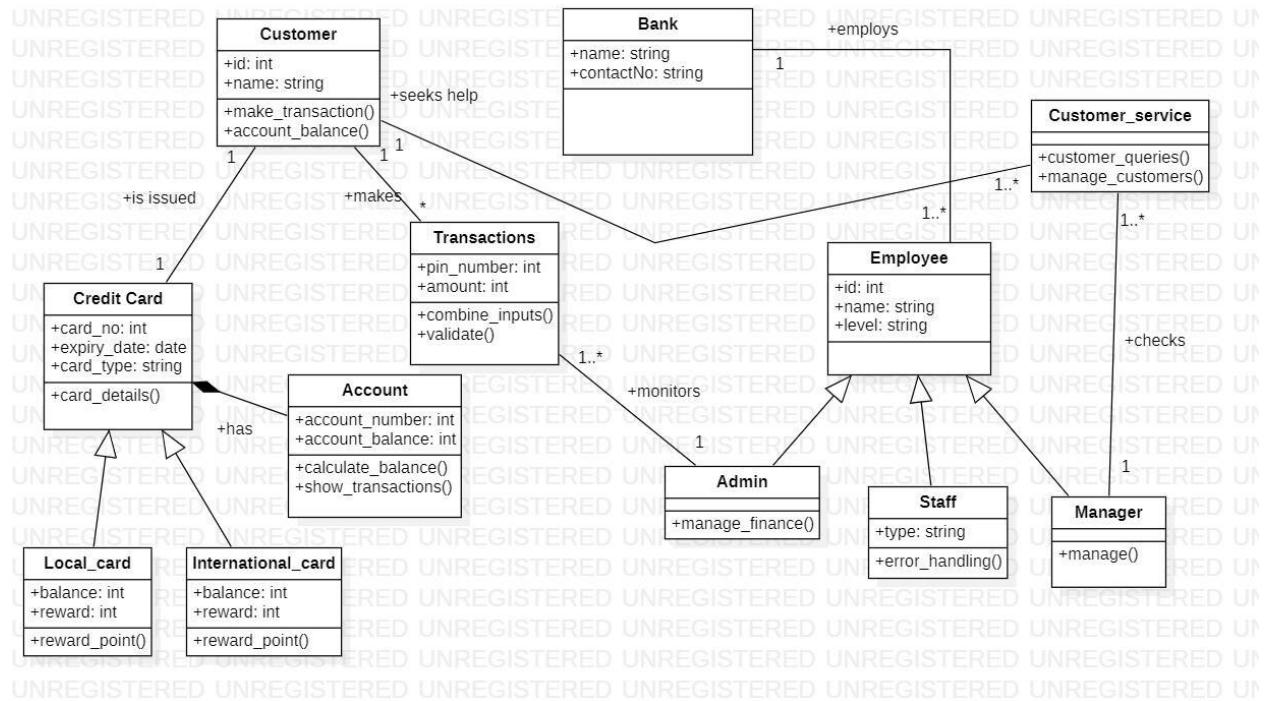
Deployment and User Training: \$5,000 - \$10,000

Maintenance and Support (annual): \$10,000 - \$20,000

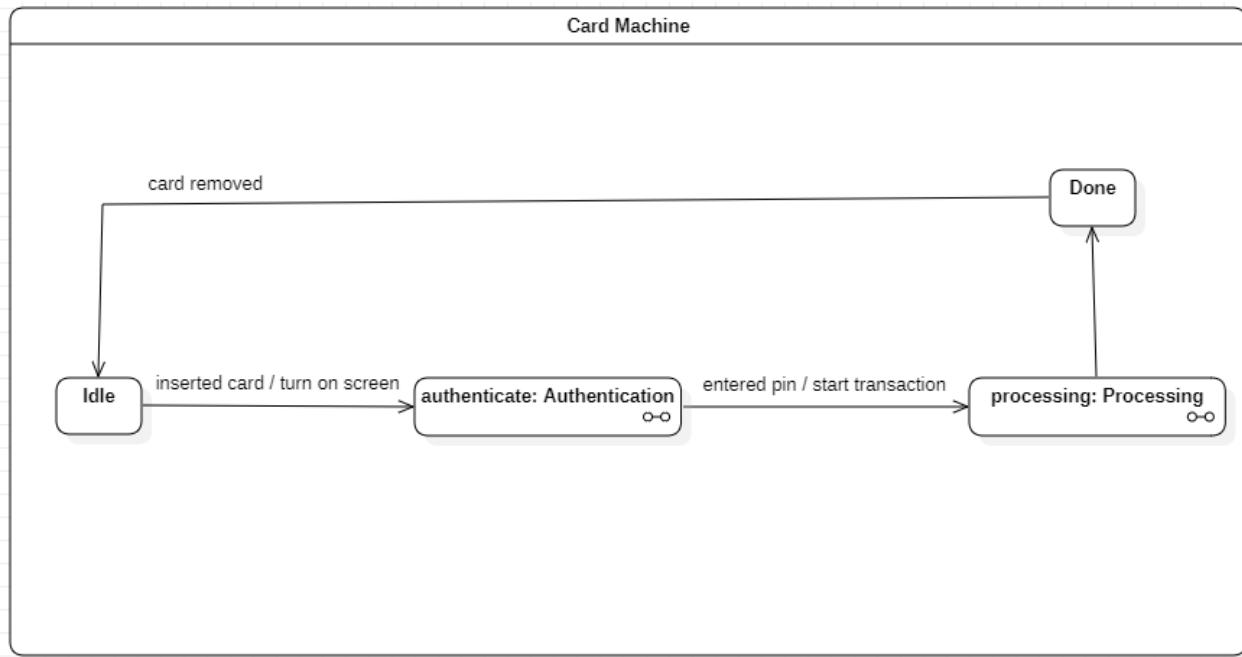
Total preliminary budget: \$90,000 - \$180,000

Note that this is just an estimate, and the actual budget may be higher or lower depending on the specific needs and requirements of the project.

