

B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU



OOMD Lab record

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:

Sameecha Sudheer

1BM20CS213

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

B.M.S COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



DECLARATION

We, Sameecha Sudheer (1BM20CS213) students of 6th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this OOMD lab has been carried out in Department of CSE, BMS College of Engineering, Bangalore during the academic semester March - July 2023. I also declare that to the best of our knowledge and belief, the OOMD Lab report is not from part of any other report by any other students.

Signature of the Candidate

Sameecha Sudheer (1BM20CS213)

BMS COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND

ENGINEERING



CERTIFICATE

This is to certify that the OOMD Laboratory Report has been carried out by **Sameecha Sudheer** (**1BM20CS213**) during the academic year 2022-2023.

Signature of the Faculty in Charge

Table of Contents

Sl No	Title	Pageno
1	Hotel Management System	
2	credit card processing	
3	Library Management System	
4	Stock Maintenance System	
5	Passport Automation System	
6	Railway Reservation System	
7	online shopping system	

Hotel Management System:

Software requirements specification (SRS)

1. Introduction

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of a Hotel Management Software. The software aims to provide a comprehensive solution for managing various aspects of hotel operations, including reservations, guest services, billing, housekeeping, and reporting. It will streamline processes, enhance guest satisfaction, and improve overall efficiency within the hotel management system.

2. Scope

The Hotel Management Software will consist of the following key modules and functionalities:

- Reservation Management: Enable the hotel staff to manage guest reservations, including check-in, check-out, room assignment, and booking modifications.
- Guest Services: Provide a platform for managing guest requests, complaints, and inquiries efficiently, ensuring prompt response and resolution.
- Billing and Invoicing: Generate accurate bills for guest stays, track payments, and automate invoicing processes for individual guests and corporate clients.
- Housekeeping: Enable housekeeping staff to manage room cleaning schedules, track room status, and allocate resources effectively.
- Reporting and Analytics: Provide comprehensive reporting and analytical capabilities to monitor occupancy rates, revenue, guest satisfaction, and other key performance indicators.

3. Functional Requirements

3.1 Reservation Management

The system should allow the hotel staff to create, modify, and cancel guest reservations.

Guests should be able to book rooms online, specifying check-in and check-out dates, room preferences, and additional services.

The software should facilitate check-in and check-out processes, including capturing guest information, verifying identification, and assigning rooms.

The system should support room availability checks across different room types and categories.

The software should send automated email confirmations to guests upon successful reservation.

3.2 Guest Services

The system should provide a centralized platform for capturing and managing guest requests, complaints, and inquiries.

Hotel staff should be able to log, assign, track, and resolve guest service requests efficiently.

Guests should have access to a self-service portal for submitting requests, checking request status, and providing feedback.

The software should send automated notifications to guests upon request completion or updates.

3.3 Billing and Invoicing

- The software should generate accurate bills for guest stays, including room charges, additional services, and taxes.
- The system should support various payment methods, such as cash, credit cards, and online payments.
- It should provide the capability to split bills between multiple guests or corporate clients.
- The software should generate invoices automatically and allow customization to include branding and relevant details.
- It should maintain a record of payment history and outstanding balances.

3.4 Housekeeping

- The system should enable housekeeping staff to view room cleaning schedules and update room statuses (e.g., clean, dirty, occupied, or out of order).
- Housekeepers should be able to log cleaning activities and report any maintenance issues or damages.
- The software should optimize housekeeping routes and allocate resources based on room status and priority.
- The system should provide alerts and notifications to housekeeping staff for urgent tasks or room status changes.

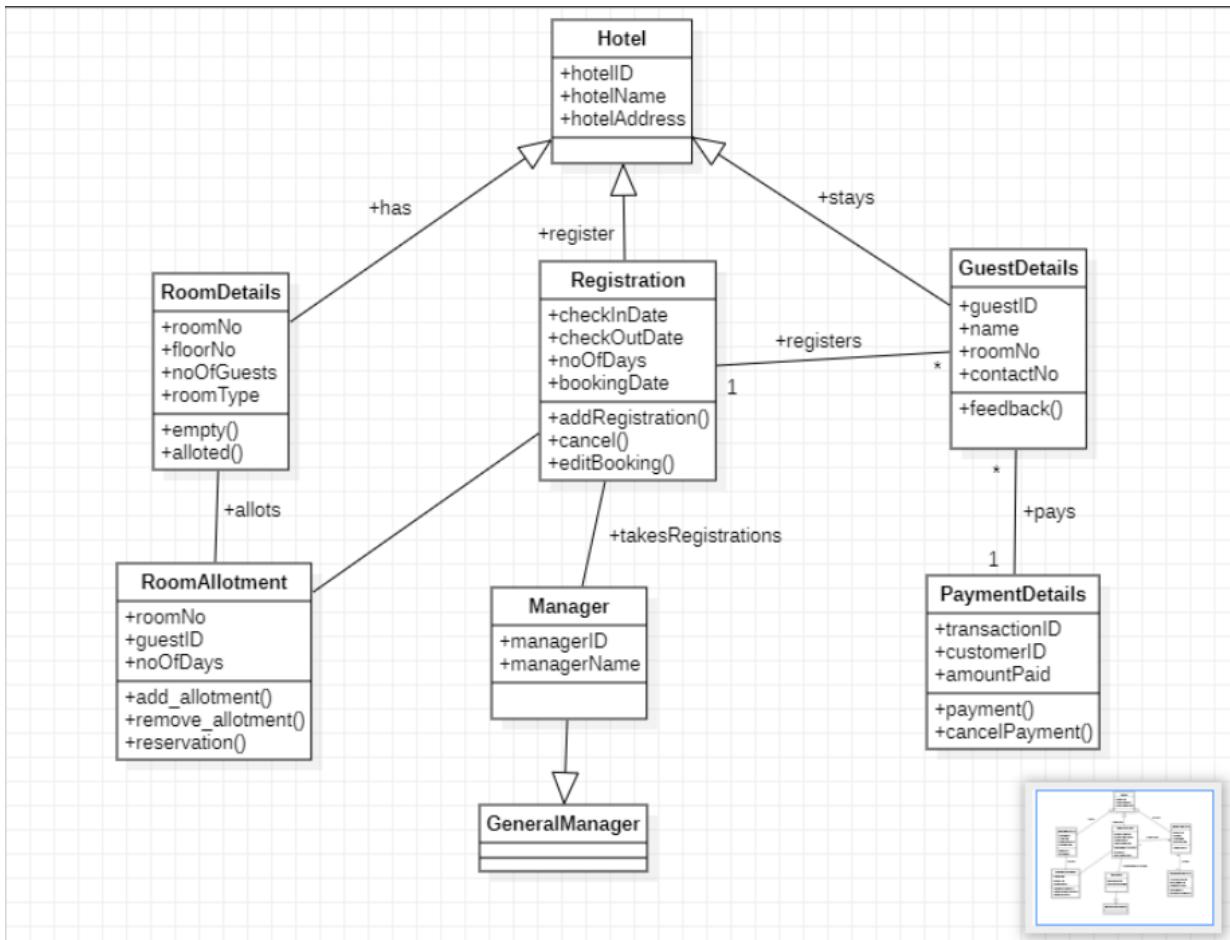
3.5 Reporting and Analytics

- The software should provide a range of pre-defined reports related to occupancy rates, revenue, guest satisfaction, and other key metrics.
- The system should allow customization of reports, including date ranges, filters, and data visualization options.
- It should support real-time monitoring of hotel performance and provide actionable insights for decision-making.
- The software should offer secure access control and user permissions to limit report availability based on roles and responsibility

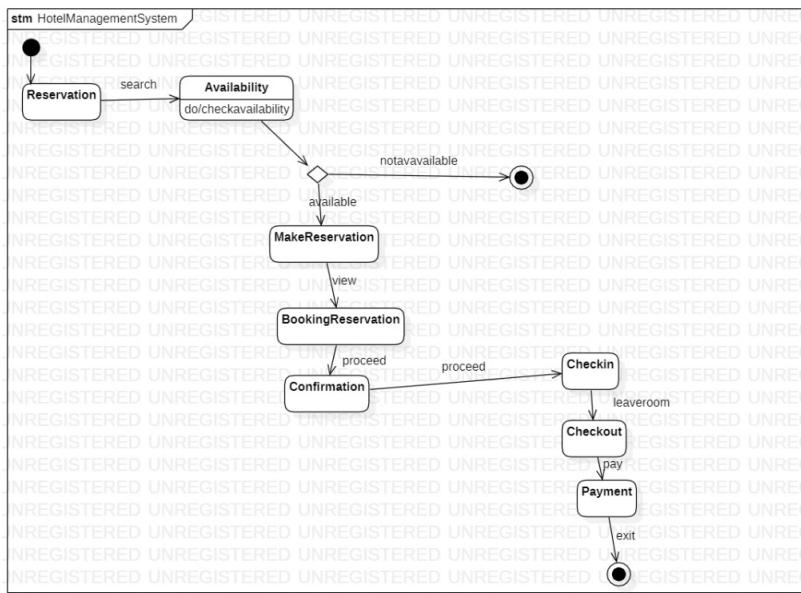
4. Non-Functional Requirements

- The Hotel Management Software should be user-friendly, intuitive, and accessible to both hotel staff and guests.
- It should be available 24/7 and maintain high availability to ensure uninterrupted operations.
- The software should be scalable to accommodate multiple hotels or expand as the hotel grows.
- It should have robust security measures in place to protect sensitive guest information and prevent unauthorized access.
- The system should be able to integrate with other existing hotel systems, such as property management systems, online travel agencies, and payment gateways.
- The software should be compatible with various devices and platforms, including desktops, tablets, and mobile devices.

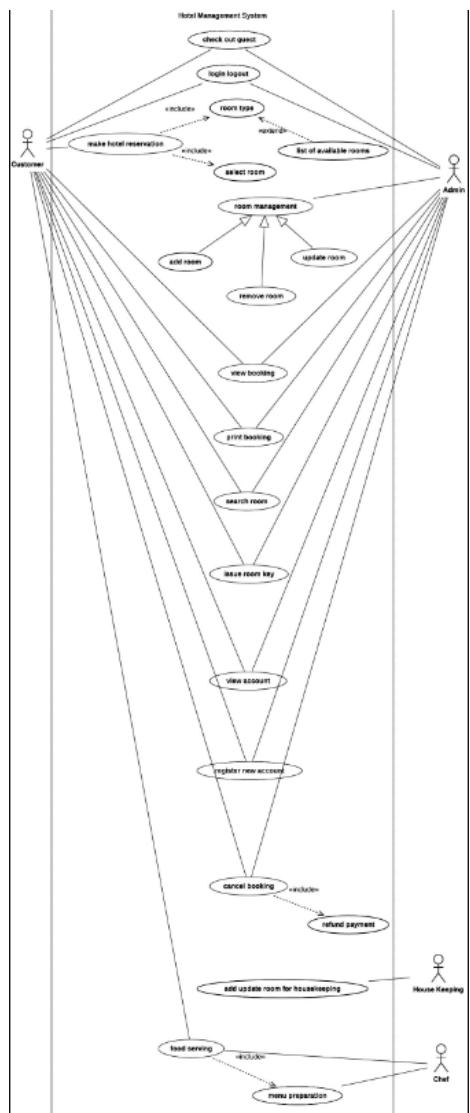
class Diagram:



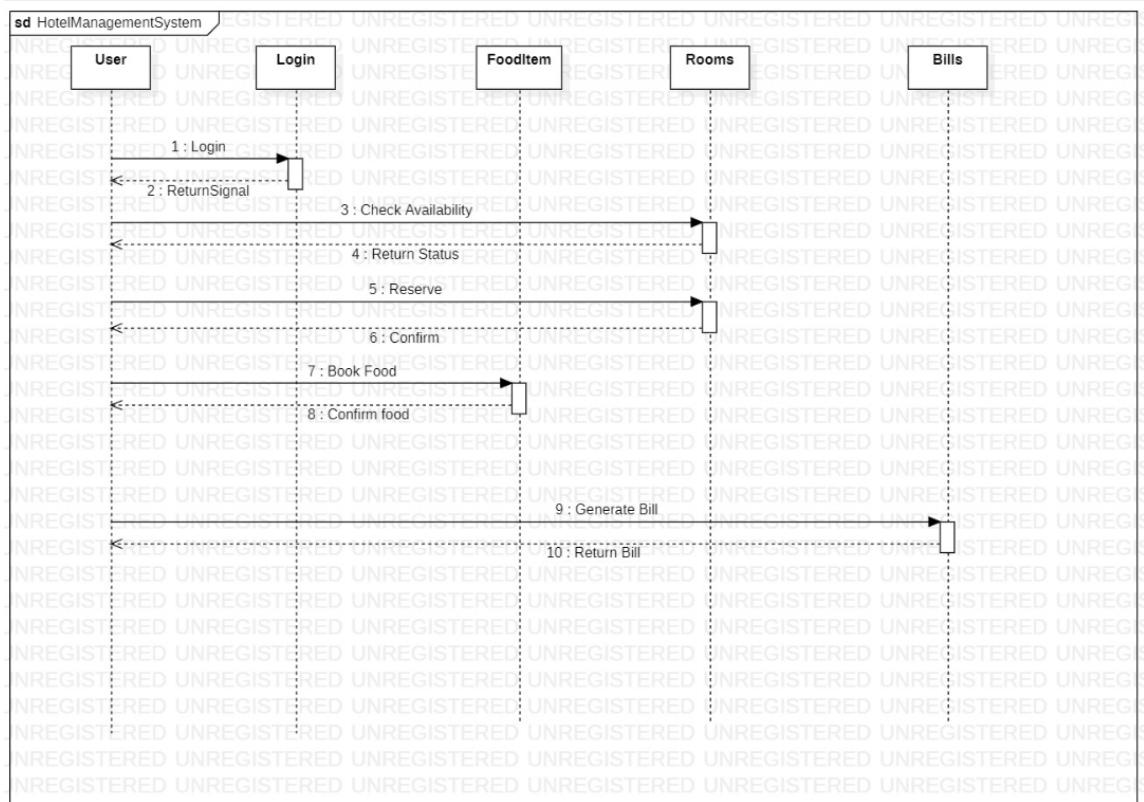
State Diagram:



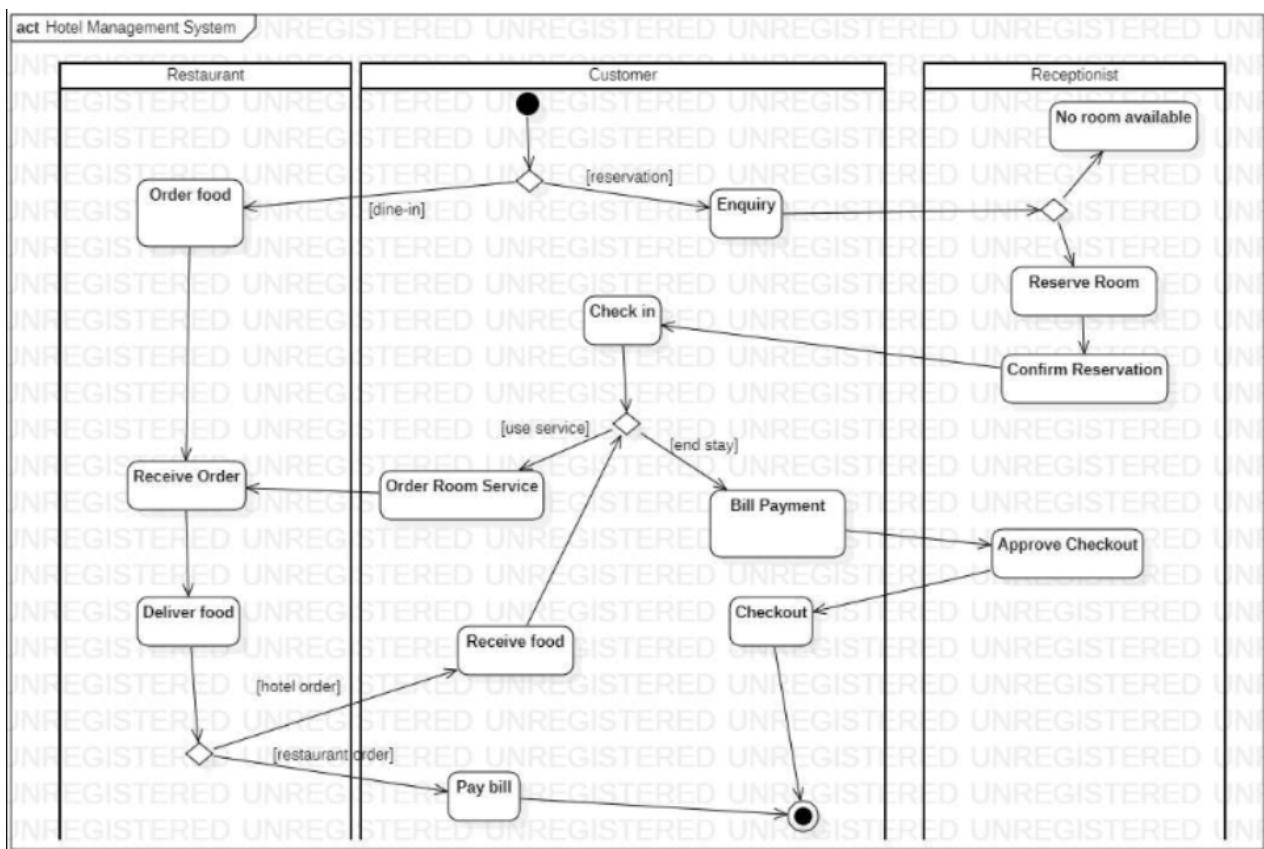
Use Case:



Sequence Diagram:



Activity Diagram:



Library Management System:

1. Introduction

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of a Library Management System. The software aims to provide a comprehensive solution for managing various aspects of library operations, including cataloging, circulation, member management, acquisitions, and reporting. It will streamline processes, enhance user experience, and improve overall efficiency within the library system.

2. Scope

- The Library Management System will consist of the following key modules and functionalities:
 -
 - Cataloging: Enable library staff to manage the library's collection, including adding, updating, and deleting books, audiovisual materials, and other resources.
 - Circulation: Facilitate the borrowing and returning of library materials, manage loan periods, and track overdue items.
 - Member Management: Provide a platform for managing library member information, including registration, authentication, and communication.
 - Acquisitions: Support the acquisition process, including budgeting, purchasing, and receiving new materials.
 - Reporting and Analytics: Provide comprehensive reporting and analytical capabilities to monitor circulation statistics, budget utilization, and collection usage.

3. Functional Requirements

3.1 Cataloging

- The system should allow library staff to add new books and other resources to the library's collection, including details such as title, author, ISBN, publication date, and category.
- The software should support batch importing of bibliographic records from external sources, such as ISBN databases.
- It should provide a search function to retrieve resources based on various criteria, including title, author, subject, or keyword.
- The system should allow librarians to update resource information, including adding or modifying additional metadata, such as summaries, reviews, or cover images.
- The software should generate unique identifiers (e.g., barcode or RFID tags) for each item in the collection to facilitate circulation and inventory management.

3.2 Circulation

- The system should support borrowing and returning processes, including checking out items to library members and updating their borrowing status.
- It should provide a self-service portal or kiosks for users to check out and return materials independently.
- The software should enforce loan periods, manage renewals, and track overdue items, sending reminders to members with approaching due dates.
- It should handle reservation requests for popular items and notify users when the requested items become available.
- The system should generate alerts for lost or damaged items and facilitate the process of charging fines or replacement fees.

3.3 Member Management

- The software should enable users to register as library members, providing necessary personal information and authentication credentials.
- The system should allow members to update their contact details, view their borrowing history, and manage their preferences.
- It should provide a login mechanism to authenticate members and grant access to personalized features and services.
- The software should support member communication, such as notifications about overdue items, reservation pickups, or library events.

- The system should offer administrative features to manage member accounts, such as suspensions, renewals, or cancellations.

3.4 Acquisitions

- The system should facilitate the acquisition process, allowing librarians to create and manage purchase orders for new materials.
- It should provide budget tracking features to monitor expenditures and ensure efficient allocation of funds.
- The software should support receiving and processing new materials, updating the catalog, and making them available for circulation.
- It should integrate with external systems or databases to retrieve bibliographic data, pricing information, and availability.
- The system should generate reports on acquisitions, budget utilization, and vendor performance.

3.5 Reporting and Analytics

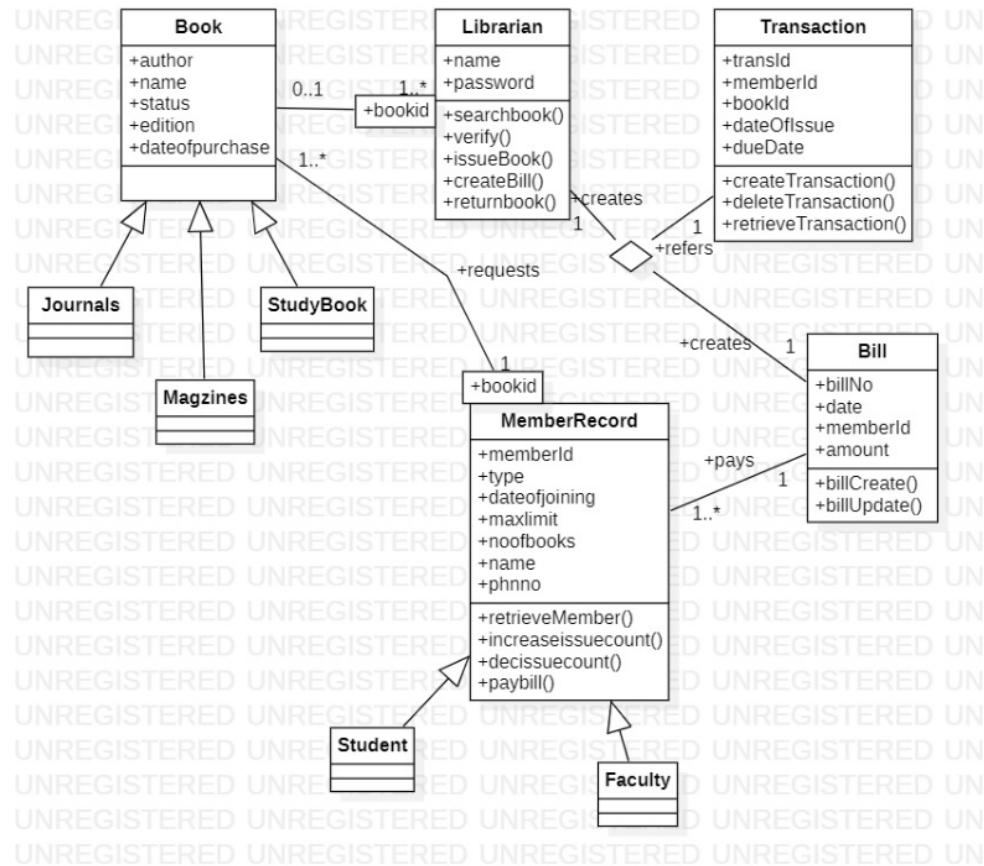
- The software should provide a range of pre-defined reports related to circulation statistics, collection usage, budget analysis, and member activity.
- The system should allow customization of reports, including date ranges, filters, and data visualization options.
- It should support real-time monitoring of library performance and provide actionable insights for decision-making.
- The software should offer secure access control and user permissions to limit report availability based on roles and responsibilities.

4. Non-Functional Requirements

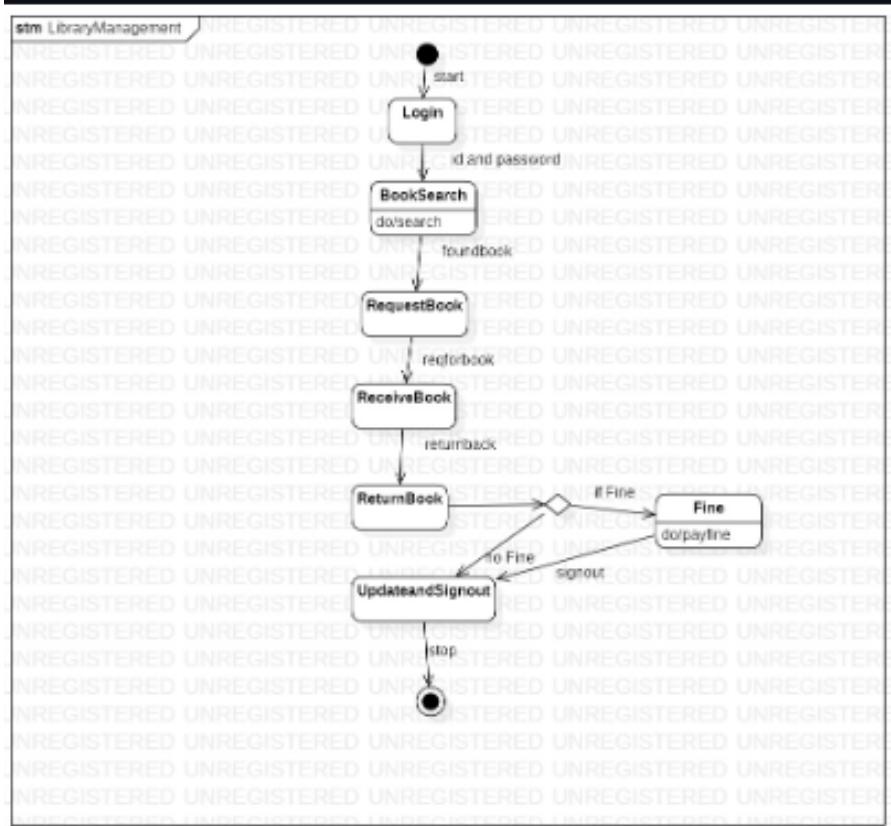
- The Library Management System should be user-friendly, intuitive, and accessible to both library staff and users.
- It should be available 24/7 and maintain high availability to ensure uninterrupted access to library services.
- The software should have robust security measures in place to protect user information, prevent unauthorized access, and secure the library's digital assets.

- The system should be scalable to accommodate the growing collection size and increasing user demands.
- It should be compatible with various devices and platforms, including desktops, tablets, and mobile devices.

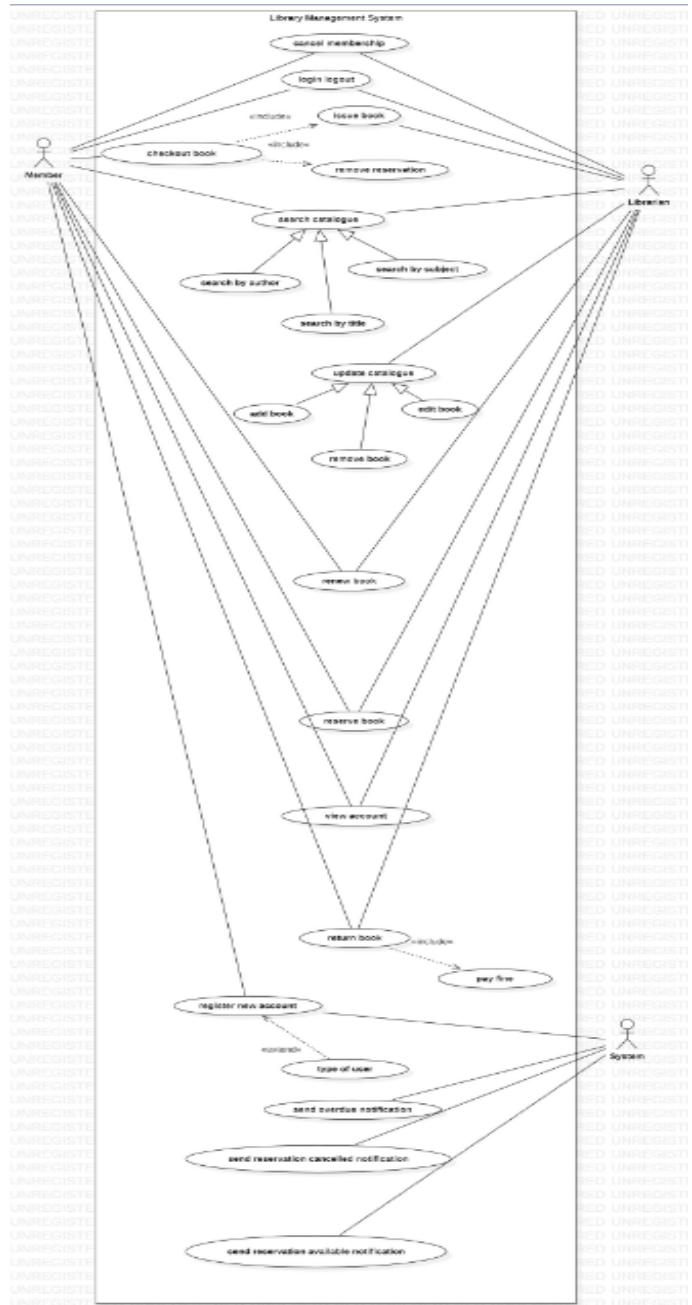
class diagram:



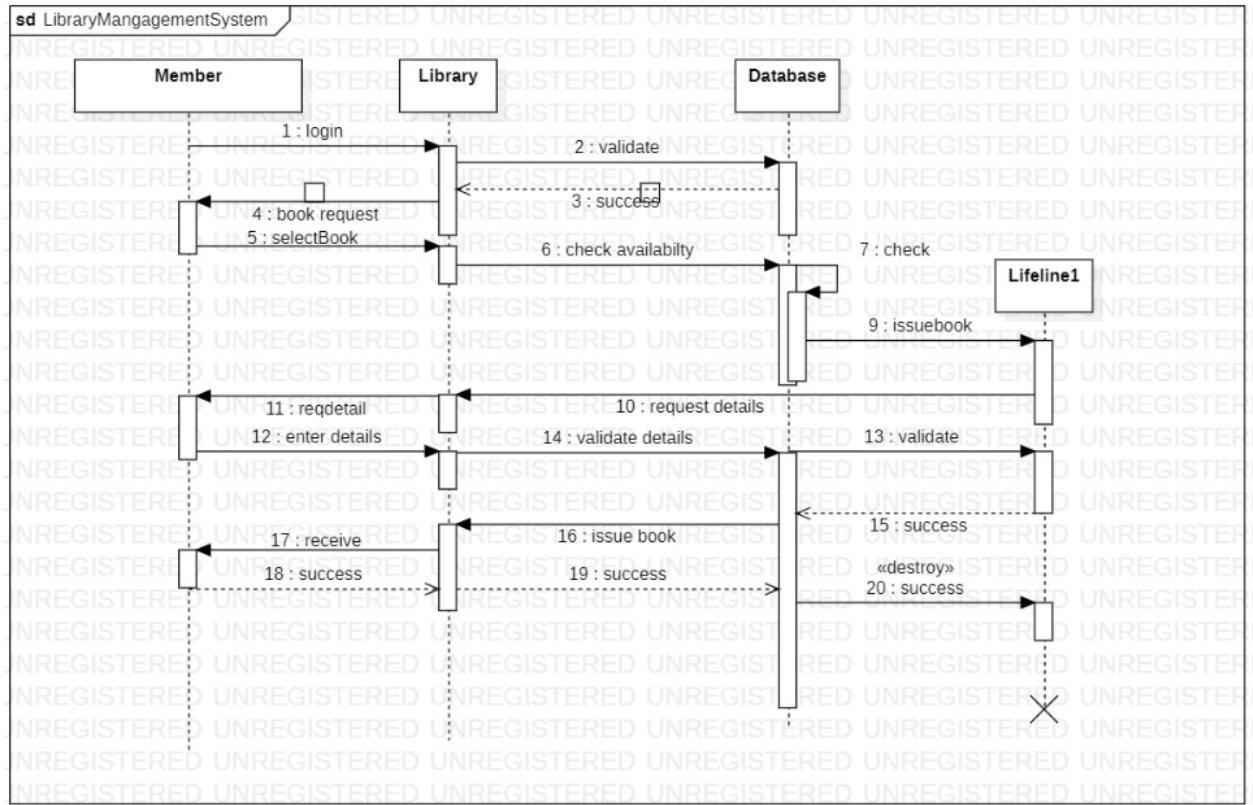
State diagram:



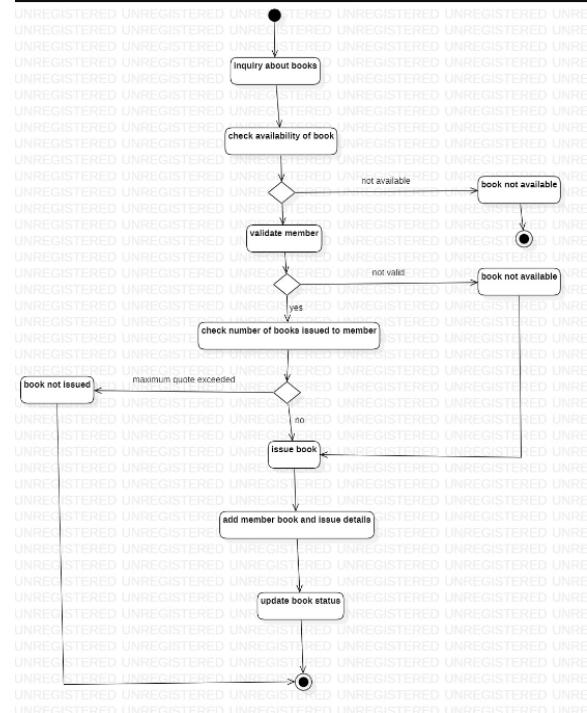
Use case diagram:



sequence diagram:



activity diagram:



Credit card processing

1. Introduction The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of a Credit Card Processing System. The software aims to provide a secure and efficient solution for processing credit card transactions, authorizing payments, and managing related financial operations. It will ensure compliance with industry standards, protect sensitive cardholder data, and streamline payment processing for merchants.

2. Scope The Credit Card Processing System will consist of the following key modules and functionalities:

- Payment Gateway Integration: Establish secure connections with payment gateways to facilitate credit card transaction processing.
- Transaction Authorization: Verify the authenticity of credit card transactions and authorize payments in real-time.
- Cardholder Data Protection: Implement robust security measures to protect cardholder data, comply with industry standards (such as PCI DSS), and prevent unauthorized access.
- Merchant Management: Enable merchants to manage their accounts, view transaction details, and generate reports.
- Reporting and Analytics: Provide comprehensive reporting and analytical capabilities to monitor transaction volumes, revenue, and other key performance indicators.

3. Functional Requirements

3.1 Payment Gateway Integration

- The system should integrate with payment gateways to securely transmit credit card transaction data for processing.
- It should support major payment methods, including credit cards, debit cards, and alternative payment options.
- The software should handle transaction data encryption and decryption during transmission to ensure data integrity and security.
- It should support tokenization or other methods to store cardholder data securely for recurring payments or future transactions.
- The system should provide error handling and transaction status notifications to merchants and customers.

3.2 Transaction Authorization

- The software should communicate with issuing banks or payment networks to verify credit card details and authorize transactions.
- It should perform real-time validation checks, including card number, expiration date, CVV, and cardholder authentication (such as 3D Secure).
- The system should handle declined transactions and provide appropriate error codes or messages for failed authorizations.

- It should support partial authorizations and split payments for transactions with insufficient funds or credit limits.
- The software should generate unique transaction identifiers and store transaction details for audit and reconciliation purposes.

3.3 Cardholder Data Protection

- The system should comply with Payment Card Industry Data Security Standard (PCI DSS) requirements for handling and storing cardholder data securely.
- It should encrypt sensitive cardholder information during transmission and storage.
- The software should implement secure user authentication and access control mechanisms to prevent unauthorized access to cardholder data.
- It should support features like address verification service (AVS) and card verification value (CVV) checks to mitigate fraudulent transactions.
- The system should maintain comprehensive audit logs to track and monitor user activities and system events.

3.4 Merchant Management

- The software should provide a user-friendly interface for merchants to manage their accounts, view transaction details, and generate reports.
- It should allow merchants to configure payment settings, such as pricing, currency options, and transaction preferences.
- The system should support account reconciliation by providing access to settlement reports and transaction summaries.
- It should enable merchants to issue refunds or process voids for authorized transactions.
- The software should provide customer support features, such as ticketing systems or knowledge bases, for merchants' inquiries and assistance.

3.5 Reporting and Analytics

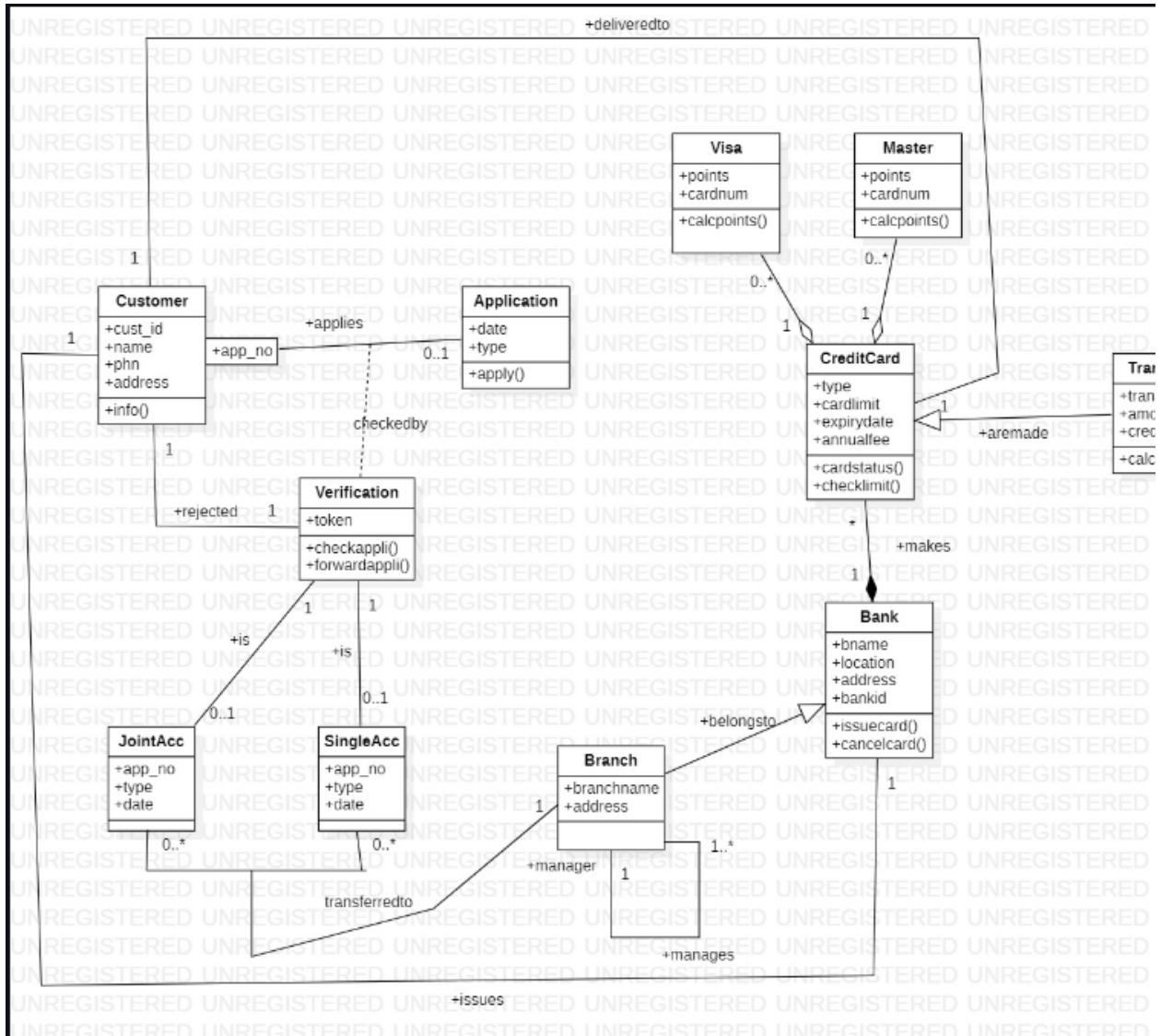
- The system should offer pre-defined and customizable reports on transaction volumes, revenue, chargebacks, and other relevant metrics.
- It should provide data visualization options, such as charts and graphs, for easy understanding and analysis.
- The software should support exporting reports in various formats (e.g., PDF, CSV) for further analysis or integration with external systems.
- It should offer real-time monitoring of transaction activity, fraud detection, and alerts for suspicious or unusual patterns.
- The system should allow role-based access control to limit report availability based on user roles and responsibilities.

4. Non-Functional Requirements

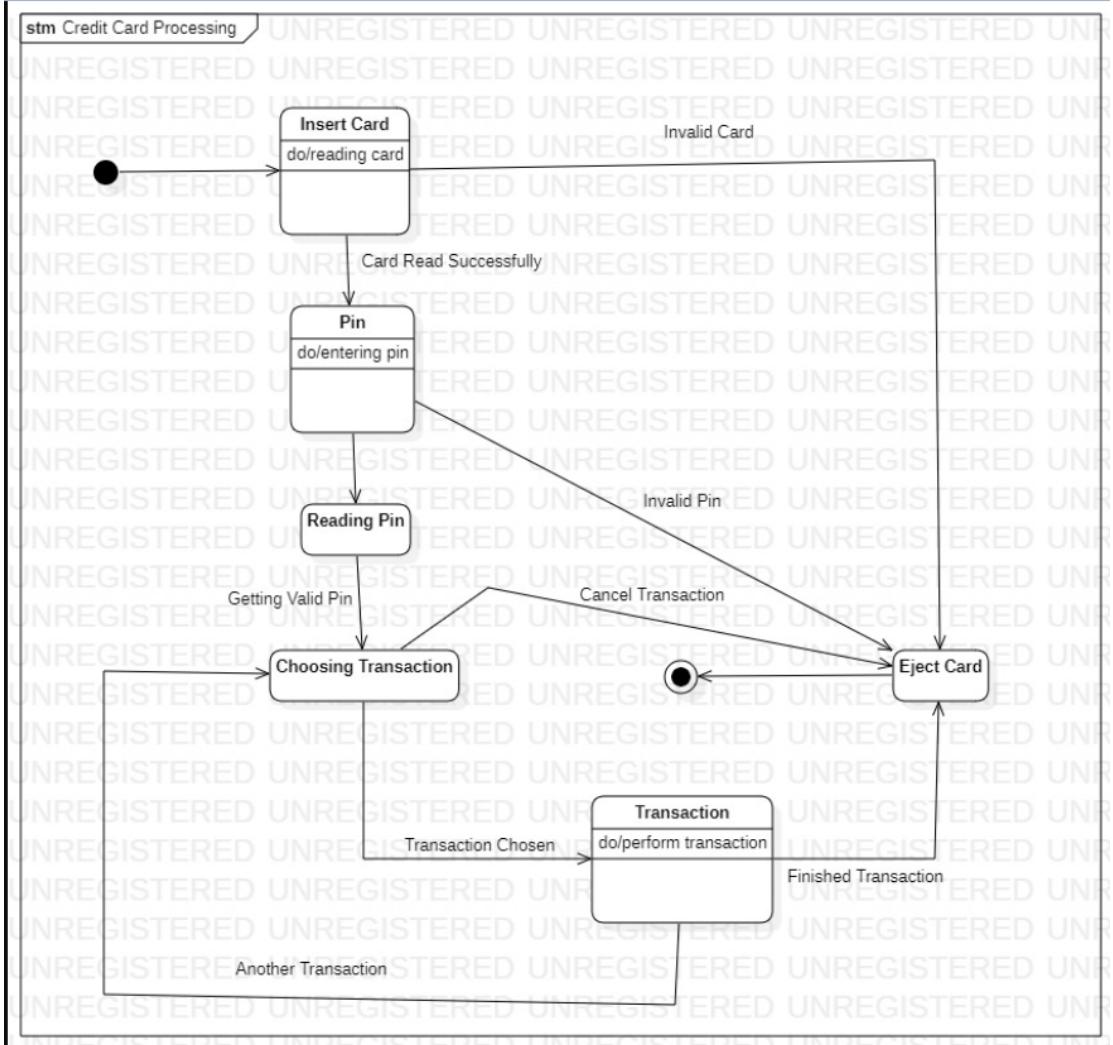
- The Credit Card Processing System should be highly secure, complying with industry standards and regulations to protect cardholder data.

