

B.M.S. COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



Lab Record

Object Oriented Analysis and Design

Submitted in partial fulfillment for the 6th Semester Laboratory

Bachelor of Technology
in
Computer Science and Engineering

Submitted by:

VIJAYA VERMA

1BM20CS187

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

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND

ENGINEERING



CERTIFICATE

This is to certify that the Object-Oriented Analysis and Design(20CS6PCOMD) laboratory has been carried out by Vijaya Verma (1BM20CS187) during the 6th Semester April-July 2023.

Signature

SEEMA PATIL

Department of Computer Science and Engineering
B.M.S. College of Engineering, Bangalore

TABLE OF CONTENTS

1.Credit Card processing System	1
2.Hotel Management System	7
3.Library Management System	13
4.Online Shopping System	19
5.Passport Authentication System	25
6.Railway reservation system	31
7.Stock Maintenance System	37

1. CREDIT CARD SYSTEM

1.1 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 operation.

1.2 Software Requirement Specification

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

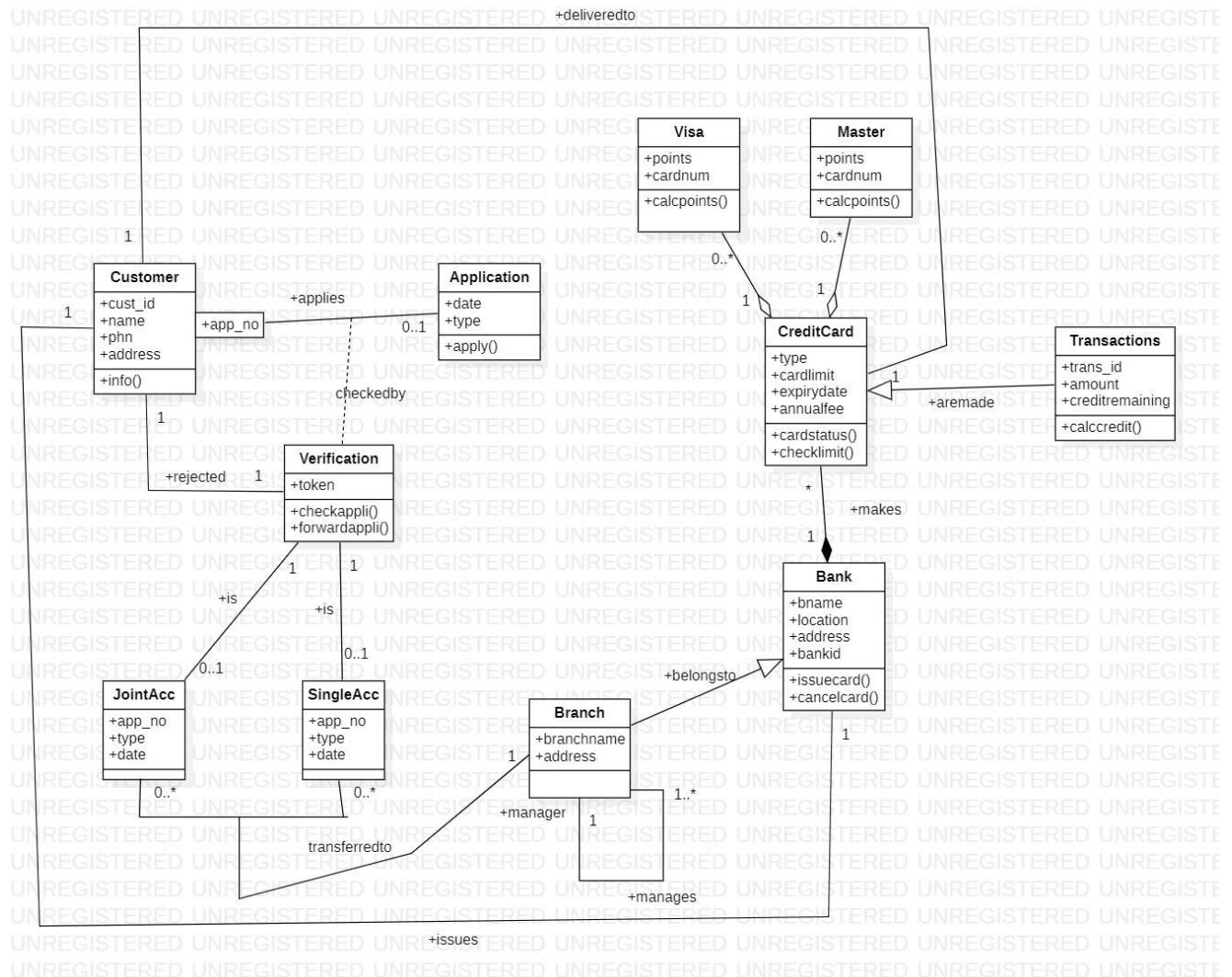
Deployment and User Training: \$5,000 - \$10,000

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

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

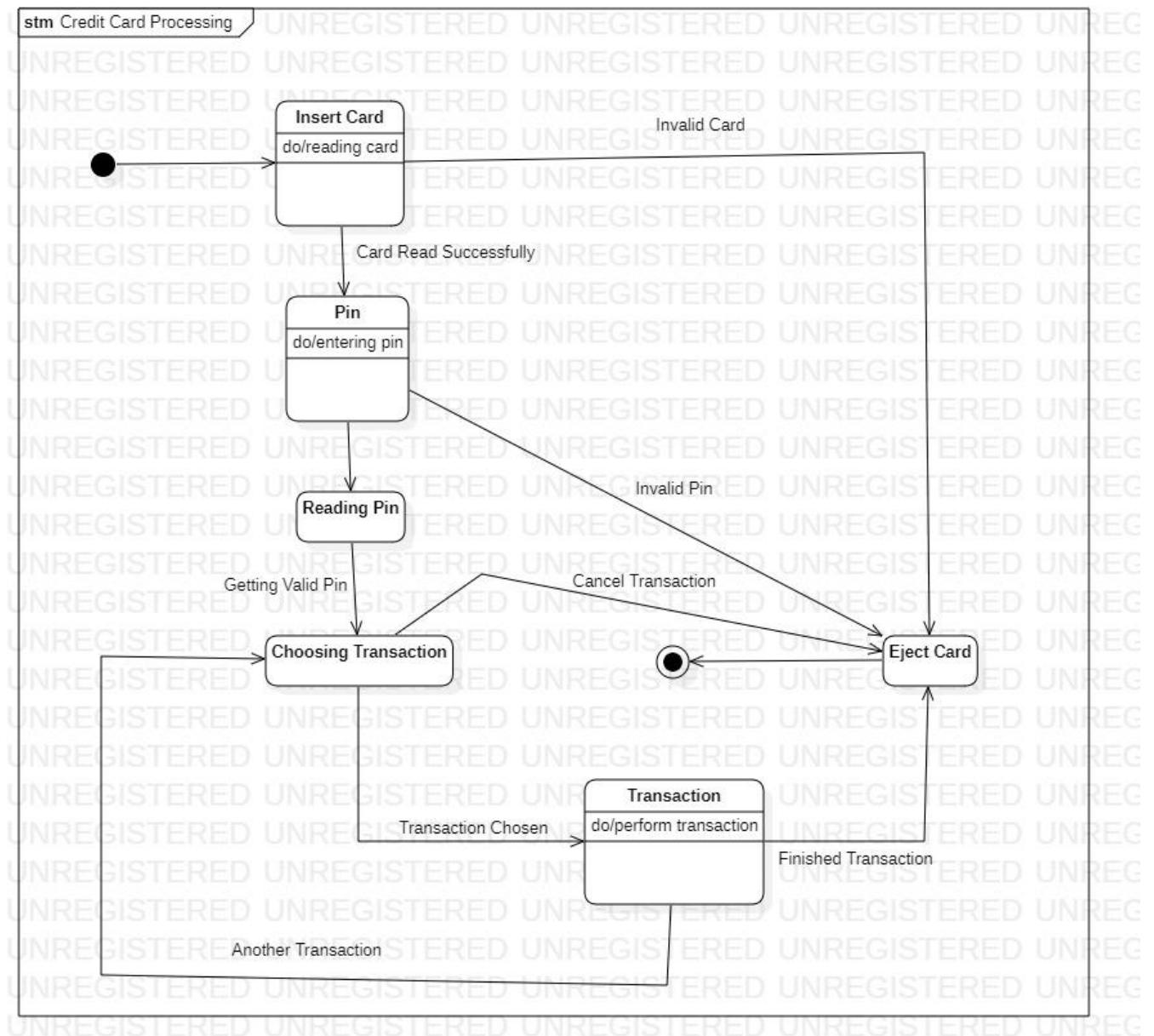
1.3 Class Diagram

The Bank can issue the card or cancel the card of a customer. Transactions happens using the card which can be either visa or master. Customer puts up application to apply for card which is verified decided which type of account they want.



1.4 State Diagram

First you insert the card which if read successfully by machine , asks pin to be entered. Pin is entered. If the pin is invalid, card is ejected. If the pin is valid, we choose the transaction process either to withdraw or to check balance, etc. Once transaction is done, card is ejected. Else customer continues with another transaction.



1.5 Use Case Diagram

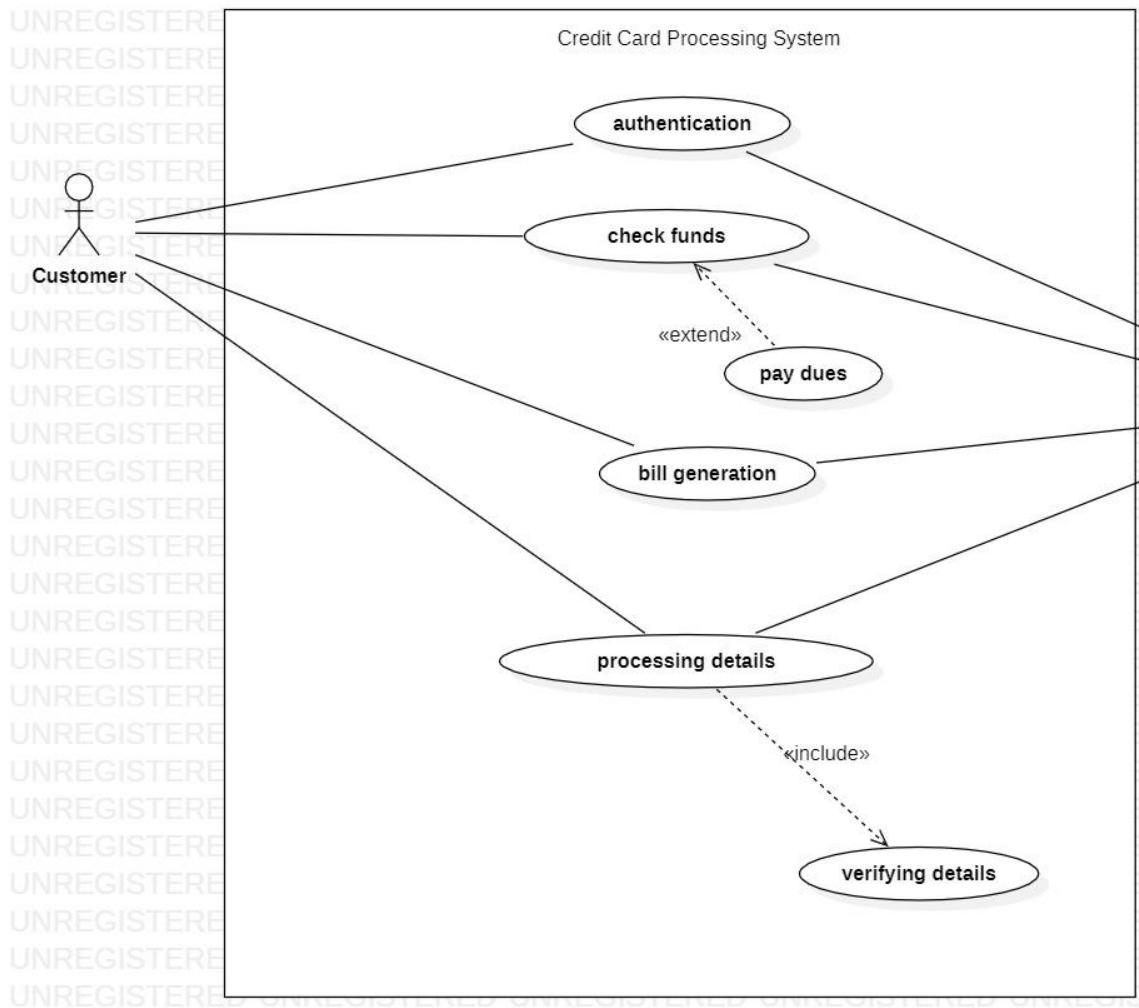


Fig 1.4

Fig 1.5

Actors:

Customer : person who is using credit card

Bank : the provider

Use Cases:

Authentication:

Check funds: customer and bank both can check balance

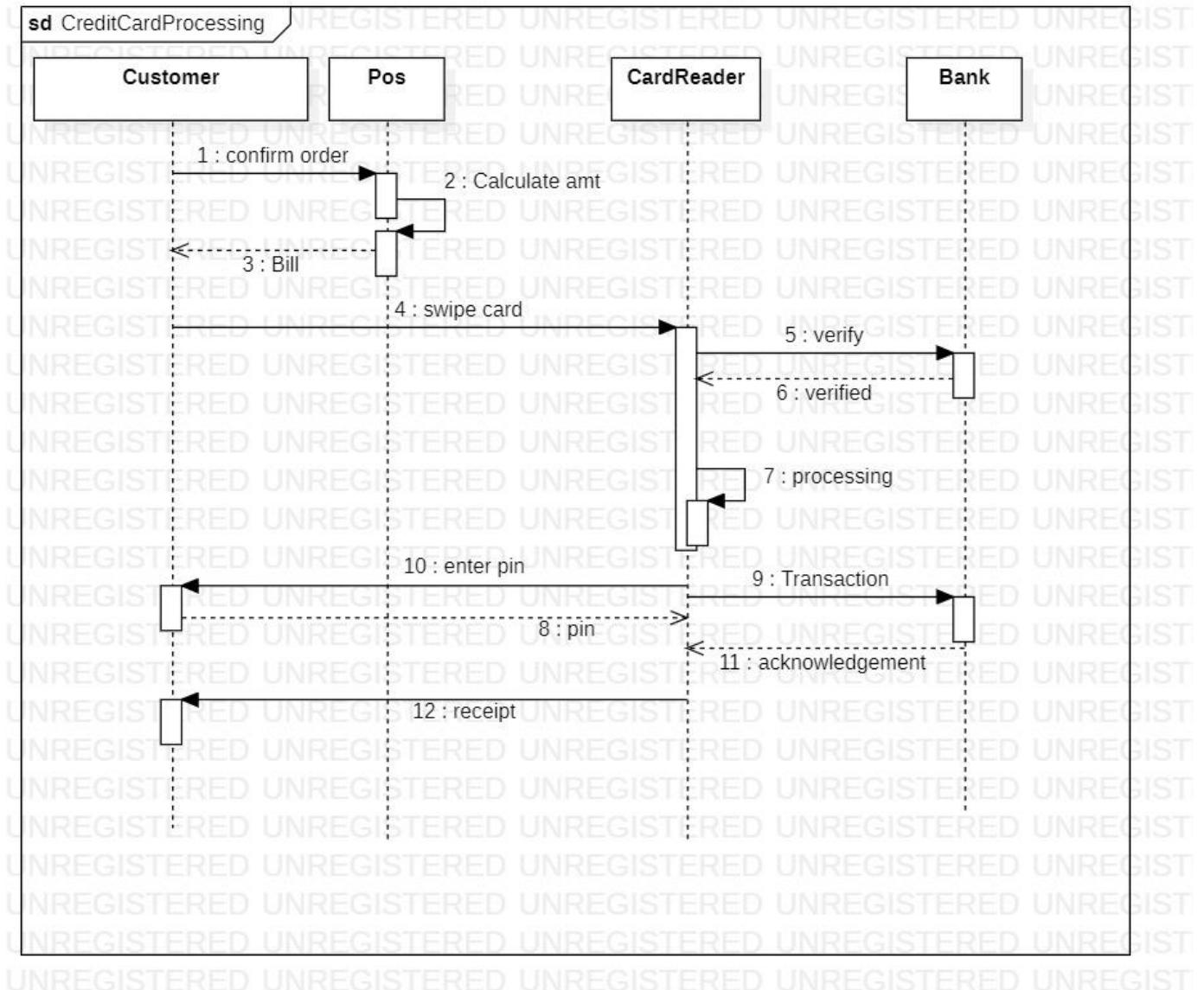
which extends to paying dues

Bill generation: bank will generate the bill and send to customer.