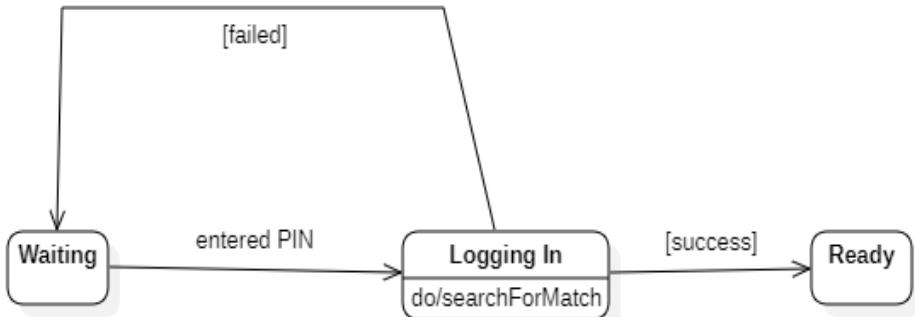
Class Diagram



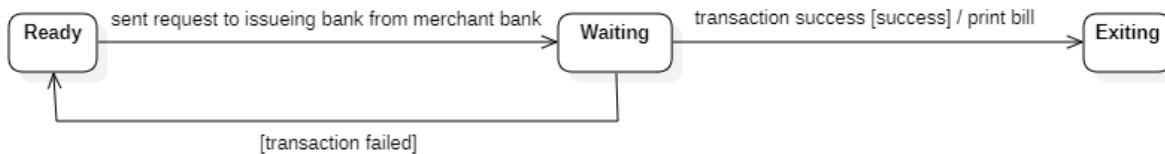
State Diagram



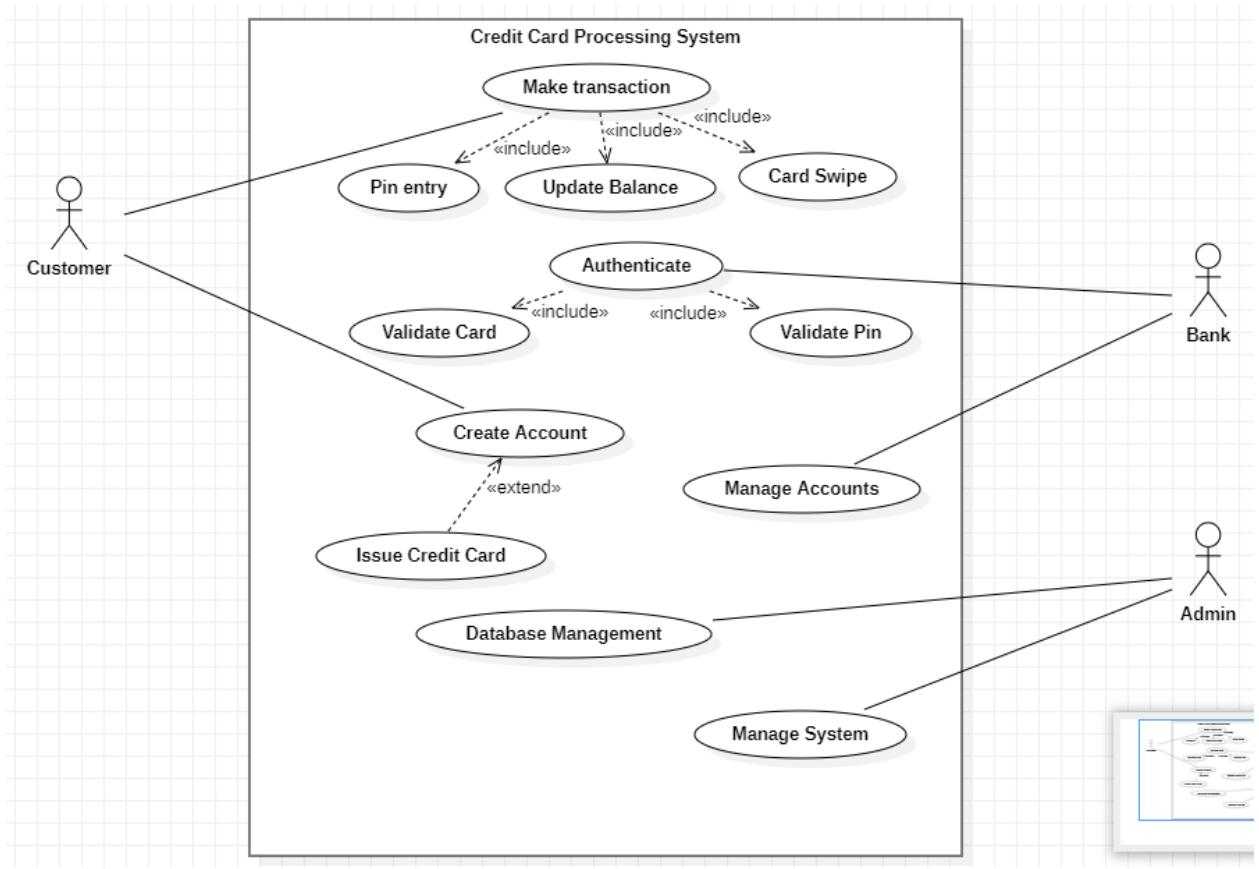
Authentication



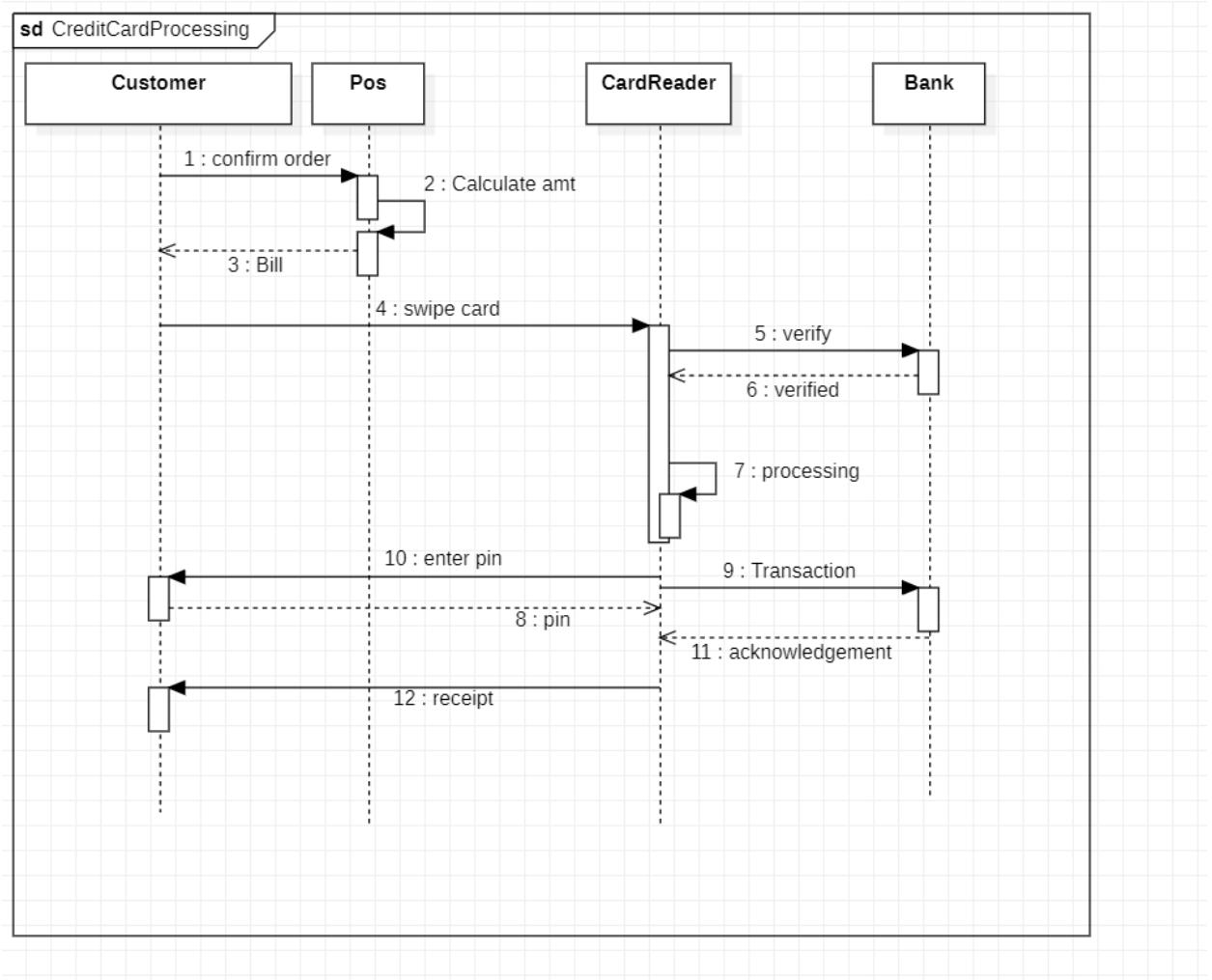
Processing



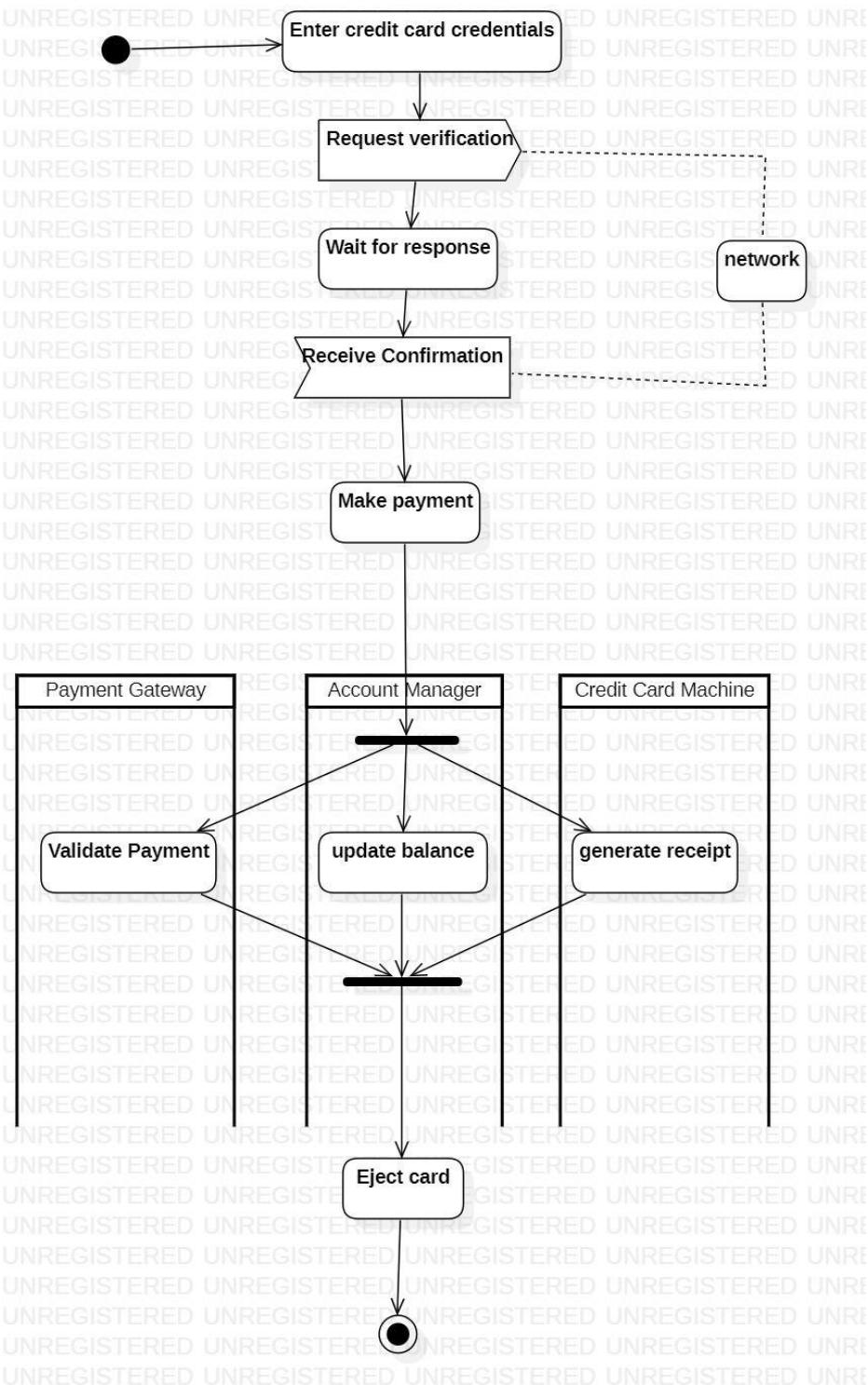
Use Case Diagram



Sequence Diagram



Activity Diagram



3. Library Management System

Problem Statement:

The Library Management System is a software application designed to manage and organize the day-to-day operations of a library. The manual system of managing library activities is time-consuming and error-prone. It can result in inefficiencies such as lost books, late returns, and difficulty in tracking book circulation. The Library Management System aims to address these problems by automating the library management process.

Software Requirement Specification (SRS)

1 Introduction

1.1 Purpose of this document

The purpose of this document is to define the requirements and specifications for the development of a Library Management System. The system should enable library staff to perform their duties efficiently and effectively, as well as provide library users with a user-friendly interface for borrowing, returning, and searching for books.

1.2 Scope of this document

The Library Management System will include the following features:

- User registration and authentication
- Cataloging of books, CDs, DVDs, and other library materials
- Borrowing and returning of library materials
- Search functionality for library materials

- Record keeping of library materials and transactions
- Reporting and analytics
- System administration functionality

1.3 Overview

The Library Management System is an online application that allows library staff to manage and organize the day-to-day operations of the library. It provides a user-friendly interface for library users to browse the catalog, search for books, and borrow and return items. The system will also generate reports and analytics to help library staff make informed decisions.

2 General Description

The Library Management System will consist of a web-based user interface for library staff and a separate user interface for library users. The system will be built using modern web development technologies such as HTML, CSS, JavaScript, and a backend language such as Python or PHP. The system will be hosted on a web server and accessed through a web browser.

3 Functional Requirements

Book Management:

- Add new books to the library's collection
 - Edit existing book information
 - Delete books from the library's collection
 - Search for books using various criteria (e.g., title, author, genre)
- Member Management:
- Add new members to the library's database

- Edit member information
 - Delete members from the database
 - Search for members using various criteria (e.g., name, address, membership ID)
- Circulation
- Check out books to members
 - Record book returns
 - Record book renewals
 - Track book due dates and send reminders
 - Calculate and record fines for overdue books
- Search Functionality
- Search for books using various criteria (e.g., title, author, genre)
 - Search for members using various criteria (e.g., name, address, membership ID)
- Reporting
- Generate reports on book circulation
 - Generate reports on overdue books
 - Generate reports on fines
- Security
- Control access to the system using login credentials
 - Protect user data by encrypting sensitive information

4 Interface Requirements

The Library Management System will have two interfaces:

- A web-based user interface for library staff

- A separate user interface for library users

5 Performance Requirements

- System Availability
 - The system should be available 24/7
 - The system should have a minimum downtime of 99%
- Response Time
 - The system should respond to user requests within 3 seconds
 - The system should handle 100 simultaneous users

6 Design Constraints

- **Technology:** The library management system will be designed using modern web technologies, including HTML, CSS, JavaScript, and PHP.
- **Database:** The system will use a relational database management system to store book and member information and track circulation data.
- **Security:** The system will be designed to ensure the security of user data by encrypting sensitive information and controlling access to the system using login credentials.

7 Non-Functional Attributes

- **Usability:** The system should be easy to use, with an intuitive user interface that is easy to navigate.
- **Reliability:** The system should be reliable and available at all times, with minimal

downtime and maximum uptime.

- **Performance:** The system should be responsive, with fast loading times and minimal lag or delay.
- **Scalability:** The system should be scalable, with the ability to handle increased user loads and data volumes.
- **Security:** The system should be secure, with measures in place to protect user data and prevent unauthorized access or breaches.
- **Accessibility:** The system should be accessible, with support for assistive technologies and compliance with accessibility guidelines.
- **Compatibility:** The system should be compatible with a range of web browsers and devices to ensure maximum accessibility and usability.
- **Maintainability:** The system should be maintainable, with a clear and well-documented codebase and easy-to-use tools for maintenance and updates.
- **Portability:** The system should be portable, with the ability to run on a range of platforms and environments.
- **Interoperability:** The system should be interoperable, with the ability to integrate with other systems and tools used by the library staff

8 Preliminary Schedule and Budget

Requirement Gathering and Analysis: 2 weeks

System Design: 4 weeks

Development and Testing: 12 weeks

Deployment and User Acceptance Testing: 2 weeks

Training and Documentation: 1 week

Total Estimated Time: 21 weeks

Budget:

Salaries: \$150,000

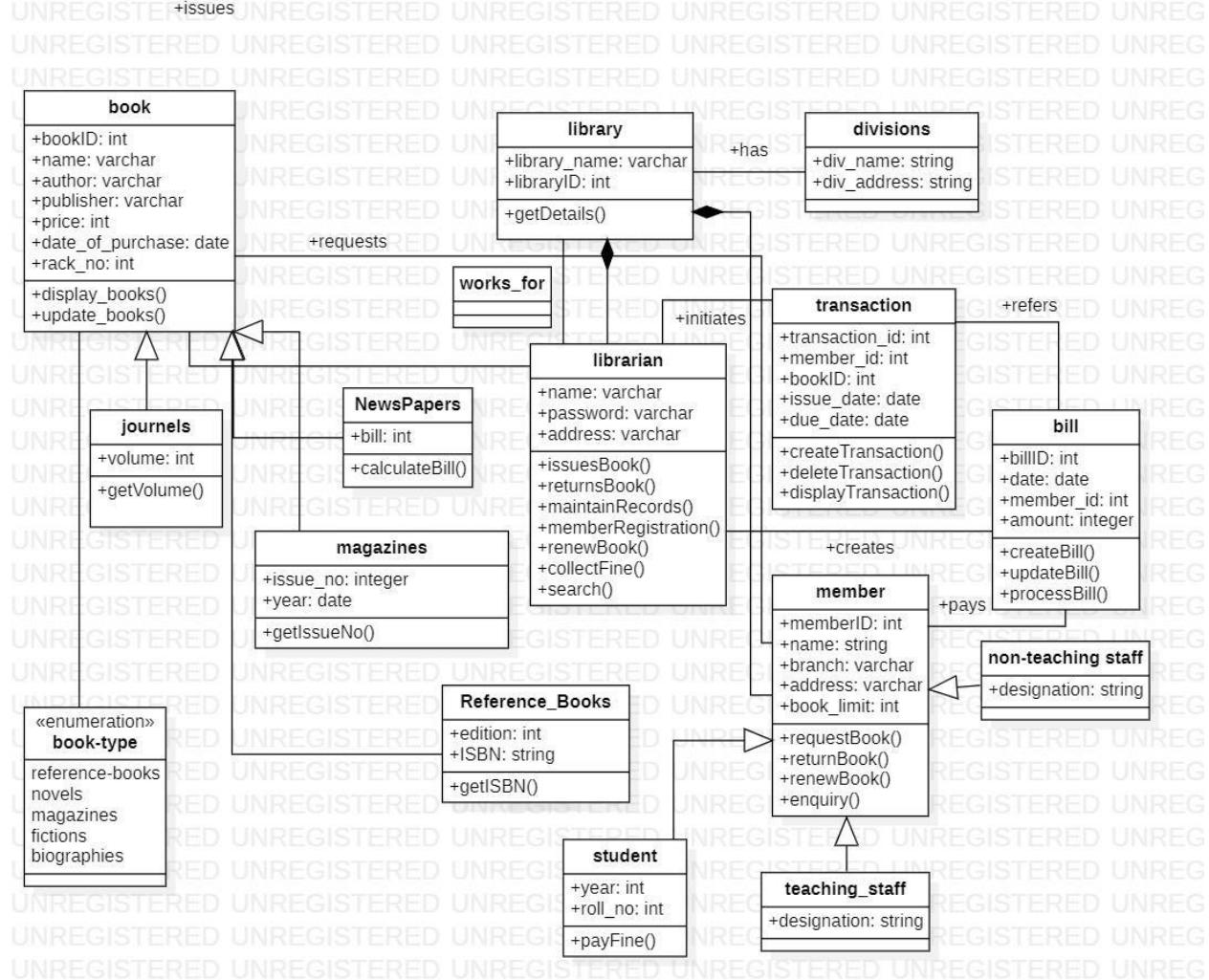
Hardware and Software: \$50,000

Database License: \$10,000

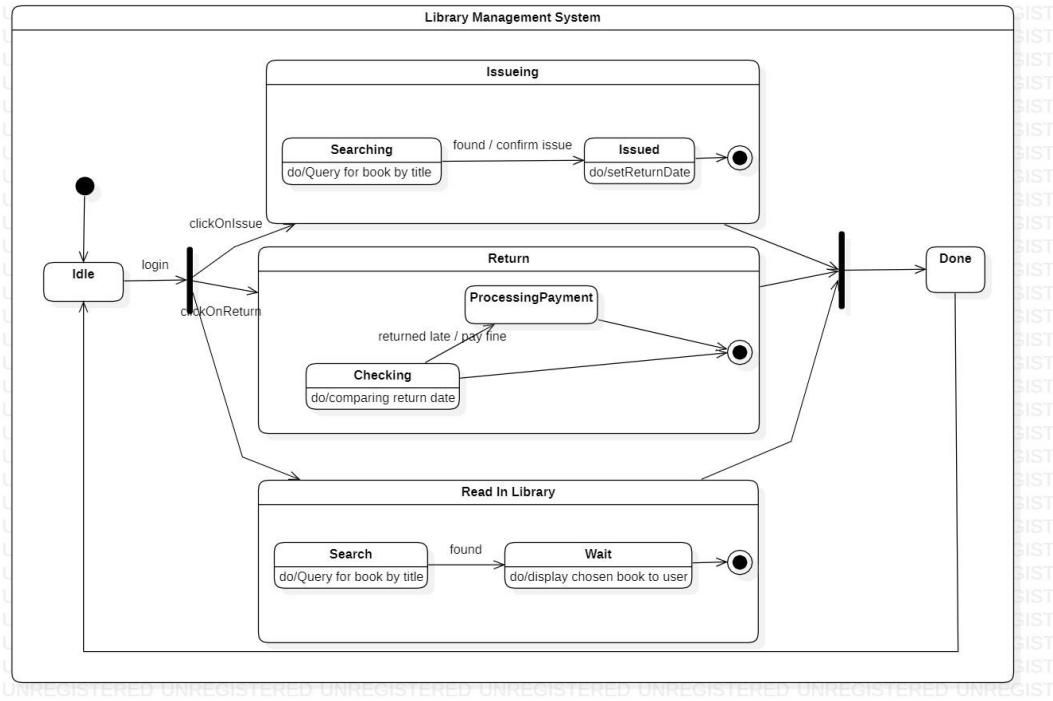
Miscellaneous Expenses: \$10,000

Total Estimated Budget: \$220,00

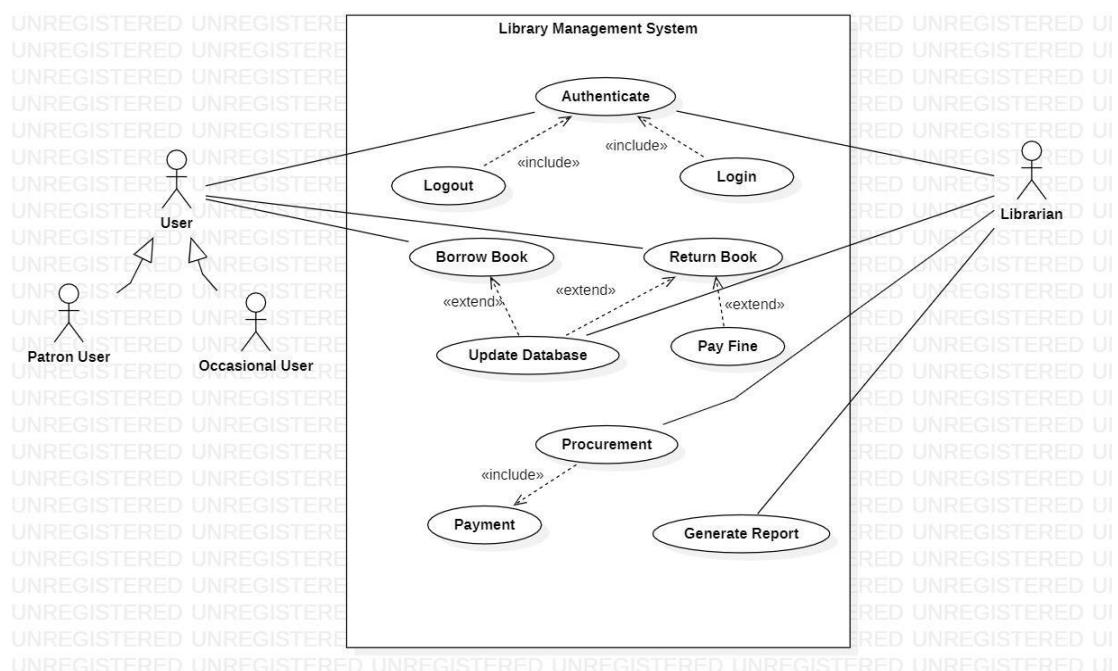
Class Diagram



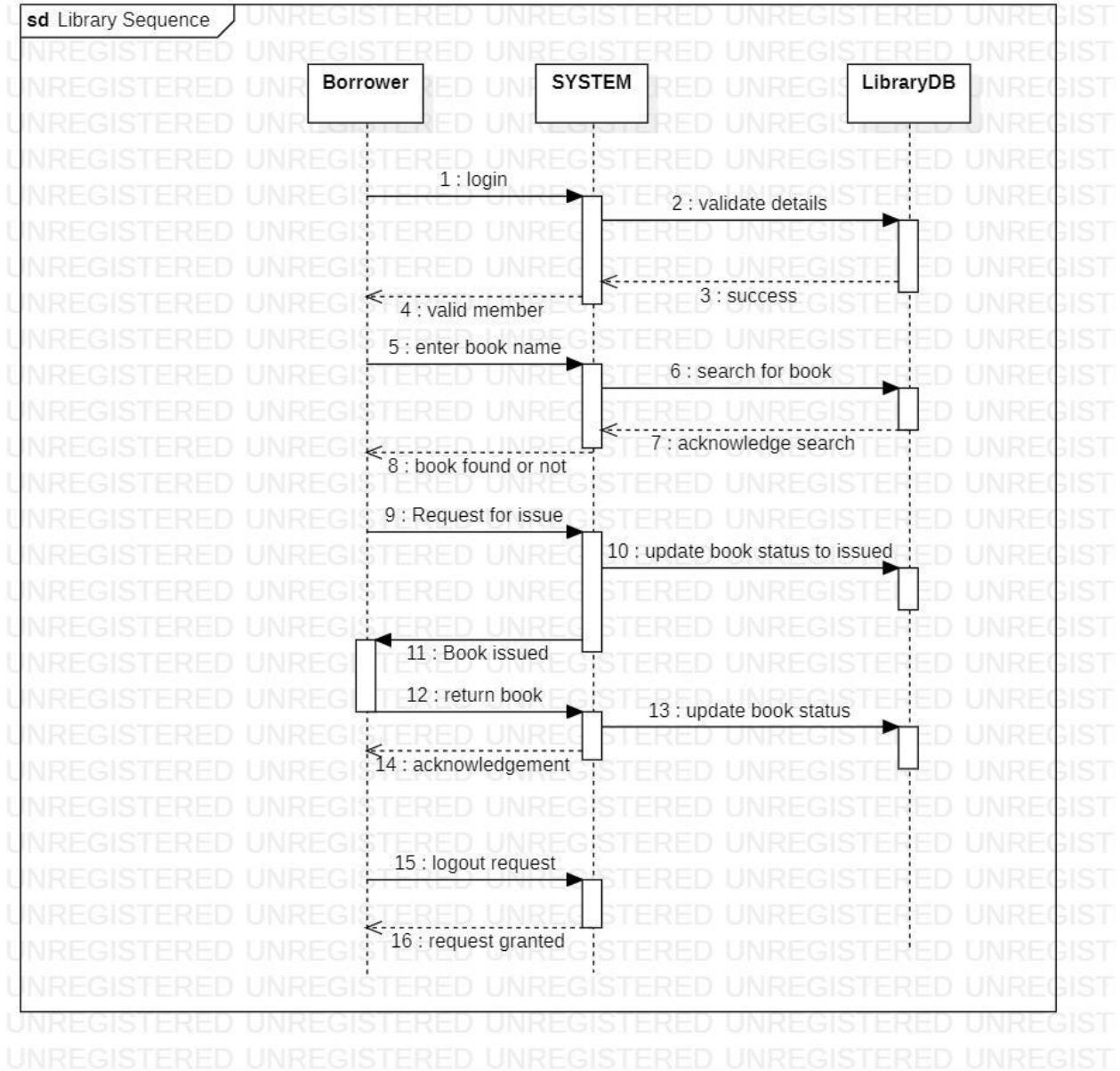
State Diagram



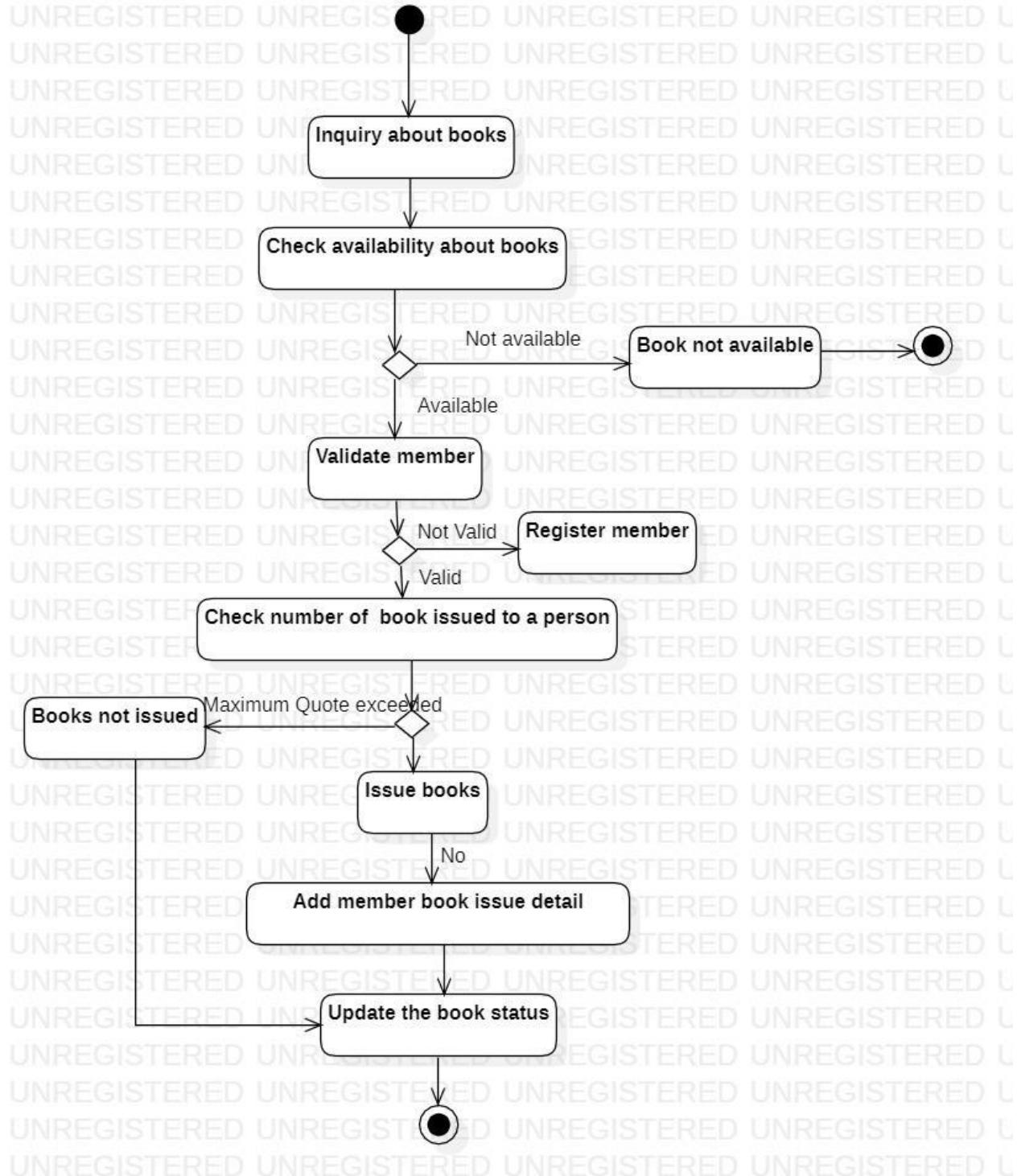
Use Case Diagram



SequenceDiagram



Activity Diagram



4. Stock Maintenance System

Problem Statement:

The client currently manages their inventory manually, which is time-consuming and prone to errors. The client needs a system to automate inventory management processes, reduce errors, and provide real-time inventory information. The system should enable the client to manage their inventory efficiently and make informed decisions based on accurate data. Therefore, there is a need for a Stock Maintenance System to automate inventory management processes and provide real-time inventory information.

Software Requirement Specification(SRS)

1. Introduction:

1.1. Purpose of this Document:

The purpose of the Stock Maintenance System is to provide an automated solution for managing inventory for a business. The system will help the client save time and effort in managing inventory, and provide accurate data for decision-making. The system aims to automate inventory management processes, reduce errors, and provide real-time inventory information. Additionally, the system should enable the client to manage their inventory efficiently, keep track of stock levels, and generate reports on inventory status, sales, and purchase history. Ultimately, the purpose of the Stock Maintenance System is to improve the overall efficiency of inventory management for the client's business.

1.2. Scope of this document

The scope of the Stock Maintenance System is to manage inventory for the client's business. The system will include functionalities for inventory management such as adding new items, updating stock levels, and generating reports. The system will also provide real-time inventory information to the client, enabling them to make informed decisions. The Stock Maintenance System will be designed to handle a large amount of data and will be able to manage multiple locations of inventory. The system will have a user-friendly interface that allows for easy navigation and will be developed using the latest web technologies. However, the Stock Maintenance System will not include functionalities for financial management or point of sale.

1.3. Overview –

The Stock Maintenance System is a web-based application developed using the latest web technologies. The system will have a userfriendly interface, making it easy for the client to use. The system will automate inventory management processes, reduce errors, and provide realtime inventory information. The system will be developed using an agile methodology, allowing for iterative development and quick deployment of new features. The Stock Maintenance System will enable the client to manage their inventory efficiently, keep track of stock levels, and generate reports on inventory status, sales, and purchase history. Ultimately, the Stock Maintenance System aims to improve the overall efficiency of inventory management for the client's business.

2 General description:

The Stock Maintenance System is an online platform designed to streamline the inventory management processes of businesses. It provides a centralized inventory management system accessible through a web browser, eliminating the need for manual record keeping and reducing errors. The system enables businesses to manage their inventory efficiently, keep track of stock levels, and generate reports on inventory status, sales, and purchase history. The Stock Maintenance System is scalable and can handle a large volume of data. It can manage inventory across multiple locations, making it suitable for businesses with warehouses or stores in different locations. The system is customizable, allowing businesses to tailor the platform to their specific needs. The system will be developed using the latest web technologies, ensuring that it is secure, reliable, and easy to use. The development process will follow an agile methodology, allowing for quick and iterative deployment of new features. The system will be thoroughly tested to ensure that it meets the client's requirements and is free of bugs. The Stock Maintenance System will be deployed on a cloud server, making it accessible from anywhere with an internet connection. The system will have a user-friendly interface, making it easy for users to add new items, update stock levels, and generate reports. However, the Stock Maintenance System will not include functionalities for financial management or point of sale.

3 Functional Requirements:

- **User Management:** The system must be able to manage user accounts for staff and customers. Staff accounts must be able to access the management functions of the system, while customer accounts will have limited access to view and manage their own reservations.
- **Reservation Management:** The system must be able to manage hotel room reservations, including the ability to check room availability, reserve rooms, modify and cancel reservations, and track payment and deposit information.
- **Room Management:** The system must be able to manage hotel rooms, including the ability to assign rooms to guests, track room occupancy, manage room cleaning and maintenance schedules, and track room availability.
- **Billing and Payment Management:** The system must be able to manage guest billing and payment information, including the ability to generate invoices, track payments, and

generate financial reports. Inventory Management: The system must be able to manage hotel inventory, including the ability to track inventory levels, reorder inventory, and generate reports on inventory usage and availability.

- **Reporting:** The system must be able to generate various reports, including financial reports, occupancy reports, inventory reports, and customer feedback reports.

4 Interface Requirements:

- **User Interface:**

The user interface of the system should be user-friendly, easy to navigate, and responsive. The interface should provide the user with the ability to manage inventory, purchase orders, sales orders, and generate reports.

- **Browser Compatibility:**

The system should be compatible with multiple browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

- **Mobile Responsiveness:**

The system should be responsive to mobile devices, enabling users to access the system from their smartphones and tablets.

- **API Integration:**

The system should be able to integrate with external APIs, such as suppliers, accounting software, and other inventory management systems.

- **Import and Export Data:**

The system should allow users to import and export data from external sources, such as spreadsheets.

- **Error Handling:**

The system should be able to handle errors and provide the user with feedback when an error occurs.

- **Language Localization:**

The system should be able to support multiple languages, enabling users to use the system in their preferred language.

- **Printing:**

The system should provide the user with the ability to print reports and documents.

- **Help and Support:**

The system should provide help and support features such as tutorials, user guides, and a knowledge base to assist the user in using the system.

5 Performance Requirements

Response Time:

The system should respond to user actions, such as adding or updating inventory items, creating purchase orders, or generating reports, within 3 seconds or less.

Concurrent Users:

The system should be able to handle concurrent users and maintain the same level of performance, regardless of the number of users accessing the system simultaneously.

Scalability:

The system should be scalable, allowing for the addition of more users and inventory items without impacting system performance.

Data Storage:

The system should be able to store large amounts of data, including inventory items, purchase orders, sales orders, and user information, without experiencing performance issues.

Backup and Recovery:

The system should include backup and recovery features to ensure data is safe and can be restored in the event of a system failure.

System Uptime:

The system should have a minimum uptime of 99.9%, ensuring that the system is available to users at all times.

System Monitoring:

The system should be monitored continuously for performance issues and system failures. The monitoring system should notify the administrator immediately in the event of an issue.

Load Testing:

The system should undergo load testing to ensure that it can handle the expected number of users and inventory items without experiencing performance issues.

Network Latency:

The system should be designed to minimize network latency, ensuring that users can access the system quickly and efficiently.

Compatibility:

The system should be compatible with different hardware and software configurations, ensuring that it can be used on a variety of devices without experiencing performance issues.

6 . Design Constraints

Response Time:

The system should respond to user actions, such as adding or updating inventory items, creating purchase orders, or generating reports, within 3 seconds or less.

Concurrent Users:

The system should be able to handle concurrent users and maintain the same level of performance, regardless of the number of users accessing the system simultaneously.

Scalability:

The system should be scalable, allowing for the addition of more users and inventory items without impacting system performance.

Data Storage:

The system should be able to store large amounts of data, including inventory items, purchase orders, sales orders, and user information, without experiencing performance issues.

Backup and Recovery:

The system should include backup and recovery features to ensure data is safe and can be restored in the event of a system failure.

System Uptime:

The system should have a minimum uptime of 99.9%, ensuring that the system is available to users at all times.

System Monitoring:

The system should be monitored continuously for performance issues and system failures. The monitoring system should notify the administrator immediately in the event of an issue.

Compatibility:

The system should be compatible with different hardware and software configurations, ensuring that it can be used on a variety of devices without experiencing performance issues.

7 Non-Functional Attributes:

Usability:

The system should be designed with usability in mind, ensuring that it is easy to use and intuitive for the end-users. This may include compliance with relevant usability standards and best practices.

Security:

The system should be designed to ensure data security and prevent unauthorized access. This may include encryption of data, secure authentication and authorization, and compliance with relevant security standards.

Performance:

The system should be designed to meet performance requirements, such as response time, concurrent user capacity, and system uptime. This may include load testing and optimization.

Reliability:

The system should be reliable, ensuring that it is available to users at all times and that data is stored and processed correctly. This may include backup and recovery features, error handling, and fault tolerance.

Scalability:

The system should be scalable, allowing for the addition of more users and inventory items without impacting system performance. This may include the ability to add more servers, load balancing, and clustering.

Maintainability:

The system should be designed to facilitate maintenance and support activities, such as updates, bug fixes, and user support. This may include modular architecture, version control, and documentation.