- It should be highly available and capable of processing transactions in real-time, ensuring minimal downtime and disruptions.
- The software should be scalable to handle increasing transaction volumes and support multiple merchants simultaneously.
- It should have efficient performance to process transactions quickly and provide timely responses to merchants and customers.
- The system should support integration with different payment gateways and external systems, such as e-commerce platforms and financial systems.

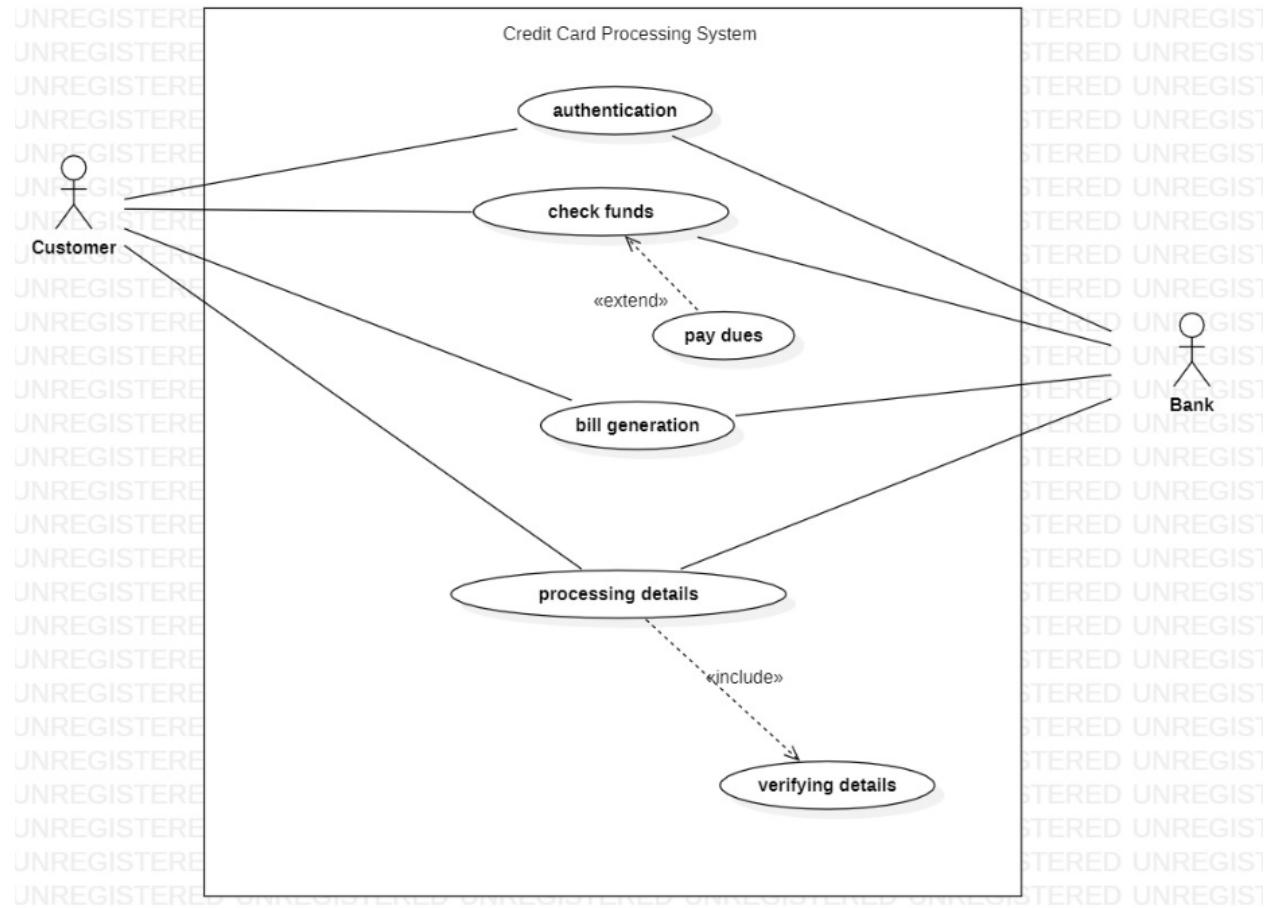
Class Diagram:



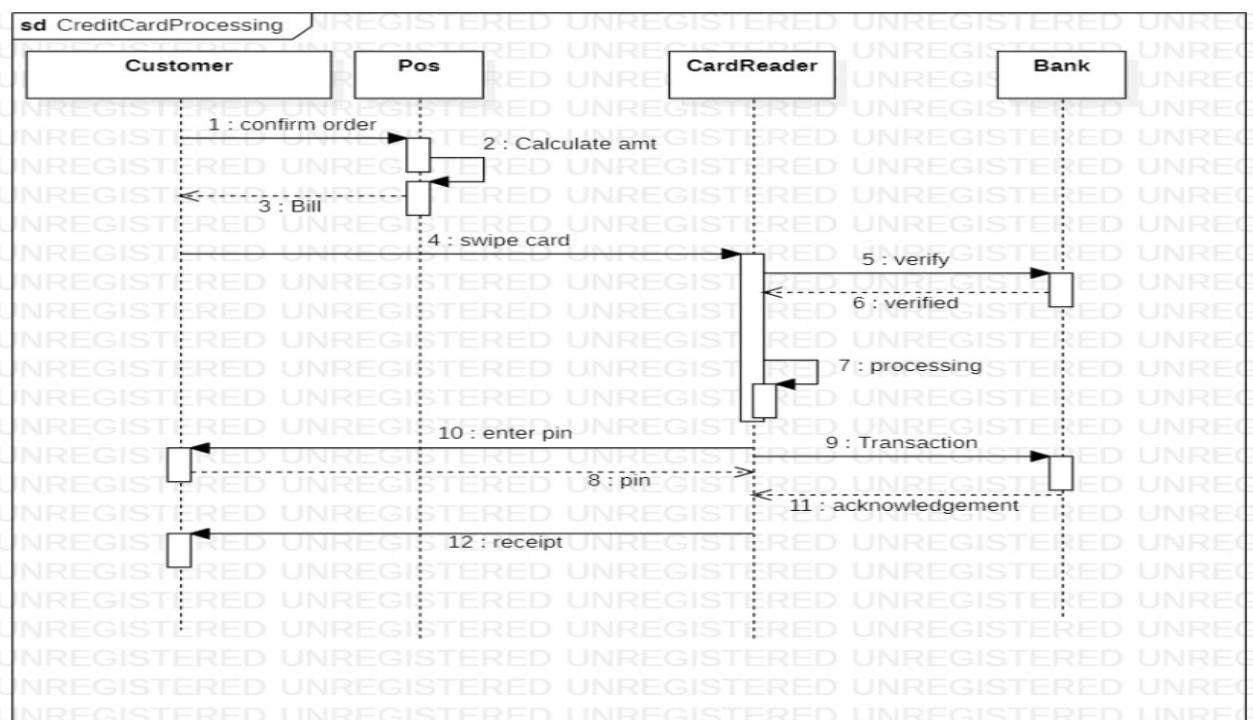
State diagram:



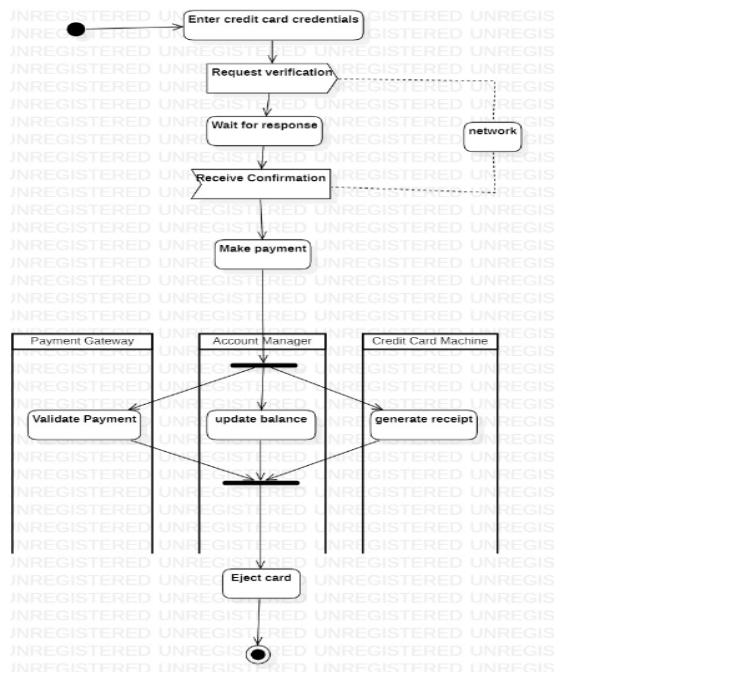
Use case diagram:



sequence diagram:



activity diagram:



Stock Maintenance System

1. Introduction The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of a Stock Maintenance System. The software aims to provide an efficient solution for managing stock inventory, tracking stock levels, and facilitating stock-related operations in an organization.

It will streamline stock management processes, improve inventory accuracy, and optimize stock availability.

2. Scope The Stock Maintenance System will consist of the following key modules and functionalities:

- Stock Inventory Management: Enable the organization to maintain a centralized stock inventory database, including details of products, quantities, locations, and other relevant attributes.
- Stock Replenishment: Facilitate the process of stock replenishment based on predefined reorder points, supplier orders, and demand forecasts.
- Stock Movement Tracking: Track the movement of stock within the organization, including stock receipts, stock transfers between locations, and stock returns.
- Reporting and Analytics: Provide comprehensive reporting and analytical capabilities to monitor stock levels, stock turnover, stock valuation, and other key performance indicators.

3. Functional Requirements

3.1 Stock Inventory Management

- The system should allow users to add, update, and delete stock items in the inventory database, including details such as product codes, descriptions, unit costs, and quantities.
- It should support categorizing stock items into different categories or groups for better organization and searchability.
- The software should provide a search function to retrieve stock items based on various criteria, such as product codes, descriptions, or categories.
- It should allow users to view and track stock details, including current stock levels, locations, and associated attributes.

- The system should support bulk imports or exports of stock data for easier data management.

3.2 Stock Replenishment

- The software should generate alerts or notifications when stock levels reach predefined reorder points, indicating the need for stock replenishment.
- It should allow users to create purchase orders or requisitions for stock replenishment, specifying the required quantities, preferred suppliers, and delivery dates.
- The system should integrate with supplier management systems or databases to facilitate the selection and retrieval of supplier details.
- It should provide functionalities to track the status of purchase orders, including order confirmation, delivery, and invoicing.
- The software should update stock levels automatically upon receiving new stock from suppliers.

3.3 Stock Movement Tracking

- The system should support recording stock movements within the organization, such as stock receipts, stock transfers between locations, and stock returns.
- It should capture relevant details for each stock movement, including timestamps, quantities, source locations, destination locations, and reasons for the movement.
- The software should update stock levels and locations accordingly based on recorded stock movements.
- It should generate reports or provide search capabilities to track the movement history of specific stock items.

3.4 Reporting and Analytics

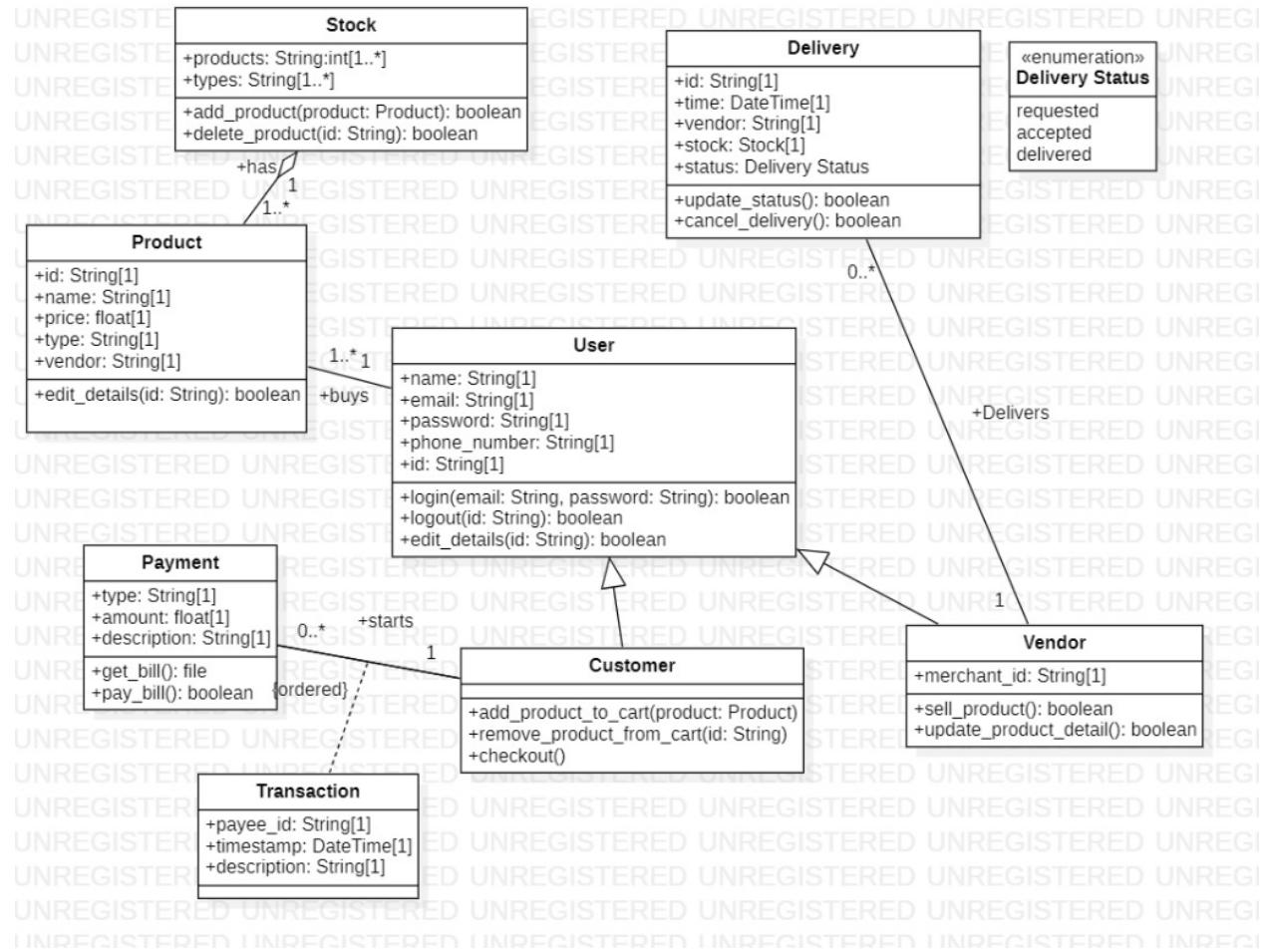
- The software should provide pre-defined and customizable reports on stock levels, stock turnover, stock valuation, and other relevant metrics.
- It should support data visualization options, such as charts and graphs, for easy understanding and analysis of stock-related data.