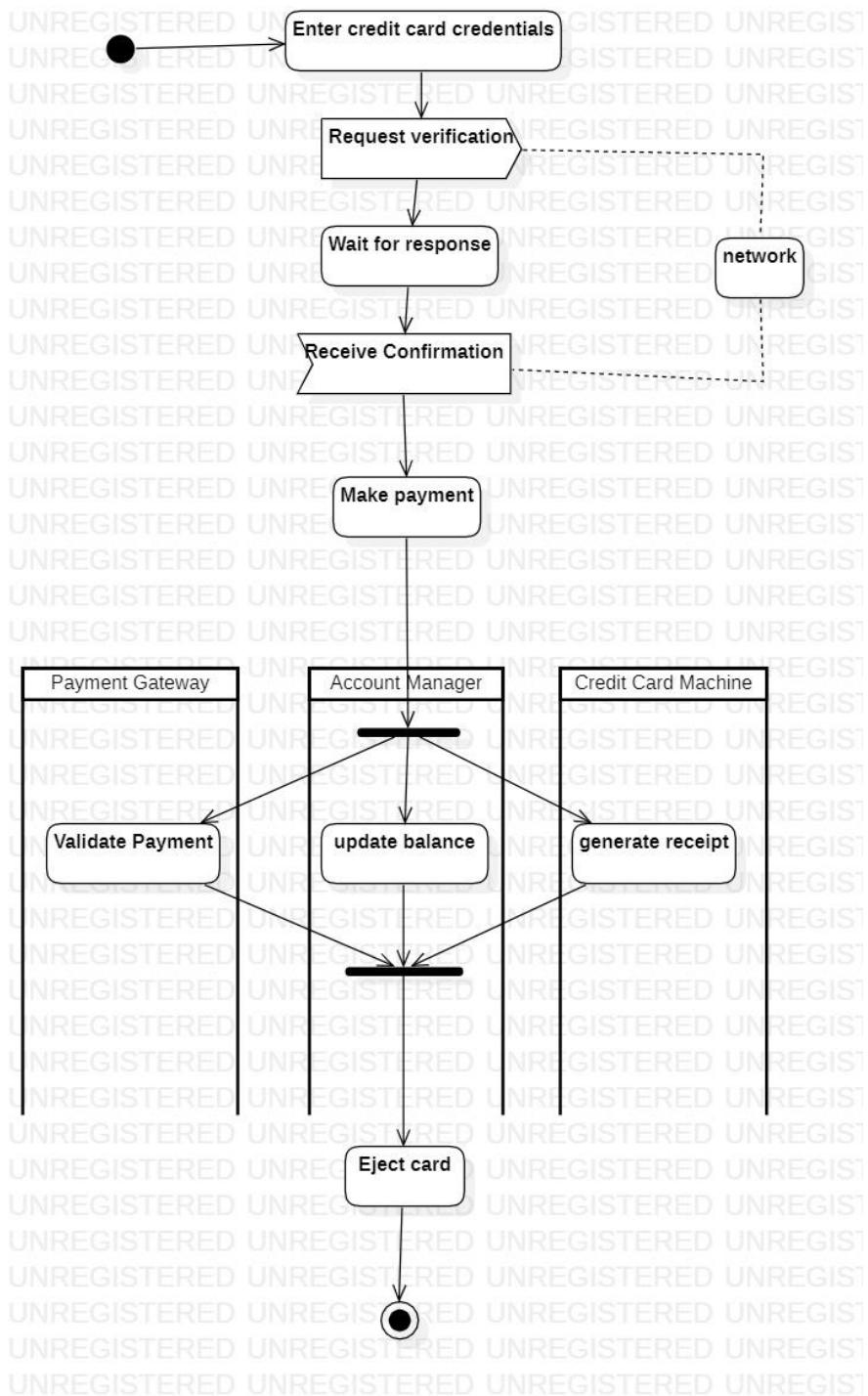
Pocessing details including verifying details.

1.6 Sequence Diagram

After order is confirmed and amount is calculated, card is swiped, verified by the bank and processed by card reader. Reader then asks customer to enter the pin. After insertion, bank acknowledges and sends the receipt.



1.7 Activity Diagram



2. HOTEL MANAGEMENT SYSTEM

2.1 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.

2.2 Software Requirement Specification

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.

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. 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.occupancy reports, inventory reports, and customer feedback reports.

3. 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.

4. 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.

5. 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.

6. 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.

7. 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

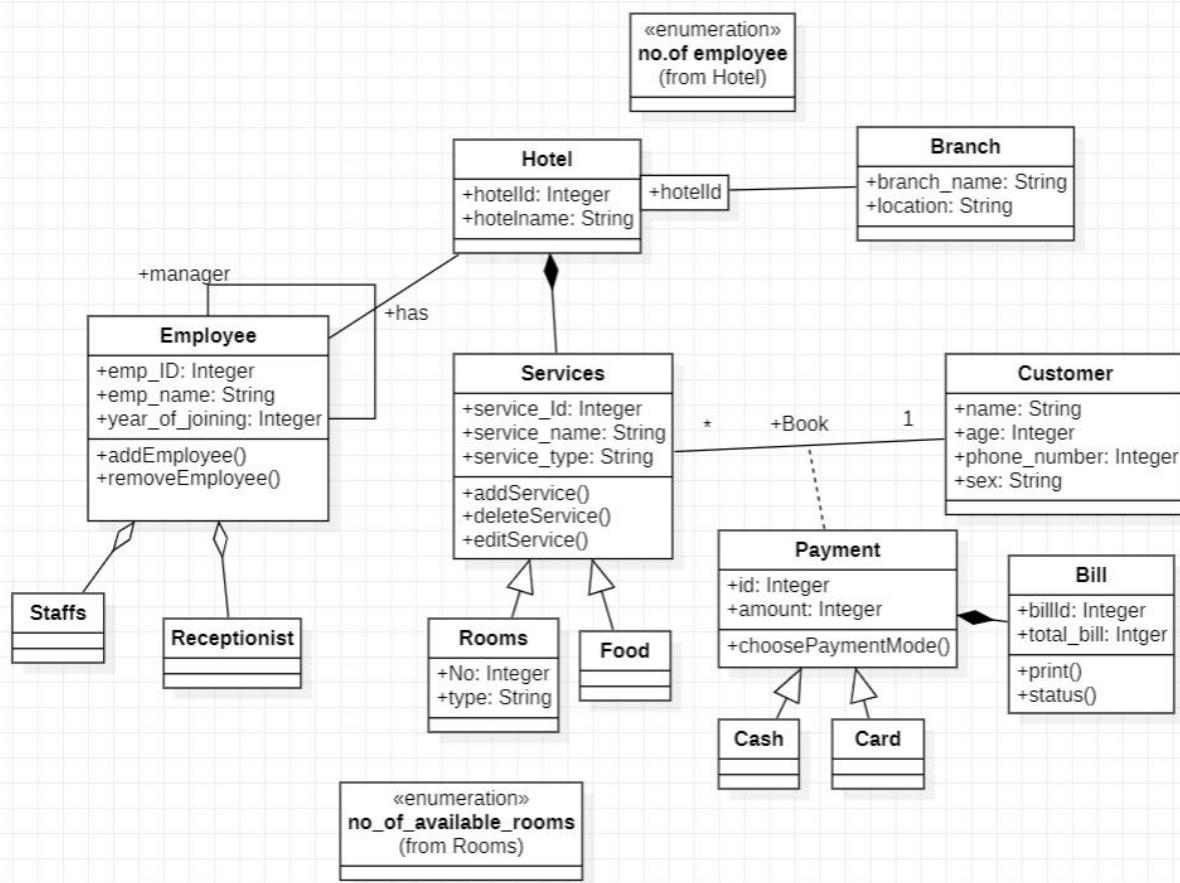
Testing and Quality Assurance: 4-6 weeks

Deployment and User Training: 2-4 weeks

Total Estimated Time: 28-40 weeks

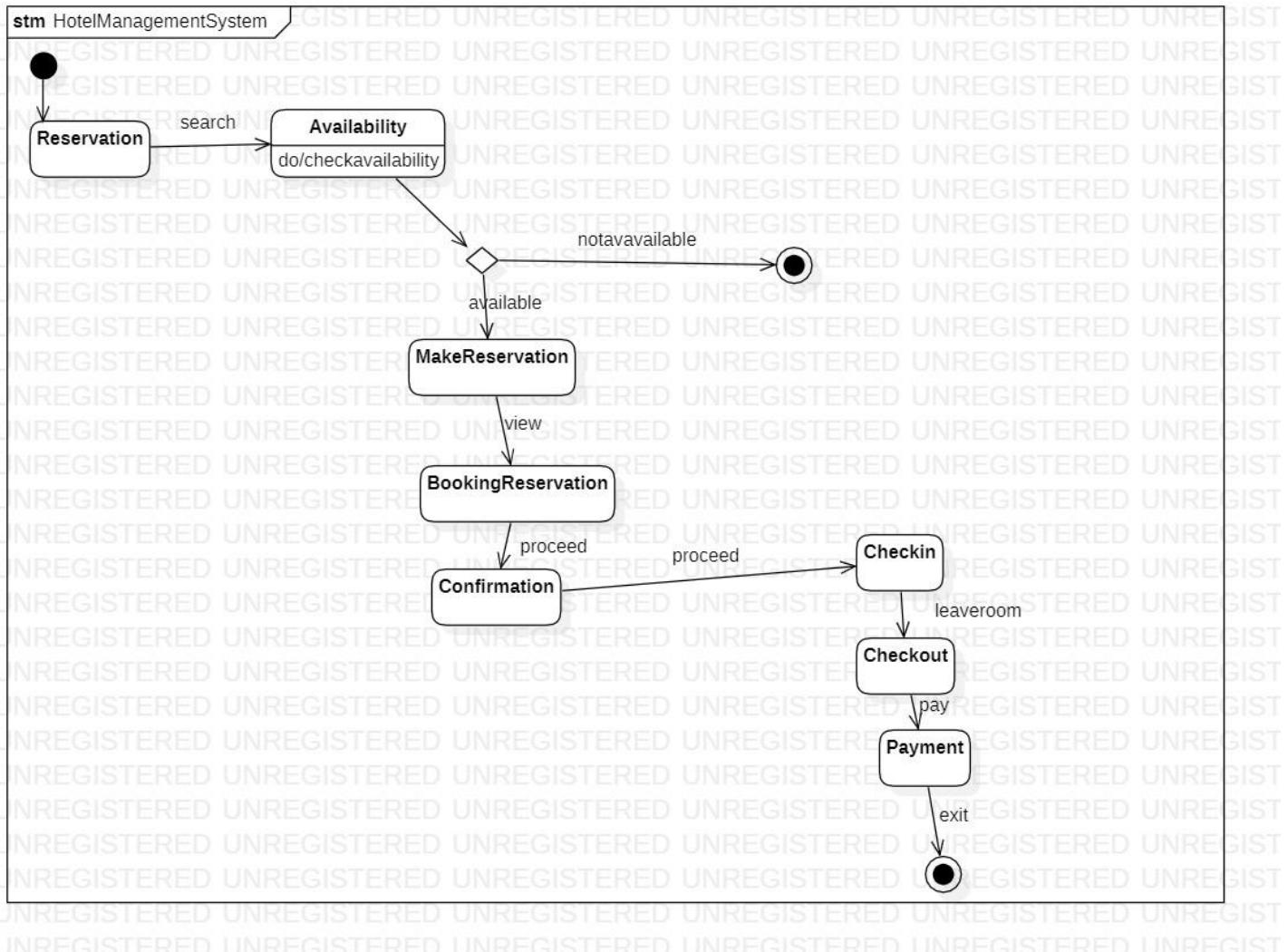
2.3 Class Diagram

Class diagrams shows following classes - Hotels has branches all over country. Hotel has employee which can be staff, receptionist,etc. Services like food, rooms, spa, etc are being provided by the hotel. Customer makes payment via cash or card after booking the room. Bill is given to the customer.

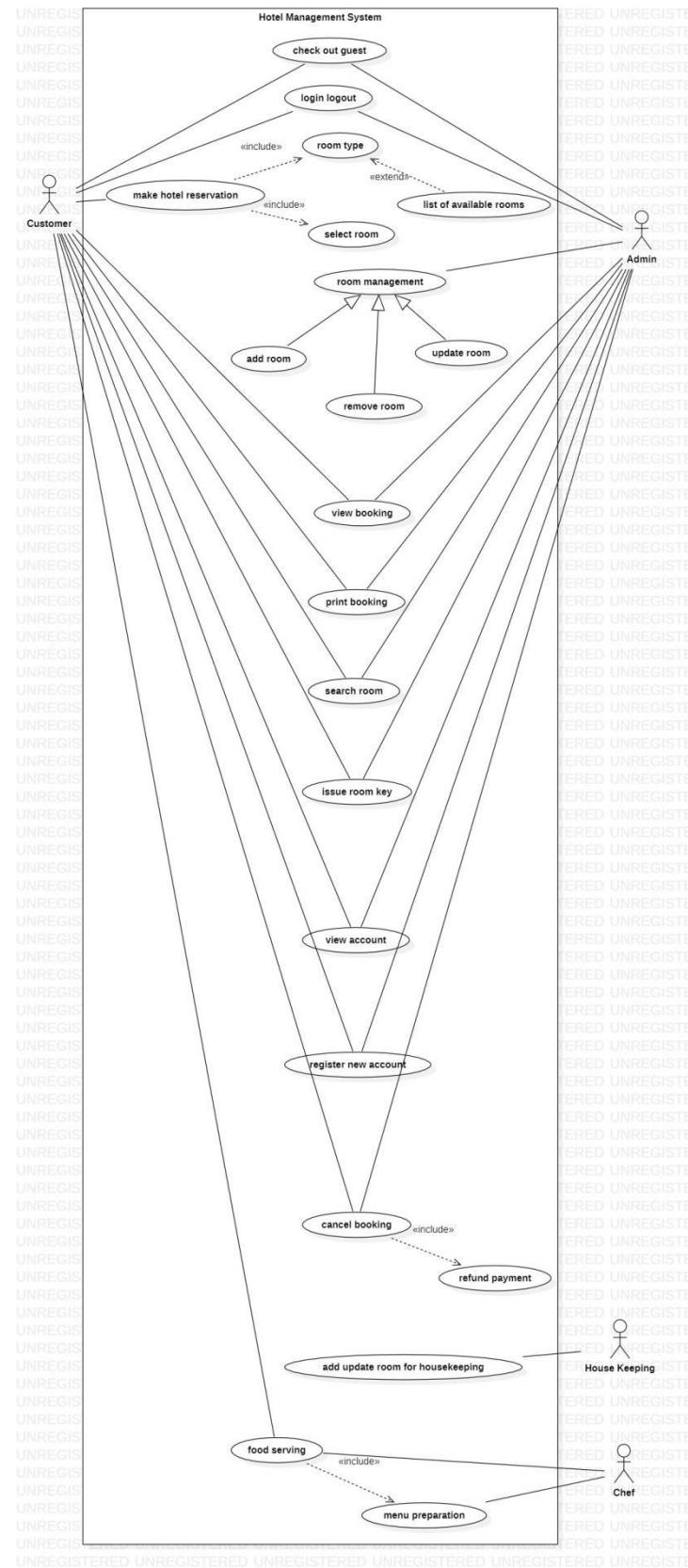


2.4 State Diagrams

Customer want to make reservation which is searched for availability. It exits if not available. If available, reservation is made by filling details and confirmation is received. Customer proceeds for check-in and payment is done after checkout.



2.5 Use Case Diagram



Actors:

Admin: the person who manages the whole systemCustomer: the person who stays in hotel.

House Keeping: the person who cleans.