Portability:

The system should be designed to be portable, allowing it to be deployed on different hardware and software configurations. This may include compliance with relevant software standards, and use of platform-independent technologies.

Accessibility:

The system should be designed to be accessible to users with disabilities, ensuring compliance with relevant accessibility standards. This may include compliance with the Web Content Accessibility Guidelines (WCAG).

Compatibility:

The system should be compatible with different hardware and software configurations, ensuring that it can be used on a variety of devices without experiencing compatibility issues. This may include compliance with relevant software standards, and use of platform-independent technologies.

Interoperability:

The system should be designed to integrate with other systems or third-party software, such as accounting software or e-commerce platforms. This may include compliance with relevant integration standards and use of open APIs.

8 Preliminary Schedule and Budget:

Project Scope Definition:

Duration: 1 week

Budget: \$5,000

Requirements Gathering:

Duration: 2 weeks

Budget: \$10,000

System Design and Architecture:

Duration: 4 weeks

Budget: \$20,000

Development:

Duration: 12 weeks

Budget: \$60,000

Quality Assurance and Testing:

Duration: 4 weeks

Budget: \$20,000

Deployment and User Training:

Duration: 2 weeks

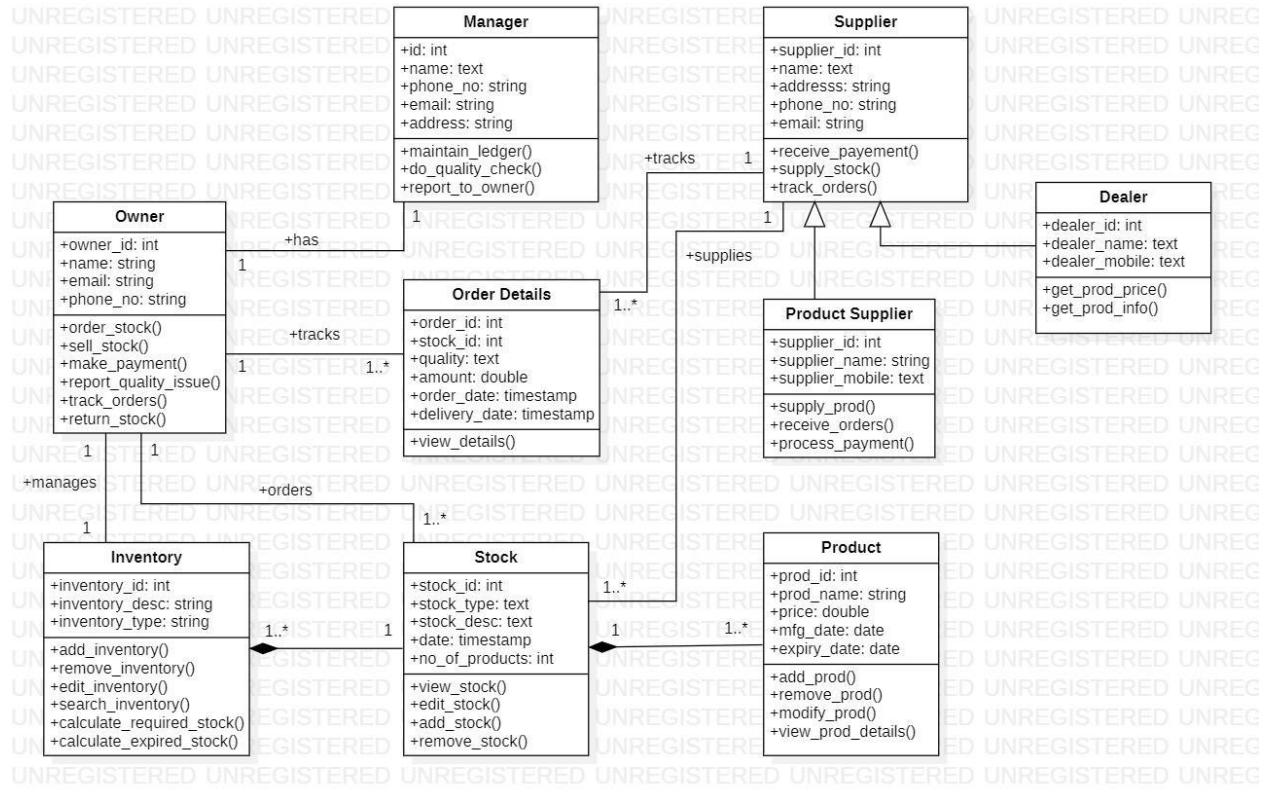
Budget: \$10,000

Total Duration: 25 weeks

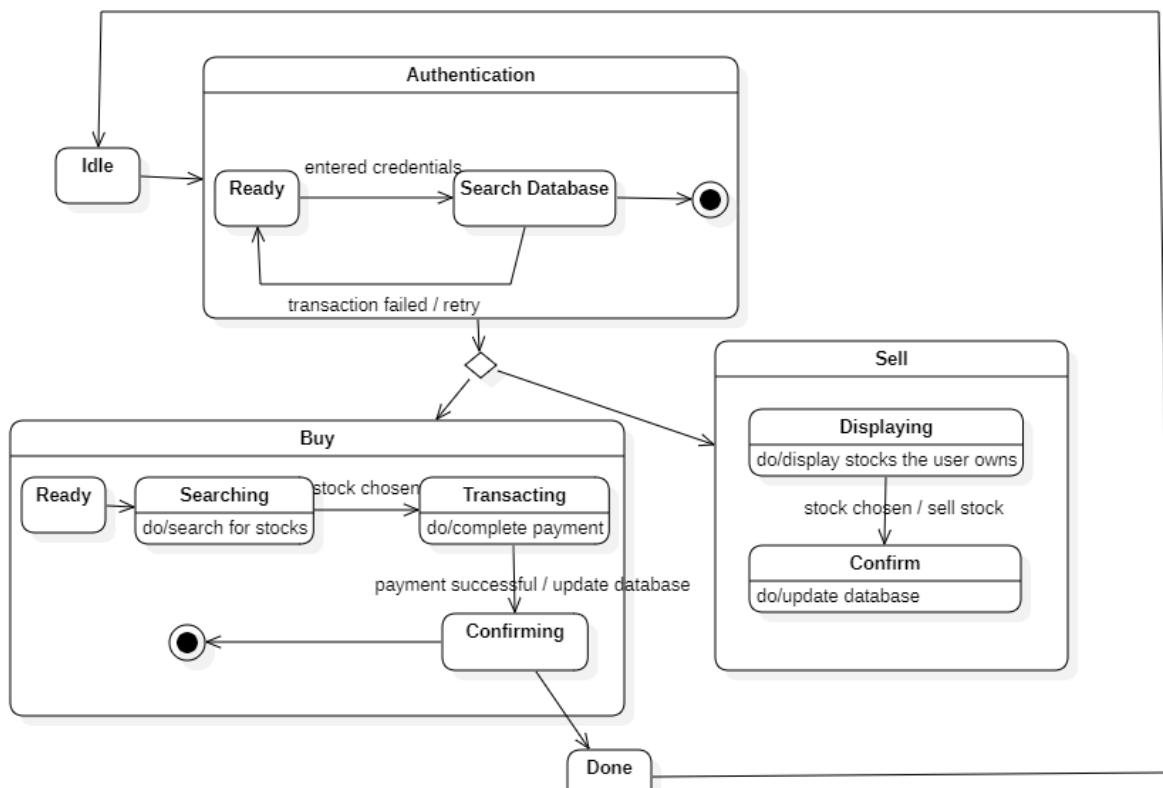
Total Budget: \$125,000

Note: The above schedule and budget are preliminary and subject to change based on further analysis and project requirements. It is also important to note that the budget may vary depending on the rates of the development team and any third-party tools or services required for the project.

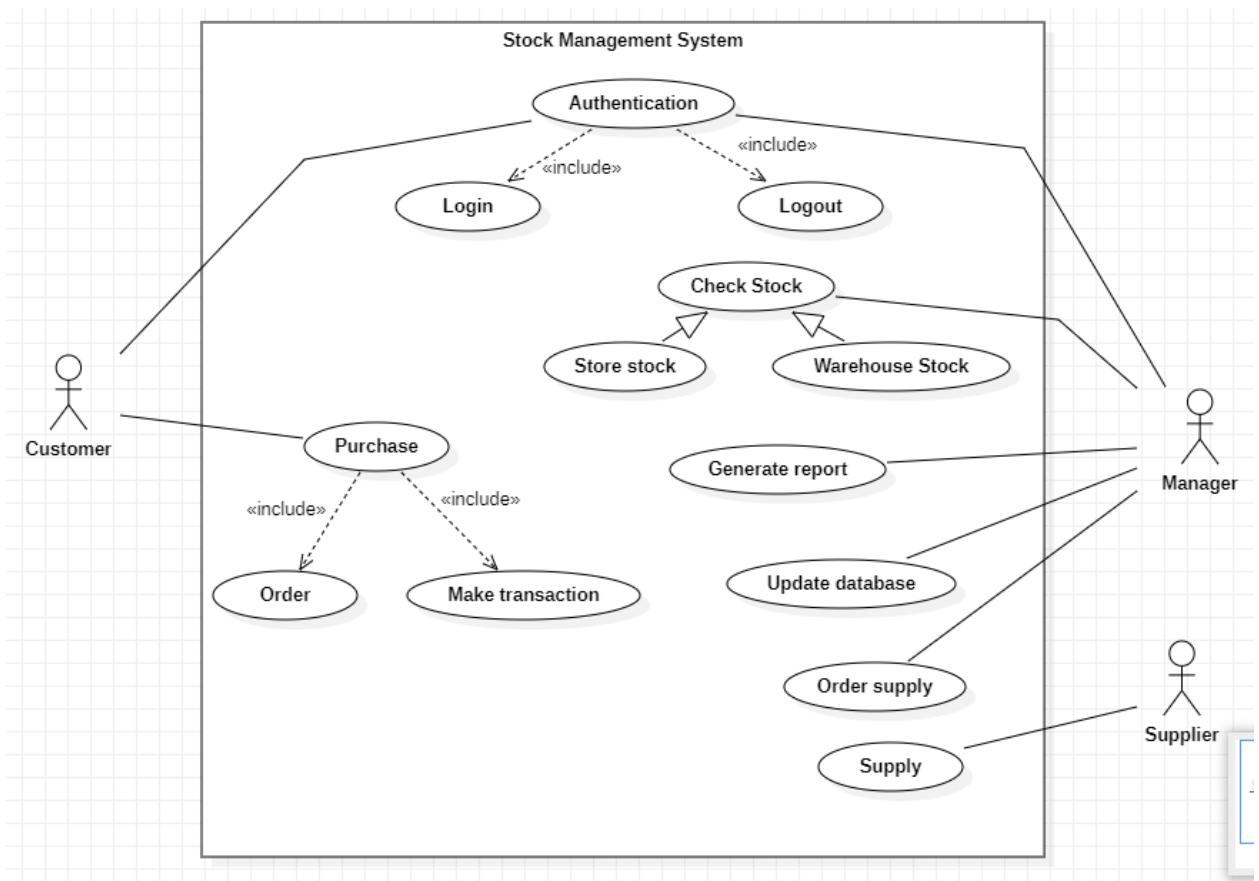
Class Diagram



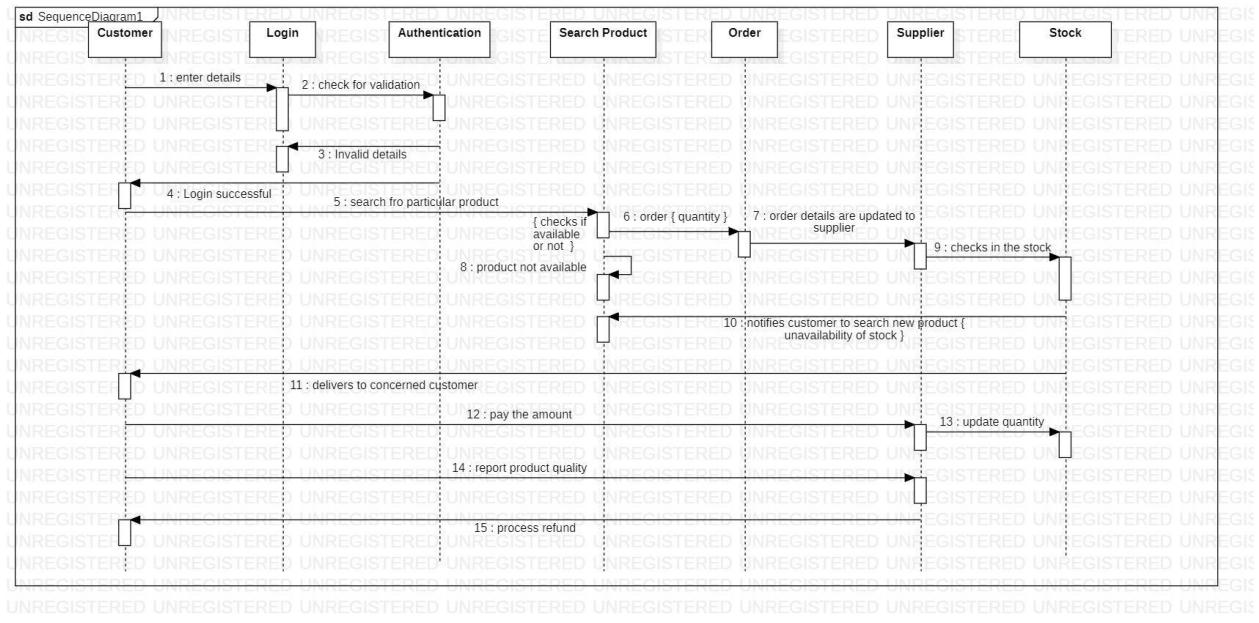
State Diagram



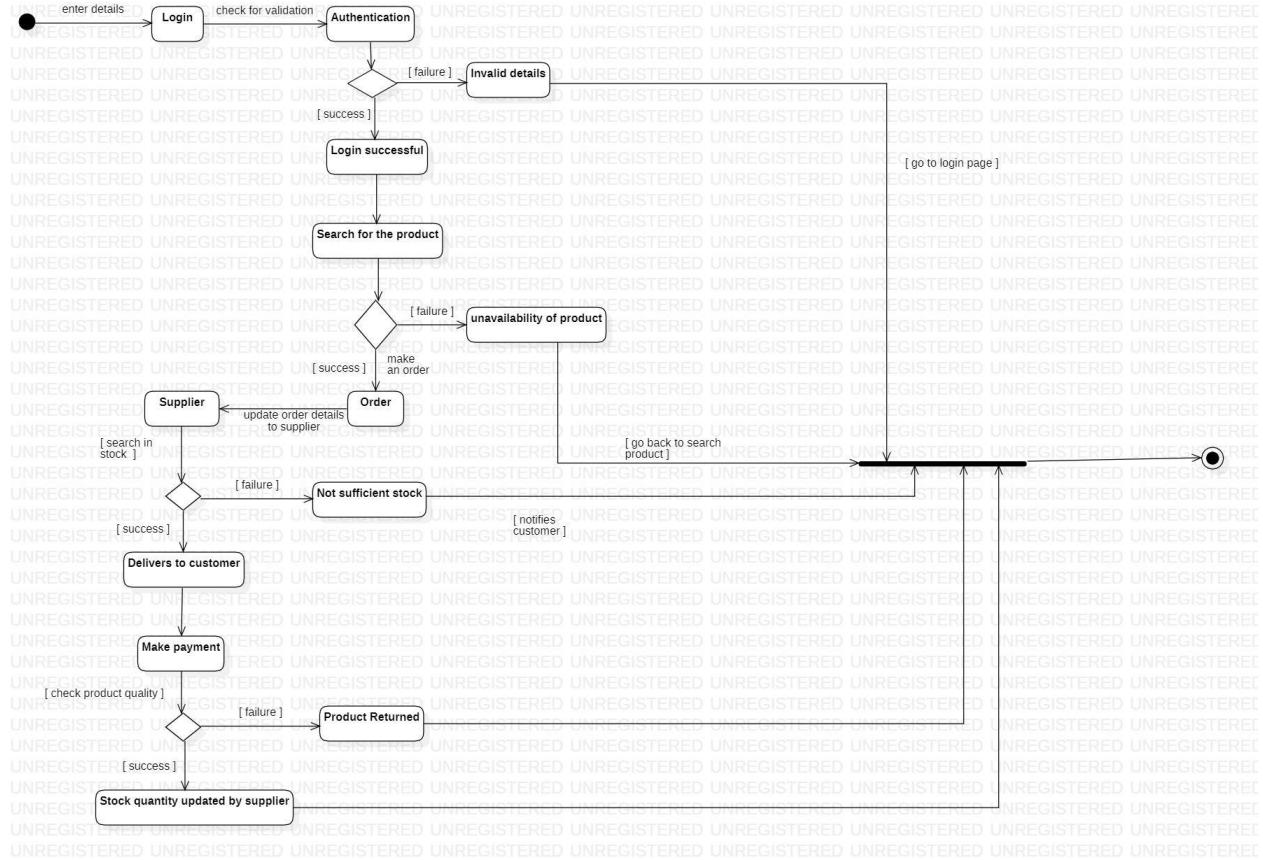
Use Case Diagram



Sequence Diagram



Activity Diagram



5. Passport Automation System

Problem Statement:

The manual passport application process is time-consuming, error-prone, and lacks efficiency. The traditional method of passport application and processing involves lengthy procedures and a considerable amount of paperwork, which can lead to delays in issuing passports. Moreover, it is difficult to track the status of the application, and there is a high risk of fraud.