- The system should allow users to filter and drill down into stock data based on various criteria, such as time periods, stock categories, or locations.
- It should offer real-time monitoring of stock levels and provide alerts or notifications for critical stock thresholds or stock discrepancies.
- The software should allow exporting reports in various formats (e.g., PDF, CSV) for further analysis or integration with external systems.

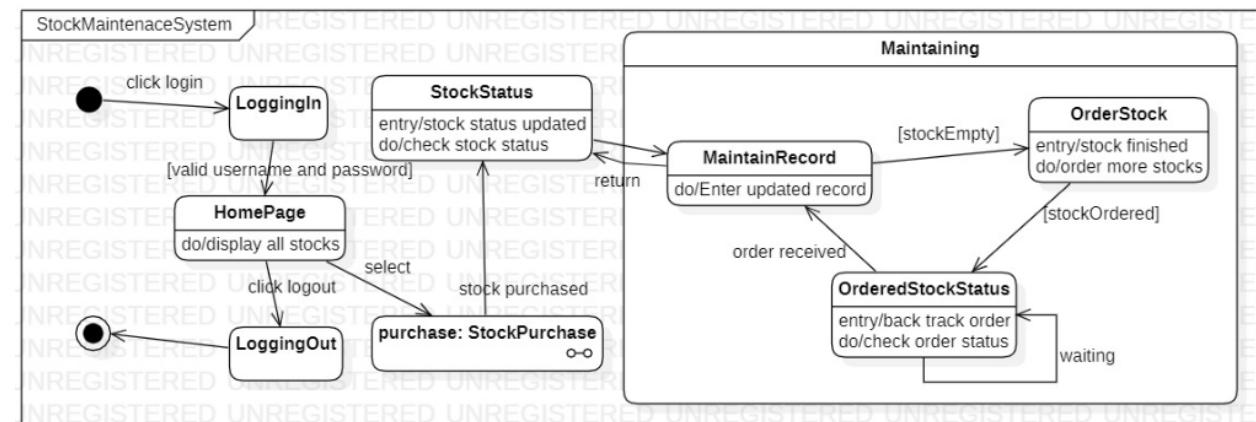
4. Non-Functional Requirements

- The Stock Maintenance System should be user-friendly, intuitive, and accessible to authorized users within the organization.
- It should be scalable to handle a large number of stock items and accommodate organizational growth.
- The software should have efficient performance to handle stock data processing and provide timely responses to user requests.
- The system should be secure, implementing appropriate access control mechanisms to protect stock data from unauthorized access or modification.
- It should support integration with other relevant systems, such as accounting systems or enterprise resource planning (ERP) systems.

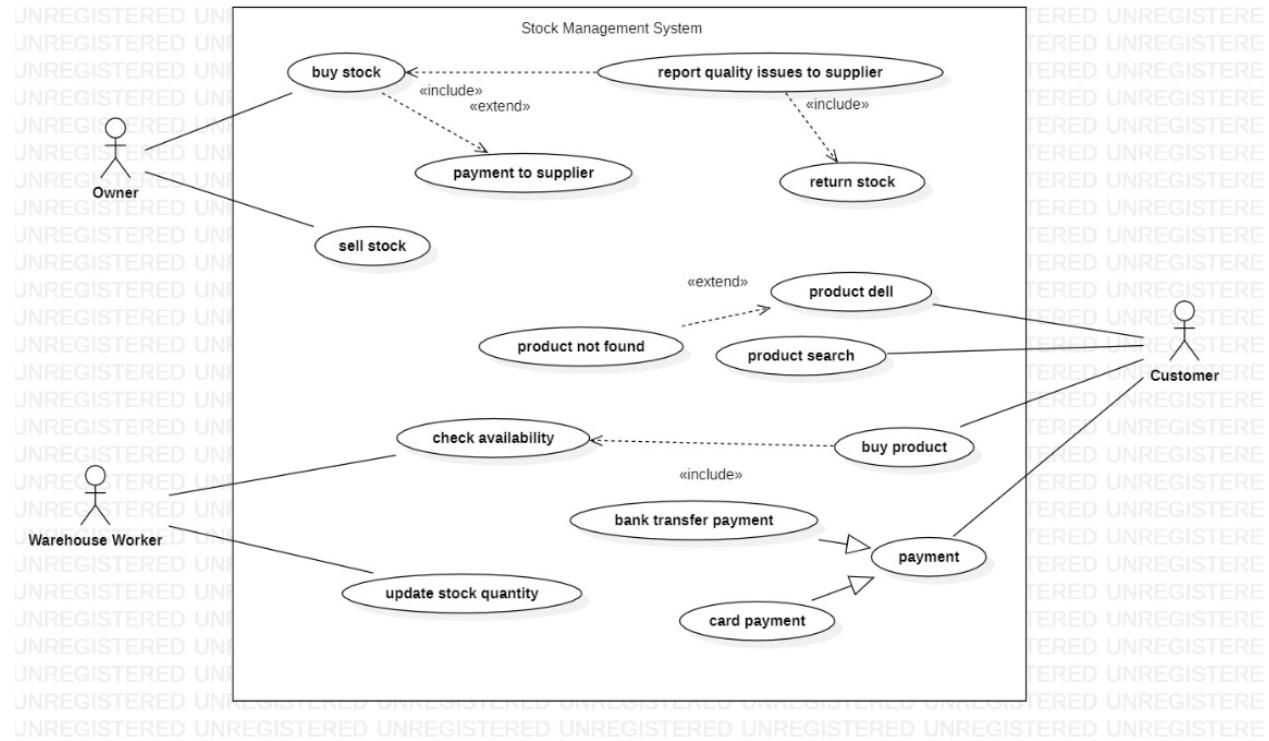
class diagram:



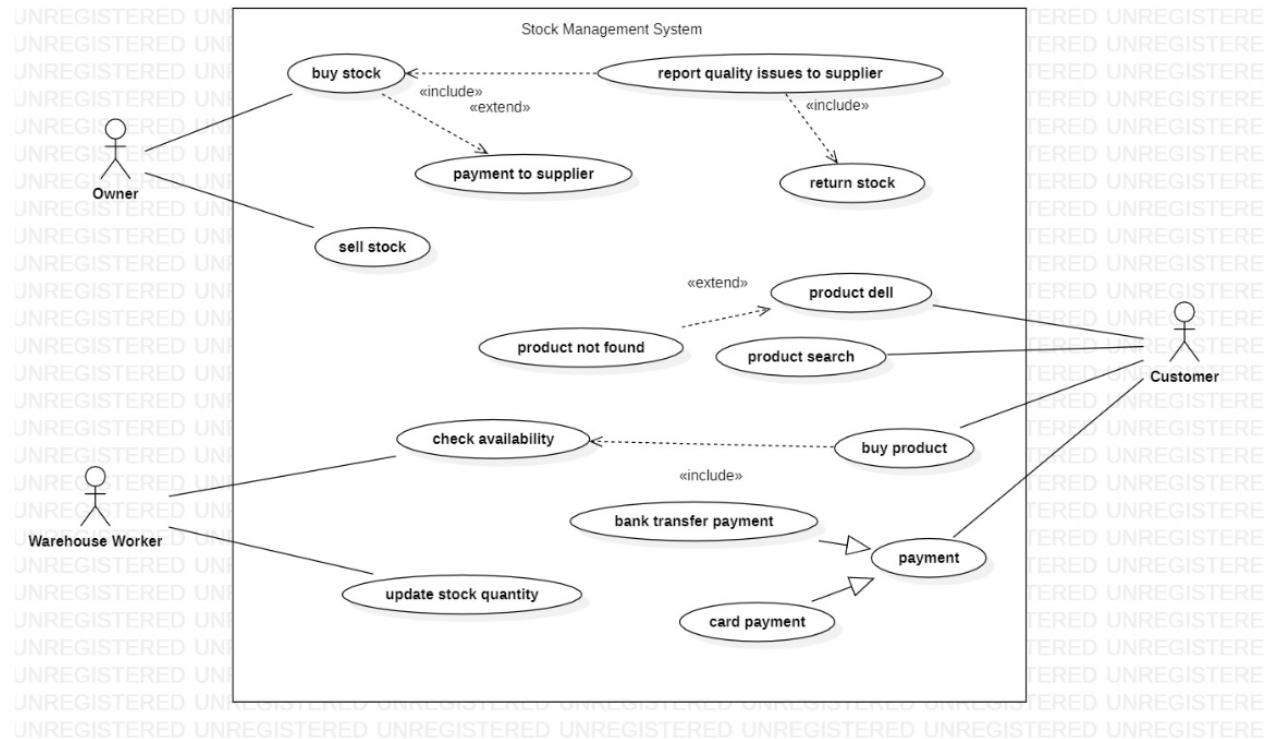
State diagram:



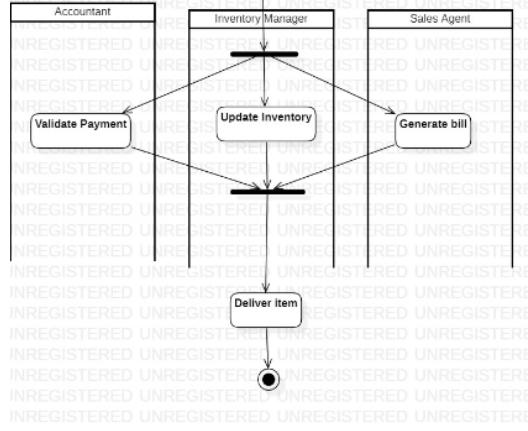
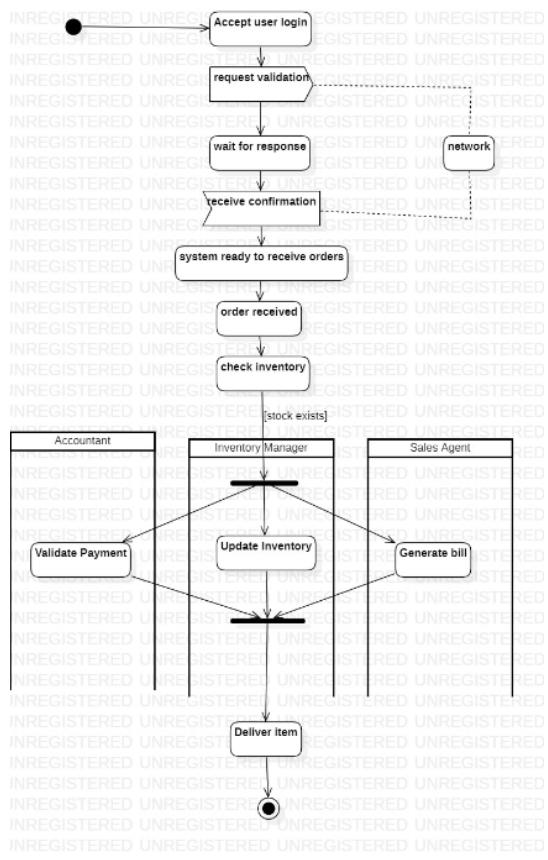
Use case diagram:



Use case diagram:



activity diagram:



Passport Automation System:

1. Introduction

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of a Passport Automation System. The software aims to automate and streamline the process of passport issuance and management, providing an efficient and user-friendly solution for both applicants and passport authorities. It will facilitate online application submission, appointment scheduling, document verification, and passport delivery, while ensuring data security and compliance with relevant regulations.

2. Scope

The Passport Automation System will consist of the following key modules and functionalities:

- Online Application Submission: Enable applicants to submit passport applications online, providing personal information, supporting documents, and selecting appointment slots.
- Appointment Scheduling: Allow applicants to schedule appointments for document verification and biometric data capture.
- Document Verification: Provide tools for passport authorities to verify the authenticity of submitted documents and conduct background checks.
- Payment Processing: Facilitate secure online payment for passport application fees and other related services.
- Passport Production and Delivery: Manage the production process of passports, including data encoding, printing, and secure delivery to applicants.
- Reporting and Analytics: Provide comprehensive reporting and analytical capabilities for monitoring application volumes, processing times, and other key performance indicators.

3. Functional Requirements

3.1 Online Application Submission

- The system should provide an online portal for applicants to create and submit passport applications, including personal details, contact information, and supporting documents.
- It should support the uploading and storage of digital copies of documents, such as identification proof, address proof, and photographs.
- The software should validate the completeness and accuracy of application data, highlighting any missing or incorrect information to applicants.

- It should generate application reference numbers and confirmation receipts for applicants to track the status of their applications.

3.2 Appointment Scheduling

- The system should allow applicants to schedule appointments for document verification and biometric data capture at designated passport offices.
- It should provide a calendar view displaying available appointment slots, allowing applicants to select their preferred date and time.
- The software should send appointment reminders and notifications to applicants through email or SMS.
- It should allow rescheduling or cancellation of appointments, considering availability and rescheduling rules.

3.3 Document Verification

- The software should provide tools for passport authorities to verify the authenticity of submitted documents, such as identity proofs and address proofs.
- It should allow passport officers to mark documents as verified or request additional documents from applicants, if necessary.
- The system should support background checks, including identity verification and criminal record checks, based on government databases or external systems.
- It should maintain a record of verification activities and status for each application.

3.4 Payment Processing

- The system should integrate with secure online payment gateways to facilitate online payment of passport application fees and other related services.
- It should support various payment methods, such as credit cards, debit cards, and online banking transfers.
- The software should generate payment receipts and update payment status for each application.
- It should provide mechanisms to handle refunds, if required, following specified refund policies.

3.5 Passport Production and Delivery

- The system should manage the production process of passports, including data encoding, printing, and secure packaging.
- It should generate unique passport numbers and encode applicant information into passports securely.
- The software should track the progress of passport production, providing status updates to applicants.
- It should facilitate secure delivery of passports to applicants, including tracking mechanisms and delivery confirmation.