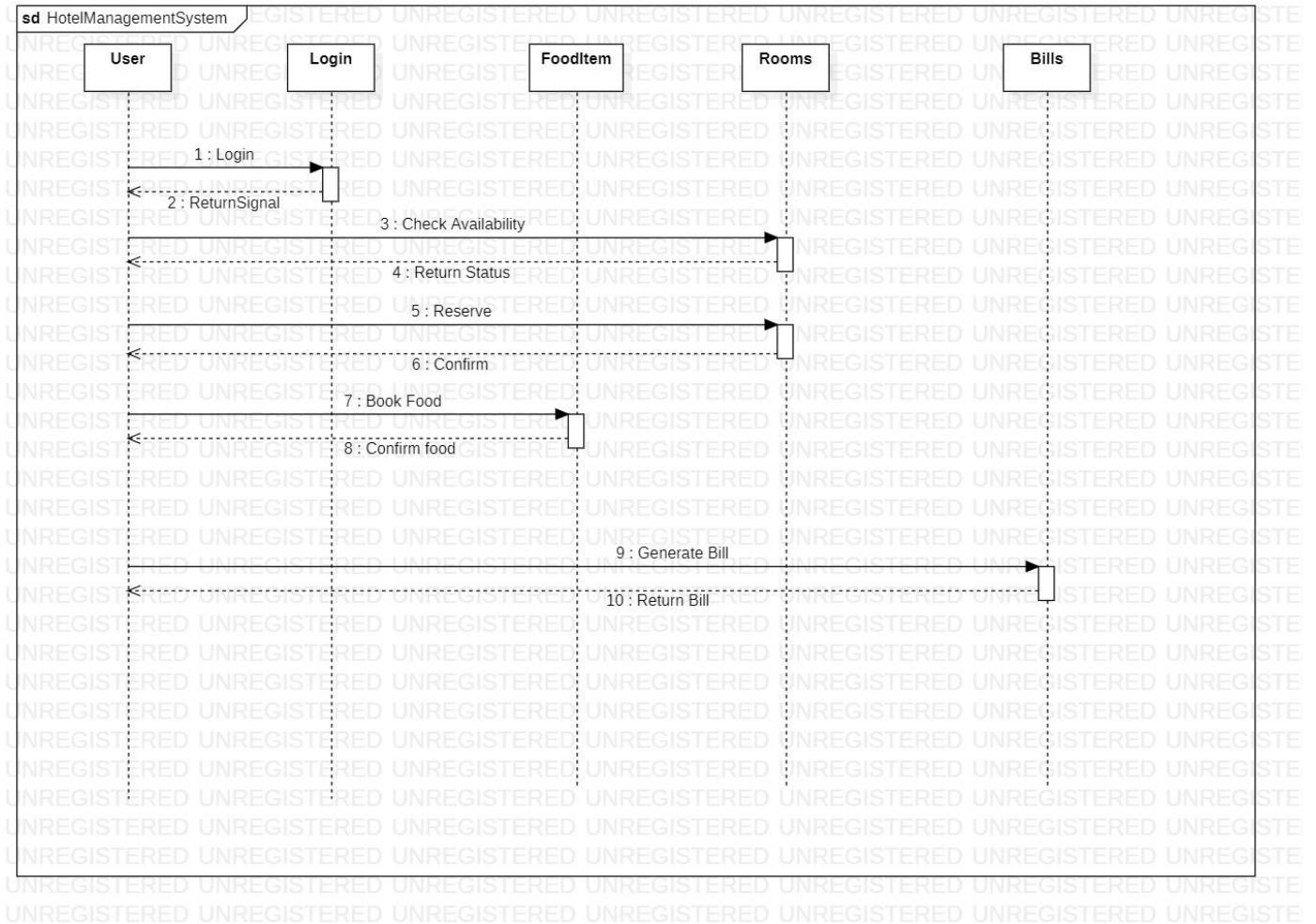
Chef : works in the kitchen

Use Cases:

Manage hostel : allows actor to update delete or add informationLogin : allows actors to login into the system. Add allottee: the students are allotted hostel rooms.

Book hostel: the student can select the hostel they wish to stay in.Pay fees: the fees payment is done by this use case.

2.6 Sequence Diagram



3. STOCK MAINTENANCE SYSTEM

3.1 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.

3.2 Software Requirement Specification

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 user-friendly interface, making it easy for the client to use. The system will automate inventory management processes, reduce errors, and provide real-time inventory information. The system will be developed using an agile methodology, allowing for iterative development and quick deployment of new features.

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.

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.

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.

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.

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.
- 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.
- 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 ad

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.

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

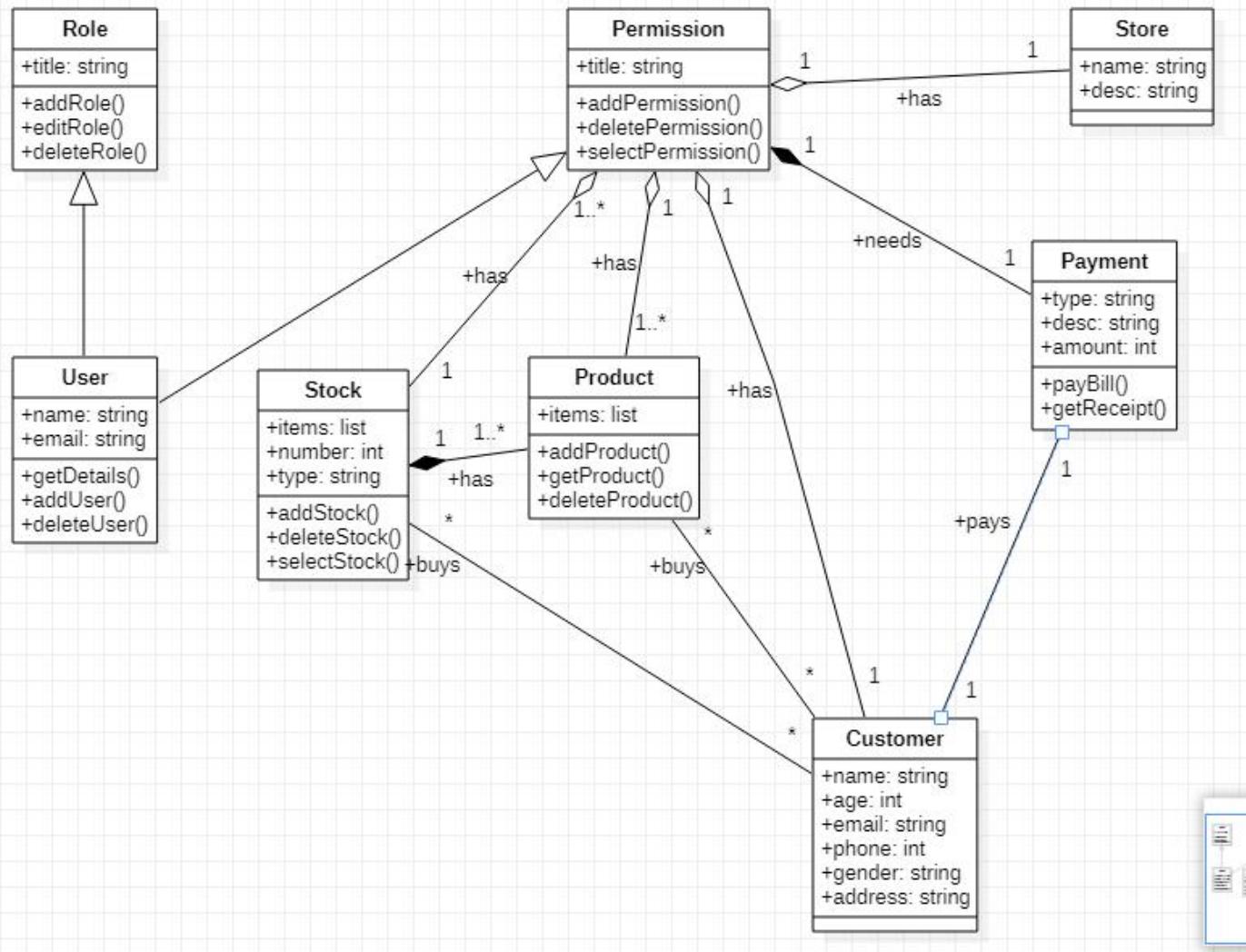
Quality Assurance and Testing: Duration: 4 weeks, Budget: \$20,000

Deployment and User Training: Duration: 2 weeks, Budget: \$10,000

Total Duration: 25 weeks

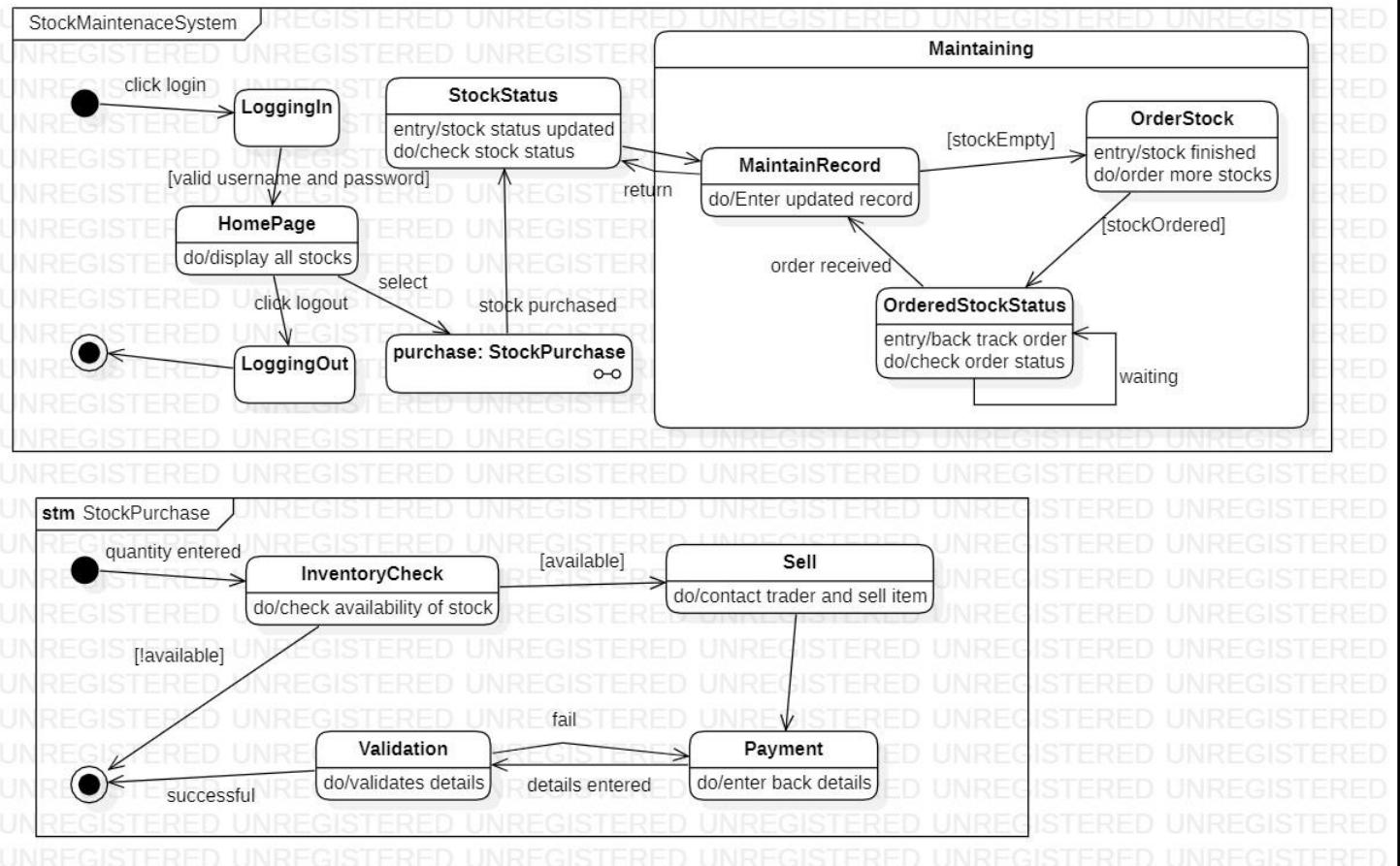
Total Budget: \$125,000

3.3 Class Diagram



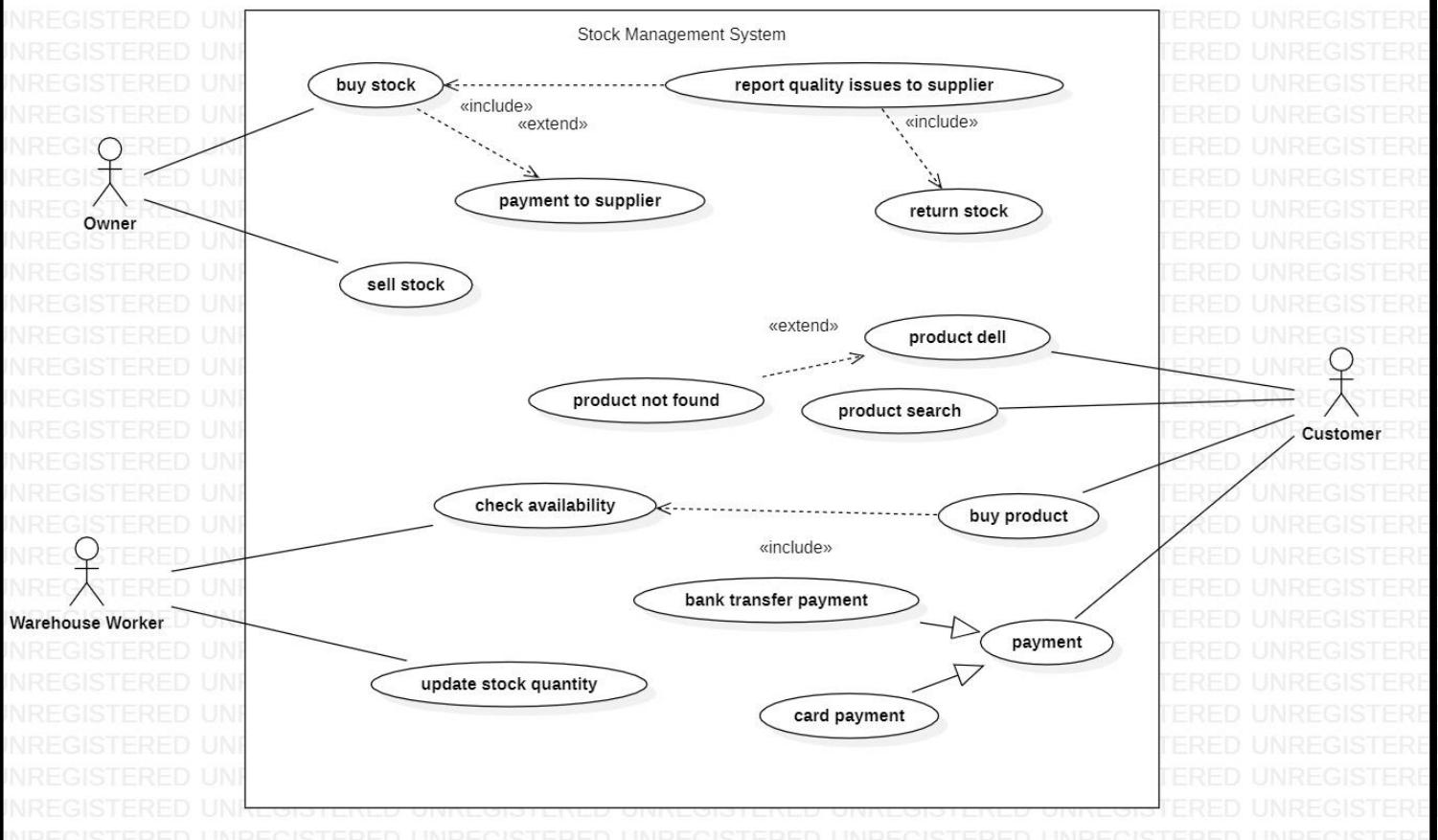
The products are displayed in stores across the city or world. All the information regarding the store are used to locate any product. The stores can be of many types. Some of them are departmental stores, super markets and ware houses where the products are kept for display. The vendor deals with the information about the details of the suppliers giving product to the organization. The stock of the products is maintained separately. The stock deals with information about the details of the product that the concern handling.

3.2 State Diagrams



The state diagram above gives us the states involved in purchasing a product and placing the order for the same. There is first an inventory check, where stock of products is noted and if the stock is less than minimum an order is placed by first searching for suitable trader. If a suitable trader is found, the order is placed and verified by the accountant. After the accountant has verified a payment is made for the products purchased.