Software Requirement Specification(SRS)

1. Introduction:

1.1. Purpose of this Document:

The purpose of the Passport Automation System is to automate the process of passport application and processing. This system aims to provide a hassle-free and efficient way of applying for and obtaining passports. The system aims to simplify the passport application process and reduce the time required for processing applications.

1.2. Scope of this document

The Passport Automation System will allow applicants to apply for a passport online, and the application will be processed electronically. The system will facilitate the processing of passport applications and the issuance of passports. The system will also enable the tracking of passport applications and provide updates on the status of the application. The system will be accessible to citizens of the country who wish to apply for a passport.

1.3. Overview

The Passport Automation System will be a web-based application that will allow citizens to apply for a passport online. The system will include an online application form that the applicants will fill out with their personal details, contact information, and other relevant information. The system will also allow the applicants to upload their passport photos and other necessary documents. Once the application is submitted, the system will automatically process the application and generate a passport application number.

2 General description:

The Passport Automation System will have two main components: the applicant module and the processing module. The applicant module will allow citizens to apply for a passport online, while the processing module will facilitate the processing of the applications and the issuance of passports. The system will also have an administrative module that will enable the system administrator to manage the system, including user accounts, system settings, and database management.

3 Functional Requirements:

- The system shall allow citizens to apply for a passport online.
- The system shall validate the applicant's personal details, contact information, and other relevant information.
- The system shall allow the applicants to upload their passport photos and other necessary documents.
- The system shall generate a passport application number for each application.
- The system shall facilitate the processing of passport applications and the issuance of passports.
- The system shall provide updates on the status of the application to the applicants.
- The system shall allow the system administrator to manage user accounts, system settings, and database management.

4 Interface Requirements:

- The system shall have a user-friendly interface for the applicants to fill out the application form.
- The system shall have a secure login system for the applicants and the system administrator.
- The system shall be accessible through a web browser.

5 Performance Requirements:

- The system shall process passport applications within 10 working days.

- The system shall allow a maximum of 10,000 concurrent users.
- The system shall have an uptime of at least 99.9%.
- The system shall have a response time of less than 3 seconds for each user request.

6 Design Constraints:

- The system shall be developed using Java programming language.
- The system shall use a MySQL database for storing and managing data.
- The system shall be developed using the Model-View-Controller (MVC) architecture.

7 Non-Functional Attributes:

- The system shall be secure and protect the privacy of the applicants' personal information.
- The system shall be scalable and able to handle an increase in the number of users.
- The system shall be reliable and robust, with a low risk of downtime or system failure.
- The system shall be easy to maintain and upgrade.

8 Preliminary Schedule and Budget:

Requirement Gathering and Analysis: 2 weeks

System Design: 4 weeks

Development and Testing: 12 weeks

Deployment and User Acceptance Testing: 2 weeks

Training and Documentation: 1 week

Total Estimated Time: 21 weeks

Budget:

Personnel: \$250,000

Hardware and software: \$100,000

Training and documentation: \$30,000

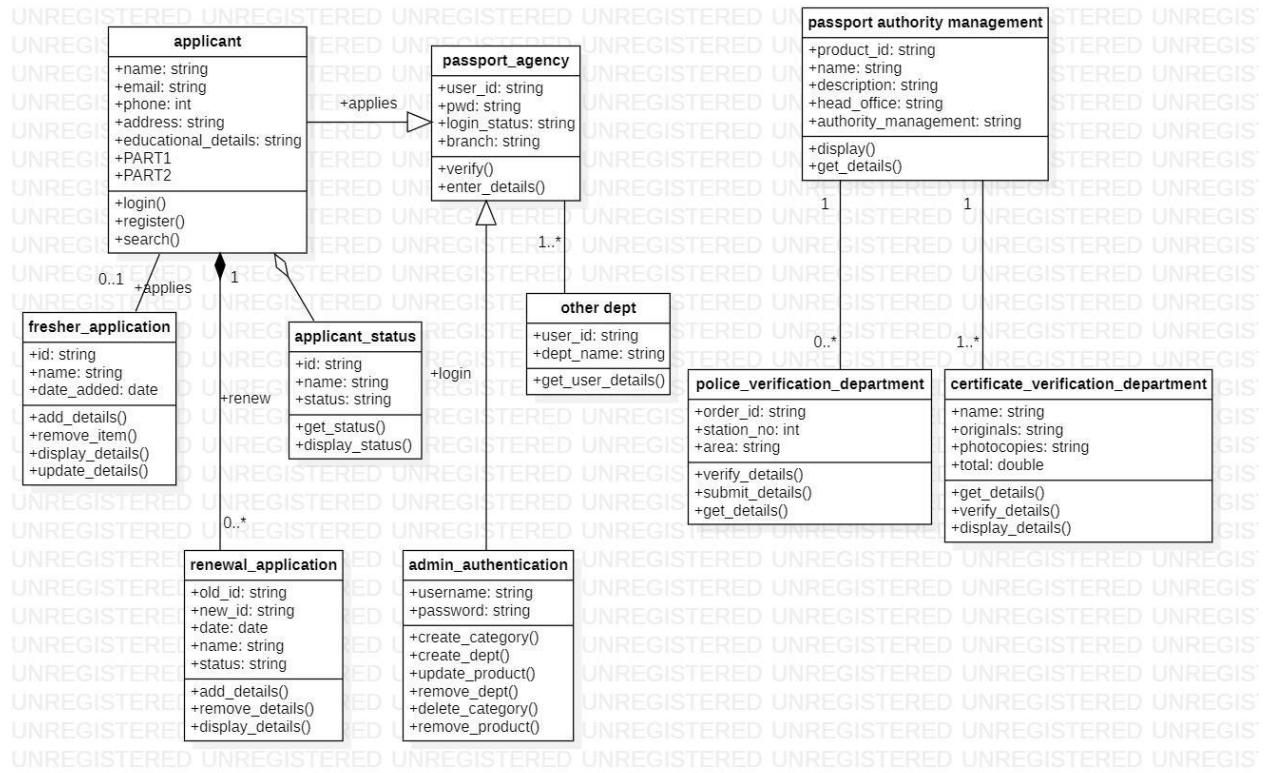
Contingency: \$50,000

Maintenance and support: \$70,000 per year

Total budget: \$500,000 (initial development and implementation) + \$70,000 per year

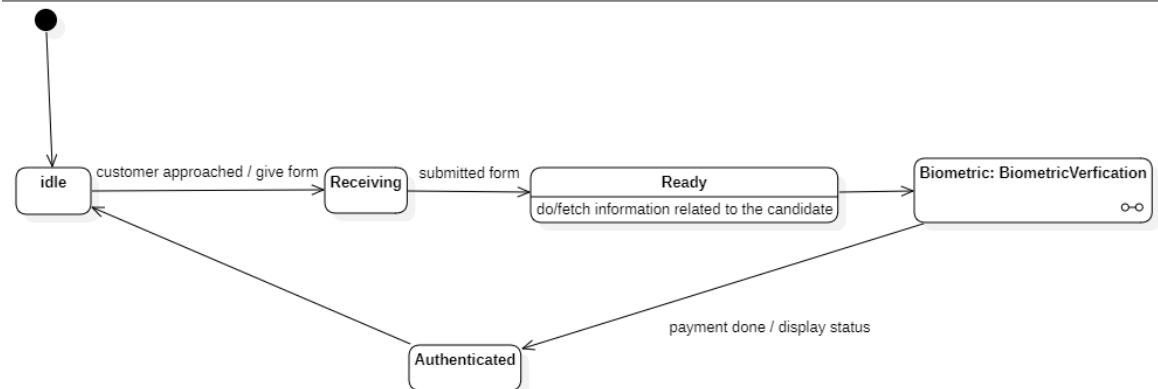
(maintenance and support)

Class Diagram

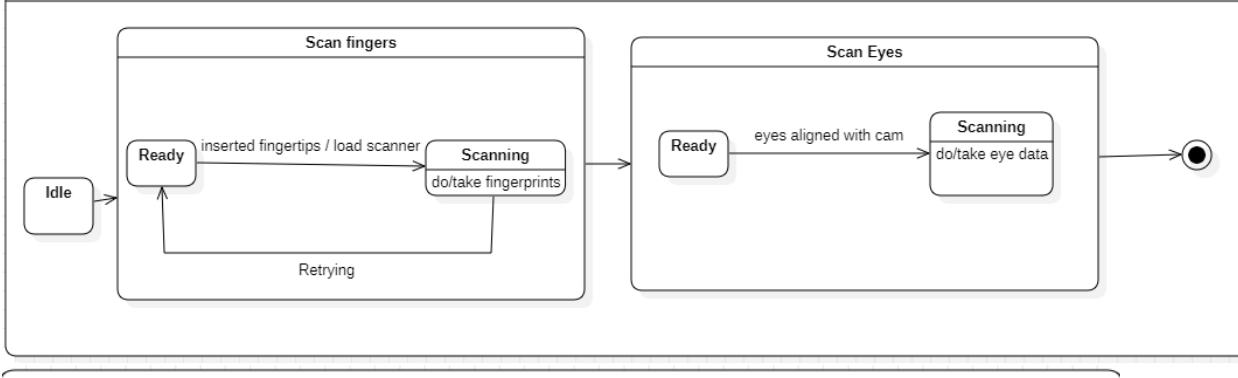


State Diagram

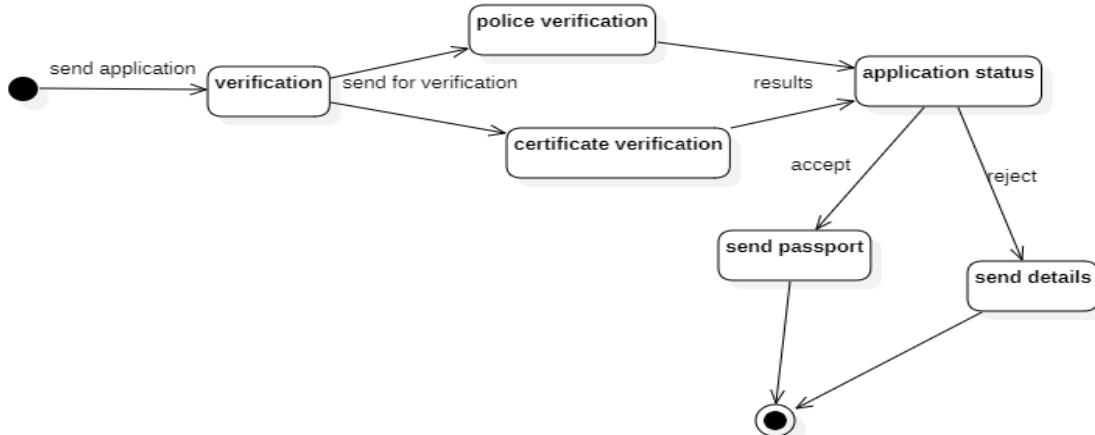
Passport Authenticator



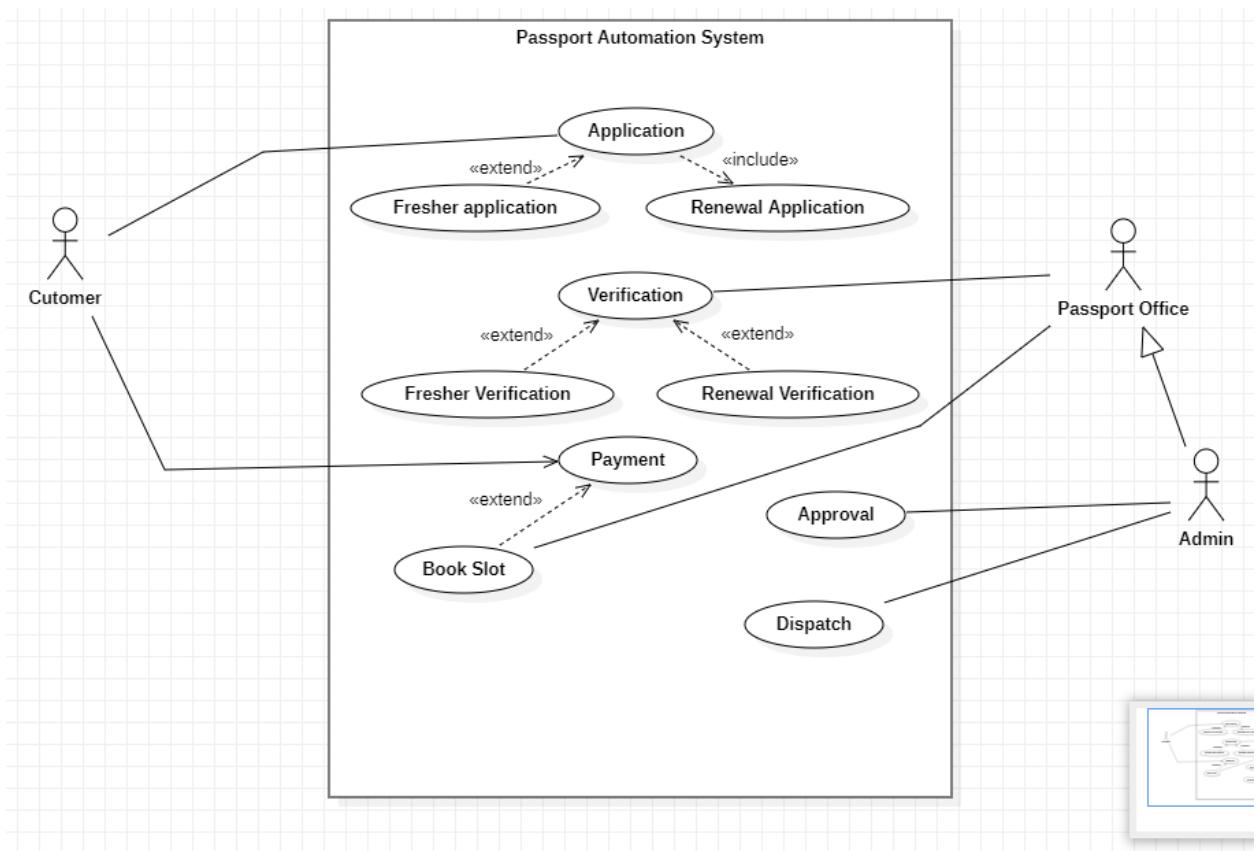
Biometric



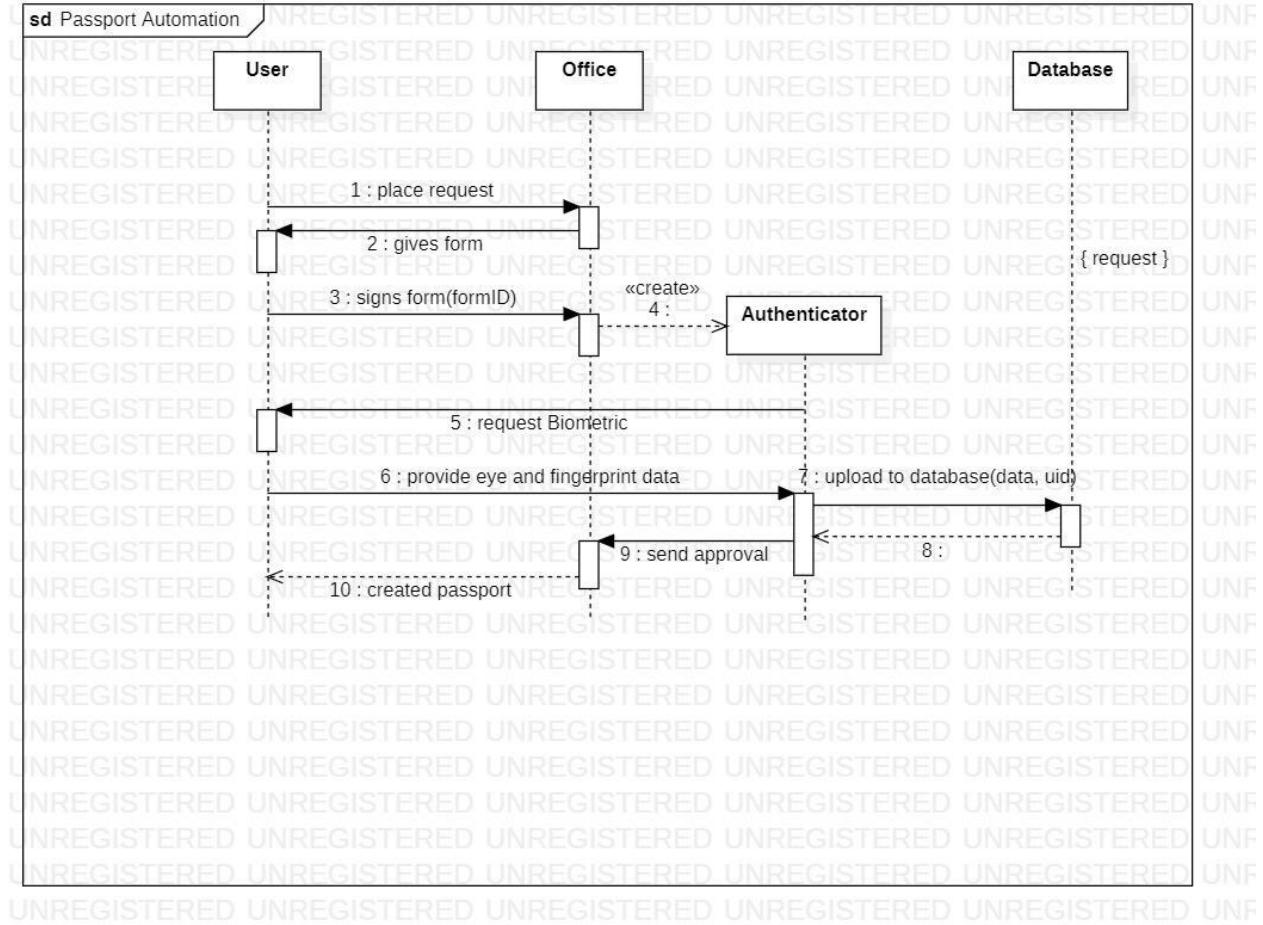
Processing



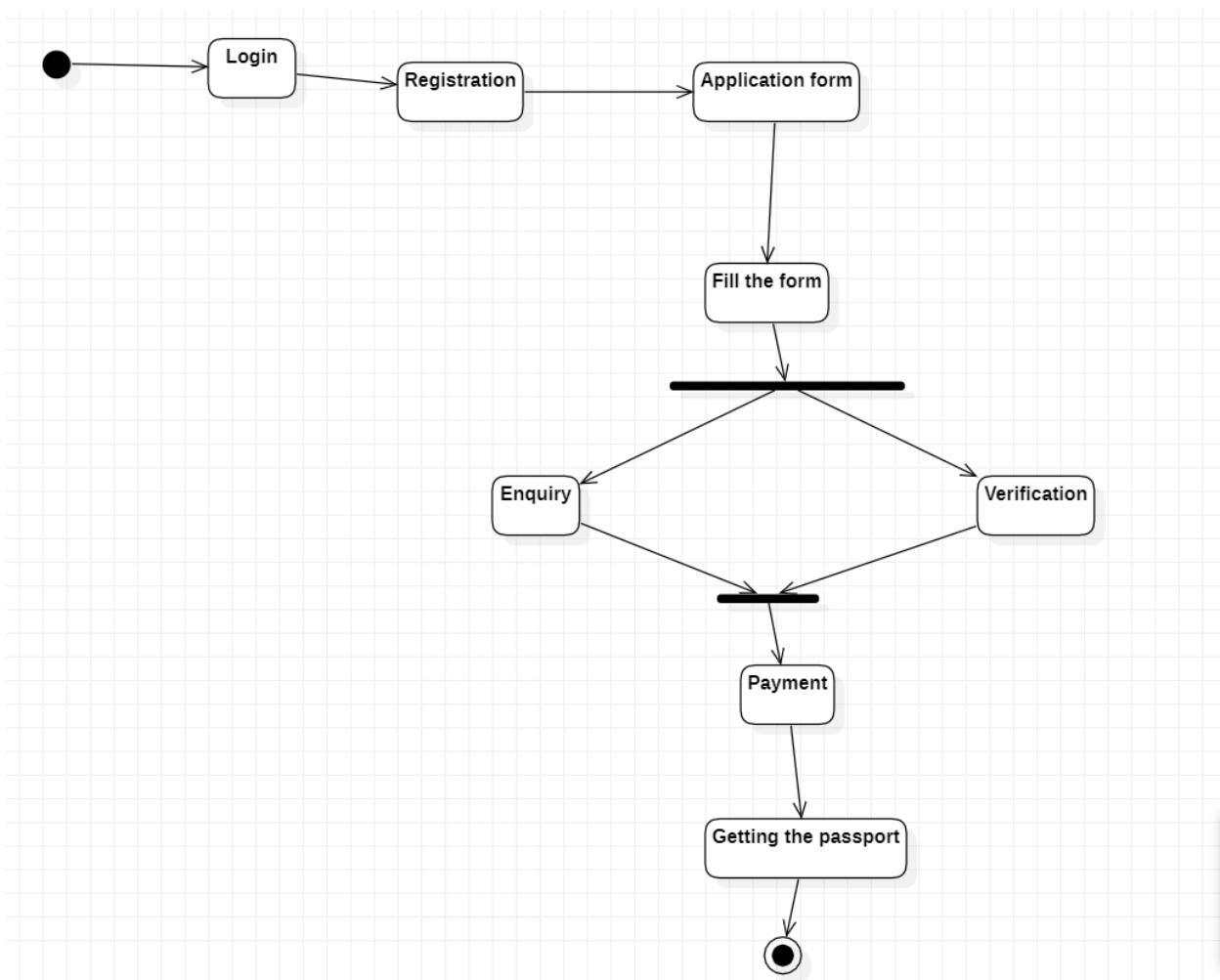
Use Case Diagram



Sequence Diagram



Activity Diagram



6. Railway Reservation System

Problem Statement: The current manual railway reservation process is time-consuming, error-prone, and inefficient, leading to long queues and delays for passengers. Railway operators struggle to manage reservations, track seat availability, and manage cancellations and refunds, resulting in a high rate of customer complaints and revenue losses. The lack of real-time updates on train schedules, delays, and cancellations leaves passengers uninformed and frustrated, leading to a poor customer experience. The manual payment processing system is also susceptible to fraud and errors, leading to revenue leakage and reputational damage for railway operators. There is a need for an automated railway reservation system that can streamline the ticket booking process, provide real-time updates, and enhance the customer experience. The system should also provide railway operators with tools to manage reservations, track seat availability, and manage cancellations and refunds in a secure and efficient manner.

Software Requirement Specification (SRS)

1 Introduction

1.1 Purpose of this document

The purpose of this document is to define the requirements and specifications for the development of a Railway Reservation System. This document is intended to guide the developers in creating a system that will enable customers to book tickets, check train schedules, and manage their travel itinerary.

1.2 Scope of this document

The scope of an automated railway reservation system is to automate the manual processes involved in railway ticket booking and management. The system is designed to simplify the ticket booking process for passengers and improve the efficiency of the reservation process

for railway operators. The scope of an automated railway reservation system typically includes the following:

Ticket booking: The system allows passengers to search for train schedules, check seat availability, and book tickets online or through automated kiosks.

Payment processing: The system should support various payment options, including online payment gateways, credit/debit cards, net banking, and mobile wallets.

Reservation management: The system should provide railway operators with tools to manage reservations, track seat availability, and manage cancellations and refunds.

Integration with other systems: The system should be able to integrate with other railway systems, including accounting systems, revenue management systems, and passenger information systems.

Security: The system should be designed with robust security features to protect customer data, prevent fraudulent activities, and ensure system reliability.

Reporting and analytics: The system should provide reports and analytics on passenger traffic, revenue, and other key performance indicators to help railway operators make informed decisions.

User experience: The system should be designed with a user-friendly interface that is easy to use and accessible to all users, including those with disabilities.

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

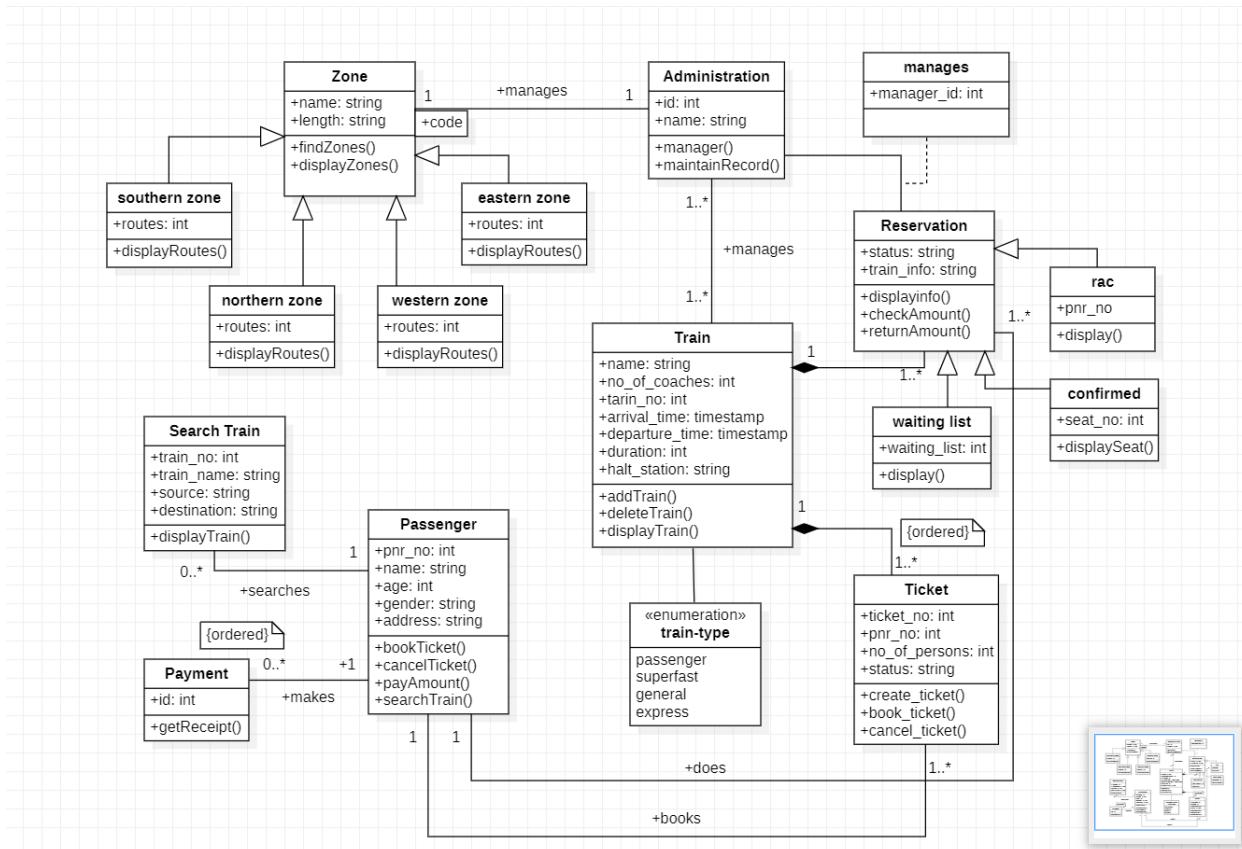
The non-functional attributes of the Railway Management System are as follows:

- **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.

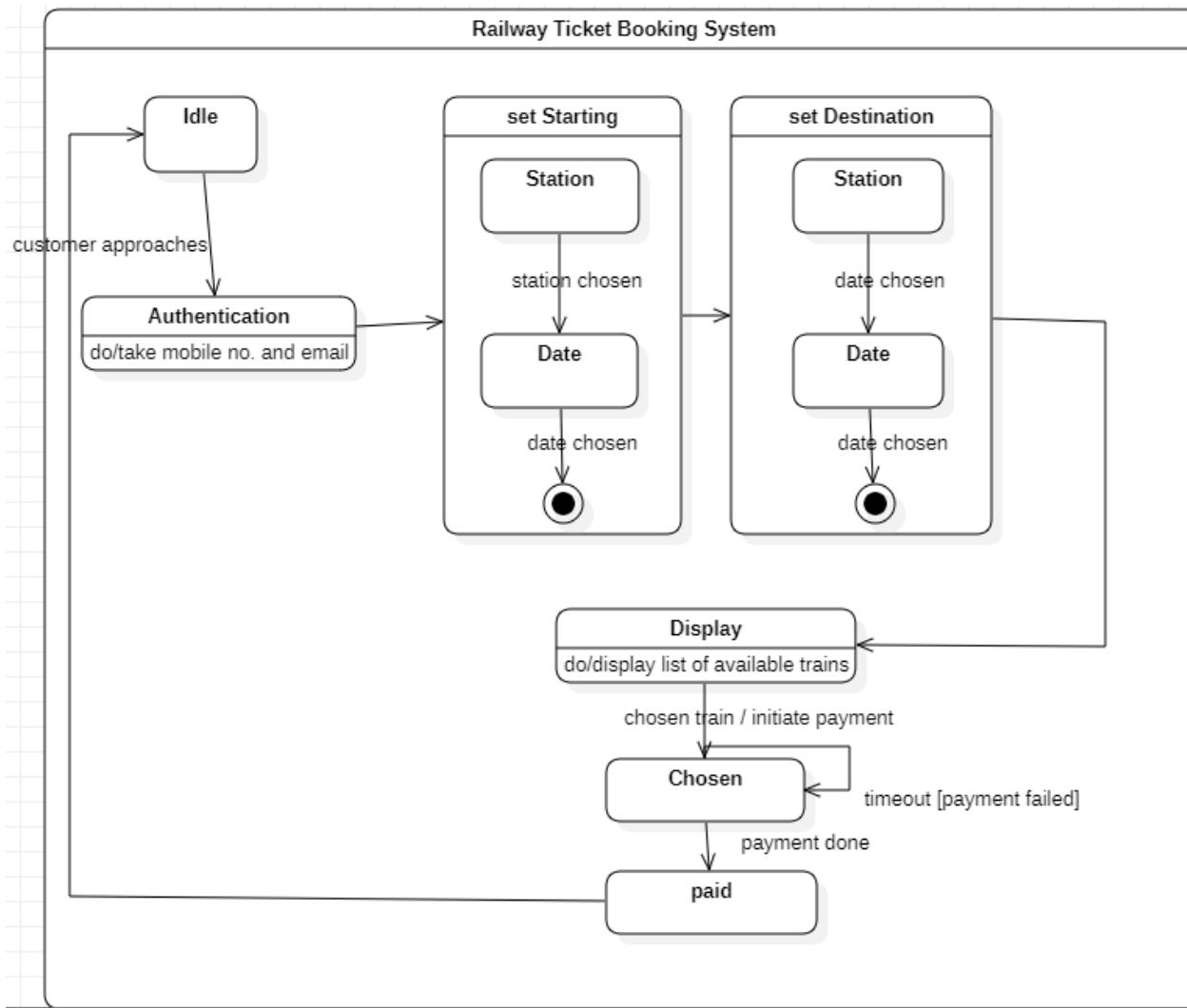
8 Preliminary Schedule and Budget

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.

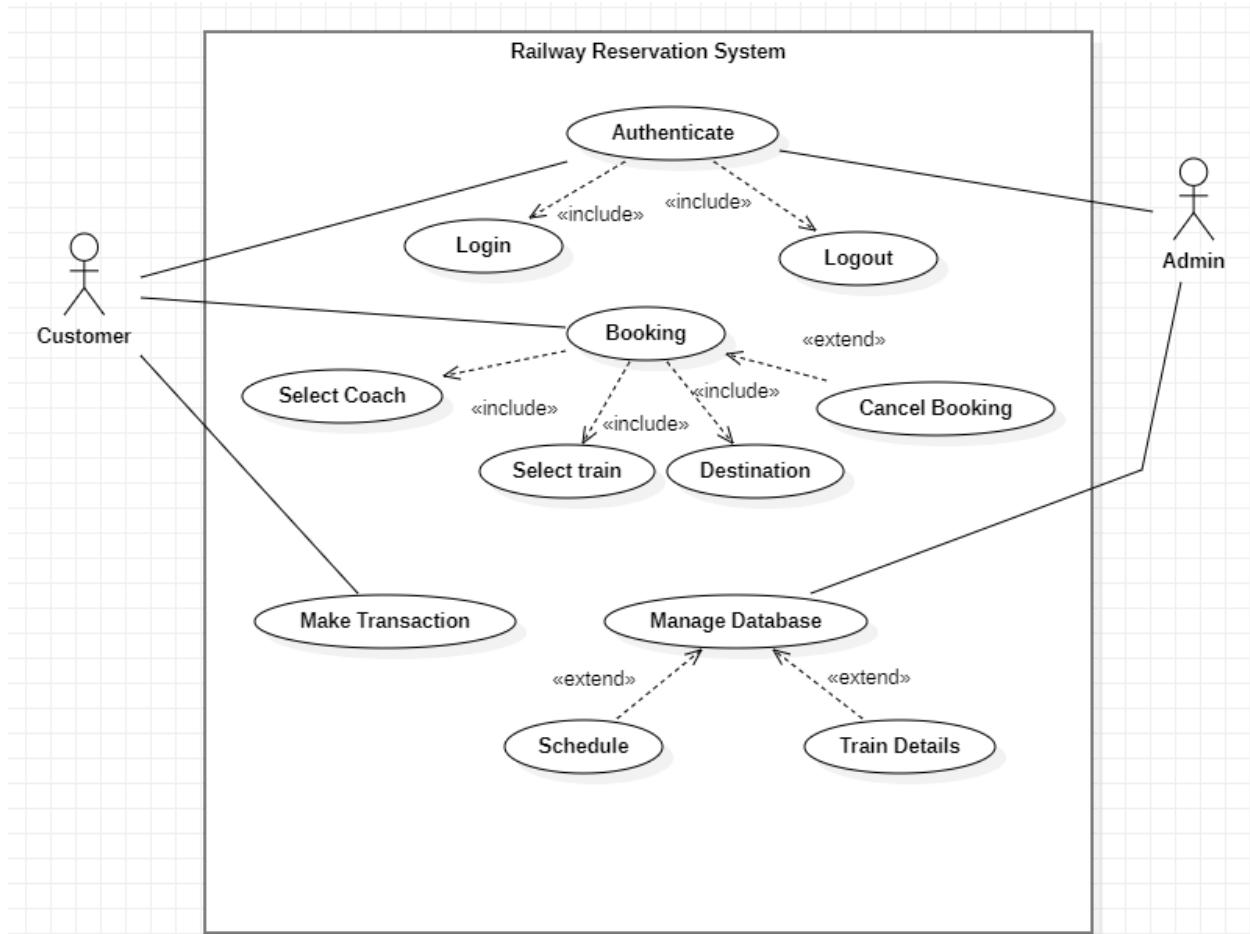
Class Diagram



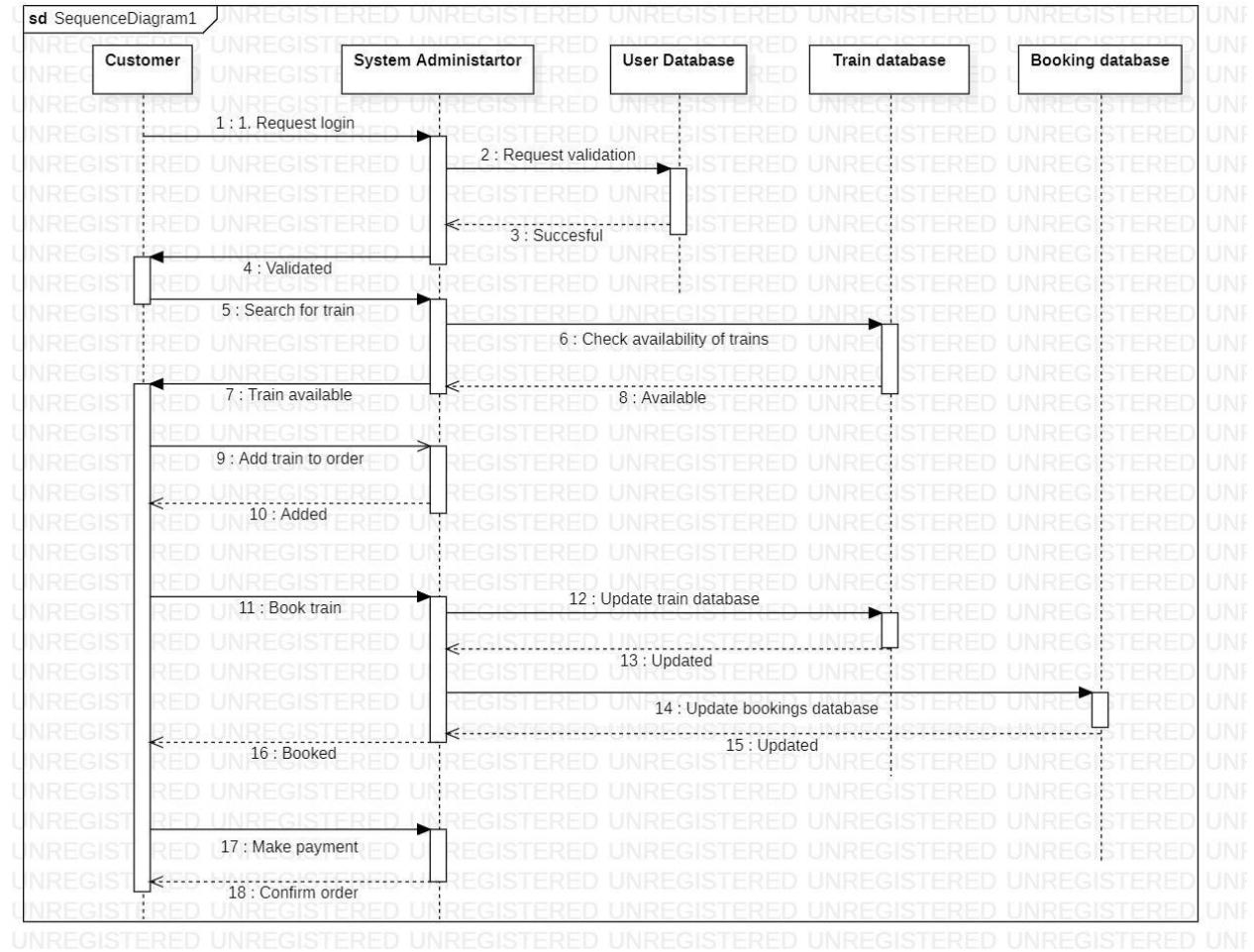
State Diagram



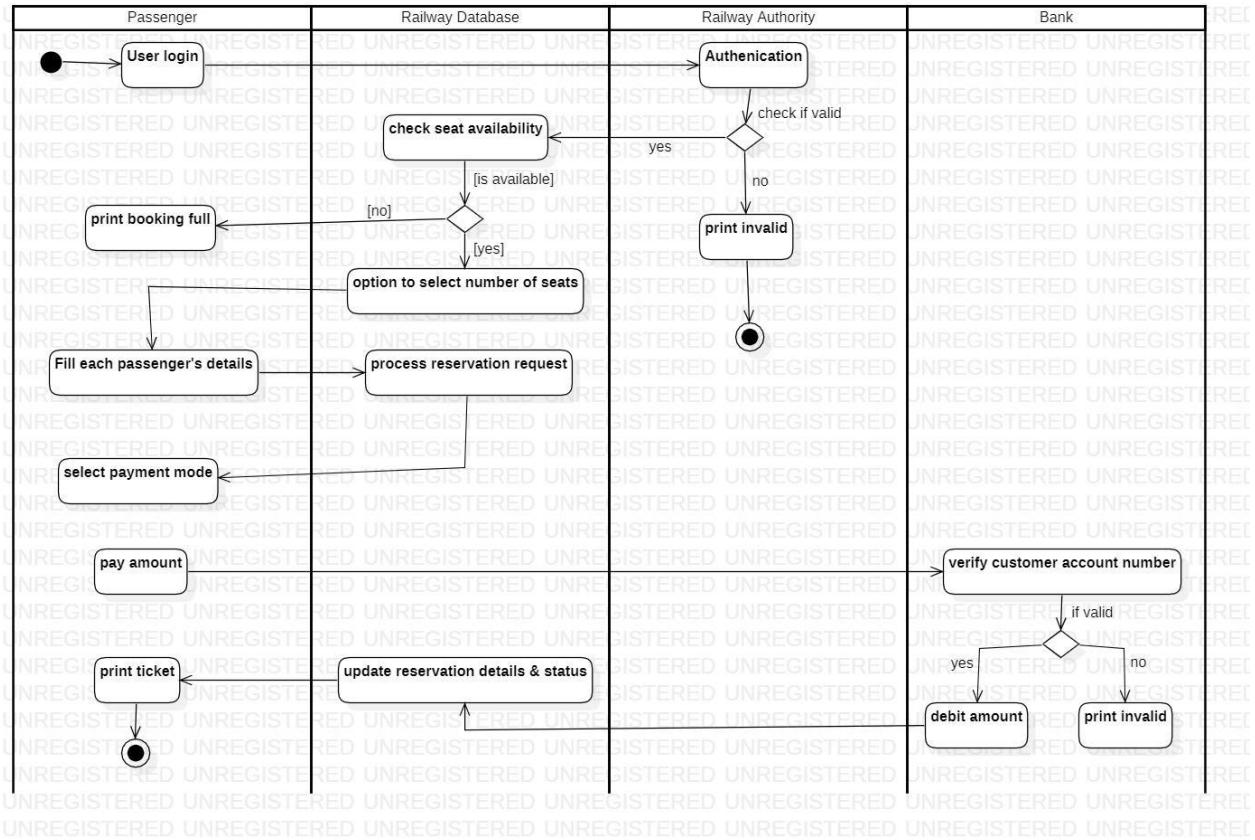
Use Case Diagram



Sequence Diagram



Activity Diagram



7. Online Shopping System

Problem Statement:

Online shopping has become increasingly popular in recent years, as customers seek the convenience of shopping from the comfort of their own homes. However, many online shopping systems suffer from a number of issues that can negatively impact the customer experience. These issues include poor system performance, security vulnerabilities, and usability challenges. As a result, customers may experience frustration and dissatisfaction when using online shopping systems, leading to lost sales and damaged brand reputation for businesses. Therefore, there is a need for an online shopping system that addresses these issues and provides customers with a seamless and secure shopping experience that meets their expectations.

Software Requirement Specification (SRS)

1 Introduction

1.1 Purpose of this document

The purpose of an online shopping system is to enable customers to purchase goods or services over the internet. This system provides a convenient and hassle-free shopping experience to customers, who can shop from anywhere and at any time. Online shopping systems also provide businesses with an additional sales channel, enabling them to reach a wider customer base and increase their revenue. .

1.2 Scope of this document

The scope of an online shopping system can vary depending on the business and the products or services being offered. However, a typical online shopping system would include the following functionality:

1. Product Catalog: The online shopping system should have a comprehensive catalog of all products or services available for purchase, including prices, descriptions, and images.

2. Shopping Cart: The shopping cart is a virtual basket where customers can add items they wish to purchase. Customers can view their shopping cart, add or remove items, and update the quantities of items they want to buy.

3. Checkout: The checkout process allows customers to review their order, select payment and shipping options, and confirm their purchase.

4. Payment Gateway: An online payment gateway is used to securely process payments made by customers using credit or debit cards.

5. Order Tracking: After making a purchase, customers should be able to track their order status, including shipping and delivery information.

1.3 Overview

The Railway Reservation System will be a comprehensive platform that will integrate multiple services such as booking tickets, checking train schedules, and managing travel itinerary. The system will be designed to be user-friendly and will provide customers with a seamless experience.

2 General Description

The system should have a user-friendly interface that is easy to navigate and understand. The system should be accessible through web and mobile applications. The system should be compatible with various web browsers and mobile devices. The system should provide customers with multiple language options.

3 Functional Requirements

- The system should allow customers to search for trains based on their travel dates and destinations.
- The system should provide customers with real-time information about train schedules, availability of seats, and estimated arrival and departure times.
- The system should enable customers to book train tickets by selecting their travel dates, destinations, and preferences.
- The system should allow customers to make payments using various payment methods such as credit cards, debit cards, and online wallets.
- The system should provide customers with a confirmation of their booking and a travel itinerary with all the relevant details.

4 Interface Requirements

- The system should have a user-friendly interface that is easy to navigate and understand.
- The system should be accessible through web and mobile applications.
- The system should be compatible with various web browsers and mobile devices.
- The system should provide customers with multiple language options.

5 Performance Requirements

- The system should be able to handle a large number of transactions simultaneously.
- The system should be able to provide real-time information about train schedules and availability of seats.
- The system should be able to process payments quickly and efficiently.

6 Design Constraints

- **Technology Constraints:** The system should be designed to work on a specific technology stack that is determined by the development team or organization. This may include specific programming languages, databases, and operating systems.
- **Budget Constraints:** The system design should consider the budget allocated for development, testing, deployment, and maintenance. The design should ensure that the system is developed within the allocated budget.
- **Time Constraints:** The design should consider the time frame for the project and ensure that the system can be developed within the allocated time frame. The design should prioritize critical functionalities and features to ensure timely delivery.
- **Integration Constraints:** The design should ensure that the system can integrate with other systems, services, and platforms that are essential for the functionality of the Railway Reservation System. This may include integration with payment gateways, SMS providers, and travel insurance providers.
- **Security Constraints:** The design should ensure that the system is secure and can protect customer data from unauthorized access. This may include implementing security protocols and encryption mechanisms to ensure data privacy and confidentiality.

7 Non-Functional Attributes

- **Usability:** The system should be easy to use and navigate, with a user-friendly interface that requires minimal training. The system should be designed to be accessible to users of all ages and technical abilities.
- **Reliability:** The system should be reliable and available 24/7, with minimal downtime. The system should be designed to handle a large volume of traffic without any performance issues.
- **Security:** The system should be designed with robust security features to ensure the safety and privacy of customer data. The system should comply with industry-standard security protocols and regulations.

- **Performance:** The system should be designed to perform efficiently, with fast response times and minimal lag. The system should be able to handle a large volume of concurrent users without any performance degradation.
- **Scalability:** The system should be designed to scale up or down depending on the demand for the service. The system should be able to handle a large volume of traffic without any degradation in performance.
- **Maintainability:** The system should be easy to maintain, with clear and well-documented code that can be easily modified or updated. The system should be designed to facilitate easy bug fixes and updates.

8 Preliminary Schedule and Budget

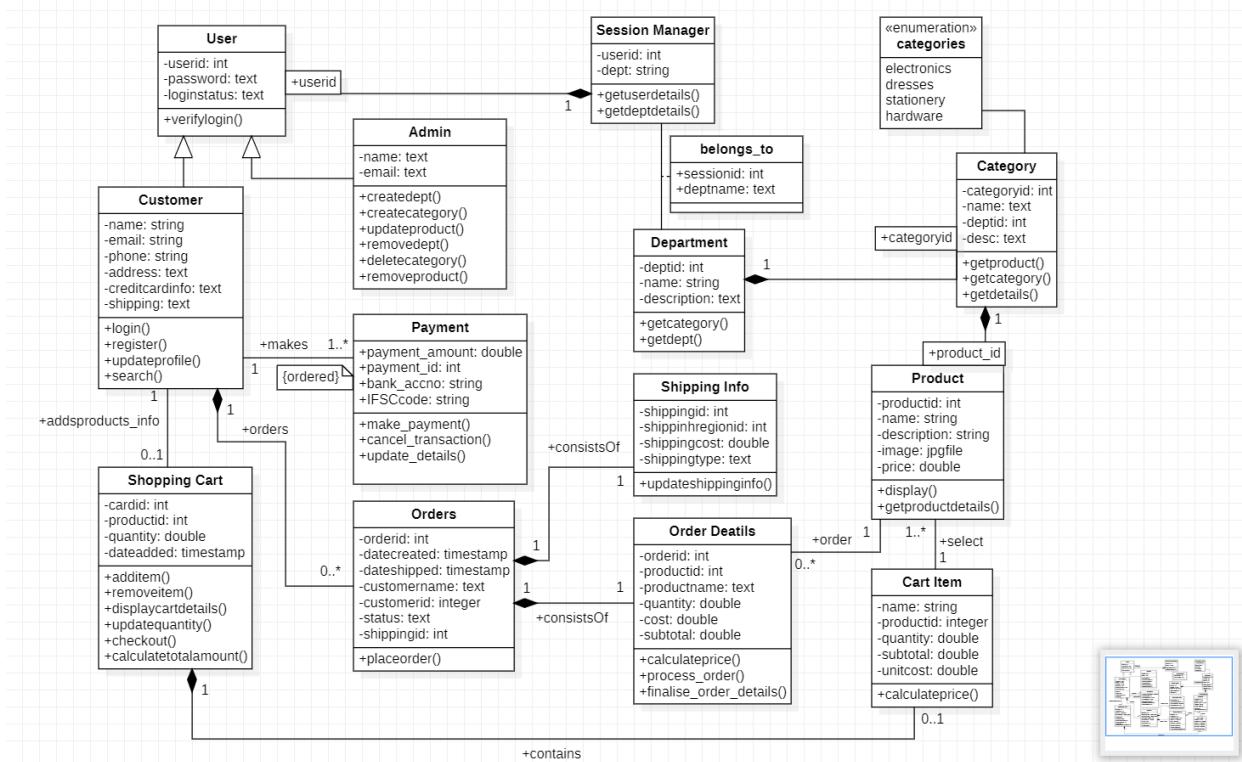
Schedule:

The development of an online shopping system can take several months, depending on the complexity of the system and the size of the development team. A preliminary schedule may include the following milestones:

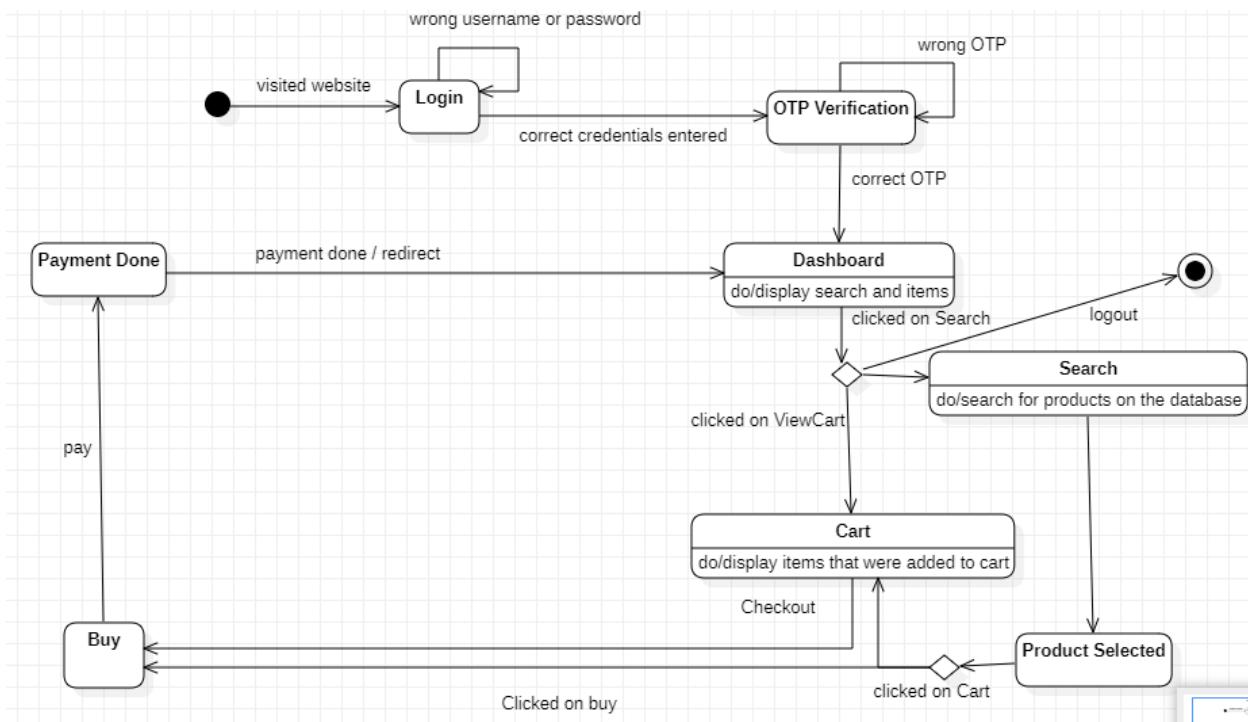
1. Requirements Gathering: 1-2 weeks
2. System Design: 2-3 weeks
3. Development: 8-12 weeks
4. Testing: 2-4 weeks
5. Deployment: 1-2 weeks

The budget for developing an online shopping system can vary significantly depending on the features and functionality required. The budget should include the cost of development, hosting, and ongoing maintenance and support. A preliminary budget estimate for a basic online shopping system could range from \$50,000 to \$100,000

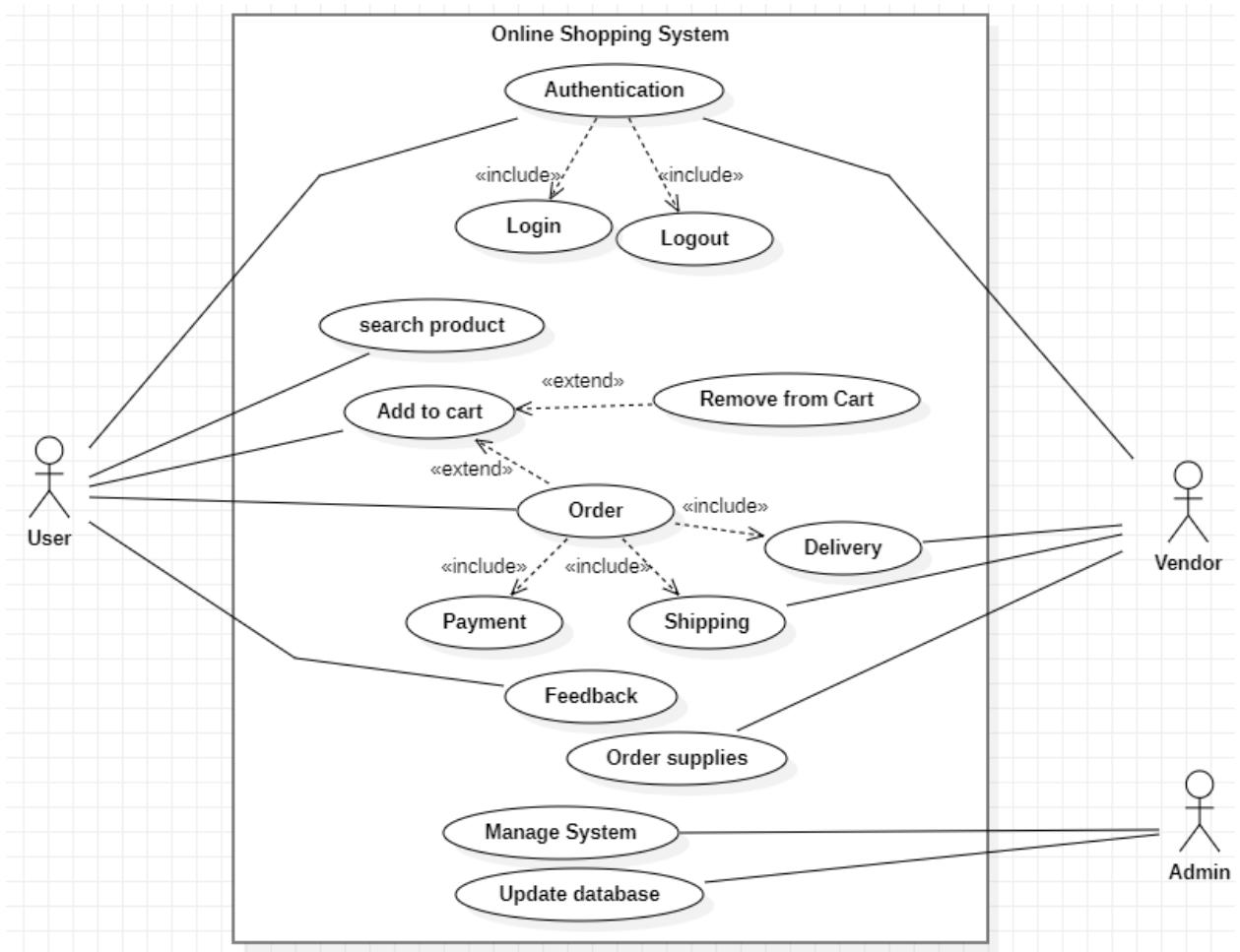
Class Diagram



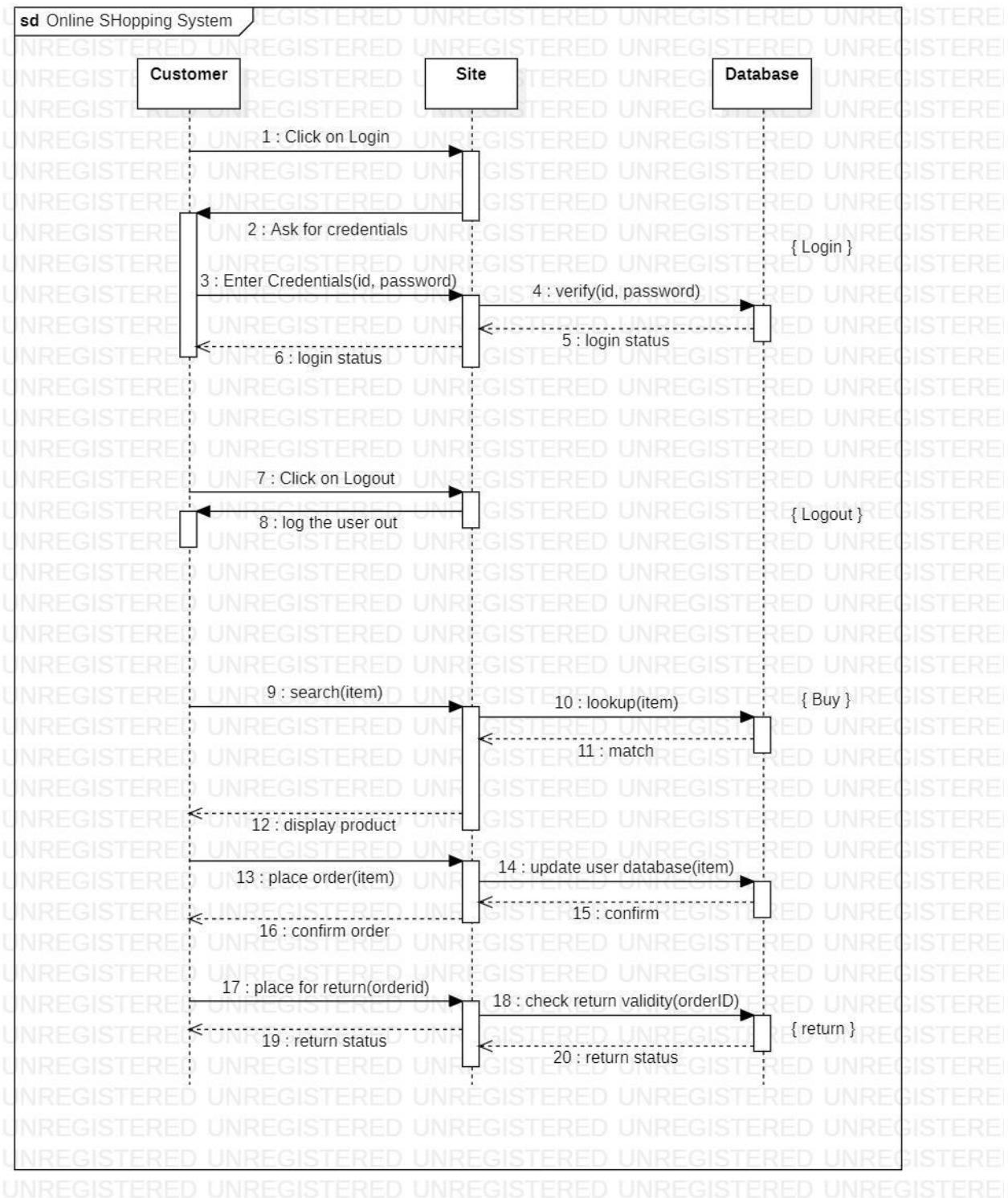
State Diagram



Use Case Diagram



Sequence Diagram



Activity Diagram