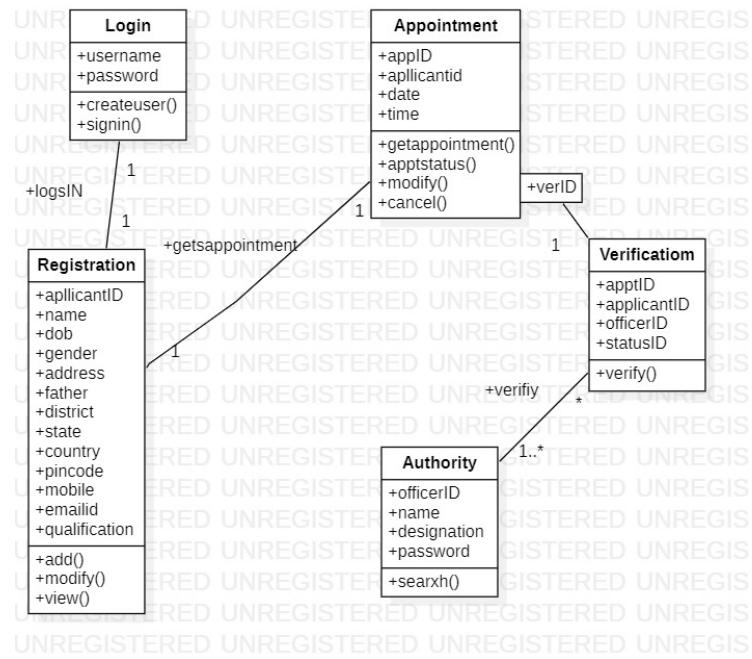
3.6 Reporting and Analytics

- The system should offer pre-defined and customizable reports on application volumes, processing times, rejection rates, and other key performance indicators.
- It should provide data visualization options, such as charts and graphs, for easy understanding and analysis.
- The software should support exporting reports in various formats (e.g., PDF, CSV) for further analysis or integration with external systems.
- It should allow authorized users to generate ad-hoc reports based on specific criteria or filters.

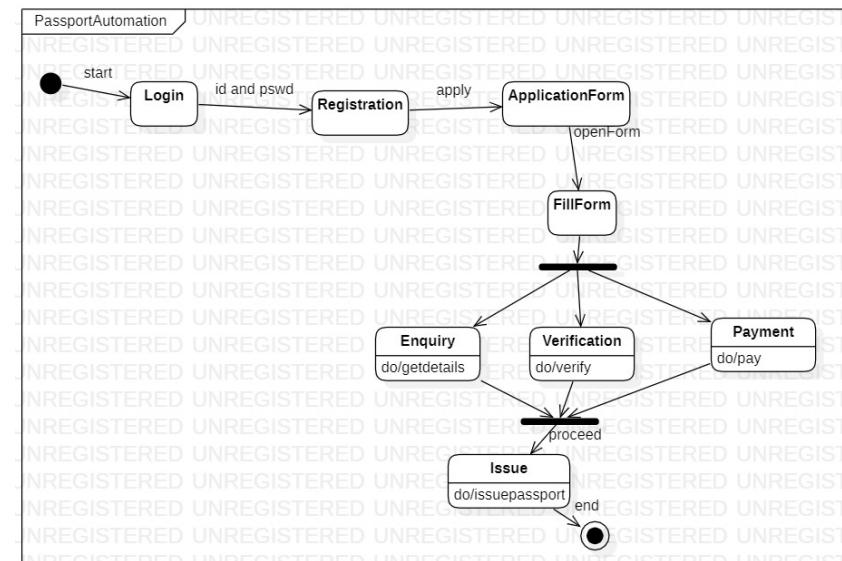
4. Non-Functional Requirements

- The Passport Automation System should be user-friendly, intuitive, and accessible to applicants with varying levels of technical expertise.
- It should have a high availability and response time to accommodate peak application periods and provide a smooth user experience.
- The software should comply with data privacy and security regulations, protecting applicant data and preventing unauthorized access or data breaches.
- It should support integration with existing government systems, such as identification databases or criminal record databases.

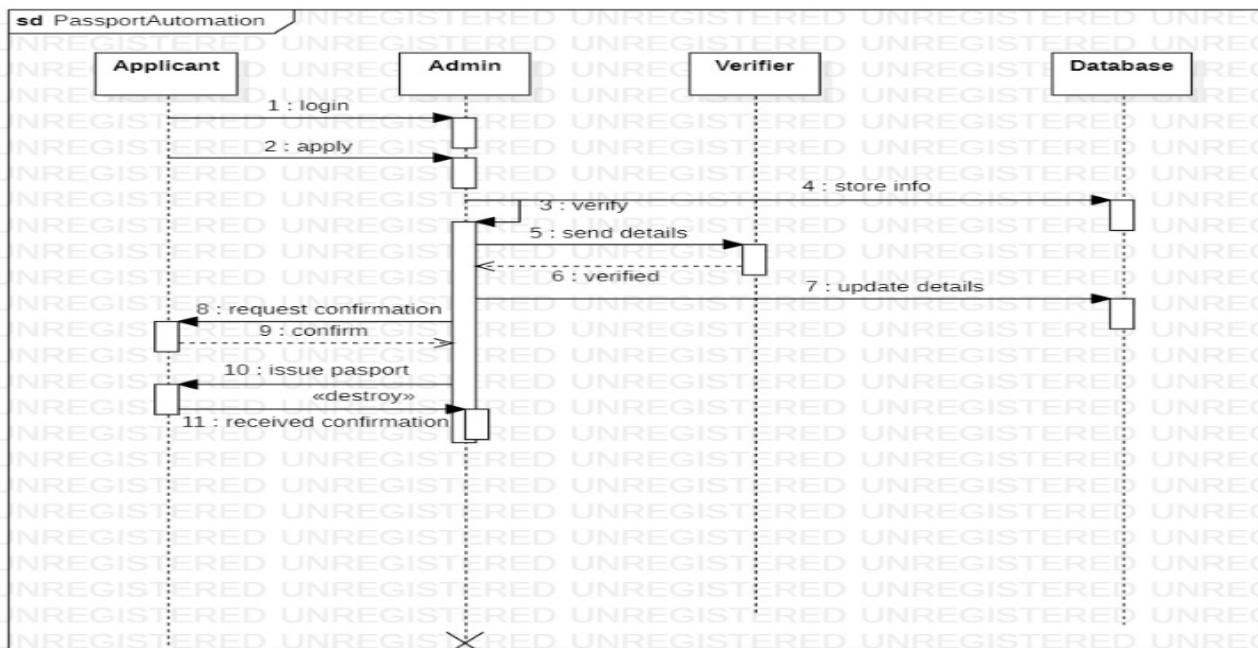
class diagram:



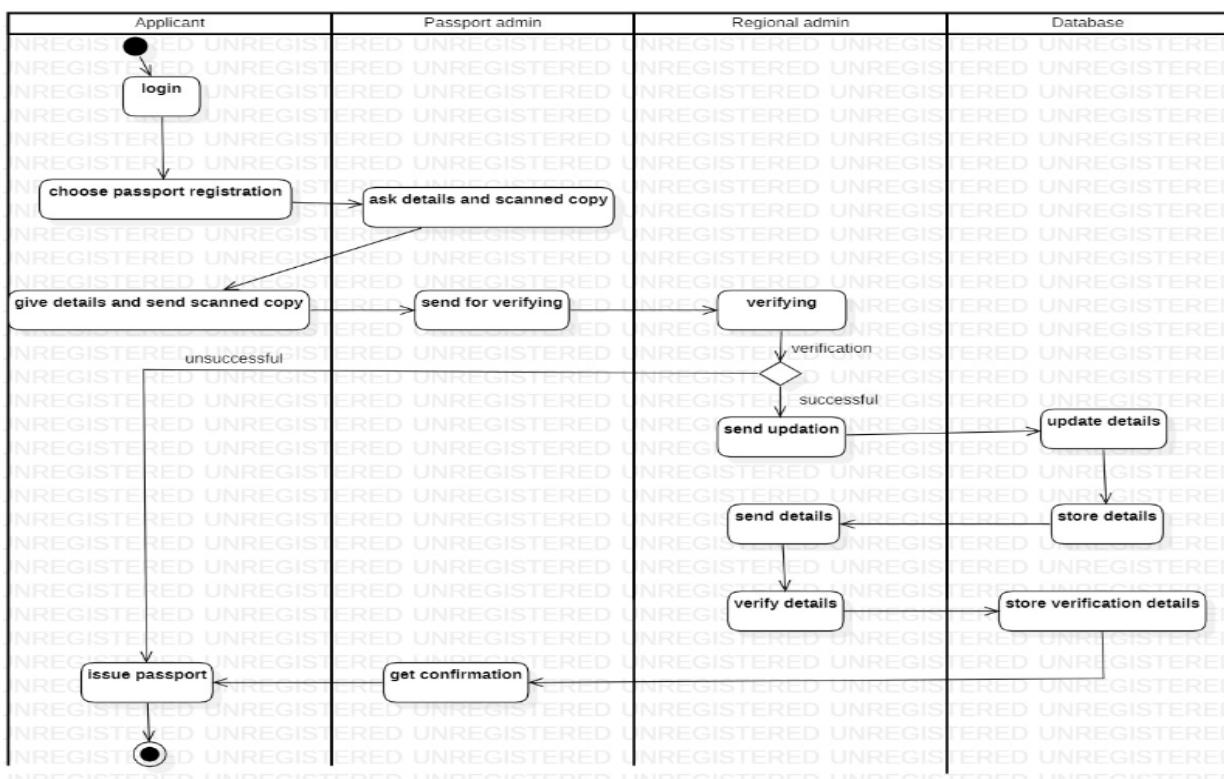
State diagram:

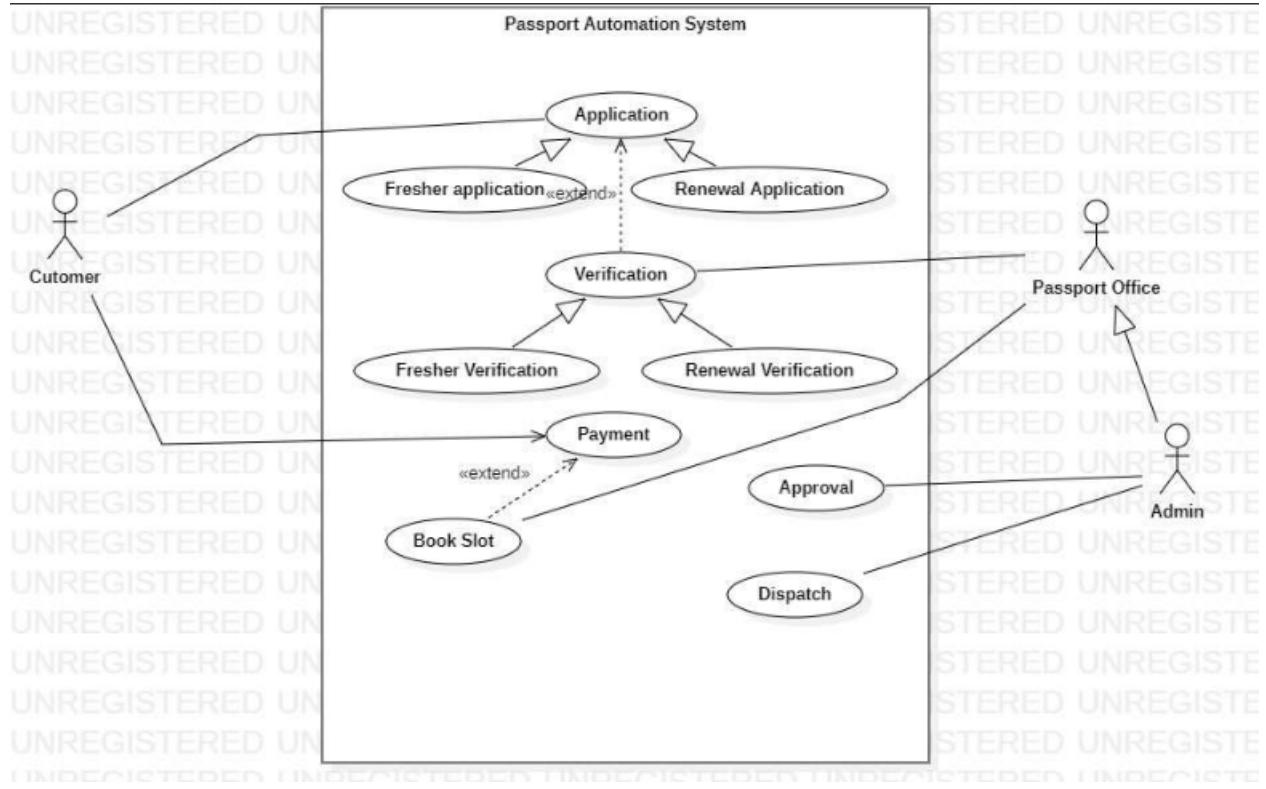


sequence diagram



activity diagram:





Railway Reservation system:

1. Introduction

The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of a Railway Reservation System. The software aims to provide a user-friendly and efficient solution for railway passengers to search, book, and manage train reservations. It will enable users to check seat availability, select preferred trains and classes, make payments, and generate tickets. The system will also provide administrative features for managing train schedules, seat inventories, and fare calculations.

2. Scope

The Railway Reservation System will consist of the following key modules and functionalities:

- Train Information: Provide a comprehensive database of train schedules, routes, stations, and fares.
- Seat Availability and Booking: Allow users to search for available seats on specific trains, select desired classes, and make reservations.
- Ticket Generation: Generate electronic tickets or e-tickets upon successful booking, containing relevant details such as passenger information, seat numbers, and travel itinerary.
- Payment Processing: Facilitate secure online payment for ticket reservations using various payment methods.
- Cancellation and Refunds: Allow users to cancel reservations and initiate refund requests according to specified cancellation policies.
- Reporting and Analytics: Provide comprehensive reporting and analytical capabilities for monitoring ticket sales, passenger demographics, and other key performance indicators.

3. Functional Requirements

3.1 Train Information

- The system should maintain an up-to-date database of train schedules, including departure and arrival times, station stops, and route information.
- It should support searching for trains based on source and destination stations, travel dates, and preferred departure or arrival times.
- The software should provide information on train classes, seat types, and fare details for each train.
- It should allow users to view train routes, including intermediate stops and estimated travel durations.

3.2 Seat Availability and Booking

- The system should enable users to check seat availability on specific trains for desired travel dates and classes.
- It should display seat availability status, such as the number of available seats, waiting list status, and coach-wise seat distribution.
- The software should allow users to select preferred trains, classes, and seat types for booking.
- It should support the reservation of multiple seats in a single transaction for groups or families traveling together.
- The system should implement seat locking mechanisms to prevent multiple users from booking the same seat simultaneously.

3.3 Ticket Generation

- The software should generate electronic tickets or e-tickets upon successful reservation, containing passenger details, seat numbers, train information, and travel itinerary.
- It should support ticket delivery via email or SMS, allowing passengers to present e-tickets for verification during travel.

- The system should provide the option to print physical tickets at designated ticket counters or kiosks for users who prefer paper tickets.
- The software should validate ticket authenticity and prevent unauthorized duplication or modification.

3.4 Payment Processing

- The system should integrate with secure online payment gateways to facilitate online payment for ticket reservations.
- It should support various payment methods, including credit cards, debit cards, net banking, and mobile wallets.
- The software should encrypt sensitive payment information during transmission and ensure secure handling of user financial data.
- It should generate payment receipts and update reservation status upon successful payment.

3.5 Cancellation and Refunds

- The system should allow users to cancel their reservations within a specified cancellation period.
- It should implement refund policies and calculate refund amounts based on predefined rules and cancellation timings.
- The software should generate cancellation confirmations and process refund requests in a timely manner.
- It should update seat availability and reservation status upon cancellation.