3.3 Use Case Diagram



Actors:

Customer: a person who purchases the products
 Owner: a person who sells the products

Warehouse Worker: a person who supplies the products

Use Cases:

Buy Stocks: allows a user to purchase the stocks
 Make payment: accepts the payment

Update stock Quantity : keeps track of the stock supplied

Product sell : sell stocks

Prepare bill:a bill for products purchased is made

3.4 Sequence Diagram

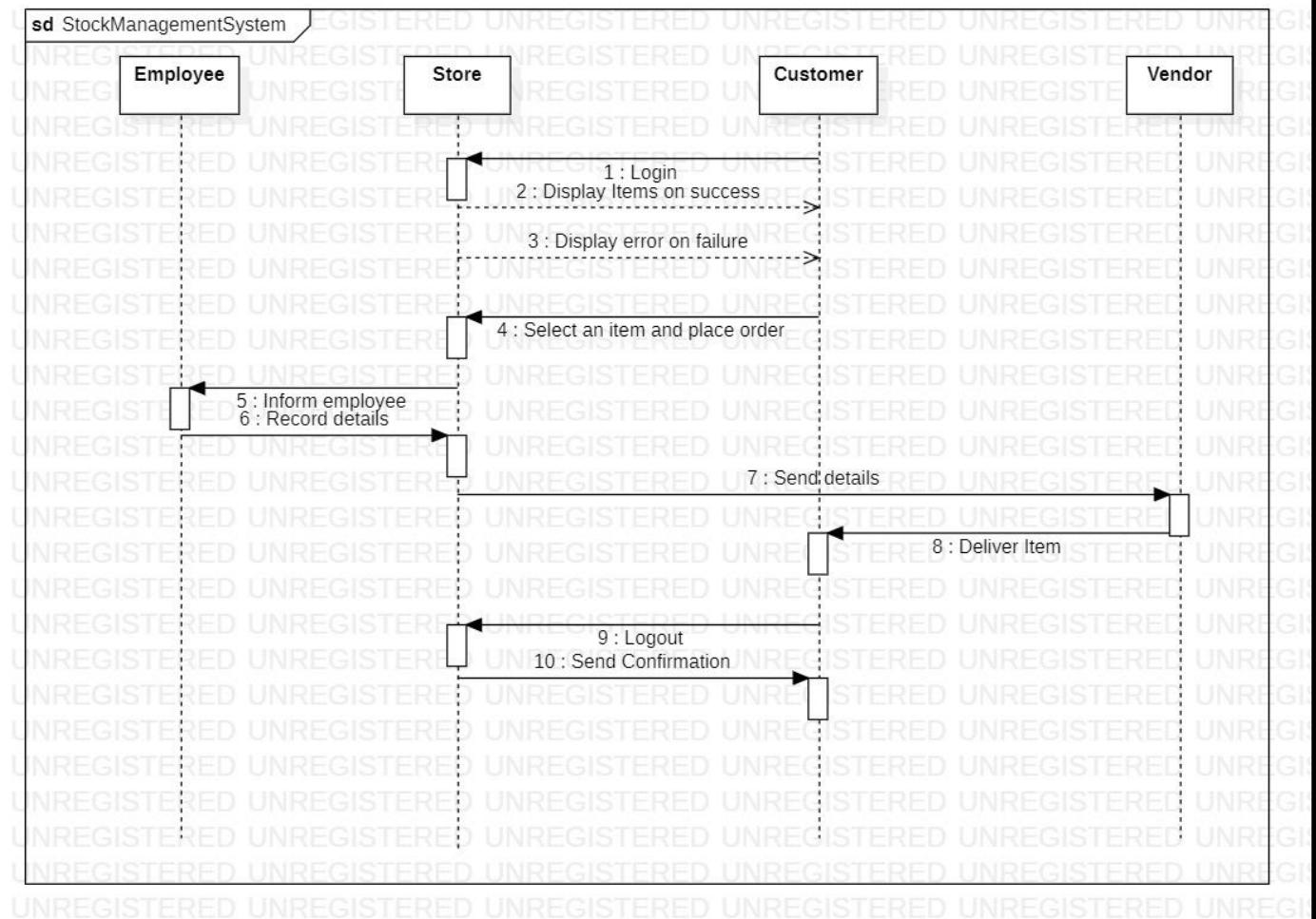
Customer will log in and store displays success/failure accordingly.

Select item to purchase and place order.

Inform the employee and the store records the details.

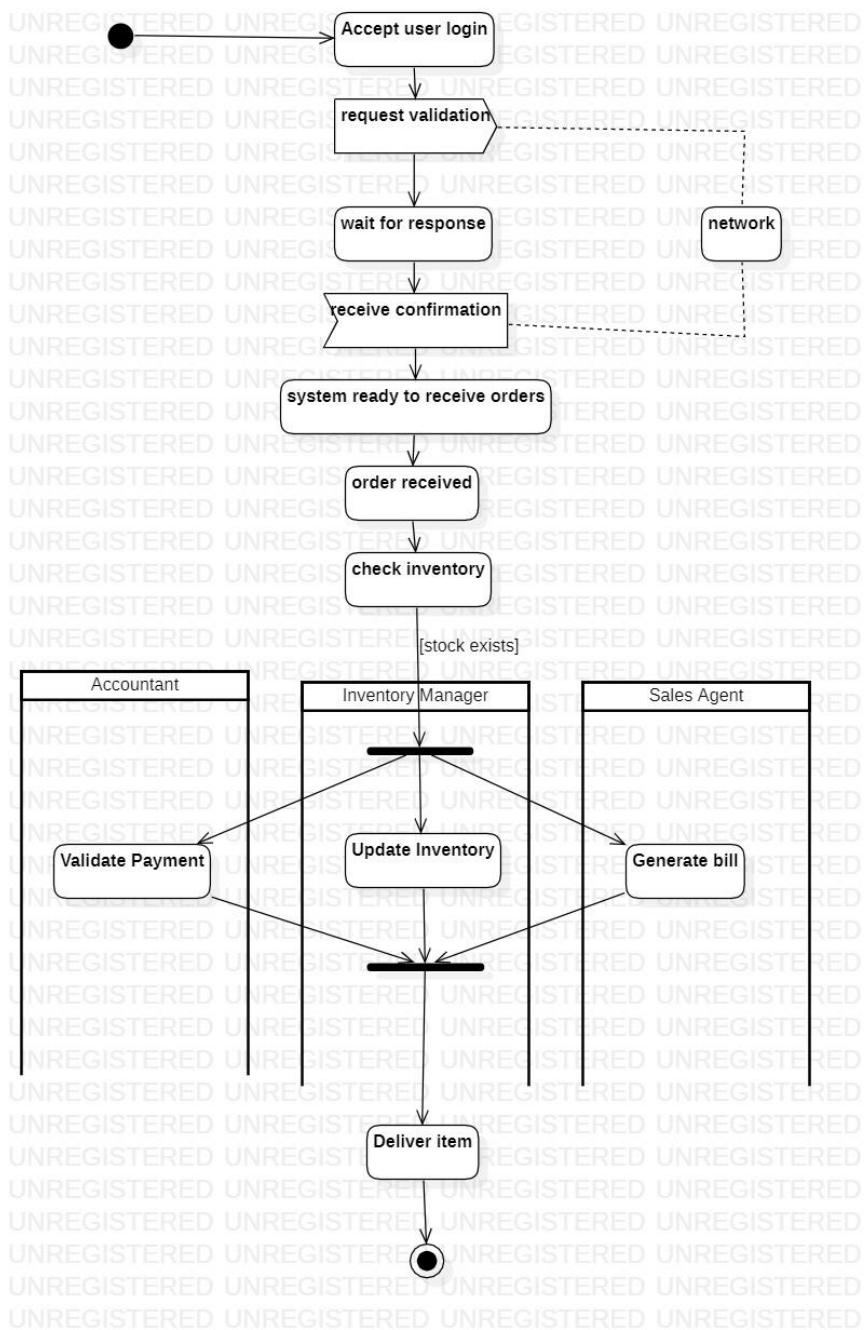
Send the details to the vendor and vendor will deliver the item to the customer.

Customer will then log out and store will send confirmation.



3.5 Activity Diagram

This diagram gives us the states involved in purchasing a product and placing the order for the same. There is first a validation check on login. System requests for validation and receives a confirmation which is shown by pentagon here. The system is then ready to receive the order. An inventory check is done, where stock of products is noted and if the stock is less than minimum an order is placed by first searching for suitable trader. After order is placed, payment validation. Updation of inventory and bill generation takes place simultaneously. Finally the item is delivered.



4. LIBRARY MANAGEMENT SYSTEM

4.1 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.

4.2 Software Requirement Specification

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 back-end 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 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

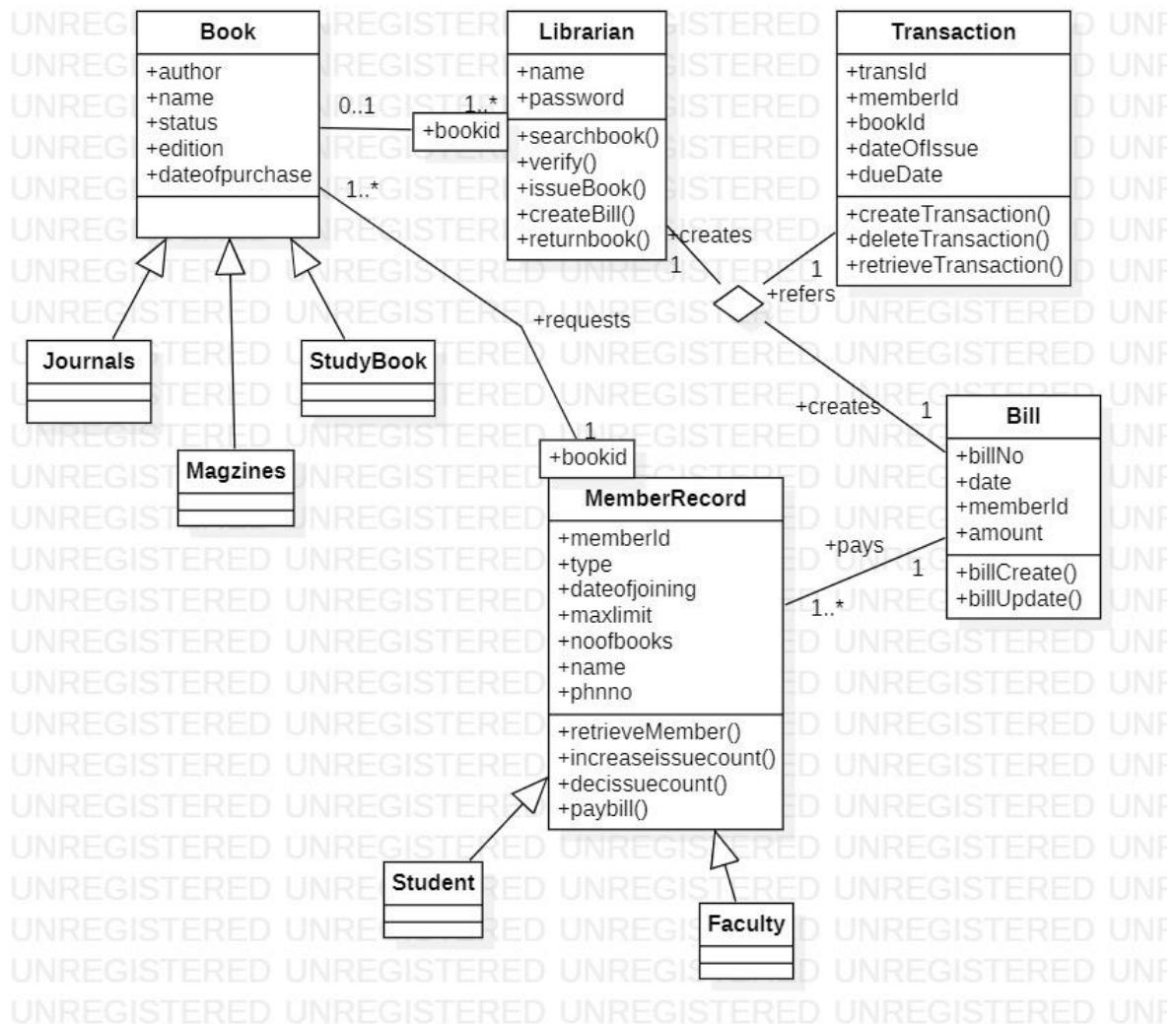
- Search for books using various criteria (e.g., title, author, genre)
- Search for members using various criteria (e.g., name, address, membership ID)

*Reporting

- Generate reports on book circulation
- Generate reports on overdue books
- Generate reports on fines

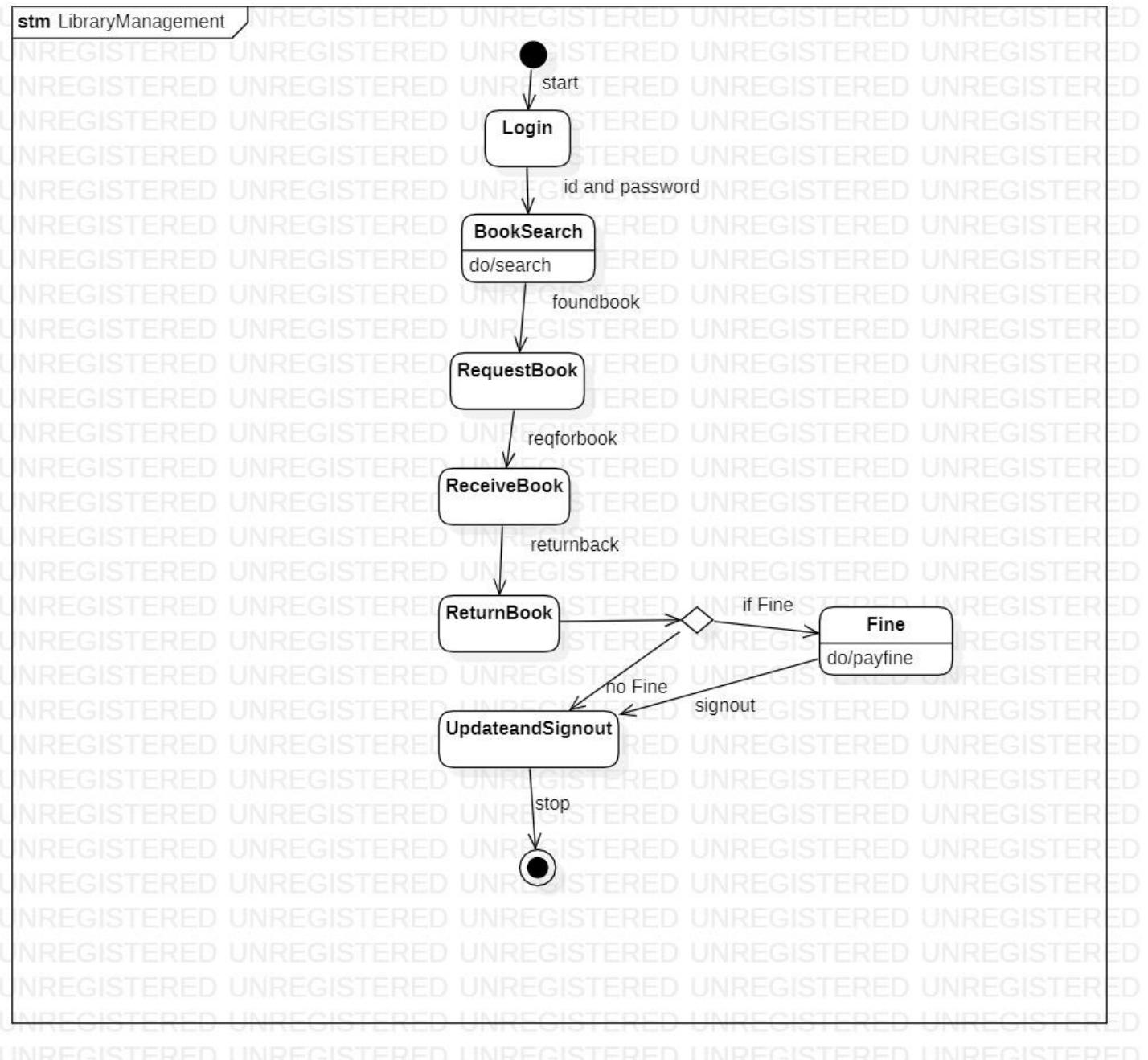
4.3 Class Diagram

Different types of books like journal, magazines and study books are in the library. These books are managed by a librarian. A record of all the members like teachers and students is maintained. The member pays the bill if there is late return or fine. Transaction is seen by librarian.