3.6 Reporting and Analytics

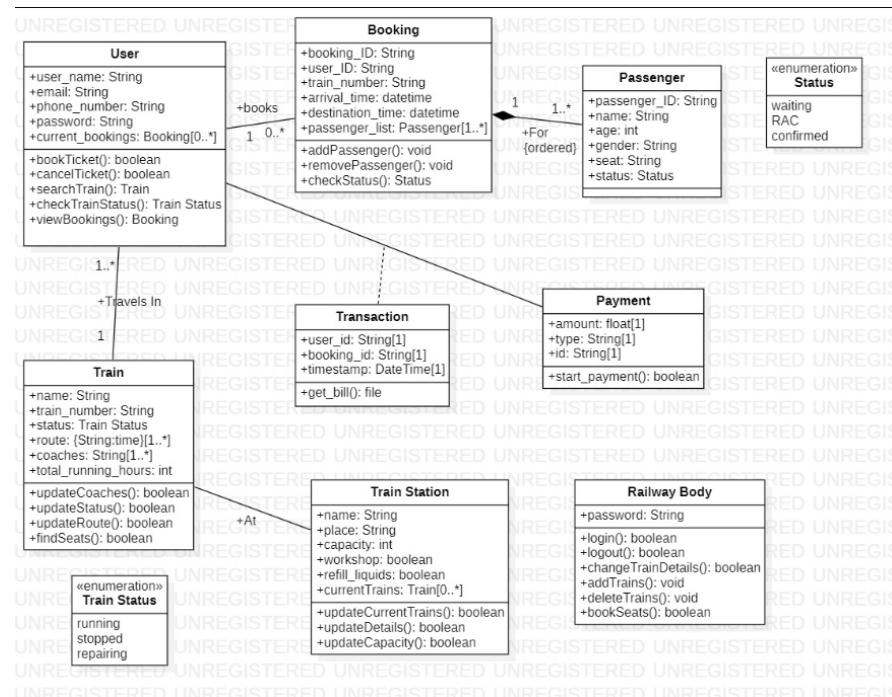
- The system should offer pre-defined and customizable reports on ticket sales, passenger demographics, revenue analysis, and other relevant metrics.
- It should provide data visualization options, such as charts and graphs, for easy understanding and analysis.
- The software should support exporting reports in various formats (e.g., PDF, CSV) for further analysis or integration with external systems.

- It should allow authorized users to generate ad-hoc reports based on specific criteria or filters.

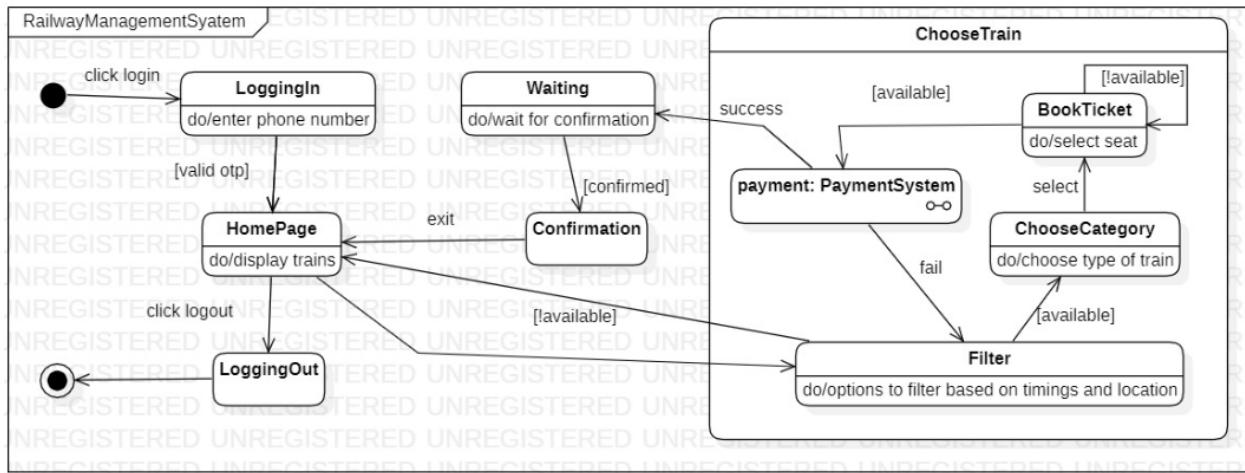
4. Non-Functional Requirements

- The Railway Reservation System should be user-friendly, intuitive, and accessible to passengers with varying levels of technical expertise.
- It should have a high availability and response time to handle concurrent user requests during peak booking periods.
- The software should be secure, implementing appropriate measures to protect user data, including encryption, access control, and data privacy compliance.
- It should support integration with other systems, such as payment gateways, passenger information systems, and seat inventory management systems.

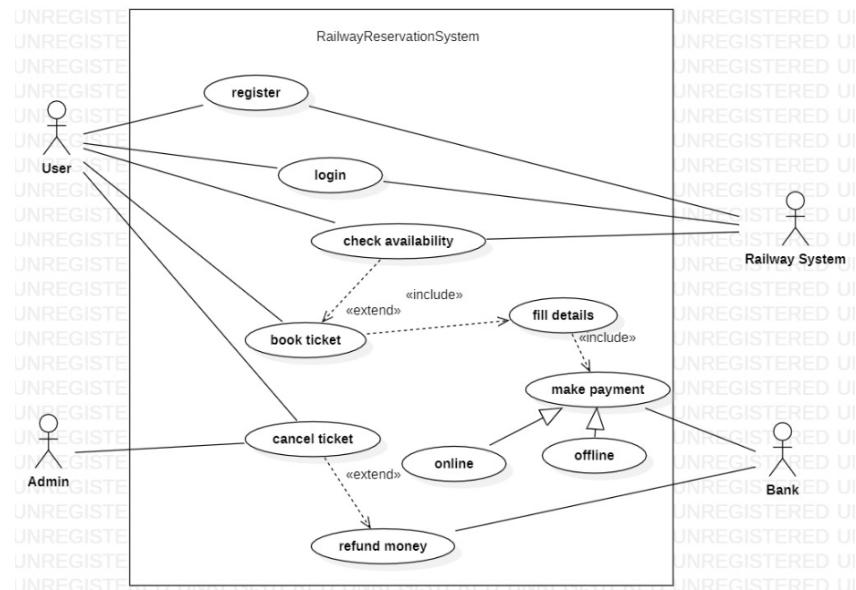
class diagram:



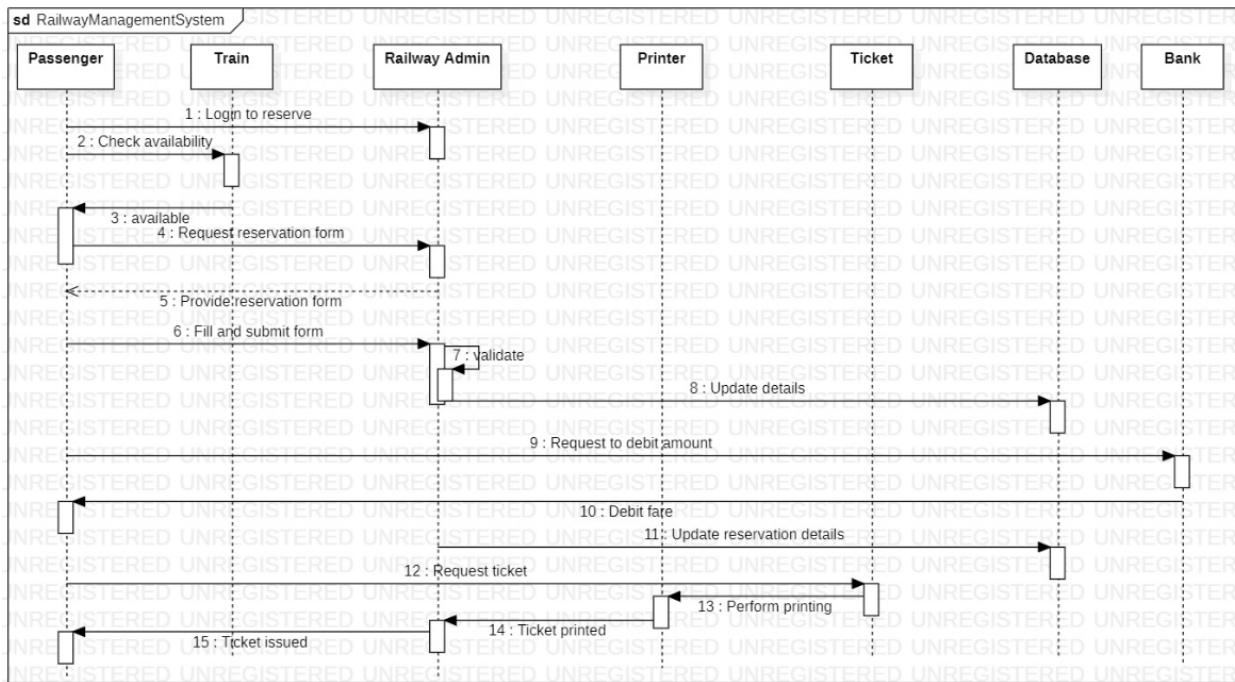
State diagram:



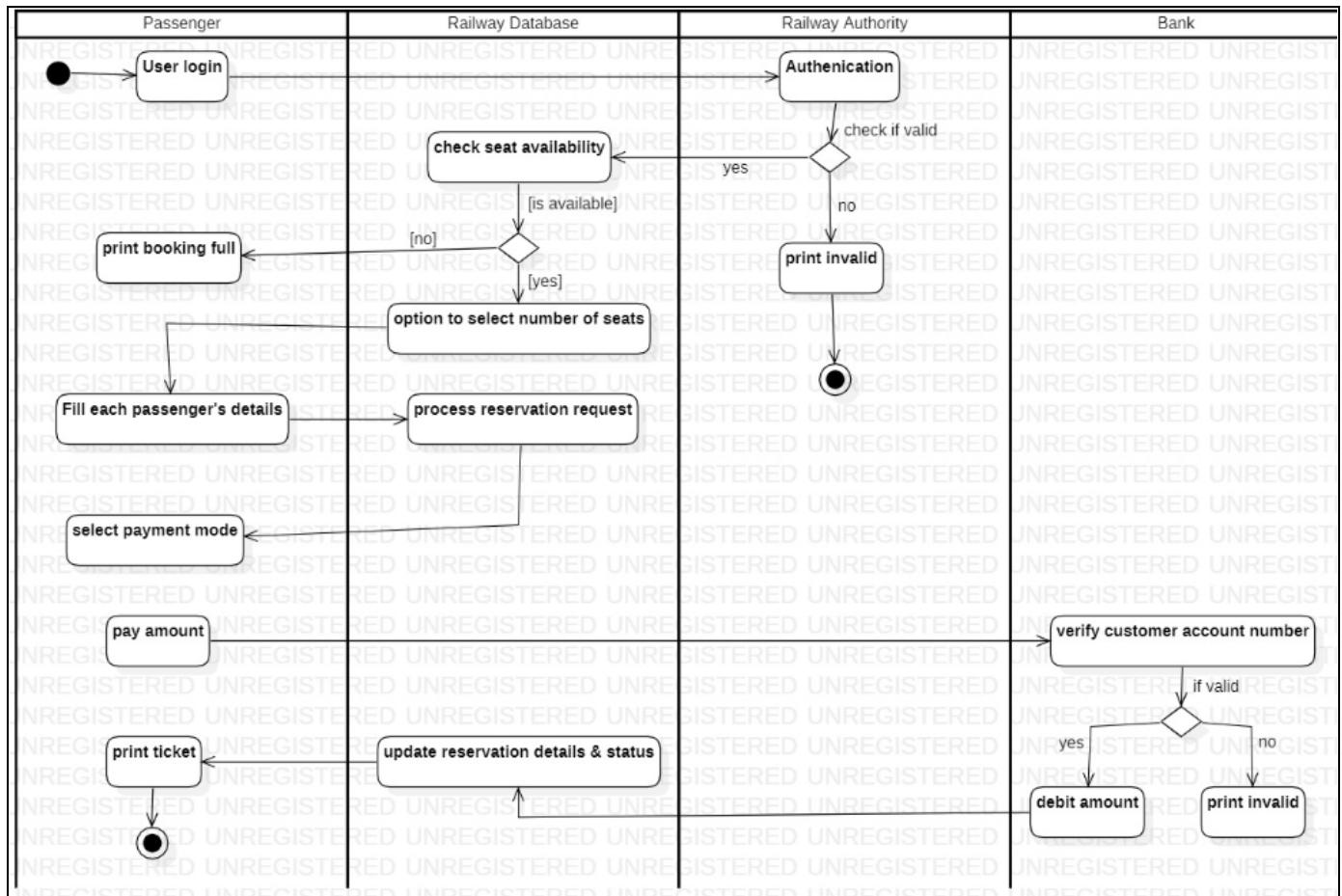
Use case diagram:



sequence diagram:



activity diagram:



online shopping system

1. Introduction The purpose of this Software Requirements Specification (SRS) document is to outline the requirements for the development of an Online Shopping System. The software aims to provide a user-friendly and secure platform for customers to browse, search, select, and purchase products online. It will facilitate efficient product management, shopping cart functionality, secure payment processing, and order fulfillment. The system will also include administrative features for managing products, inventory, customer data, and order processing.

2. Scope The Online Shopping System will consist of the following key modules and functionalities:

- Product Catalog: Display a comprehensive catalog of products with details, images, prices, and user reviews.
- Search and Filtering: Provide search functionality to allow customers to search for products based on various criteria and apply filters.
- Shopping Cart: Allow customers to add products to a shopping cart, update quantities, and proceed to checkout.
- Secure Payment Processing: Integrate with secure payment gateways to facilitate online payments using various payment methods.
- Order Management: Manage customer orders, order processing, and order tracking for efficient order fulfillment.
- Customer Accounts: Enable customers to create accounts, manage personal information, and track order history.
- Reporting and Analytics: Provide reporting and analytical capabilities to monitor sales, customer behavior, and inventory levels.

3. Functional Requirements

3.1 Product Catalog

- The system should maintain a centralized product catalog, including product names, descriptions, prices, images, and stock availability.
- It should support categorizing products into different categories or departments for better organization and searchability.
- The software should allow customers to view detailed product information, including specifications, features, and user reviews.
- It should provide options for customers to sort products based on popularity, price, rating, or other relevant criteria.

3.2 Search and Filtering

- The system should provide a search functionality allowing customers to search for products using keywords, product names, or other relevant attributes.
- It should support advanced filtering options to refine search results based on categories, price ranges, brands, sizes, or other product attributes.
- The software should display search results in a clear and organized manner, with options for pagination or infinite scrolling for large result sets.
- It should provide sorting options to order search results based on relevance, price, popularity, or customer ratings.