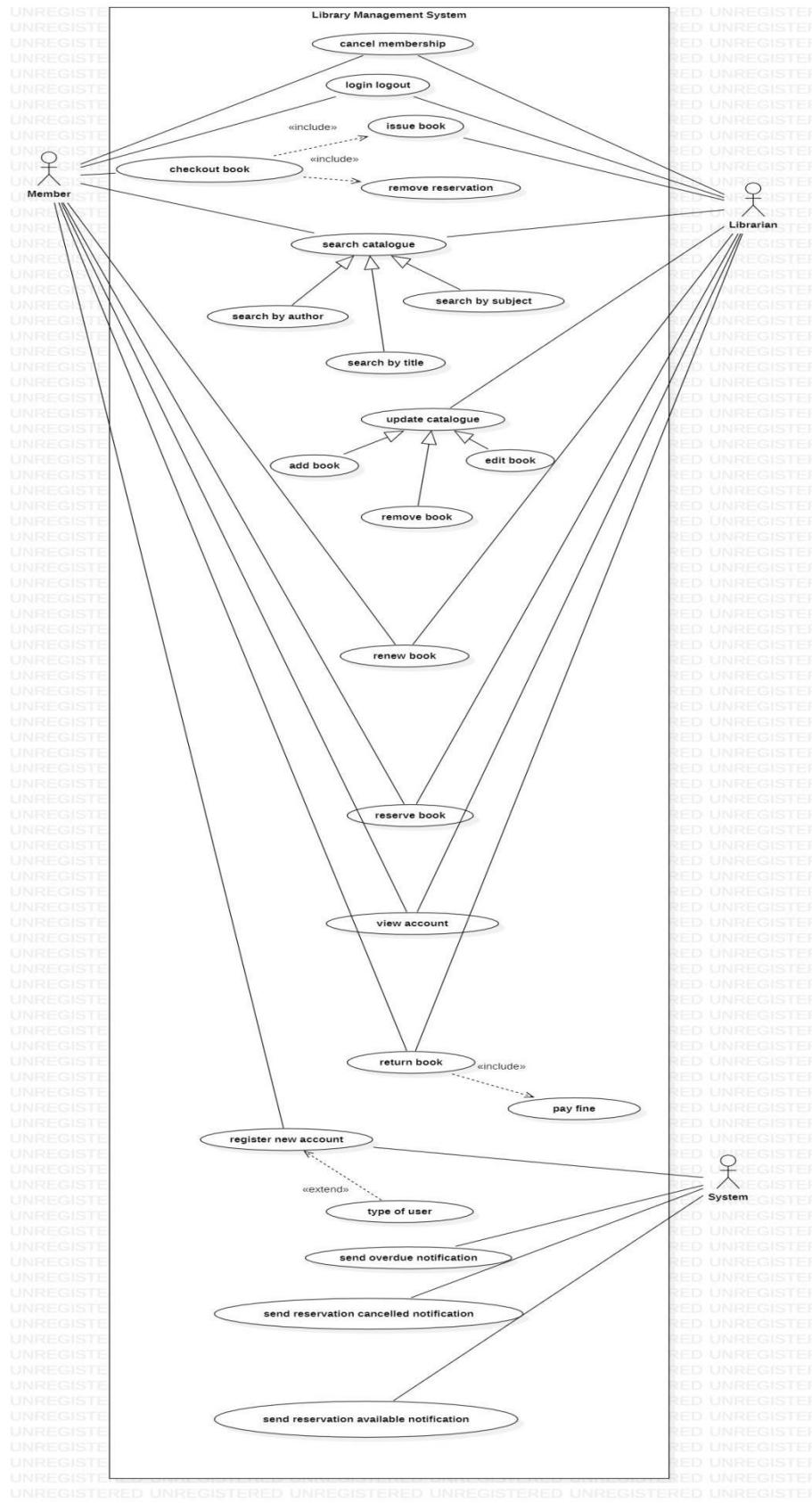


4.4 State Diagrams

The member will login using valid id and password and search for books they need. If the book is found they request the book and receives the book by librarian. They return the book before the due date , update the database and sign out. If not returned on time, fine is paid by them.



4.5 Use Case Diagram



Actor:

Librarian: who manages the whole system

System : the management system

Member: person using the library services

Use Cases :

Searching catalogue by author, title, or subject.

Login/Logout

Create account

Cancel the membership

Update the catalogue is done by librarian. They add,remove or edit the books.

Reserve book

Renew book before due date.

View account

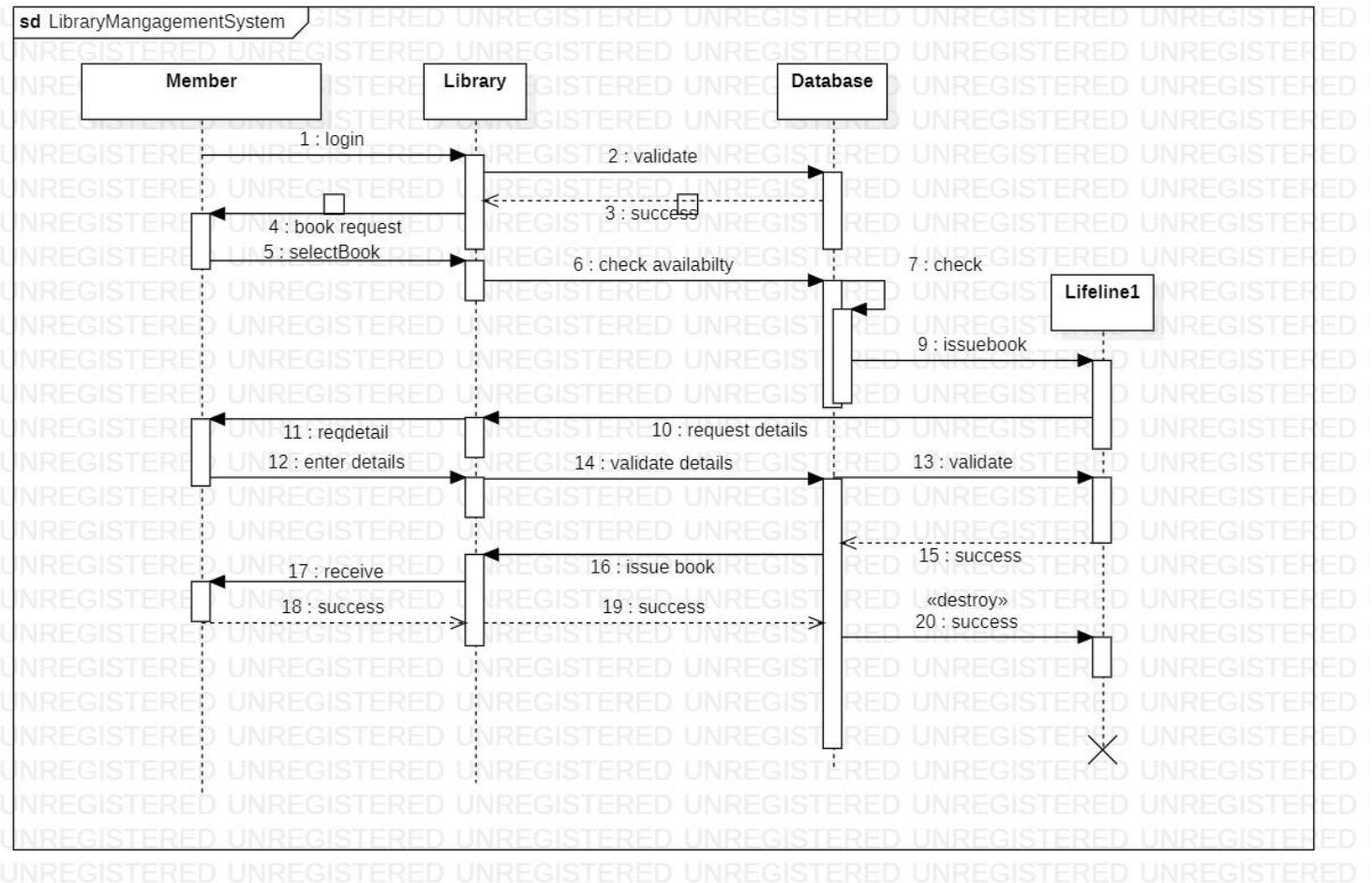
Return book. If not on time, pay fine.

4.6 Sequence Diagram

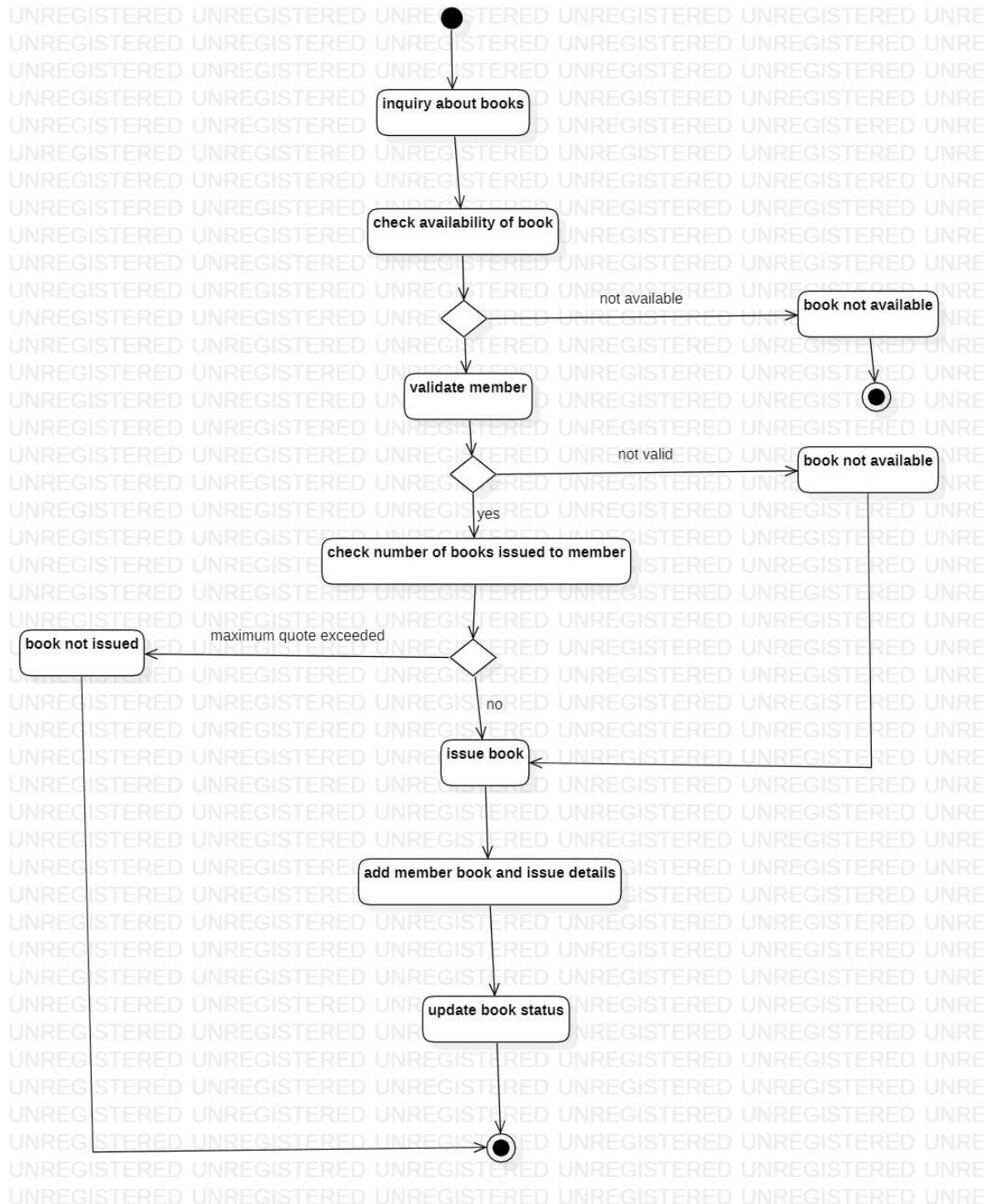
Member logs in and selects book. Library checks availability and issues book.

Database requests the member details. Member enters the details which is validated by system..

Library then issues the book and updates the database.



4.7 Activity Diagram



5. ONLINE SHOPPING SYSTEM

5.1 Problem statement

The Online Shopping System for all kind of products web application is intended to provide complete solutions for vendors as well as customers through a single get way using the internet. It will enable vendors to setup online shops, customer to browse through the shop and purchase them online without having to visit the shop physically. The administration module will enable a system administrator to approve and reject requests for new shops and maintain various lists of shop category. This system allows the customer's to maintain their cart for add or remove the product over the internet.

5.2 Software Requirement Specification

1.1 Purpose: 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: 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: An online shopping system typically operates on a web-based platform, accessible through a website or mobile application. It provides customers with a seamless shopping experience, enabling them to browse products, add items to their cart, and checkout securely. The system also provides businesses with a way to manage their inventory, process payments, and track orders. Overall, online shopping systems have revolutionized the way people shop and do business, providing a convenient, cost-effective, and efficient solution for both customers and businesses.

2. General description:

An online shopping system is a software application that enables customers to purchase goods or services over the internet. The system typically consists of a web-based platform, accessible through a website or mobile application, that provides customers with a seamless shopping experience. The system also provides businesses with a way to manage their inventory, process payments, and track orders.

3. Functional Requirements:

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.
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.

4. Interface Requirements:

1. User Interface: The online shopping system should have an intuitive and user-friendly interface that allows customers to easily navigate and complete their purchase.
2. Payment Interface: The payment gateway should have a secure interface for customers to enter their payment information.

3. Order Tracking Interface: The system should provide an interface for customers to track their order status.

5. Performance Requirements:

1. System Availability: The online shopping system should be available 24/7 without any significant downtime.
2. Response Time: The system should respond quickly to user requests, ensuring a smooth and seamless shopping experience.
3. Scalability: The system should be scalable, allowing for a large number of users to access it simultaneously without any performance degradation.

6. Design Constraints:

1. Security: The system should be designed with security in mind, ensuring that customer information and payment details are kept secure.
2. Compatibility: The system should be compatible with different browsers and devices to ensure that customers can access it from any device.
3. Accessibility: The system should be designed to be accessible to all users, including those with disabilities.

7. Non-Functional Attributes:

Non-functional requirements of an online shopping system refer to the characteristics that describe how the system should behave, rather than what the system should do. The non-functional requirements of an online shopping system typically include the following:

1. Usability: The system should be easy to use and navigate, with an intuitive user interface that does not require significant technical knowledge.
2. Performance: The system should be designed to handle a large number of users simultaneously without any significant slowdowns or delays.
3. Security: The system should be designed with robust security features, including encryption of sensitive data and protection against hacking and cyber attacks.
4. Reliability: The system should be reliable and available at all times, with minimal downtime or system failures.

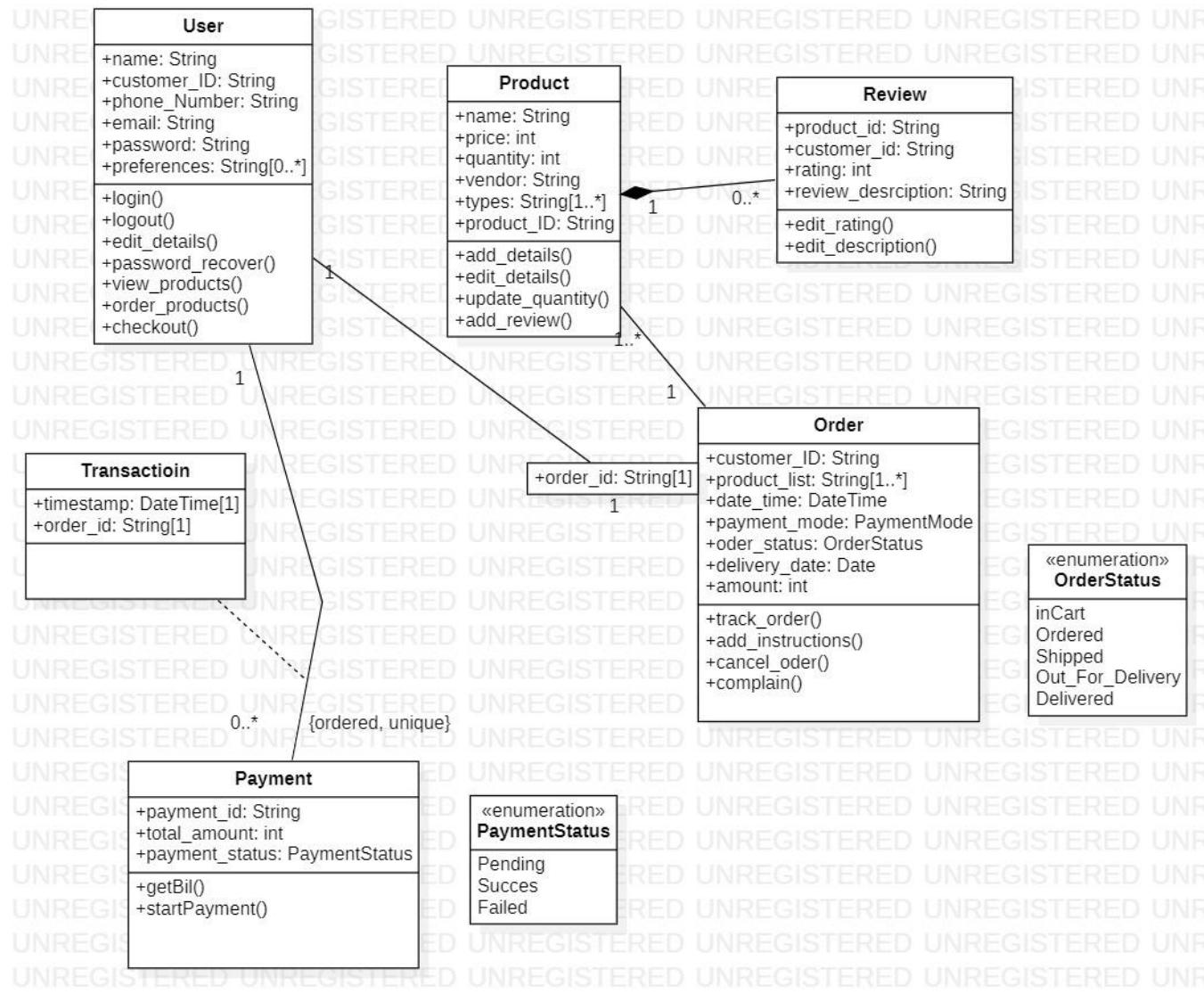
8. Preliminary Schedule and Budget:

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

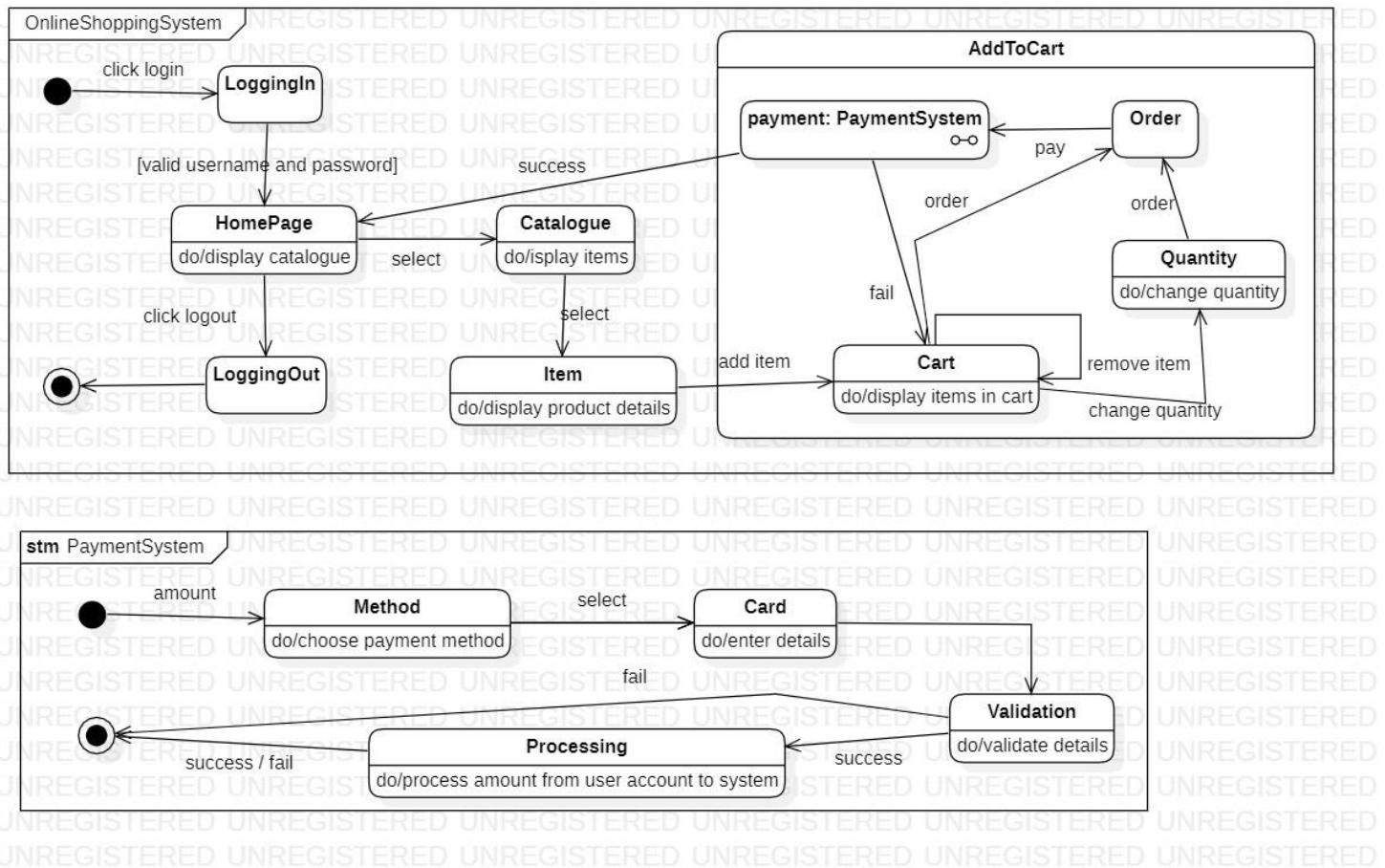
The budget for developing an online shopping system can vary significantly depending on the features and functionality required. The budget should include the cost of development, hosting, and ongoing maintenance and support. A preliminary budget estimate for a basic online shopping system could range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

5.3 Class Diagram



The online shopping system has customers who must have an account in the online website where he/she can purchase products. If customer wants to buy the product then he/she must be registered, unregistered user can't go to the shopping cart. Customer login to the system by entering valid user id and password for the shopping. The products sold for customers are sold for various categories like men, women, kids and home products. After the payment or surf the product the customer will logged out.

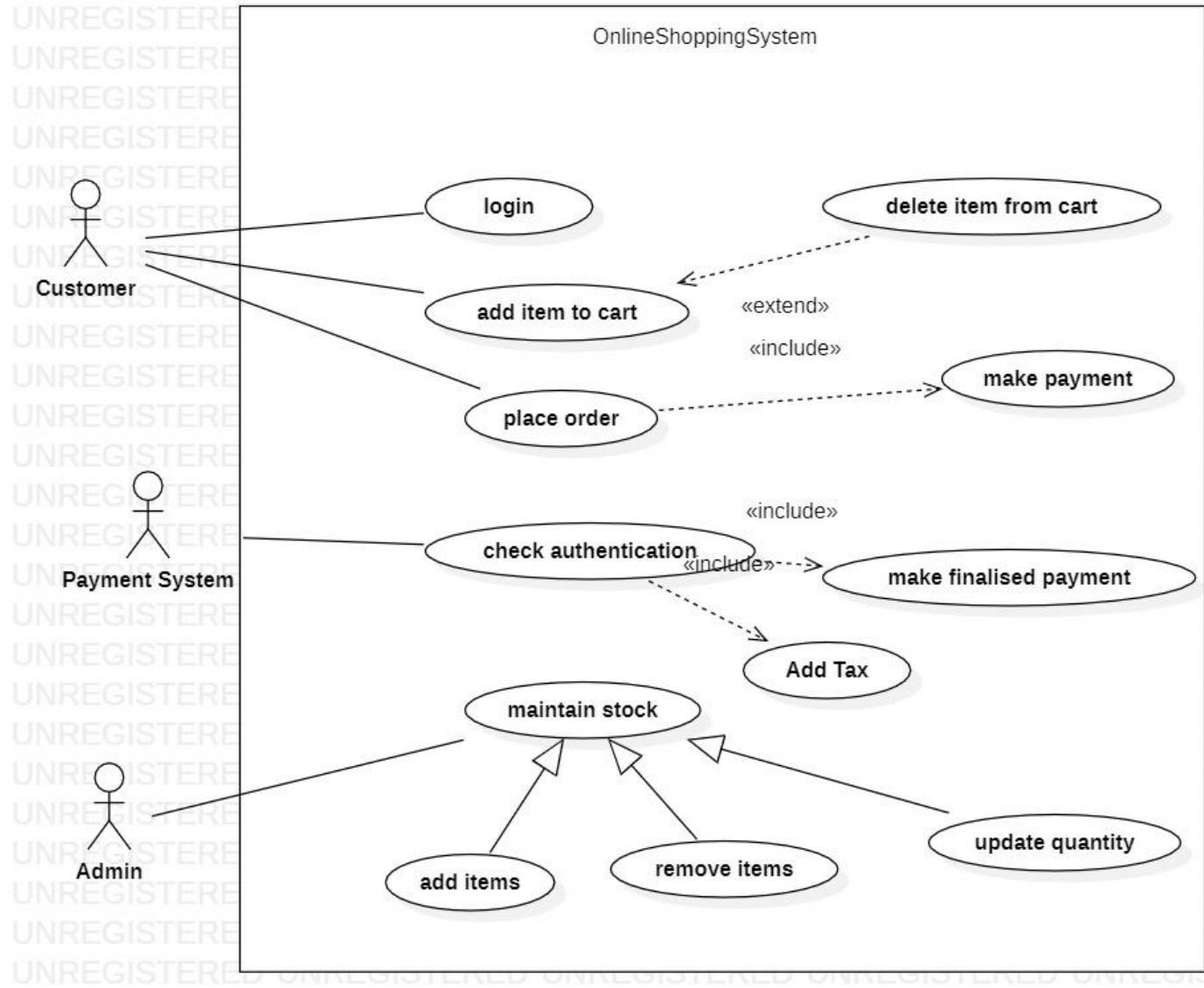
5.4 State Diagrams



The simple state chart diagram gives us states in purchasing a order and paying for the order. The customer is first mad to register and then login into their account. Then the items are displayed,where they can select their choice and add them to cart or reserve or order them. The transaction details are displayed.

The advanced state chart diagram has states explaining the product purchase and payment. It has two sub machines i.e product selection and checkout product. Product selection allows us to select products and add them to cart. Checkout product has states explaining the payment methods and validating the methods.

5.5 Use Case Diagram



Actors:

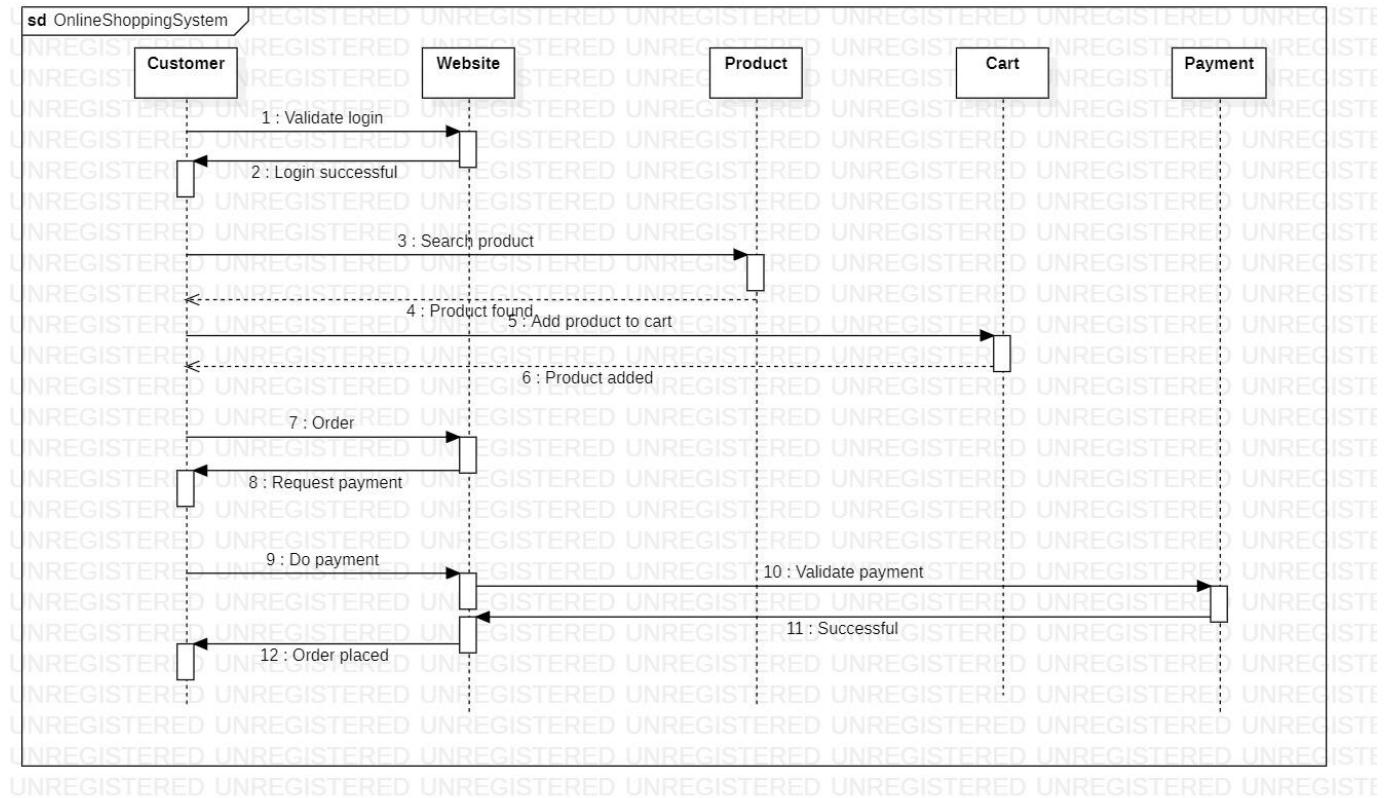
Customer: a person who uses the online shopping system
 Admin: person who supplies products and keep stocks ,
 Payment System: person who handles the payment

Use Case:

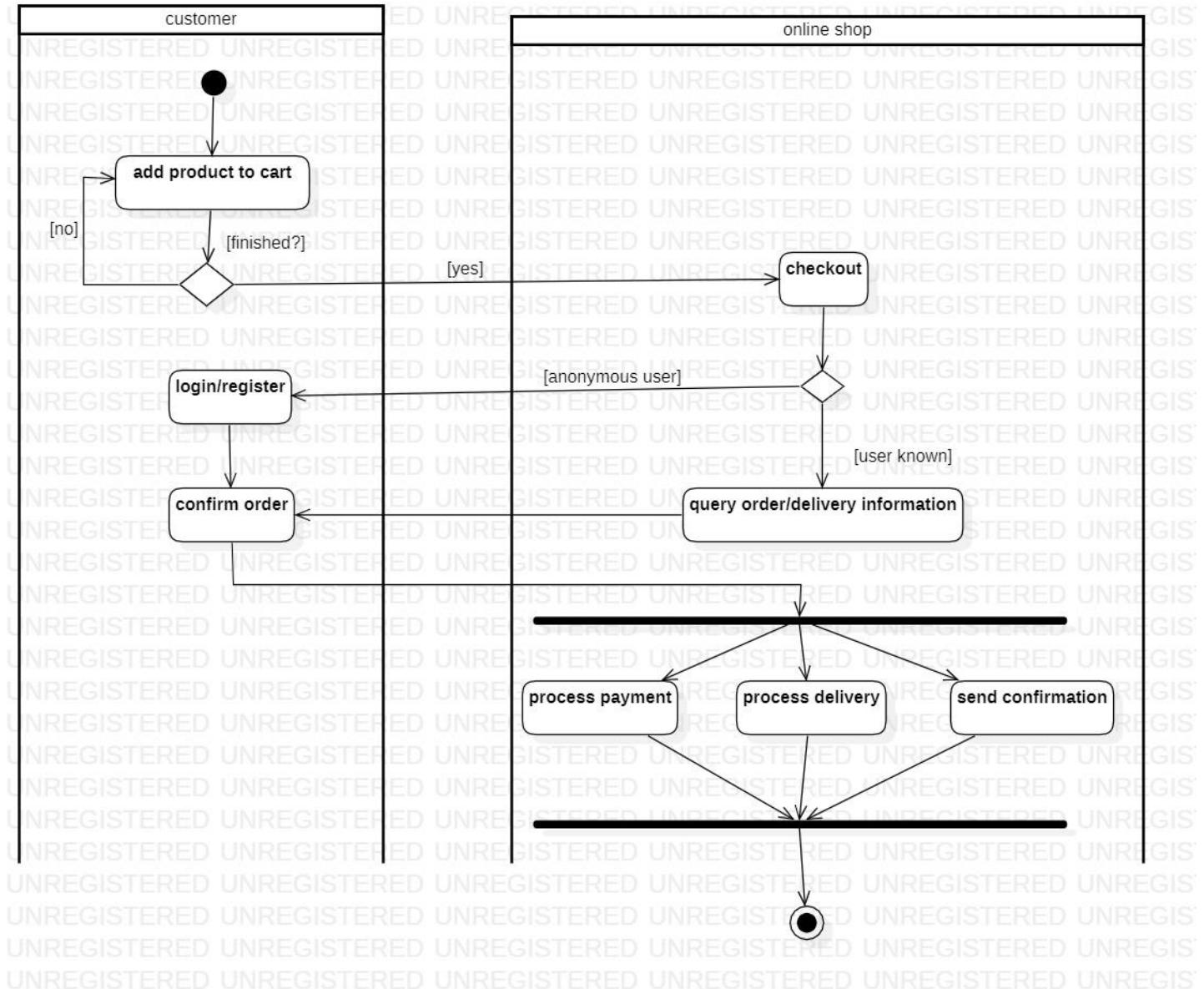
Login: customer logins, View product details : displays all product details
 Place order : order the items present in the cart, Make payment : accepts payment for the products purchased
 Deliver product : delivery of the product is handled, Supply Product : product supply is maintained.

5.6 Sequence Diagram

Customer logs in and searches for product to buy. He adds it to cart and continues ordering. The website requests payment which is then done by customer. Payment is validated and order is placed.



5.7 Activity Diagram



6. RAILWAY RESERVATION SYSTEM

6.1 Problem statement

Railway Reservation System is a system used for booking tickets over internet. Any Customer Can book tickets for different trains. Software has to be developed for automating the manual reservation system of railway. The system should be standalone in nature. It should be designed to provide functionalities like booking of tickets in which a user should be able to apply for tickets of any train and of any class. The software takes the current system date and time as the date of issue and calculates the amount to be paid by the user. It also provides the functionality of cancellation of tickets.

6.2 Software Requirement Specification

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.

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.
- 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 the Railway Reservation System is expected to take approximately 6 months, with the project divided into several phases. The estimated time for each phase is as follows:

Requirements gathering (1 month),

Design (2 months),

Development (2 months),

Testing (1 month), and

Deployment (2 weeks).

Budget:

The estimated budget for the development of the Railway Reservation System is \$500,000.

The budget will cover the cost of development, testing, deployment, and maintenance of the system for the first year. The breakdown of the budget is as follows:

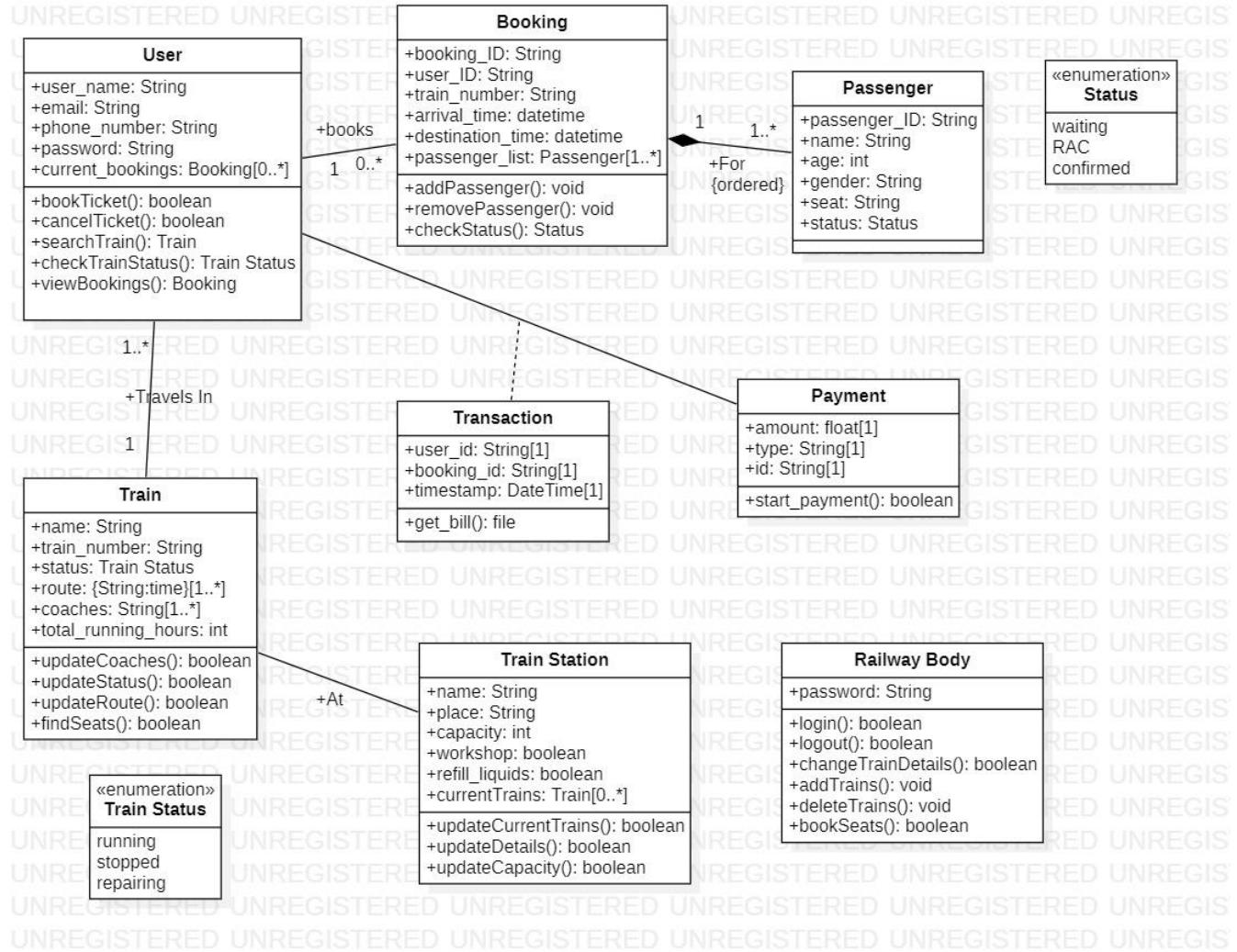
Development (\$350,000),

Testing (\$50,000),

Deployment (\$50,000), and

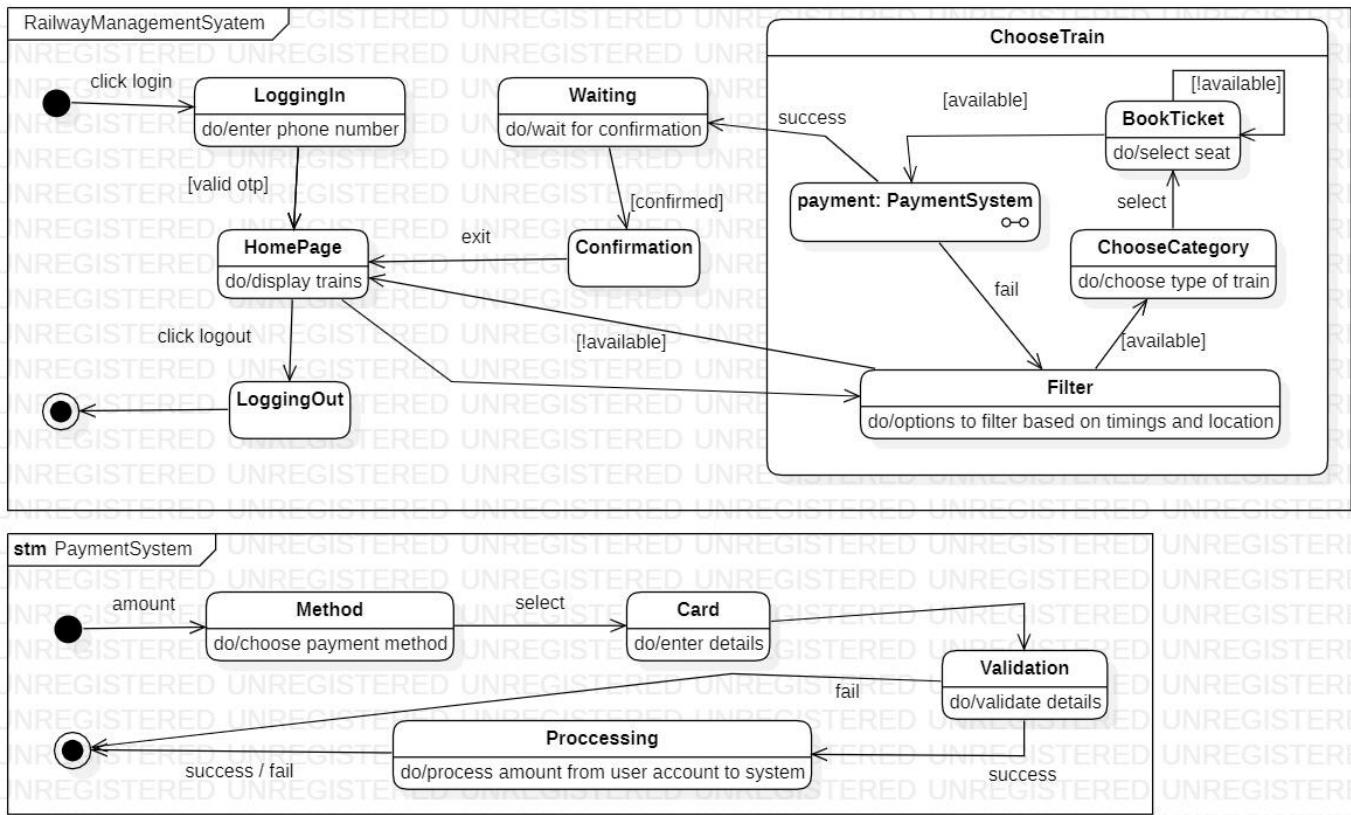
Maintenance (\$50,000).

6.3 Class Diagram



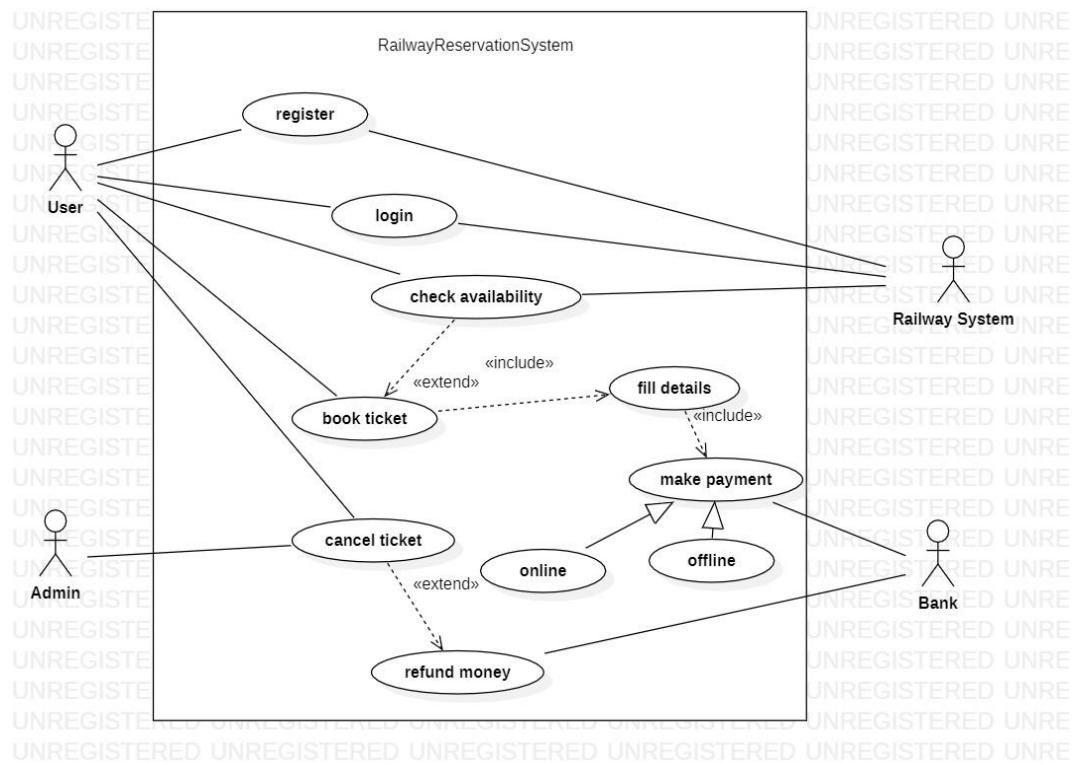
The admin manages the trains and reservation related to railway reservation system. There are three types of reservation, I.e RAC,waiting and confirmed.The passengers with a reservation goes to one or the other reservation.A train consists of coaches and engine.A passenger pays for the ticket booked .Tickets can be booked in two ways by i-ticket or by e-ticket booking.

6.4 State Diagrams



The simple state diagram gives the states involved in booking a train ticket and paying for the same. The user can see the train details and book a train for a particular source and destination . on timeout an error message is displayed and redirected to the main page. The user can then select a train and make payment for it.

6.5 Use Case Diagram



Actors :

User: person who has to do the reservation

Admin: person who does the reservation

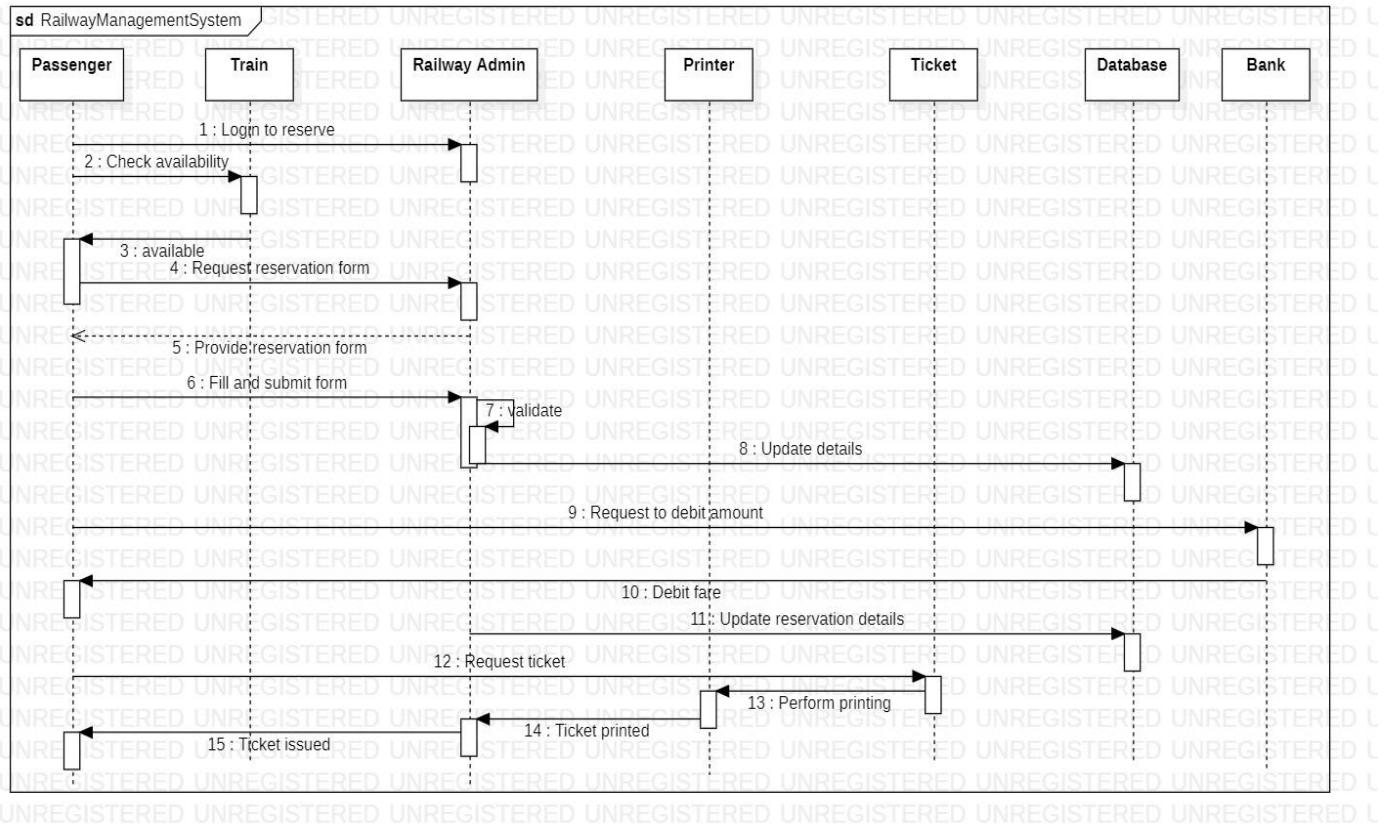
System; platform for reservation

Bank: for payments and validation

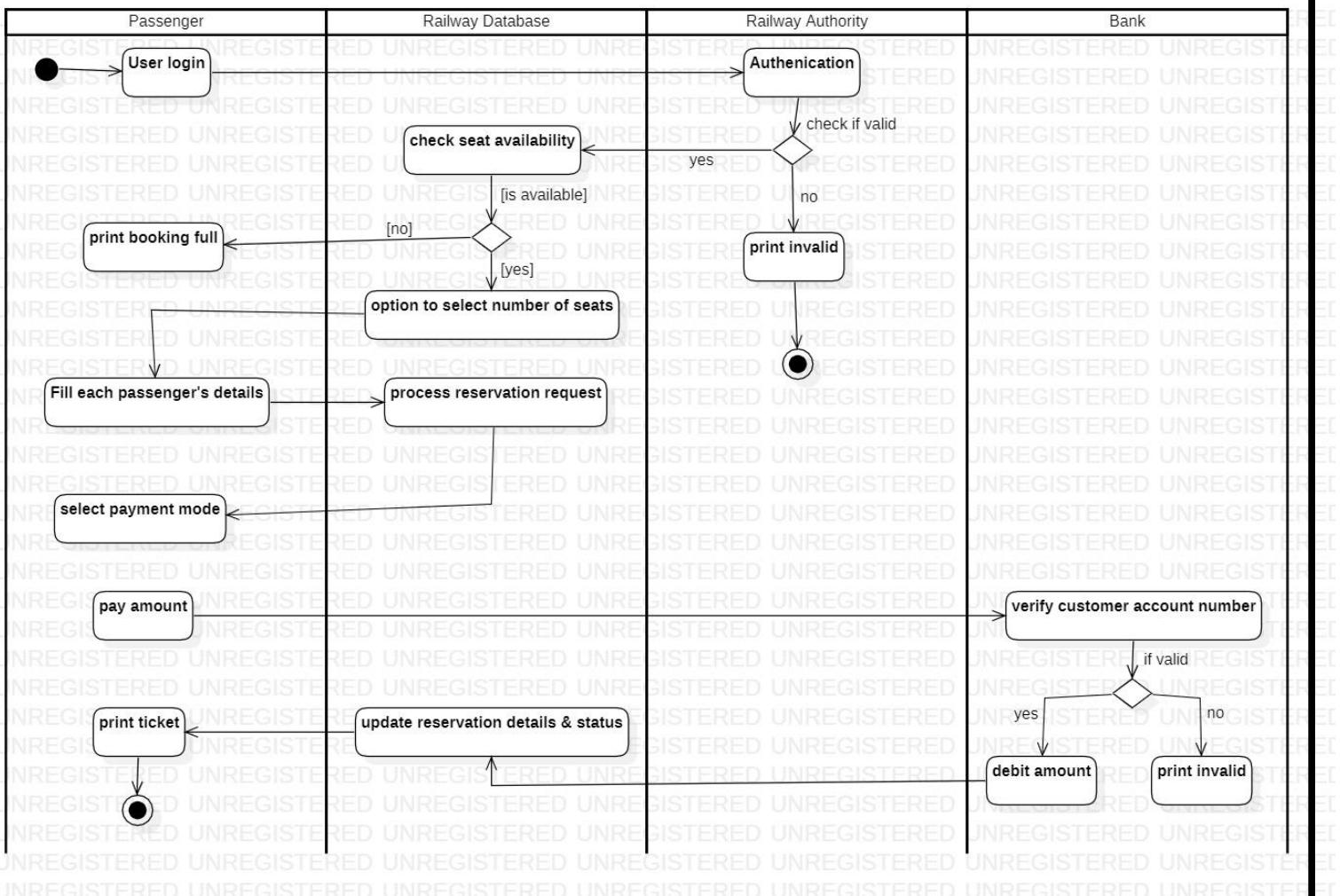
Use Case:

Register, Login, checking availability, booking tickets, canceling tickets, making payments.

6.6 Sequence Diagram



6.7 Activity Diagram



The activity diagram tells about the steps happening while canceling a ticket which is booked. The user first needs to login and select his ticket, confirm cancellation, request refund and print the canceled ticket and logout.

7. PASSPORT AUTOMATION SYSTEM

7.1 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.

7.2 Software Requirement Specification

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.

4. 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.

8. 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.

9. 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

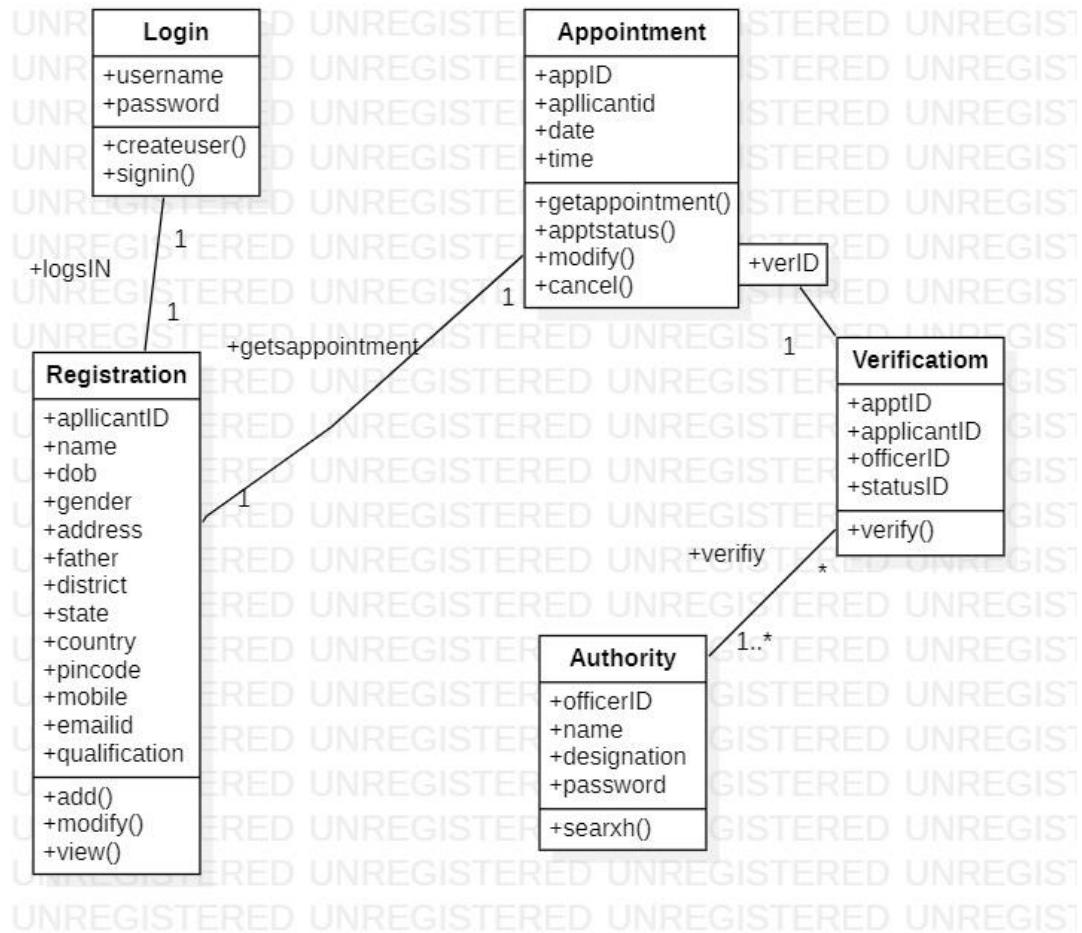
Contingency: \$50,000

Maintenance and support: \$70,000 per year

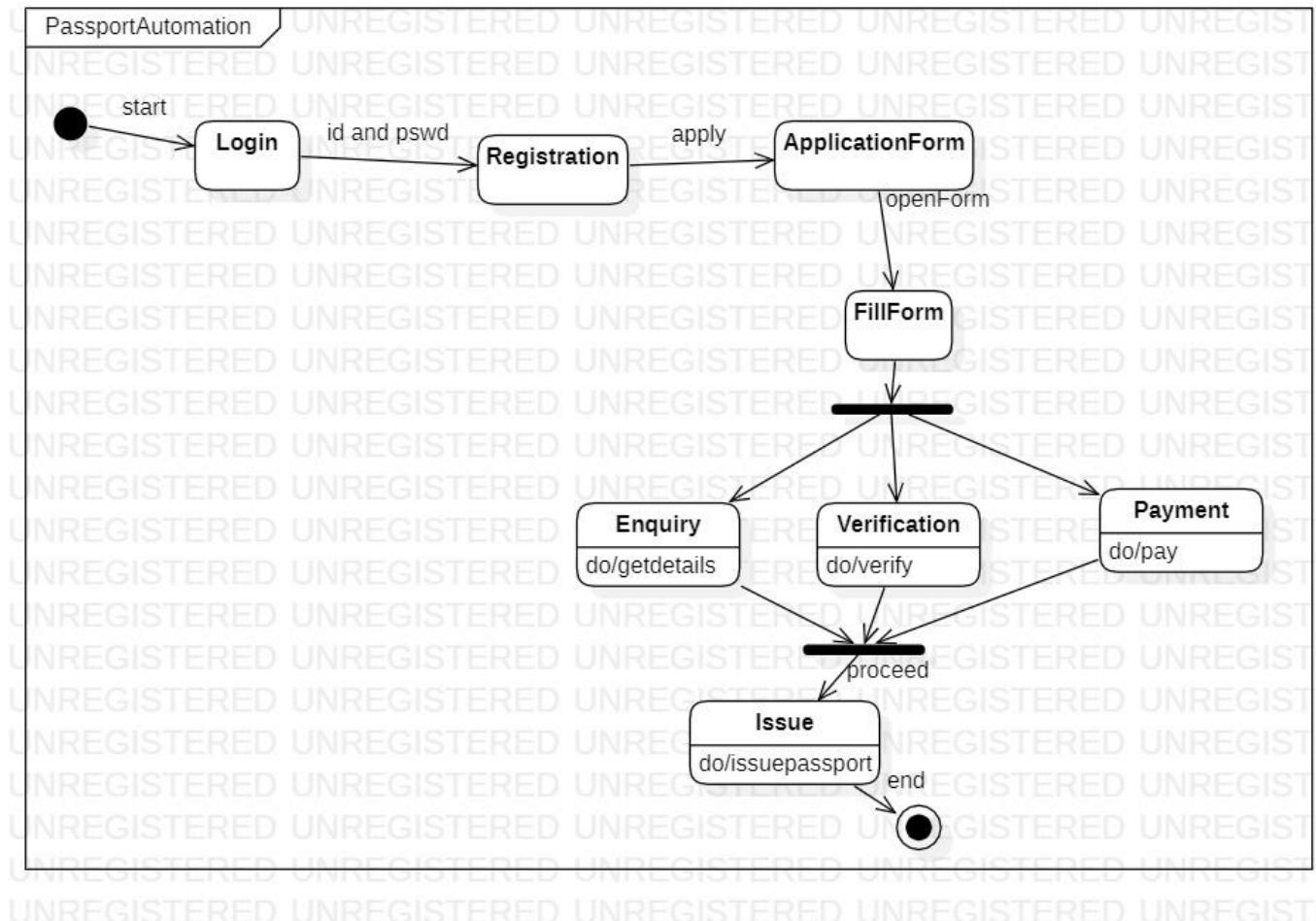
Total budget: \$500,000 (initial development and implementation) + \$70,000 per year

(maintenance and support)

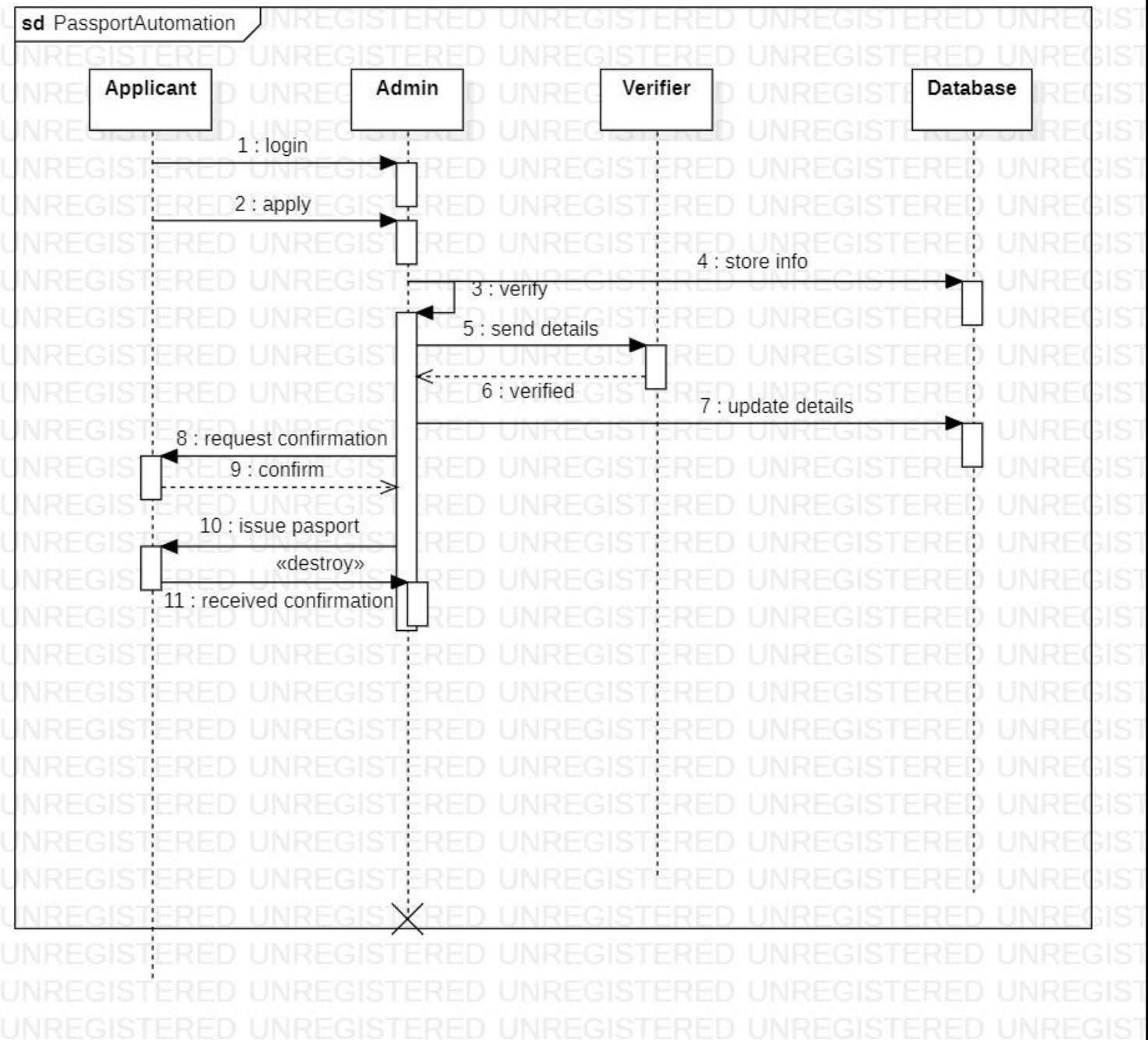
7.3 Class Diagram



7.4 State Diagrams



7.5 Sequence Diagram



7.6 Activity Diagram