3.3 Shopping Cart

- The system should allow customers to add products to a virtual shopping cart, view the contents, and modify quantities or remove items.
- It should display the total order summary, including the subtotal, applicable taxes, shipping charges, and any discounts or promotions.
- The software should support saving the shopping cart for future sessions, allowing customers to resume their shopping from where they left off.
- It should provide options for customers to apply discount codes or gift vouchers to their orders, if applicable.

3.4 Secure Payment Processing

- The system should integrate with secure payment gateways to facilitate online payment for customer orders.
- It should support various payment methods, such as credit cards, debit cards, net banking, mobile wallets, or other digital payment options.
- The software should encrypt sensitive payment information during transmission and ensure secure handling of customer financial data.
- It should provide payment confirmation and order status updates to customers upon successful payment.

3.5 Order Management

- The system should capture and store customer orders, including order details, customer information, shipping addresses, and payment records.
- It should support order processing workflows, including order confirmation, order fulfillment, and shipment tracking.
- The software should allow authorized users to manage order statuses, update shipping information, and generate invoices or shipping labels.
- It should send automated order notifications to customers, providing updates on order status, shipment tracking, and delivery estimates.

3.6 Customer Accounts

- The system should allow customers to create user accounts, providing personal information, contact details, and shipping addresses.
- It should provide a secure authentication mechanism to protect customer accounts and ensure authorized access.
- The software should allow customers to view and update their account information, including password changes, address updates, and communication preferences.

- It should maintain a history of customer orders, allowing customers to view and track their order history and shipment status.

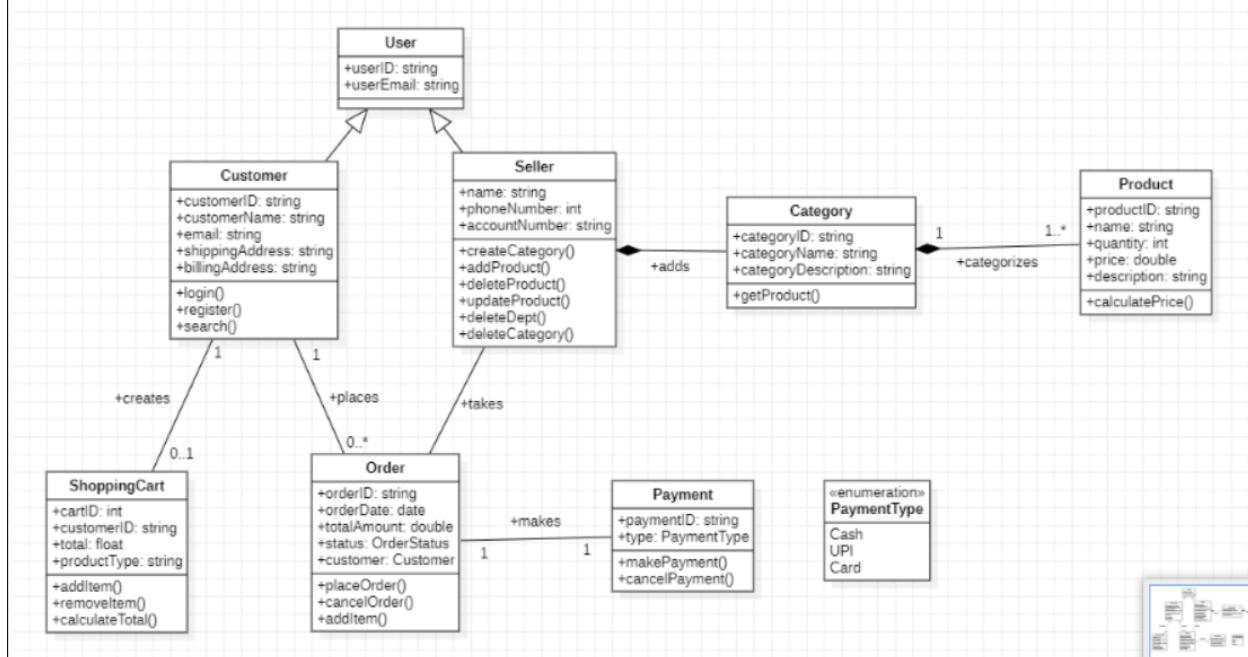
3.7 Reporting and Analytics

- The system should offer pre-defined and customizable reports on sales, customer behavior, product popularity, and other key performance indicators.
- It should provide data visualization options, such as charts and graphs, for easy understanding and analysis.
- The software should support exporting reports in various formats (e.g., PDF, CSV) for further analysis or integration with external systems.
- It should allow authorized users to generate ad-hoc reports based on specific criteria or filters.

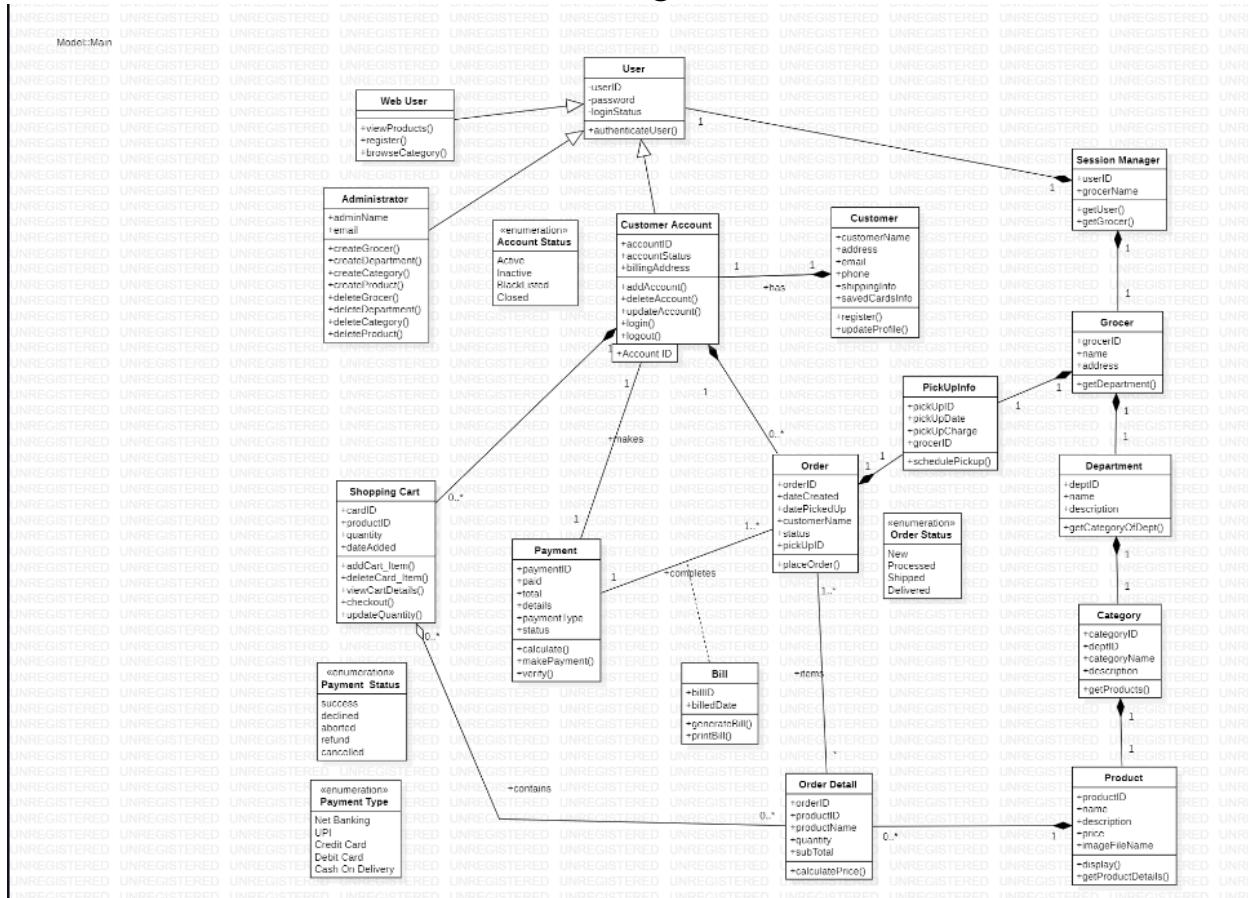
4. Non-Functional Requirements

- The Online Shopping System should be user-friendly, responsive, and accessible across different devices and screen sizes.
- It should have a high availability and response time to handle concurrent user requests and provide a seamless shopping experience.
- The software should be secure, implementing appropriate measures to protect customer data, including encryption, access control, and compliance with data privacy regulations.
- It should support scalability to handle a growing number of products, customers, and transactions.

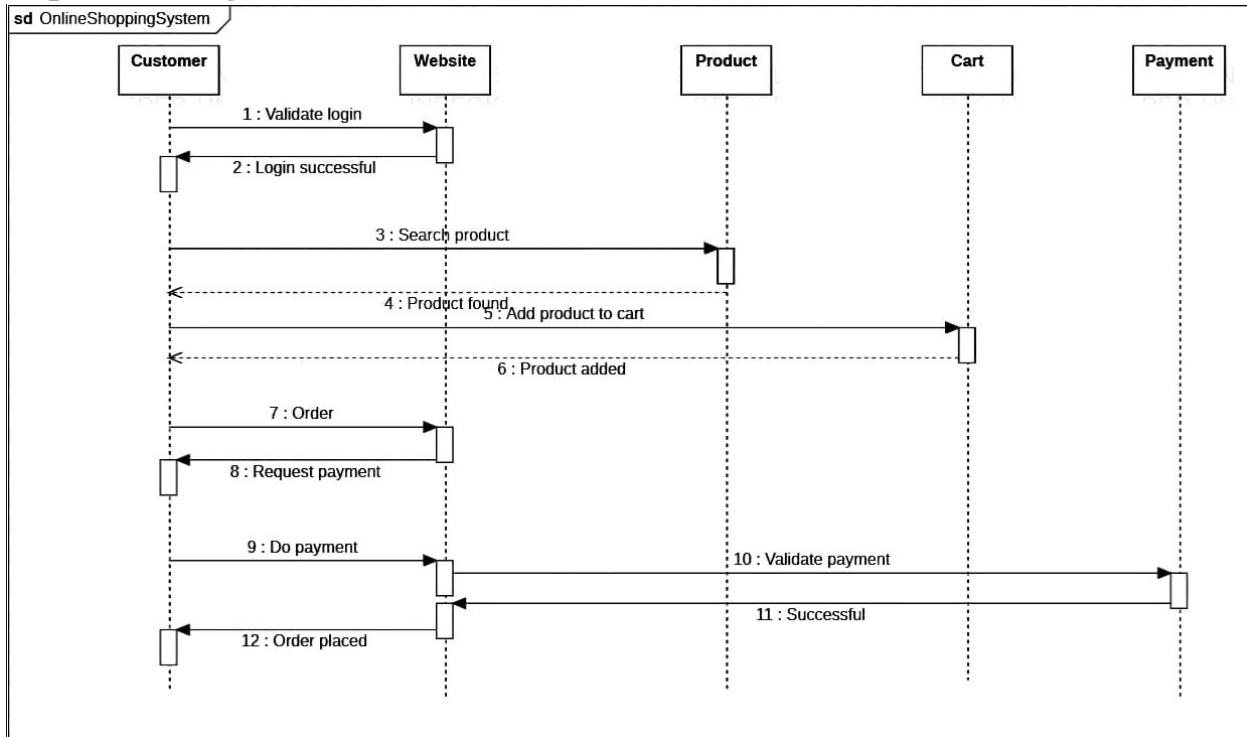
class diagram:



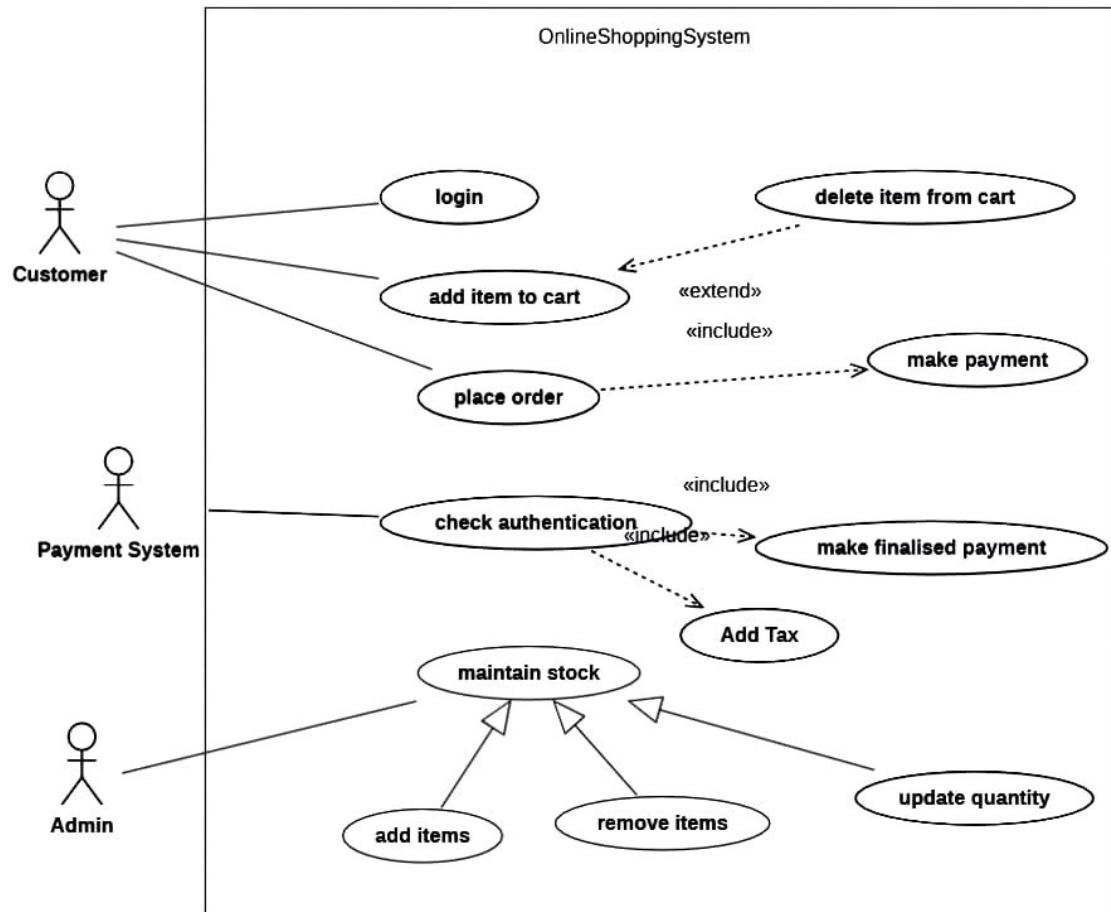
State diagram:



sequence diagram:



use case diagram:



Activity diagram:

