

B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU



OOMD Lab Report

Submitted in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted by:
Shruti Dasamandam

1BM20CS216

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

I, Shruti Dasamandam (1BM20CS216) student of 6th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this Object Orientation Modelling laboratory Practicals 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

Shruti Dasamandam 1BM20CS216

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 Shruti Dasamandam (1BM20CS216) during the academic year 2022-2023.

Signature of the Faculty in Charge

Table of Contents

Sl No	Title	Page no
1	SRS Document	5-25
2	Ch 1: Class diagrams	26-32
3	Ch 2: State Diagrams	33-39
4	Ch 3: Activity Diagrams	40-46
5	Ch 4: Use case Diagrams	47-53
6	Ch 5: Sequence diagrams	54-60

SRS Document

1. Hotel Management

1. Introduction

The Hotel Management System is a software application designed to streamline and automate various operations in a hotel. It aims to provide an efficient and user-friendly interface for managing hotel bookings, guest information, room services, billing, and other related tasks.

2. Scope

The Hotel Management System will encompass the following functionalities:

Guest management: Registering guest details, maintaining guest records, and managing reservations.

Room management: Managing room availability, room types, and assigning rooms to guests.

Booking management: Handling guest bookings, cancellations, and modifications.

Check-in/check-out: Managing guest arrivals and departures, including room allocation and key management.

Billing and invoicing: Generating bills, managing payments, and generating invoices.

Reporting: Generating various reports, including occupancy, revenue, and guest statistics.

Admin functionality: User management, access control, and system configuration.

3. System Requirements

3.1 Functional Requirements

3.1.1 Guest Management

The system shall allow the front desk staff to register new guests by capturing their personal information such as name, address, contact details, and identification documents. The system shall provide a search functionality to retrieve guest information based on criteria such as name, booking ID, or check-in/check-out dates.

3.1.2 Room Management

The system shall maintain an inventory of available rooms, including room types, rates, and occupancy status.

The system shall allow the front desk staff to assign rooms to guests based on availability and guest preferences.

The system shall provide a mechanism to block or reserve specific rooms for maintenance or special purposes.

3.1.3 Booking Management

The system shall enable guests or staff to make new bookings by specifying check-in and check-out dates, room type, and additional services.

The system shall support modification and cancellation of bookings while ensuring data integrity and appropriate notifications to guests and staff.

3.1.4 Check-in/Check-out

The system shall facilitate the check-in process by verifying guest details, allocating rooms, and issuing room keys or access cards.

The system shall handle the check-out process by updating room status, calculating charges, and generating bills.

3.1.5 Billing and Invoicing

The system shall calculate charges based on room rates, additional services, and any applicable discounts or taxes.

The system shall generate itemized bills for guests, including a breakdown of charges, taxes, and payment instructions.

The system shall support different payment methods, such as cash, credit cards, and online transfers.

3.1.6 Reporting

The system shall generate reports on room occupancy, revenue, guest statistics, and other relevant metrics.

The system shall provide filters and options for customizing reports based on specific criteria and time periods.

3.1.7 Admin Functionality

The system shall allow administrators to manage user accounts, assign roles and permissions, and reset passwords.

The system shall provide configuration options for setting up room types, rates, taxes, and other system parameters.

3.2 Non-Functional Requirements

The system shall have a user-friendly interface with intuitive navigation and responsive design.

The system shall maintain data security and privacy by implementing appropriate authentication and access control mechanisms.

The system shall be available 24/7 with minimal downtime for maintenance and upgrades.

The system shall be scalable to handle a large number of guests, rooms, and transactions. The system shall generate reports and perform operations in a timely manner to ensure efficient decision-making.

4. System Architecture

The Hotel Management System will be developed as a web-based application using a three-tier architecture consisting of a presentation layer, application layer, and data layer. The

system will use a relational database management system (RDBMS) to store and retrieve data.

5. Assumptions and Constraints

The Hotel Management System will be developed using the Java programming language and the Spring Framework.

The system will be compatible with modern web browsers and responsive to different screen sizes.

2. Library Management

1. Introduction

The Library Management System is a software application designed to automate and streamline various library operations, including book management, member management, borrowing and returning books, and generating reports. It aims to provide an efficient and user-friendly interface for both library staff and patrons.

2. Scope

The Library Management System will encompass the following functionalities:

Book management: Adding, updating, and deleting books from the library catalog, including information such as title, author, ISBN, publication year, and availability status.

Member management: Registering library members, capturing their personal information, and managing member records.

Borrowing and returning books: Facilitating the borrowing and returning process, including checking book availability, managing due dates, and issuing fines for overdue books.

Reservation system: Allowing patrons to reserve books that are currently on loan by other members.

Reporting: Generating various reports, including book inventory, member statistics, overdue books, and financial transactions.

Admin functionality: User management, access control, and system configuration.

3. System Requirements

3.1 Functional Requirements

3.1.1 Book Management

The system shall allow library staff to add new books to the library catalog by entering details such as title, author, ISBN, publication year, and quantity.

The system shall support updating book information and marking books as available or unavailable.

The system shall provide search functionality to retrieve books based on criteria such as title, author, or ISBN.

3.1.2 Member Management

The system shall enable library staff to register new members by capturing their personal information, such as name, address, contact details, and membership type. The system shall allow staff to manage member records, including updates, suspensions, and cancellations.

3.1.3 Borrowing and Returning Books

The system shall facilitate the borrowing process by checking the availability of books, recording borrower information, and updating the book's status to "on loan." The system shall enforce borrowing limits, due dates, and handle renewals and

reservations.

The system shall manage the returning process, update book availability, calculate fines for overdue books, and handle book reservations.

3.1.4 Reservation System

The system shall allow library patrons to reserve books that are currently checked out by other members.

The system shall maintain a reservation queue and notify patrons when the reserved books become available.

The system shall manage the reservation process, including cancellation and expiration of reservations.

3.1.5 Reporting

The system shall generate reports on book inventory, member statistics, borrowing history, overdue books, fines, and other relevant metrics.

The system shall provide filters and options for customizing reports based on specific criteria and time periods.

3.1.6 Admin Functionality

The system shall allow administrators to manage user accounts, assign roles and permissions, and reset passwords.

The system shall provide configuration options for setting up membership types, borrowing rules, fines, and other system parameters.

3.2 Non-Functional Requirements

The system shall have a user-friendly interface with intuitive navigation and responsive design.

The system shall maintain data security and privacy by implementing appropriate authentication and access control mechanisms.

The system shall be available 24/7 with minimal downtime for maintenance and upgrades.

The system shall be scalable to handle a large number of books, members, and transactions.

The system shall be able to integrate with external systems for book information retrieval and online catalog searches.

4. System Architecture

The Library Management System will be developed as a web-based application using a three-tier architecture consisting of a presentation layer, application layer, and data layer. The

system will use a relational database management system (RDBMS) to store and retrieve data.

5. Assumptions and Constraints

The Library Management System will be developed using a modern programming language and a suitable framework.

The system will be compatible with major web browsers and responsive to different screen sizes.

The system will integrate with external APIs or databases for book information retrieval. The hardware and network infrastructure required to host and deploy the system will be provided separately.

3. Credit Card Processing

1. Introduction

The Credit Card Processing System is a software application designed to securely process credit card transactions for businesses. It facilitates the authorization, encryption, and routing of credit card payments, ensuring a seamless and secure payment experience for customers.

2. Scope

The Credit Card Processing System will encompass the following functionalities:

Payment gateway integration: Integration with a payment gateway to securely process credit card transactions.

Authorization and encryption: Verification of cardholder information, including card number, expiration date, CVV, and address verification.

Transaction routing: Routing of authorized transactions to acquiring banks or payment service providers (PSPs).

Settlement and fund transfer: Transfer of funds from the customer's account to the merchant's account upon successful transaction settlement.

Security and compliance: Compliance with Payment Card Industry Data Security Standard (PCI DSS) requirements to protect cardholder data.

Error handling and reporting: Handling of transaction errors, generation of error reports, and notification to appropriate parties.

Integration and customization: Integration with business applications and customization options for payment workflows and user interfaces.

3. System Requirements

3.1 Functional Requirements

3.1.1 Payment Gateway Integration

The system shall integrate with a compatible payment gateway to process credit card transactions.

The system shall support secure communication with the payment gateway using industry-standard encryption protocols.

The system shall handle the transfer of transaction data to the payment gateway and receive response codes for transaction status.

3.1.2 Authorization and Encryption

The system shall verify cardholder information, including card number, expiration date, CVV, and address verification.

The system shall communicate with the card networks (Visa, Mastercard, etc.) to authorize transactions and perform risk assessments.

The system shall encrypt sensitive cardholder data during transmission and storage to ensure data security.

3.1.3 Transaction Routing

The system shall route authorized transactions to the appropriate acquiring banks or payment service providers (PSPs) based on configured rules.

The system shall handle the secure transmission of transaction data to acquiring banks or PSPs for settlement processing.

3.1.4 Settlement and Fund Transfer

The system shall initiate the transfer of funds from the customer's account to the merchant's account upon successful transaction settlement.

The system shall support automated settlement processes based on predefined settlement cycles and rules.

The system shall generate settlement reports for reconciliation and financial reporting purposes.

3.1.5 Security and Compliance

The system shall comply with Payment Card Industry Data Security Standard (PCI DSS) requirements to protect cardholder data.

The system shall implement secure authentication mechanisms, access controls, and encryption methods to ensure data confidentiality and integrity.

The system shall support periodic security audits and vulnerability assessments to maintain compliance.

3.1.6 Error Handling and Reporting

The system shall handle transaction errors, such as declined transactions, communication failures, or processing errors.

The system shall generate error reports, including detailed error codes and descriptions, to assist in troubleshooting and resolution.

The system shall notify appropriate parties, such as administrators or support personnel, of critical errors and exceptions.

3.1.7 Integration and Customization

The system shall provide integration options, such as APIs or SDKs, for seamless integration with business applications, websites, or mobile apps.

The system shall support customization of payment workflows, user interfaces, and branding to align with the business's requirements and branding guidelines.

3.2 Non-Functional Requirements

The system shall have a high level of availability and reliability to ensure uninterrupted payment processing.

The system shall provide robust security measures to protect sensitive cardholder data

and prevent unauthorized access.

The system shall have a user-friendly interface for easy configuration, monitoring, and reporting of transactions.

The system shall be scalable to handle high volumes of transactions and accommodate business growth.

The system shall provide comprehensive documentation, including installation guides, user manuals, and API documentation.

4. System Architecture

The Credit Card Processing System will be developed as a web-based application using a three-tier architecture consisting of a presentation layer, application layer, and data layer. The system will utilize industry-standard encryption protocols to ensure secure communication and data storage.

5. Assumptions and Constraints

The Credit Card Processing System will be developed using a modern programming language and suitable frameworks.

The system will integrate with compatible payment gateways and acquiring banks or payment service providers (PSPs).

The system will comply with applicable industry regulations, including Payment Card Industry Data Security Standard (PCI DSS) requirements.

The hardware and network infrastructure required to host and deploy the system will be provided separately.

4. Stock Maintenance

1. Introduction

The Stock Maintenance System is a software application designed to manage and track inventory for a business. It provides functionalities to monitor stock levels, track item movements, update inventory records, and generate reports. The system aims to optimize stock control and improve overall inventory management efficiency.

2. Scope

The Stock Maintenance System will encompass the following functionalities:

Stock management: Adding, updating, and deleting stock items, including details such as item name, description, quantity, unit price, and supplier information.

Stock movement tracking: Recording stock inflows (e.g., purchases, transfers) and outflows (e.g., sales, returns) to maintain accurate stock levels.

Stock status monitoring: Tracking stock quantities, including available, reserved, and sold quantities, and generating alerts for low stock levels.

Stock valuation: Calculating the total value of the stock based on unit prices and quantities.

Reporting and analysis: Generating reports on stock levels, stock movements, stock valuations, and other relevant metrics for decision-making.

Notifications and alerts: Sending notifications and alerts to appropriate users when stock levels reach predefined thresholds or when stock movements occur.

3. System Requirements

3.1 Functional Requirements

3.1.1 Stock Management

The system shall allow authorized users to add new stock items by entering details such as item name, description, quantity, unit price, and supplier information.

The system shall provide options to update stock item information, including quantity, unit price, and supplier details.

The system shall support deleting stock items from the inventory.

3.1.2 Stock Movement Tracking

The system shall record stock inflows, such as purchases from suppliers and stock transfers from other locations or warehouses.

The system shall record stock outflows, such as sales to customers, returns, and stock transfers to other locations or warehouses.

The system shall update stock quantities based on stock movements to maintain accurate inventory records.

3.1.3 Stock Status Monitoring

The system shall track stock quantities, including available, reserved, and sold quantities, for each stock item.

The system shall provide real-time visibility of stock levels and generate alerts when stock levels fall below predefined thresholds.

The system shall allow users to search and view stock availability for specific items.

3.1.4 Stock Valuation

The system shall calculate the total value of the stock based on the unit prices and quantities of individual stock items.

The system shall provide options to generate stock valuation reports for a specific time period or as of a particular date.

3.1.5 Reporting and Analysis

The system shall generate reports on stock levels, stock movements, stock valuations, and other relevant metrics.

The system shall provide filtering and sorting options for customized reports based on specific criteria and time periods.

The system shall support exporting reports in common formats, such as PDF or

3.1.6 Notifications and Alerts

The system shall send notifications or alerts to appropriate users or groups when stock levels reach predefined thresholds or when significant stock movements occur. The system shall allow users to configure notification preferences and recipients.

3.2 Non-Functional Requirements

The system shall have a user-friendly interface with intuitive navigation and responsive design.

The system shall ensure data security and privacy by implementing appropriate authentication and access control mechanisms.

The system shall be available 24/7 with minimal downtime for maintenance and upgrades.

The system shall be scalable to handle a large number of stock items and transactions. The system shall provide data backup and recovery mechanisms to prevent data loss.

4. System Architecture

The Stock Maintenance System will be developed as a web-based application using a three-tier architecture comprising a presentation layer, application layer, and data layer. The system will utilize a relational database management system (RDBMS) to store and retrieve inventory data.

5. Assumptions and Constraints

The Stock Maintenance System will be developed using a modern programming language and suitable frameworks.

The system will be compatible with major web browsers and responsive to different screen sizes.

The hardware and network infrastructure required to host and deploy the system will be provided separately

5. Online Shopping

1. Introduction

The Online Shopping System is a web-based application designed to facilitate the process of buying products or services online. It provides a platform for customers to browse, search, select, and purchase items, as well as manage their orders and payments. The system aims to provide a seamless and user-friendly online shopping experience.

2. Scope

The Online Shopping System will encompass the following functionalities:

User registration and authentication: Allow customers to create accounts, log in, and manage their profiles.

Product catalog browsing: Display a catalog of available products with details such as images, descriptions, prices, and ratings.

Product search and filtering: Provide search functionality to help customers find specific products based on various criteria (e.g., keywords, categories, prices).

Shopping cart management: Enable customers to add products to their shopping carts, modify quantities, and remove items.

Order placement and tracking: Allow customers to place orders, select shipping methods, and track the status of their orders.

Payment processing: Support secure and convenient payment methods, such as credit/debit cards, digital wallets, or online banking.

Order history and customer reviews: Provide a history of past orders for customers to reference, and allow customers to leave reviews and ratings for products. User notifications: Send notifications to customers regarding order status updates, promotions, and other relevant information.

3. System Requirements

3.1 Functional Requirements

3.1.1 User Registration and Authentication

The system shall provide a user registration process, allowing customers to create accounts with their personal information.

The system shall support authentication mechanisms to verify the identity of registered customers during login.

3.1.2 Product Catalog Browsing

The system shall display a catalog of products, including details such as images, descriptions, prices, and ratings.

The system shall support pagination or infinite scrolling to handle large product catalogs.

3.1.3 Product Search and Filtering

The system shall provide a search functionality that allows customers to search for products based on keywords, categories, or other specified criteria.

The system shall allow customers to filter products based on attributes such as price range, brand, size, color, etc.

3.1.4 Shopping Cart Management

The system shall allow customers to add products to their shopping carts, specify quantities, and remove items.

The system shall calculate the subtotal, including taxes and shipping costs, and display the total amount in the shopping cart.

3.1.5 Order Placement and Tracking

The system shall enable customers to place orders, select shipping methods, and provide shipping addresses.

The system shall generate unique order IDs and provide customers with order confirmation details.

The system shall allow customers to track the status of their orders, including shipping updates and estimated delivery dates.

3.1.6 Payment Processing

The system shall support secure payment processing, integrating with one or more payment gateways to handle transactions.

The system shall allow customers to select from various payment methods, such as credit/debit cards, digital wallets, or online banking.

3.1.7 Order History and Customer Reviews

The system shall maintain a history of customers' past orders for reference purposes. The system shall allow customers to leave reviews and ratings for products they have purchased.

3.1.8 User Notifications

The system shall send notifications to customers regarding order status updates, promotions, and other relevant information.

The system shall provide options for customers to manage their notification preferences.

3.2 Non-Functional Requirements

The system shall have a user-friendly interface with intuitive navigation and responsive design for seamless user experience across different devices.

The system shall ensure data security and privacy by implementing appropriate authentication, encryption, and access control mechanisms.

The system shall be scalable to handle a large number of products, customers, and concurrent users.

The system shall have a fast and efficient search functionality to provide quick and accurate product search results.

The system shall support multiple languages, currencies, and international shipping options.

The system shall provide comprehensive documentation, including user manuals and FAQs, to assist customers.

4. System Architecture

The Online Shopping System will be developed as a web-based application using a three-tier architecture, consisting of a presentation layer, application layer, and data layer. The system will utilize a relational database management system (RDBMS) to store and retrieve data.

5. Assumptions and Constraints

The Online Shopping System will be developed using a modern programming language and suitable frameworks.

The system will be compatible with major web browsers and responsive to different screen sizes.

The hardware and network infrastructure required to host and deploy the system will be provided separately.

6. Railway Management

1. Introduction

The Railway Management System is a software application designed to manage various aspects of railway operations, including ticketing, scheduling, reservation, train management, and passenger information. The system aims to streamline railway operations, enhance passenger experience, and improve overall efficiency.

2. Scope

The Railway Management System will encompass the following functionalities:

Ticketing and reservation: Provide a platform for customers to book tickets, check seat availability, and make reservations for train journeys.

Train scheduling and routing: Manage train schedules, assign platforms, and determine optimal routes for efficient train operations.

Passenger information: Provide real-time information to passengers regarding train schedules, delays, platform changes, and other relevant updates.

Train management: Track and manage trains, including tracking train locations, monitoring their statuses, and managing maintenance schedules.

Fare management: Calculate fares based on distance, class, and other parameters, and support fare adjustments and discounts.

Reporting and analytics: Generate reports and analytics on passenger traffic, revenue, train performance, and other key metrics for decision-making.

Integration with external systems: Integrate with external systems such as payment gateways, station information systems, and passenger information display systems.

3. System Requirements

3.1 Functional Requirements

3.1.1 Ticketing and Reservation

The system shall allow customers to search and book train tickets for specific journeys, specifying the source and destination stations, date, and class.

The system shall provide real-time information on seat availability, including seat maps and reserved/unreserved seat indications.

The system shall generate unique ticket numbers and provide customers with e-tickets or print-at-home tickets.

3.1.2 Train Scheduling and Routing

The system shall manage train schedules, including departure and arrival times, platform assignments, and stoppage durations.

The system shall determine optimal routes for trains, considering factors such as distance, speed, and track availability.

The system shall support the modification of train schedules and routes as needed.

3.1.3 Passenger Information

The system shall provide real-time information to passengers regarding train schedules, delays, platform changes, and other relevant updates.

The system shall display information on station facilities, amenities, and services available at each station.

The system shall support announcements, notifications, and display boards at stations to communicate important information to passengers.

3.1.4 Train Management

The system shall track train locations in real-time using GPS or other tracking mechanisms.

The system shall monitor train statuses, including on-time performance, delays, and cancellations.

The system shall schedule and manage train maintenance activities, including regular inspections and repairs.

3.1.5 Fare Management

The system shall calculate fares based on distance, class, and other parameters specified by railway authorities.

The system shall support fare adjustments, discounts, and promotions as per railway policies.

The system shall provide fare information to customers during the ticket booking process.

3.1.6 Reporting and Analytics

The system shall generate reports and analytics on passenger traffic, revenue, train performance, and other key metrics.

The system shall support customizable reports based on specified criteria, time periods, and geographical regions.

The system shall provide data visualization tools to present analytics in a user-friendly and informative manner.

3.1.7 Integration with External Systems

The system shall integrate with external systems, such as payment gateways, to process ticket payments securely.

The system shall integrate with station information systems to provide accurate and up-to-date information to passengers.

The system shall integrate with passenger information display systems at stations to deliver real-time updates to passengers.

3.2 Non-Functional Requirements

The system shall have a user-friendly interface with intuitive navigation and responsive design for seamless user experience across different devices.

The system shall ensure data security and privacy, implementing appropriate authentication, encryption, and access control mechanisms.

The system shall be scalable to handle a large number of users, trains, and transactions. The system shall be highly available, with minimal downtime for maintenance and upgrades.

The system shall provide comprehensive documentation, including user manuals and technical guides, to assist users and administrators.

4. System Architecture

The Railway Management System will be developed as a web-based application using a three-tier architecture, consisting of a presentation layer, application layer, and data layer. The system will utilize a relational database management system (RDBMS) to store and retrieve data.

5. Assumptions and Constraints

The Railway Management System will be developed using a modern programming language and suitable frameworks.

The system will be compatible with major web browsers and responsive to different screen sizes.

The hardware and network infrastructure required to host and deploy the system will be provided separately.

7. Passport Automation

1. Introduction

The Passport Automation System is a software application designed to streamline and automate the process of issuing passports. It provides functionalities to facilitate the application, verification, and issuance of passports, as well as managing passport-related data and appointments. The system aims to enhance the efficiency, accuracy, and security of the passport issuance process.

2. Scope

The Passport Automation System will encompass the following functionalities:

Passport application submission: Allow applicants to submit passport applications online, providing necessary personal information, supporting documents, and appointment preferences.
Application verification and processing: Automate the verification and processing of passport applications, including identity verification, background checks, and document validation.

Appointment scheduling: Provide a platform for applicants to schedule appointments for biometric data capture and document submission.

Passport issuance: Generate passport booklets and personalize them with applicant information, photographs, and other required details.

Data management: Maintain a secure and centralized database of passport-related information, including applicant details, application statuses, and issuance records. Reporting and analytics: Generate reports and analytics on passport application trends, processing times, and other key metrics for monitoring and decision-making. Integration with external systems: Integrate with identity verification systems, document verification databases, and payment gateways for seamless information exchange and transaction processing.

3. System Requirements

3.1 Functional Requirements

3.1.1 Passport Application Submission

The system shall provide an online platform for applicants to fill and submit passport applications, providing personal information, travel details, and supporting documents. The system shall validate application data for completeness, accuracy, and adherence to specified requirements.

The system shall generate unique application numbers and provide applicants with acknowledgments and references.

3.1.2 Application Verification and Processing

The system shall automate the verification and processing of passport applications, including identity verification, background checks, and document validation.

The system shall integrate with external systems or databases to perform identity verification and background checks.

The system shall maintain an audit trail of verification activities and decisions.

3.1.3 Appointment Scheduling

The system shall allow applicants to schedule appointments for biometric data capture and document submission.

The system shall provide options for applicants to select preferred dates, time slots, and locations for appointments.

The system shall send appointment confirmations and reminders to applicants via email or SMS.

3.1.4 Passport Issuance

The system shall generate passport booklets with personalized applicant information, photographs, and other required details.

The system shall ensure the security and integrity of passport booklets during the printing and personalization process.

The system shall maintain a record of issued passports, including details such as passport numbers, issuance dates, and expiration dates.

3.1.5 Data Management

The system shall maintain a centralized database to store and manage passport-related information, including applicant details, application statuses, and issuance records. The system shall implement appropriate security measures to protect sensitive passport data from unauthorized access or manipulation.

The system shall provide search and retrieval functionalities to access passport information based on specified criteria.

3.1.6 Reporting and Analytics

The system shall generate reports and analytics on passport application trends, processing times, and other key metrics.

The system shall support customizable reports based on specified criteria, time periods, and geographical regions.

The system shall provide data visualization tools to present analytics in a user-friendly and informative manner.

3.1.7 Integration with External Systems

The system shall integrate with external systems or databases for identity verification, document validation, and background checks.

The system shall integrate with payment gateways to process passport application fees securely.

The system shall ensure seamless information exchange and data synchronization with external systems.

3.2 Non-Functional Requirements

The system shall have a user-friendly interface with intuitive navigation and responsive design for seamless user experience across different devices.

The system shall ensure data security and privacy, implementing appropriate authentication, encryption, and access control mechanisms.

The system shall be scalable to handle a large number of applicants, transactions, and concurrent users.

The system shall be highly available, with minimal downtime for maintenance and upgrades.

The system shall provide comprehensive documentation, including user manuals and technical guides, to assist users and administrators.

4. System Architecture

The Passport Automation System will be developed as a web-based application using a three-tier architecture, consisting of a presentation layer, application layer, and data layer. The system will utilize a relational database management system (RDBMS) to store and retrieve data.

5. Assumptions and Constraints

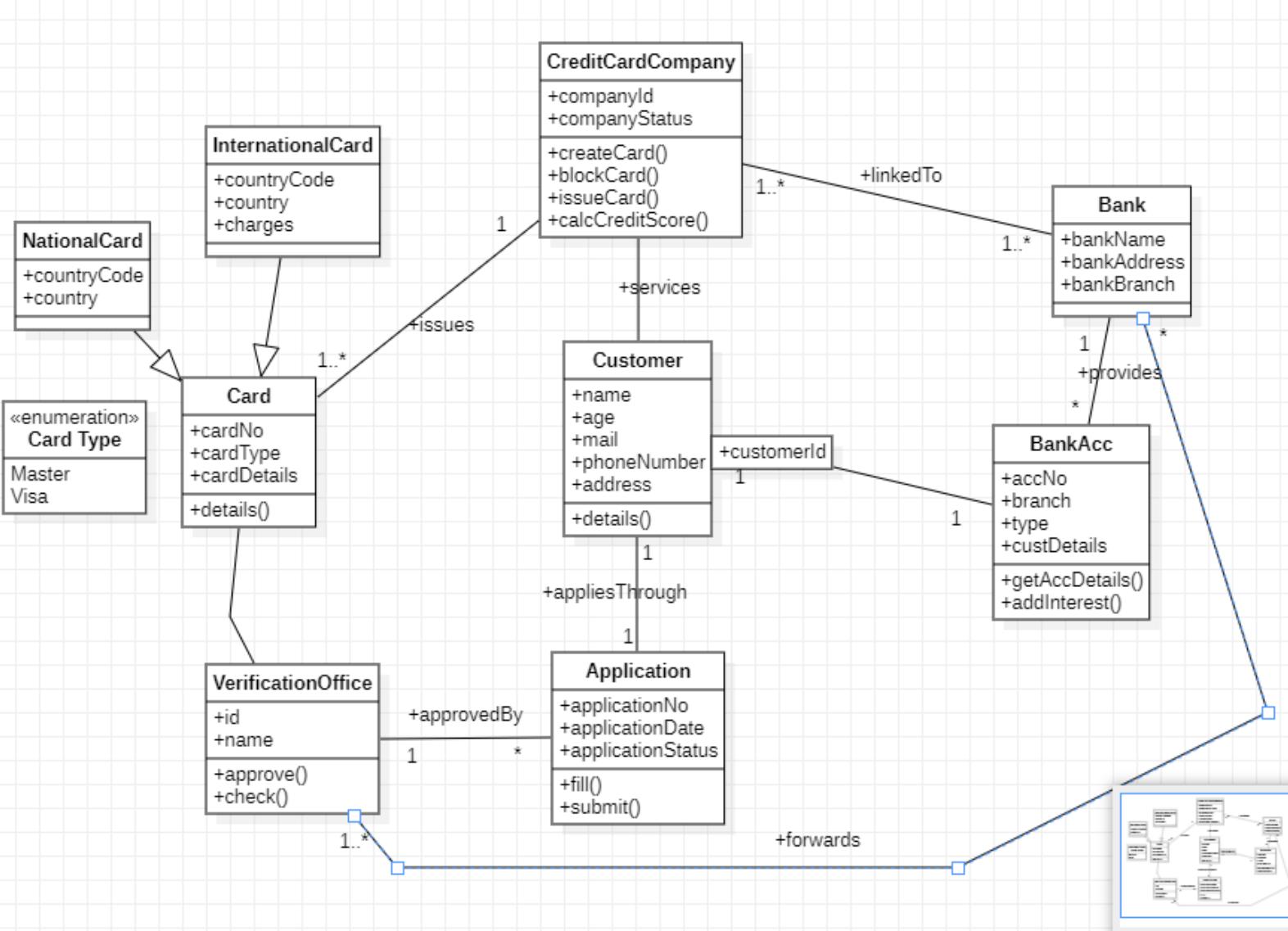
The Passport Automation System will be developed using a modern programming language and suitable frameworks.

The system will be compatible with major web browsers and responsive to different screen sizes.

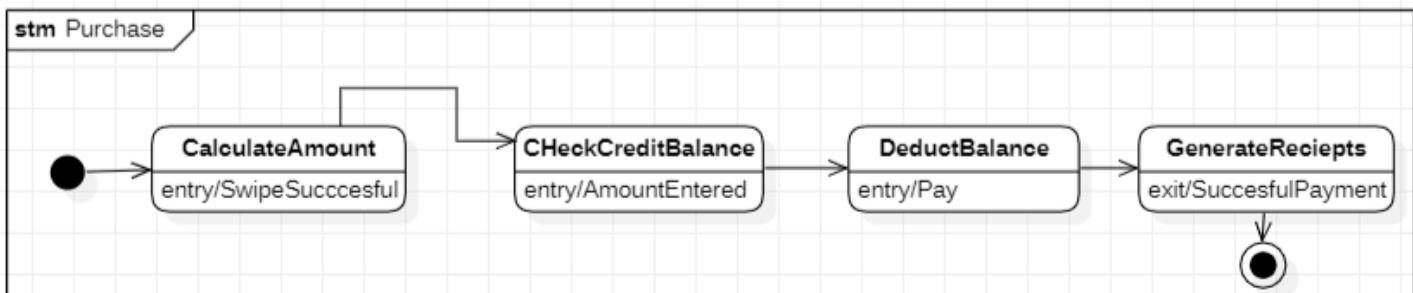
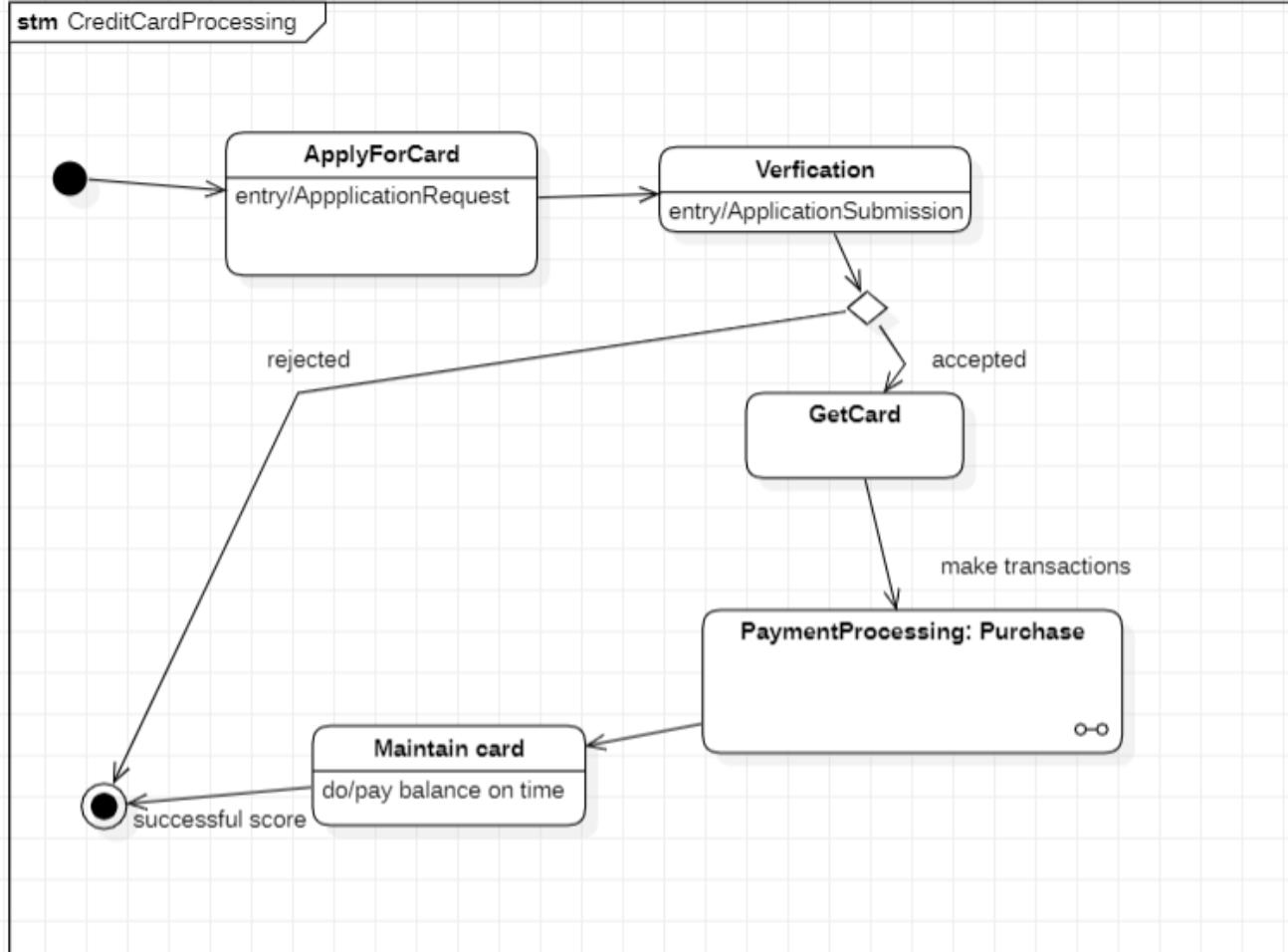
The hardware and network infrastructure required to host and deploy the system will be provided separately

Credit Card Processing

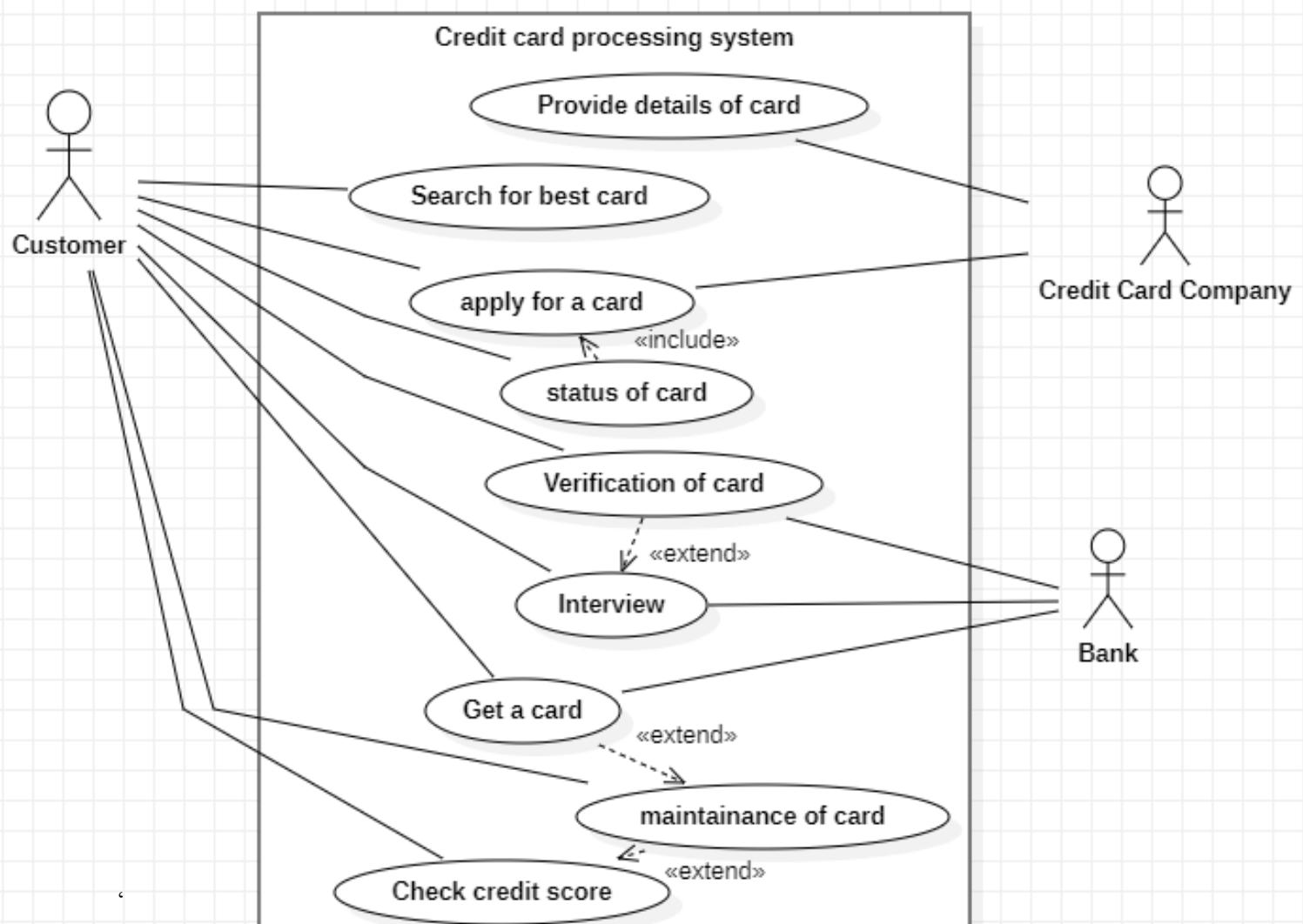
Class Diagram -



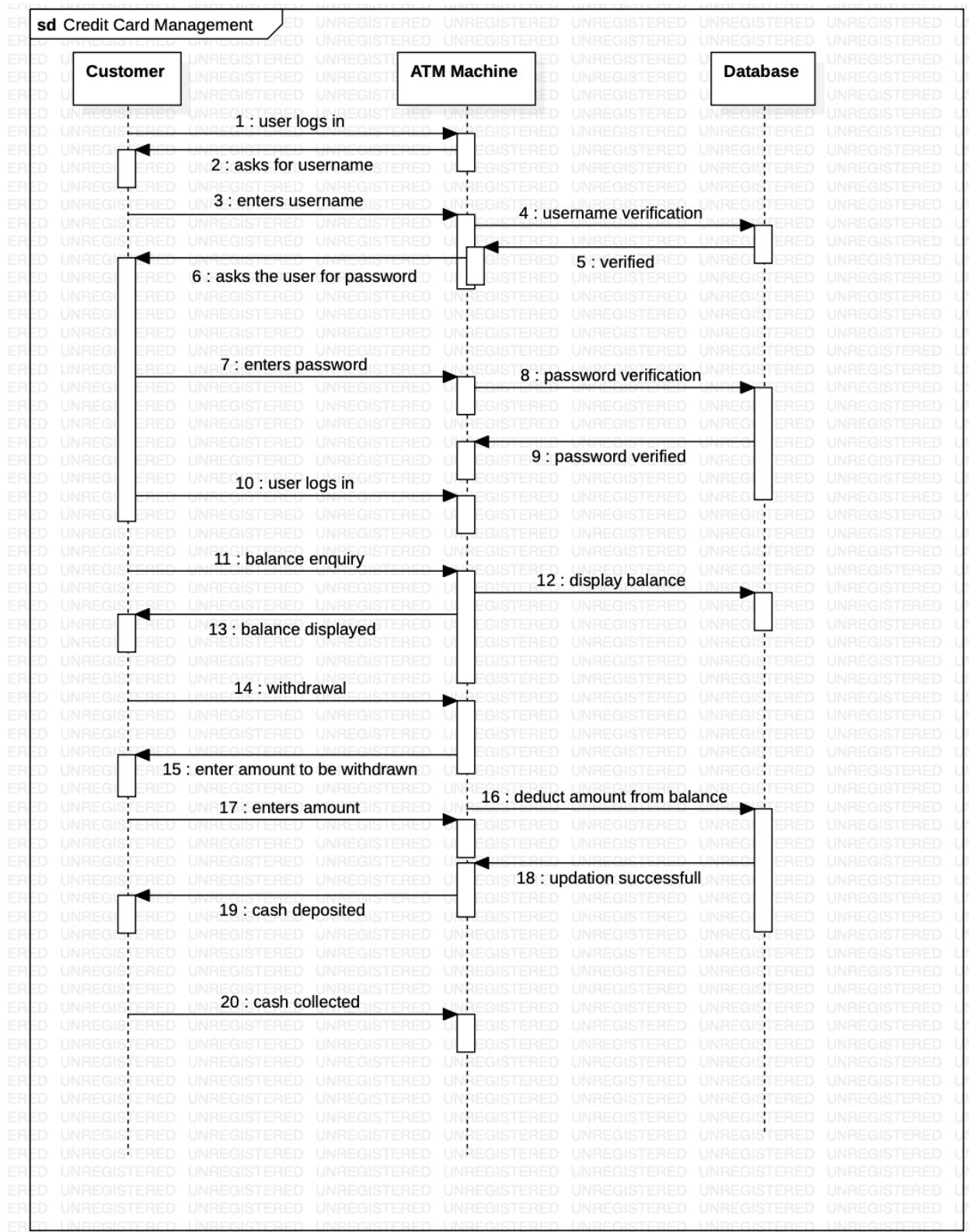
State Diagram -



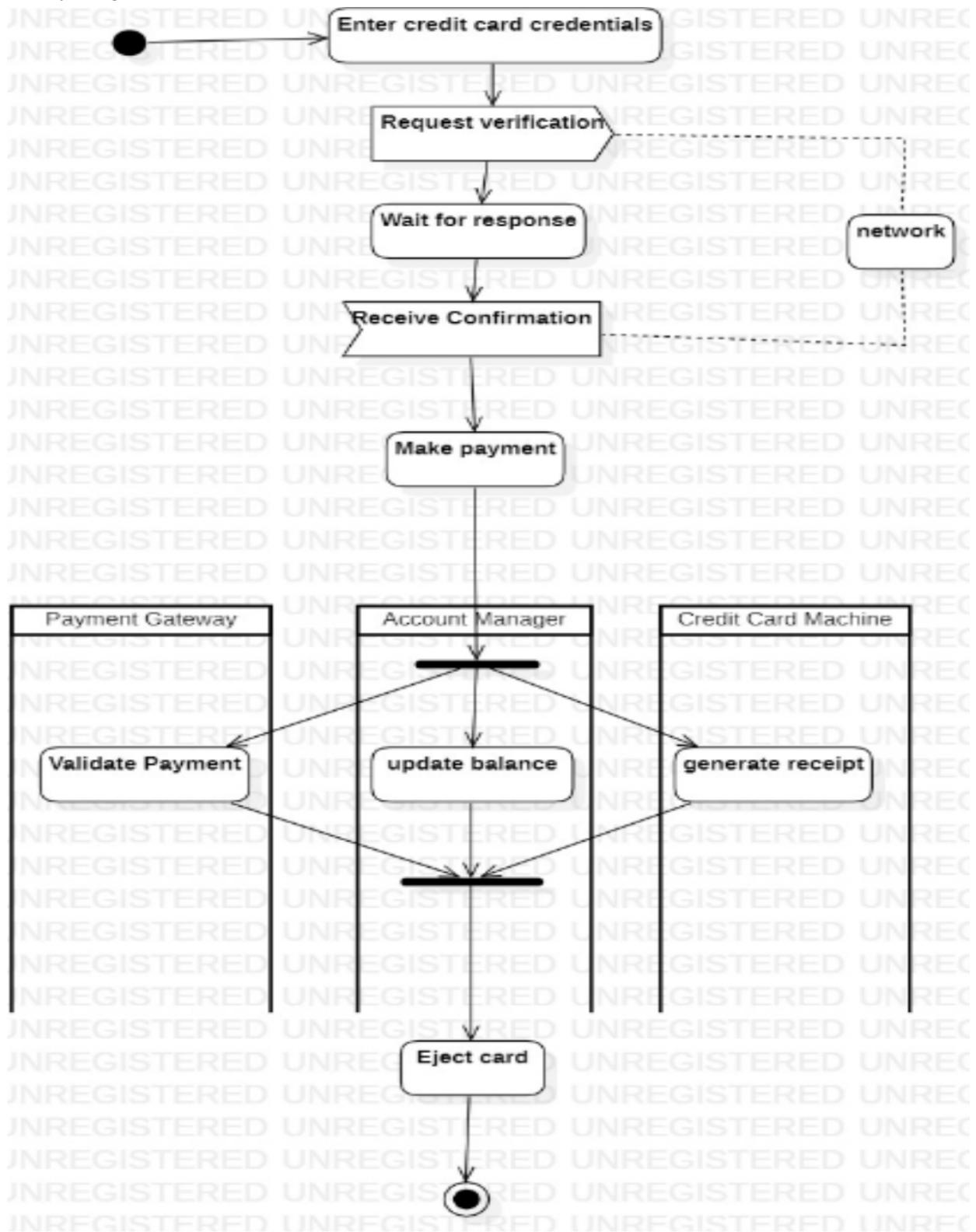
Use Case Diagram -



Sequence Diagram -

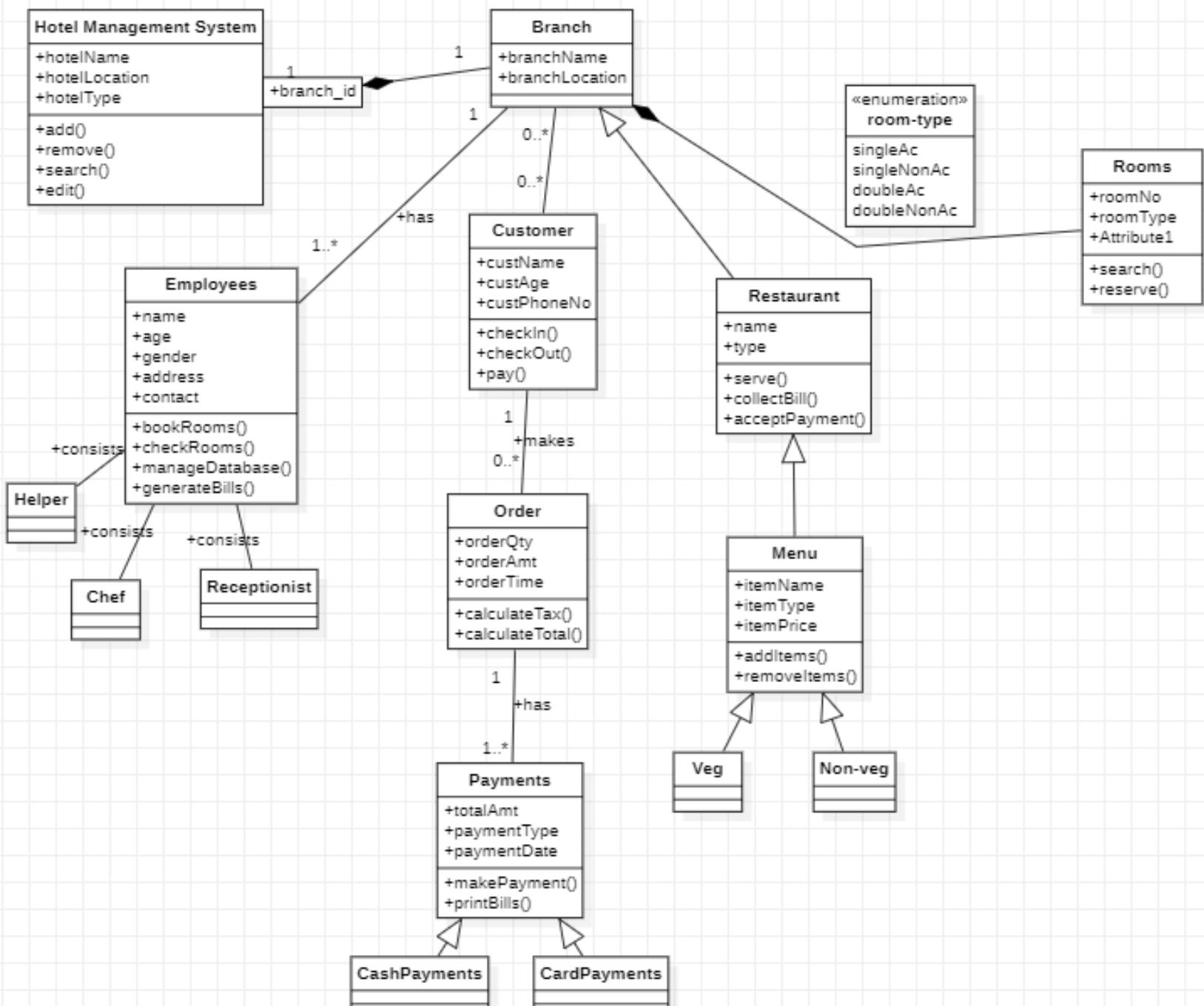


Activity Diagram -

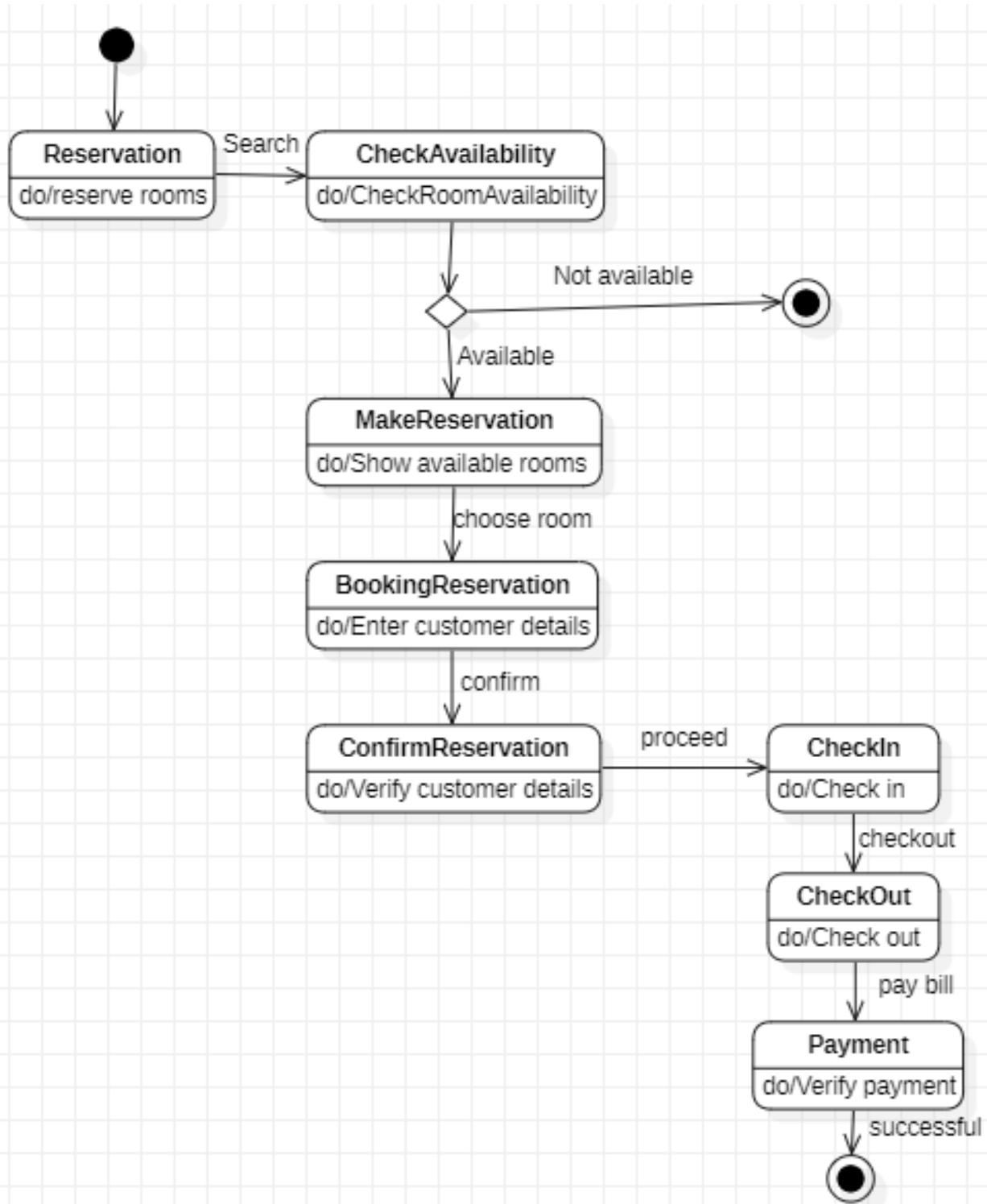


Hotel Management:-

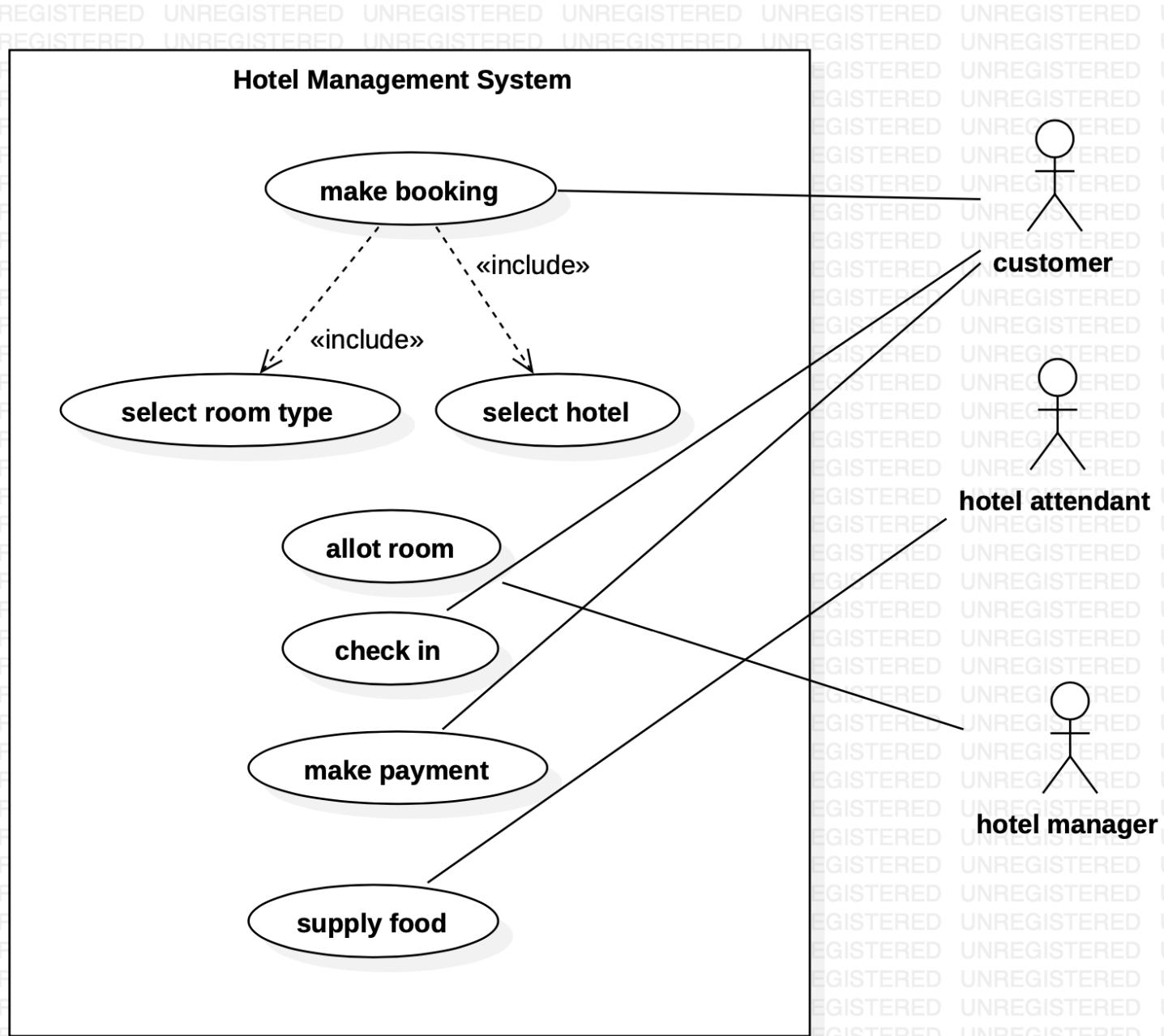
Class Model



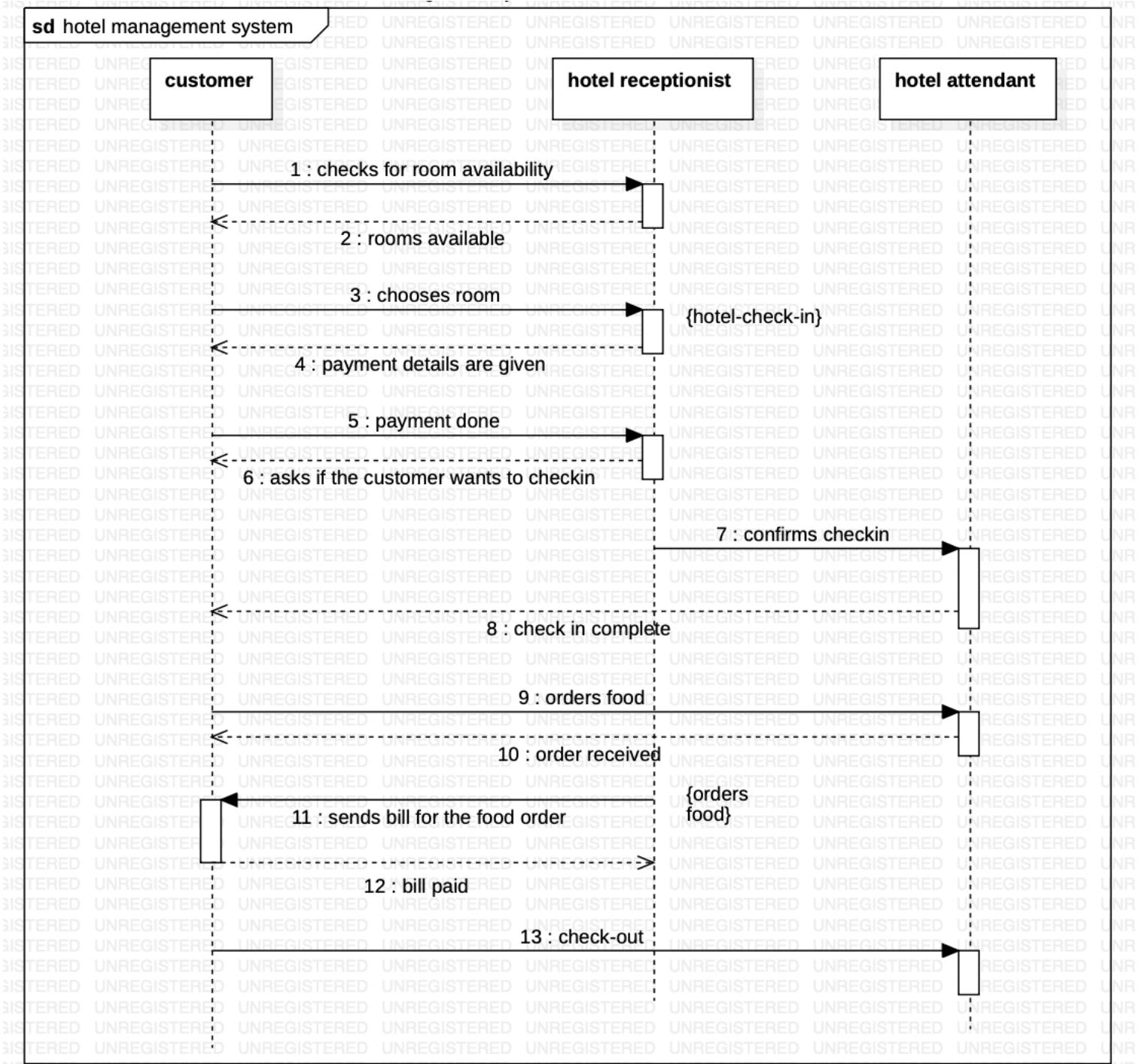
State Diagram -



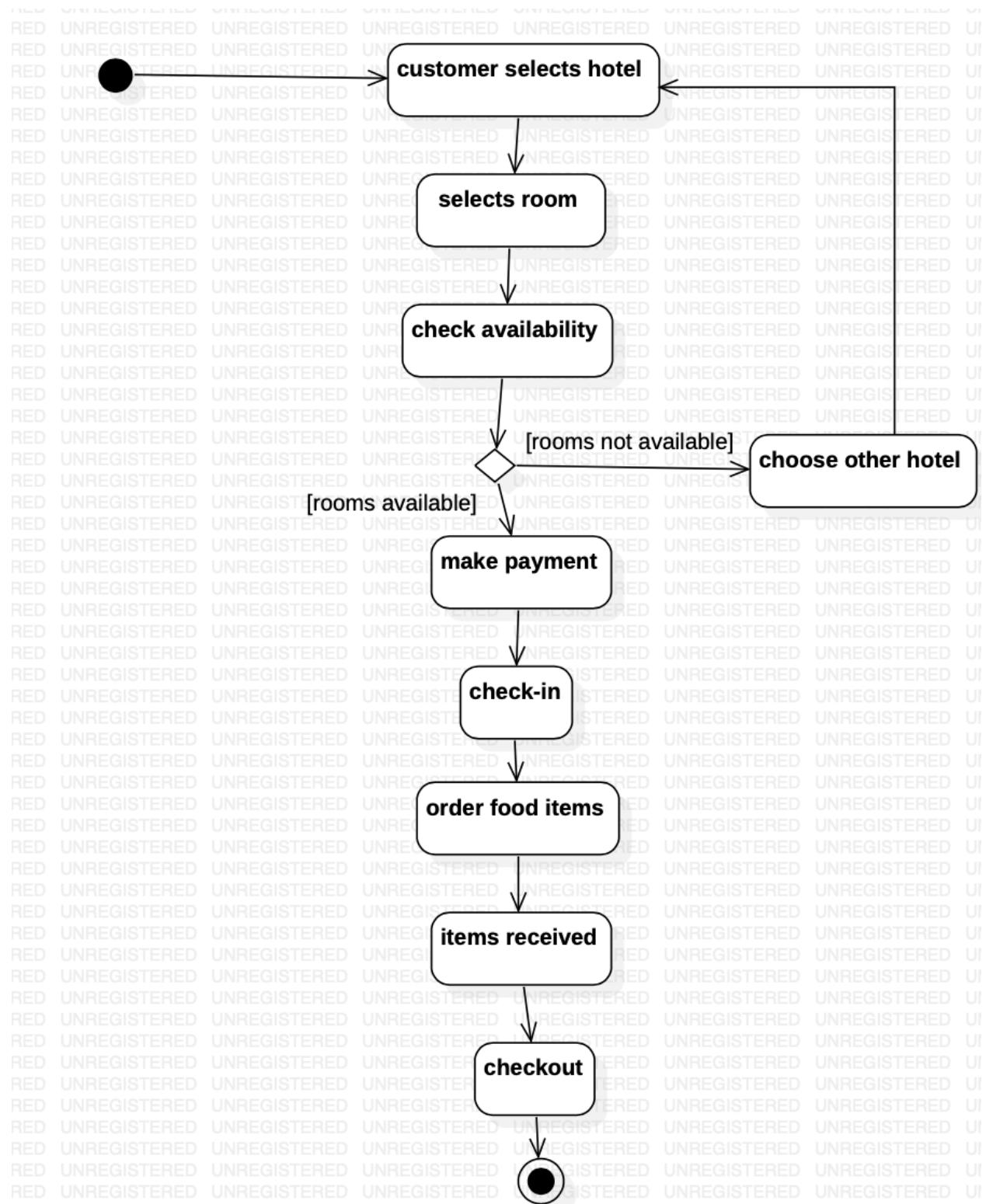
Use Case -



Sequence -

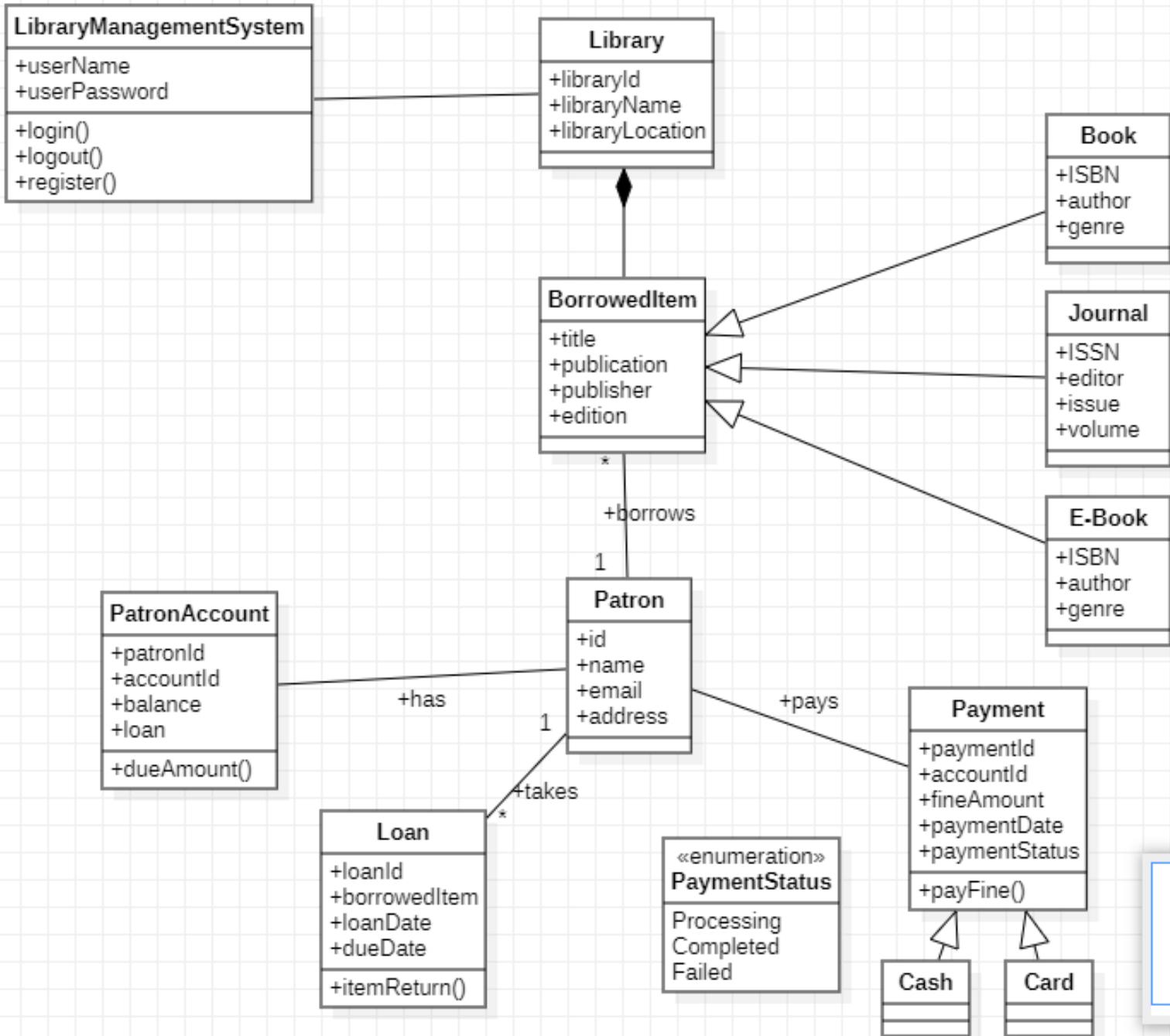


Activity

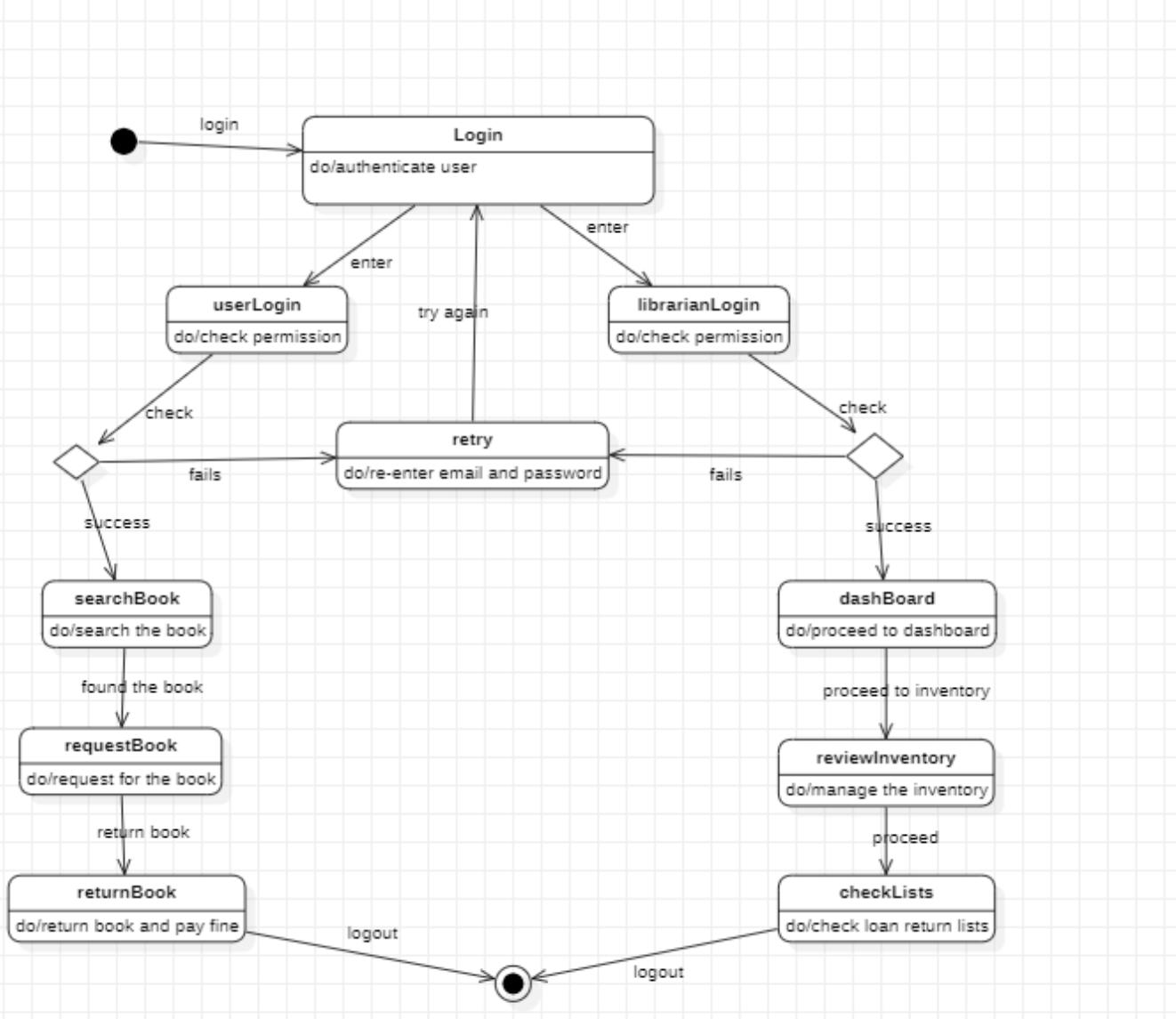


Library Management System: -

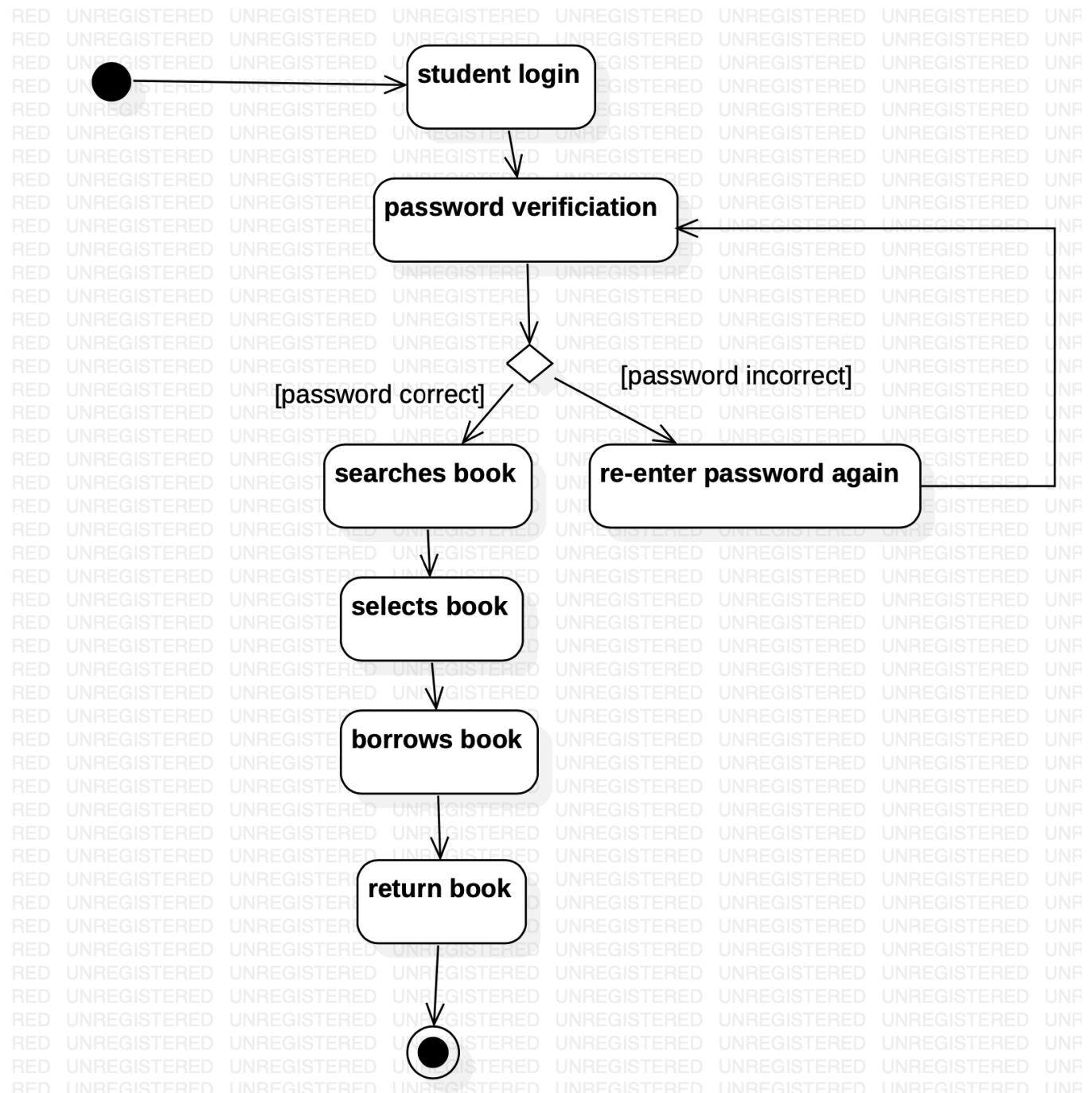
Class Model -



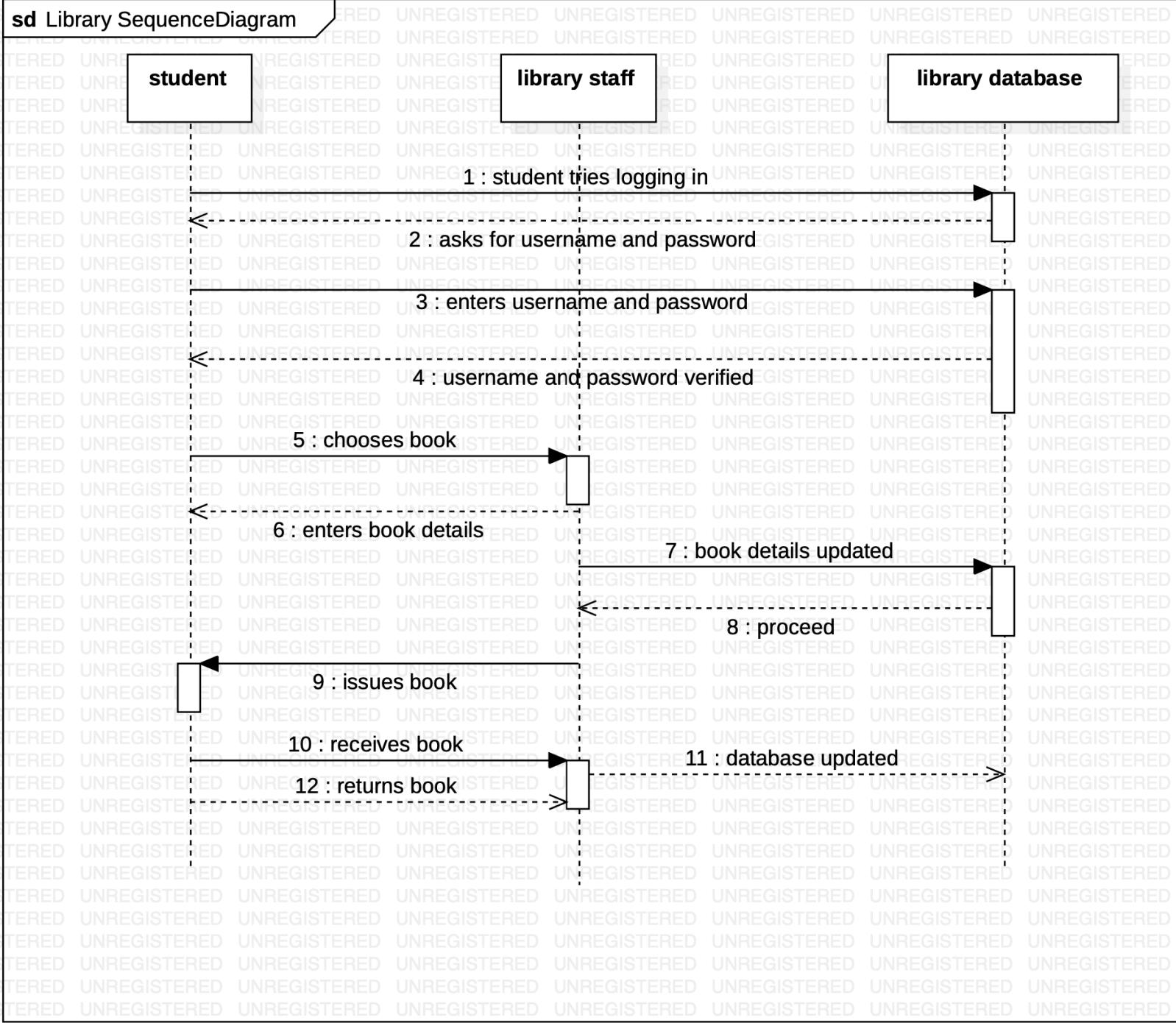
State Model -



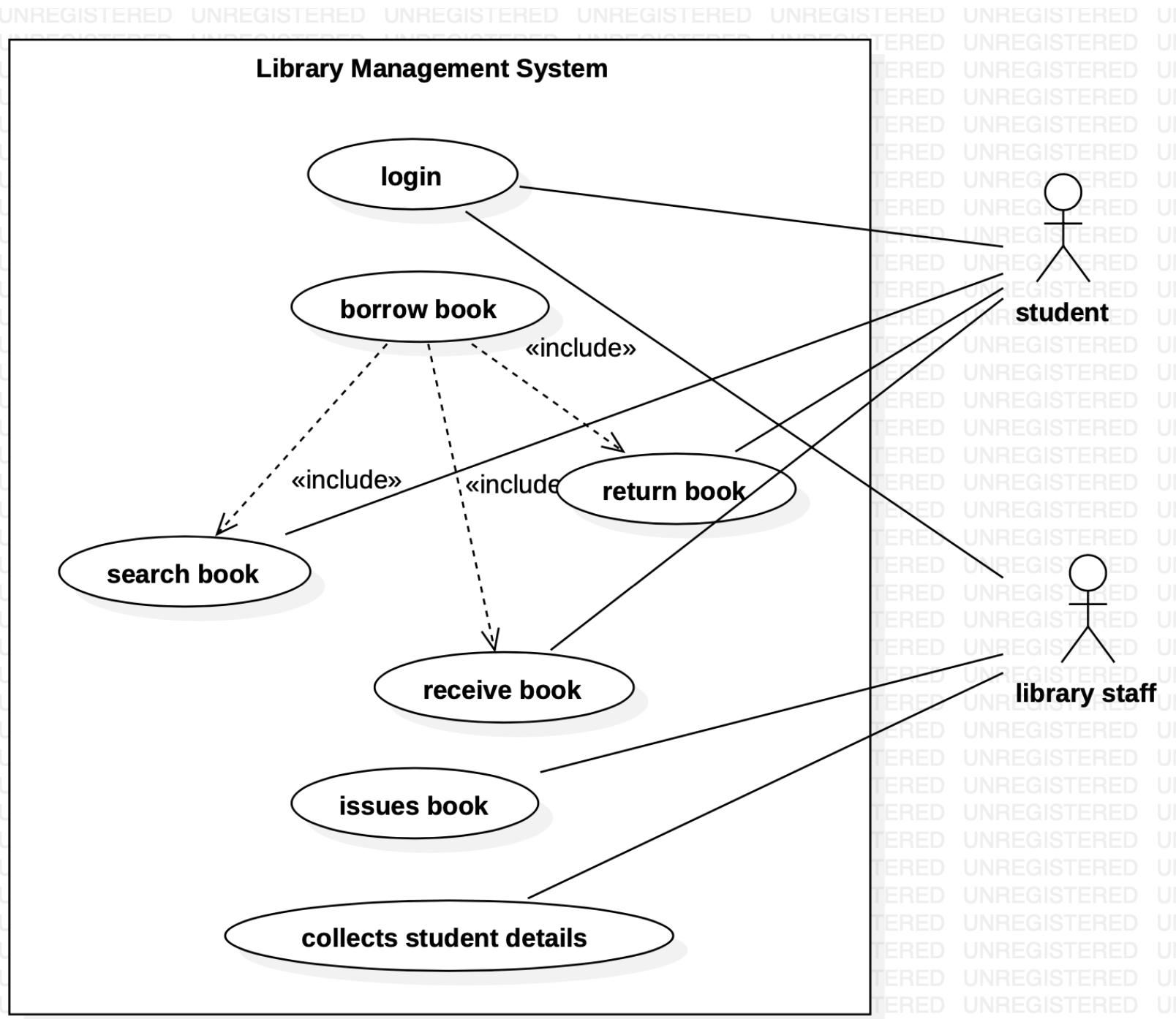
Activity Diagram -



Sequence Diagram -

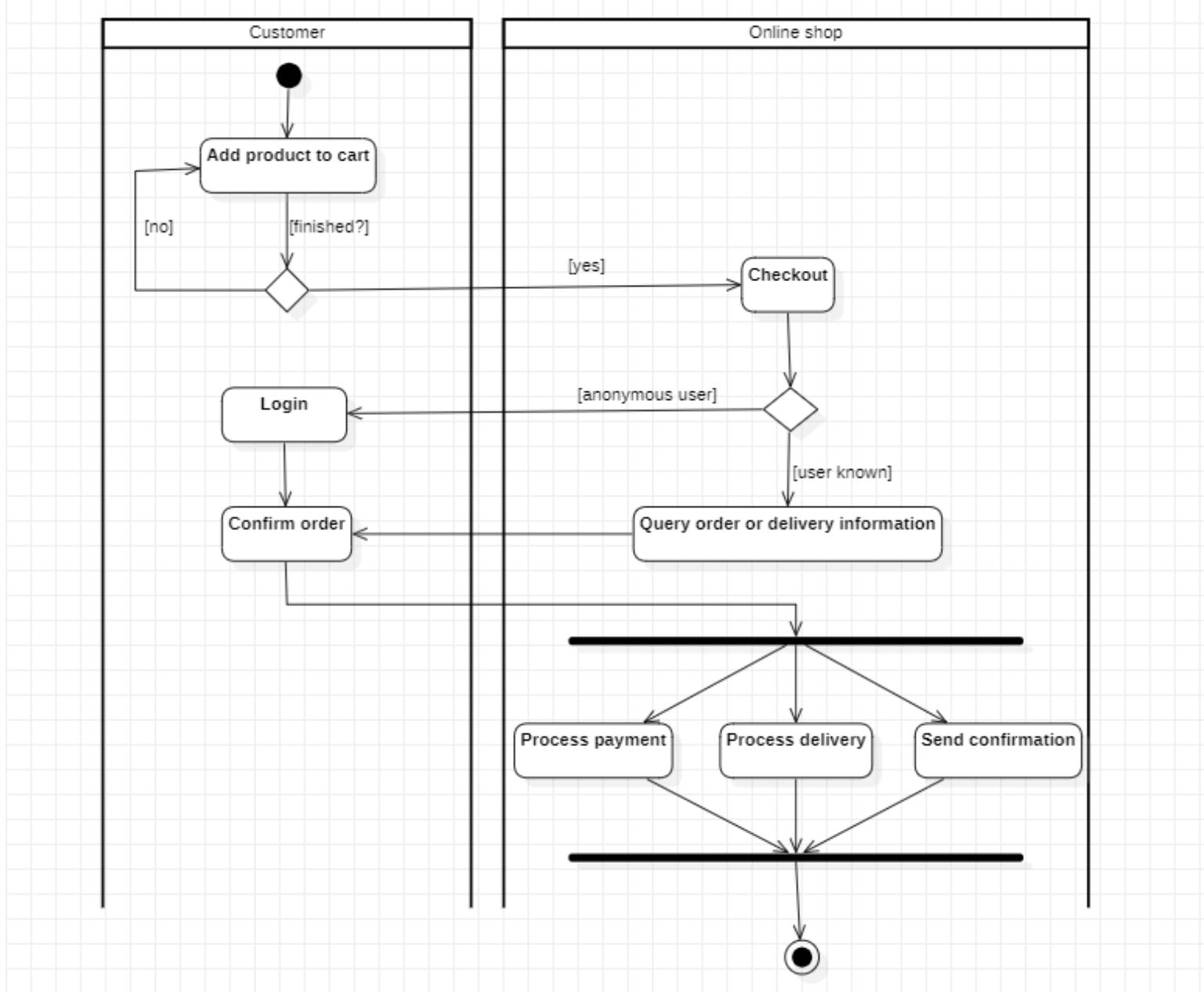


Use Case

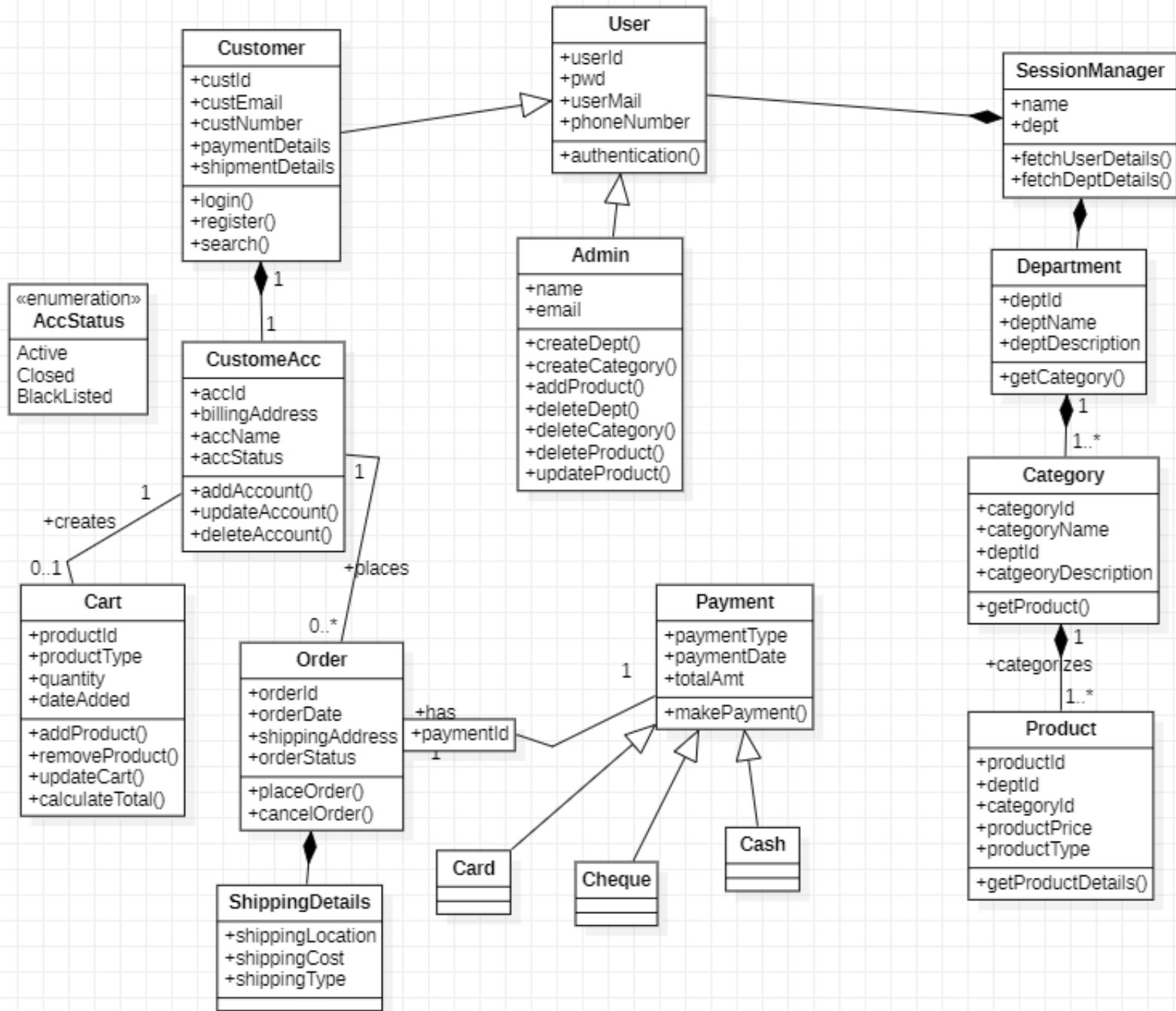


Online Shopping System -

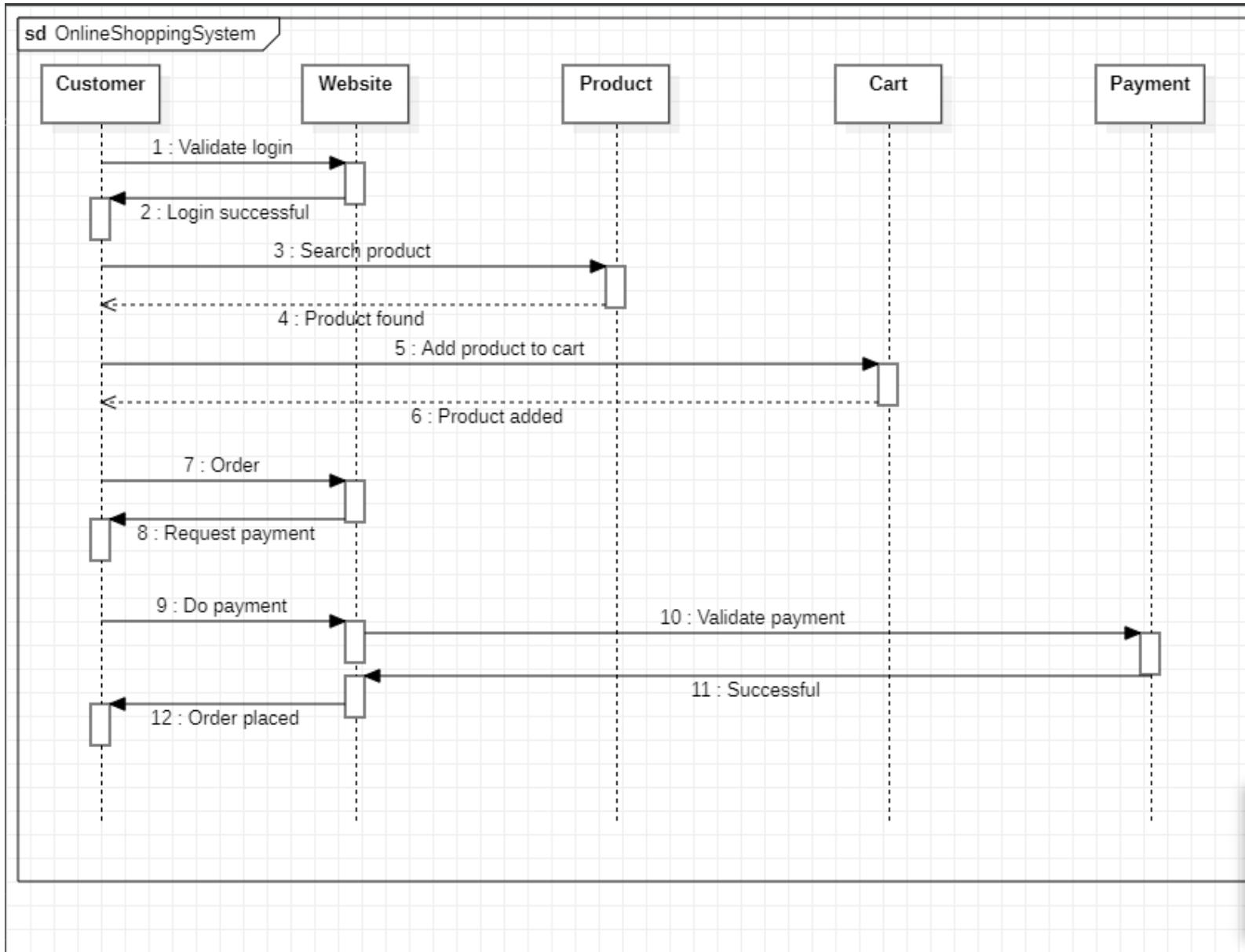
Activity Diagram -



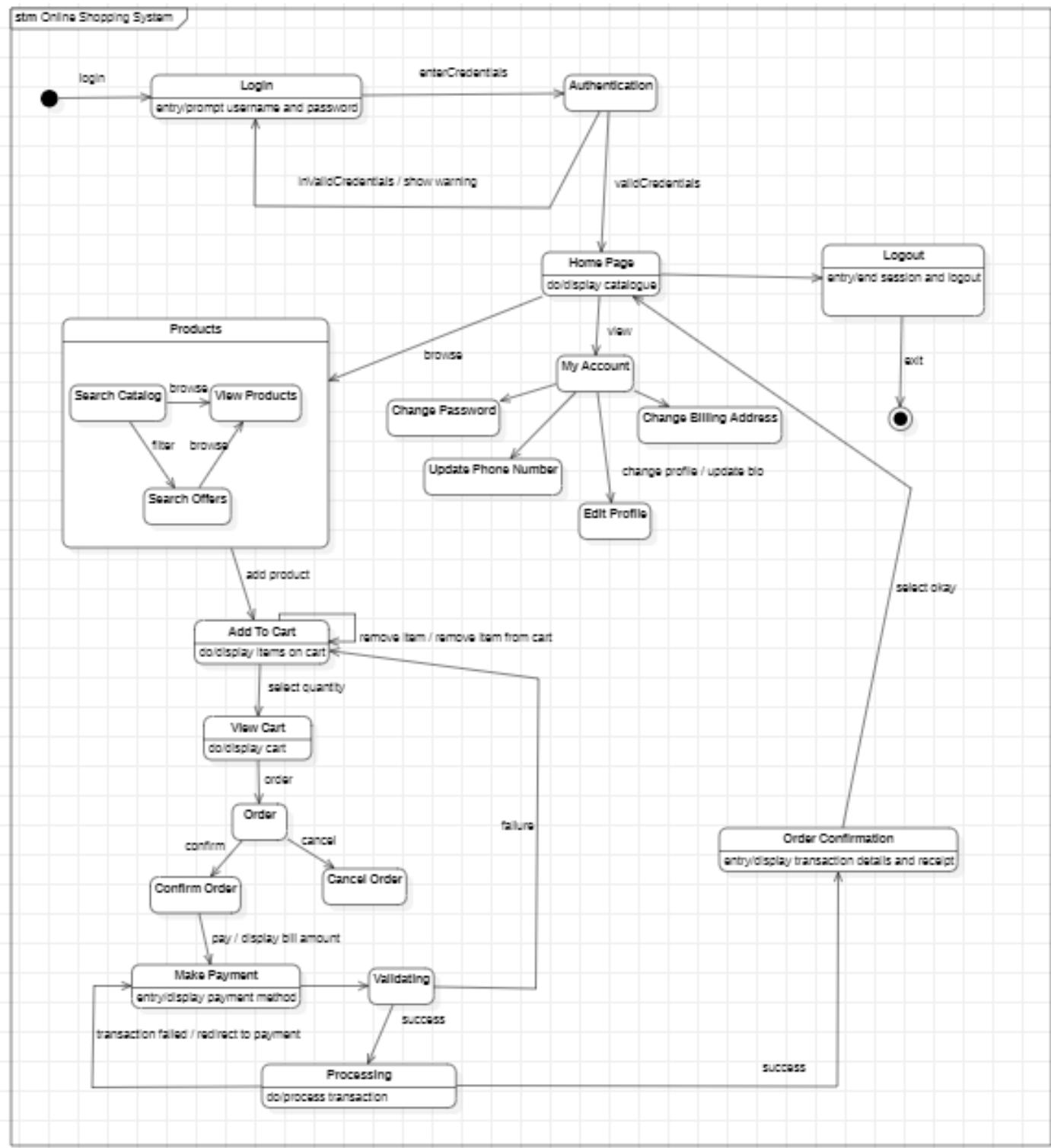
Class Model -



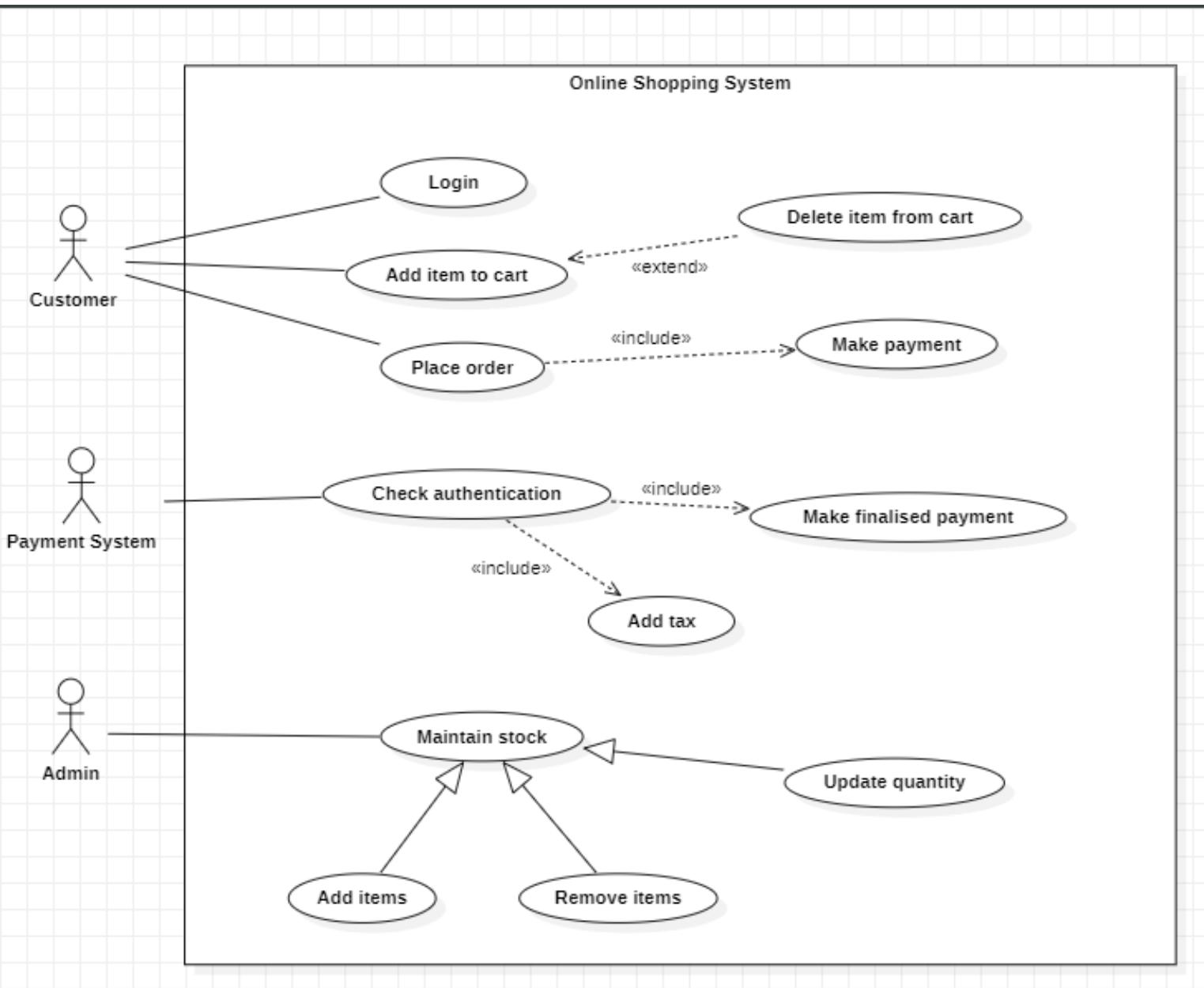
Sequence Diagram -



State Diagram -

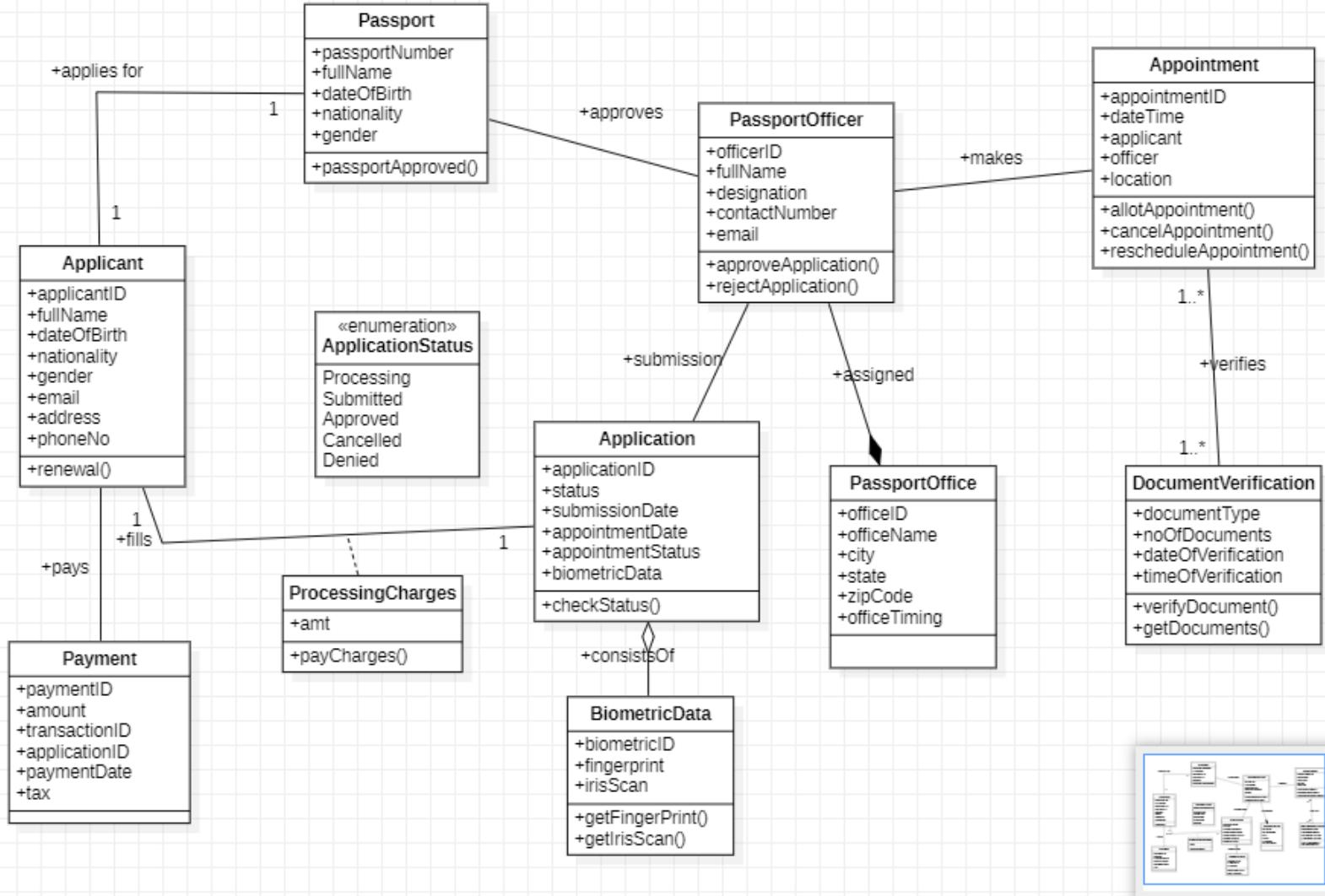


Use Case -

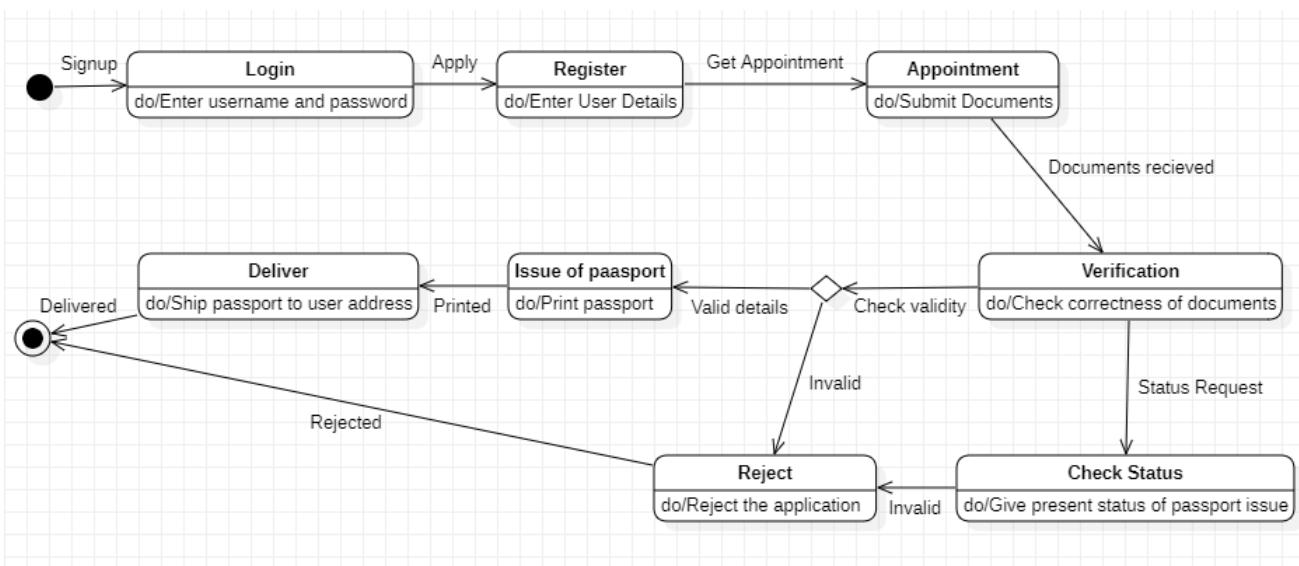


Passport Automation System

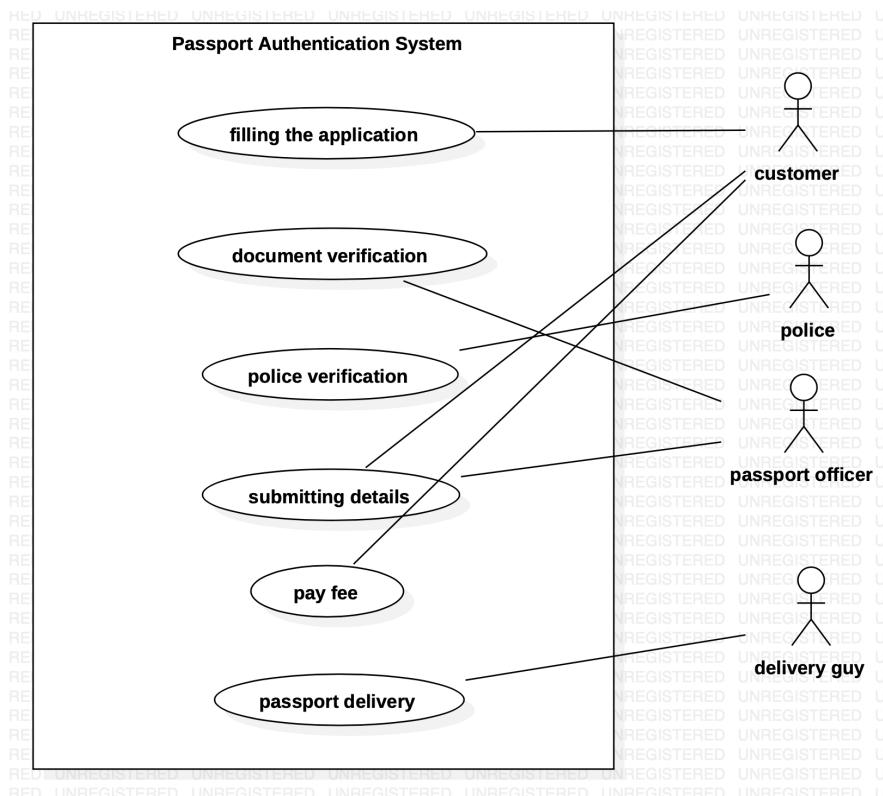
Class Model -



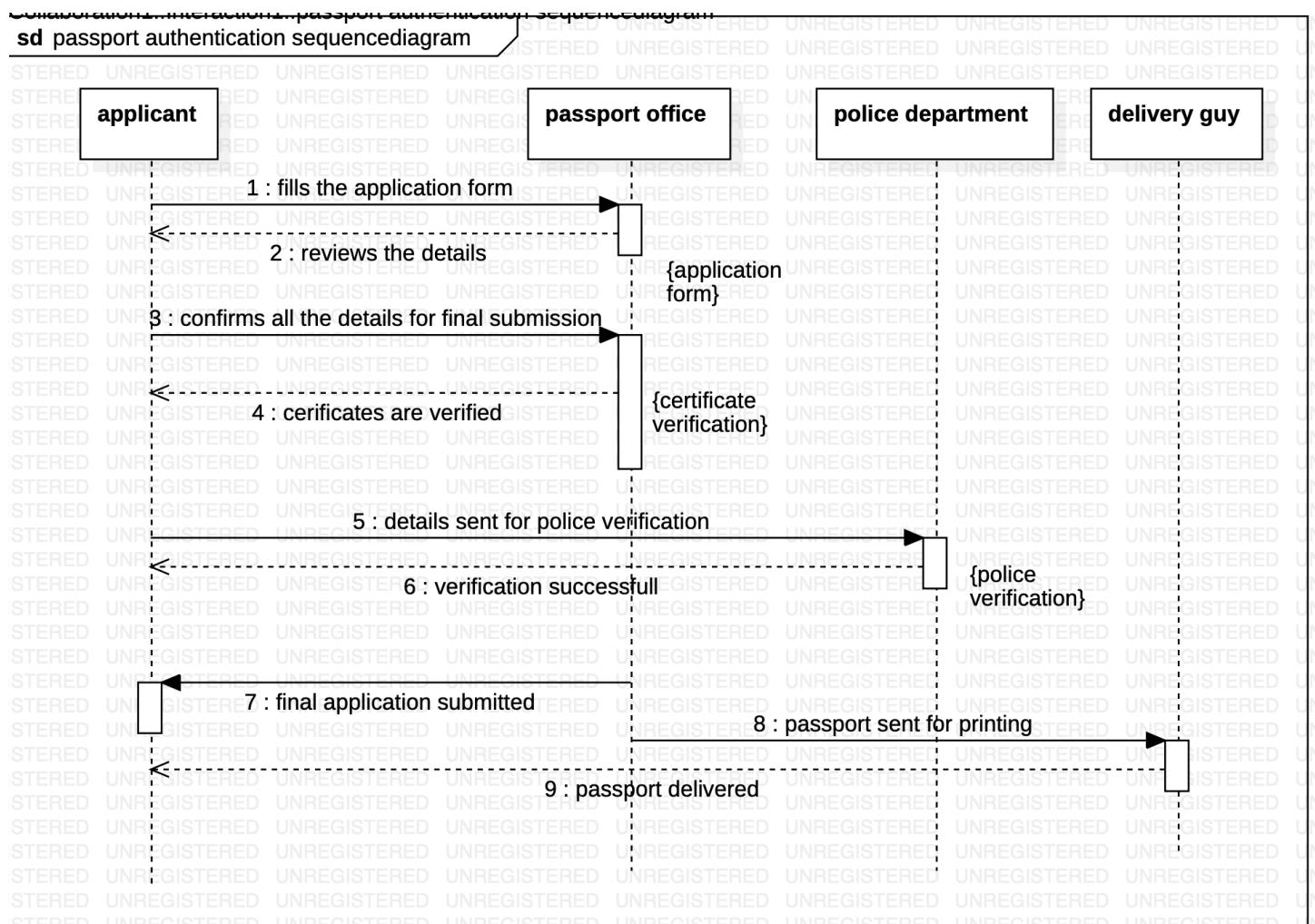
State Diagram -



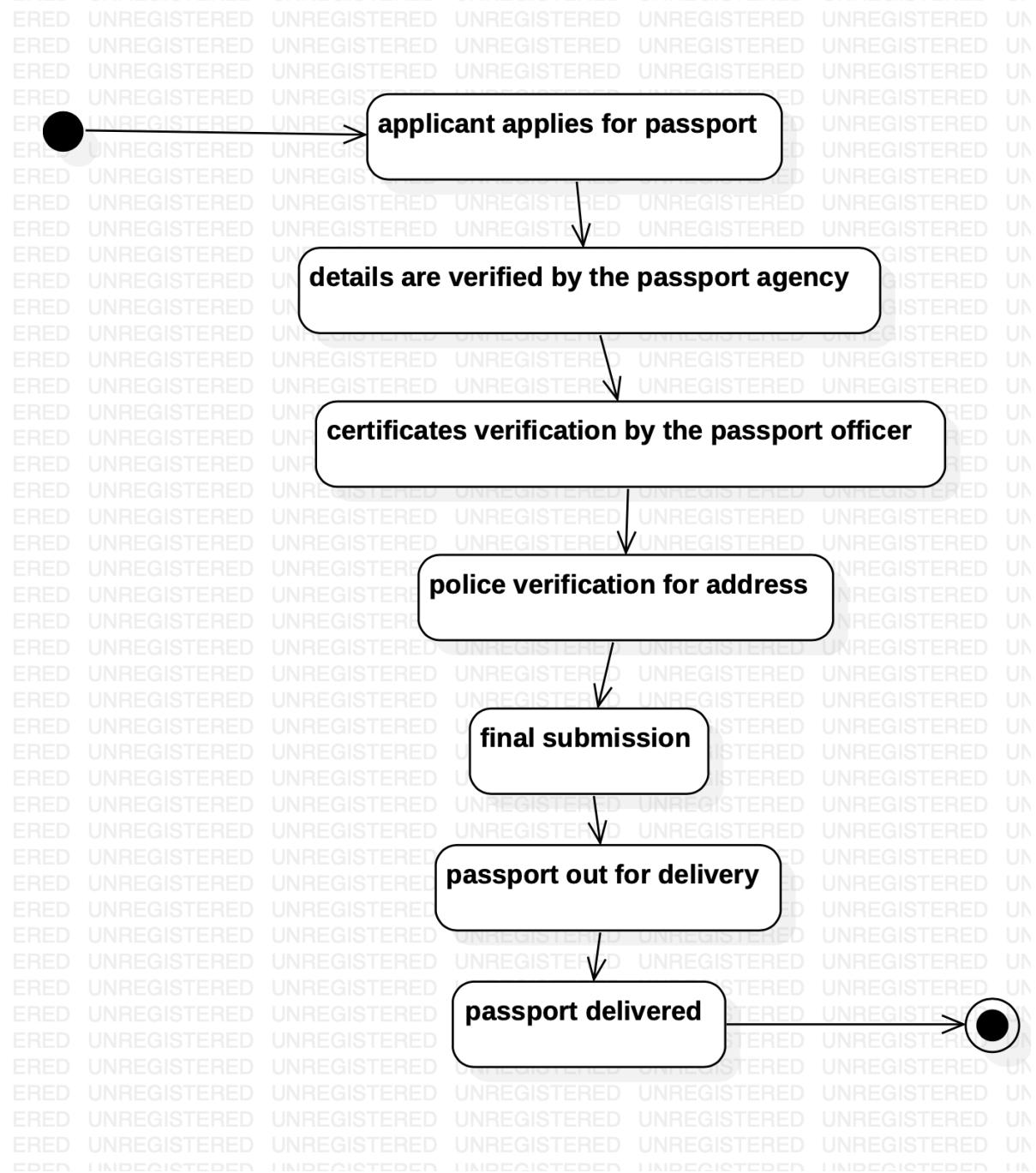
Use Case -



Sequence -

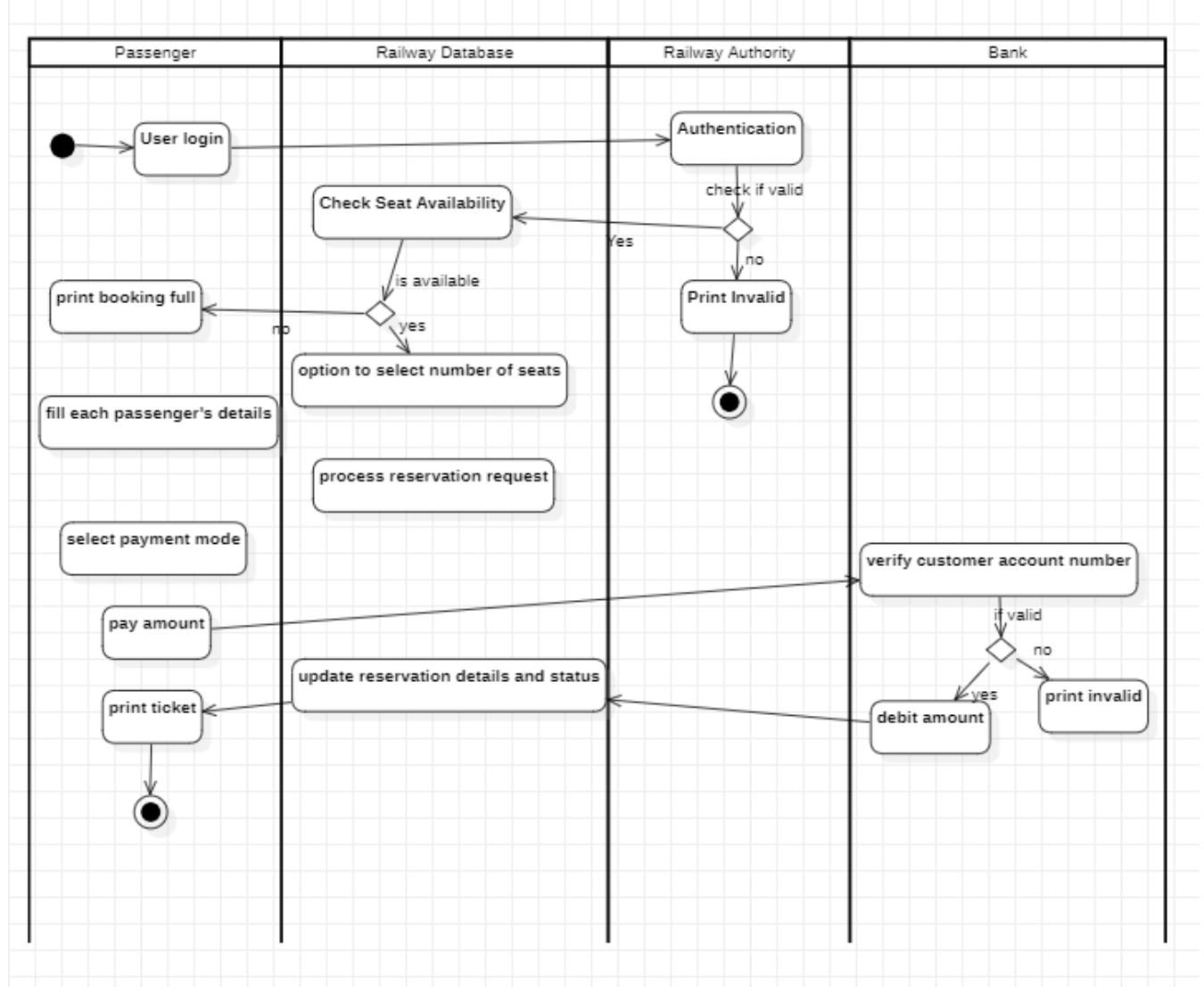


Activity -

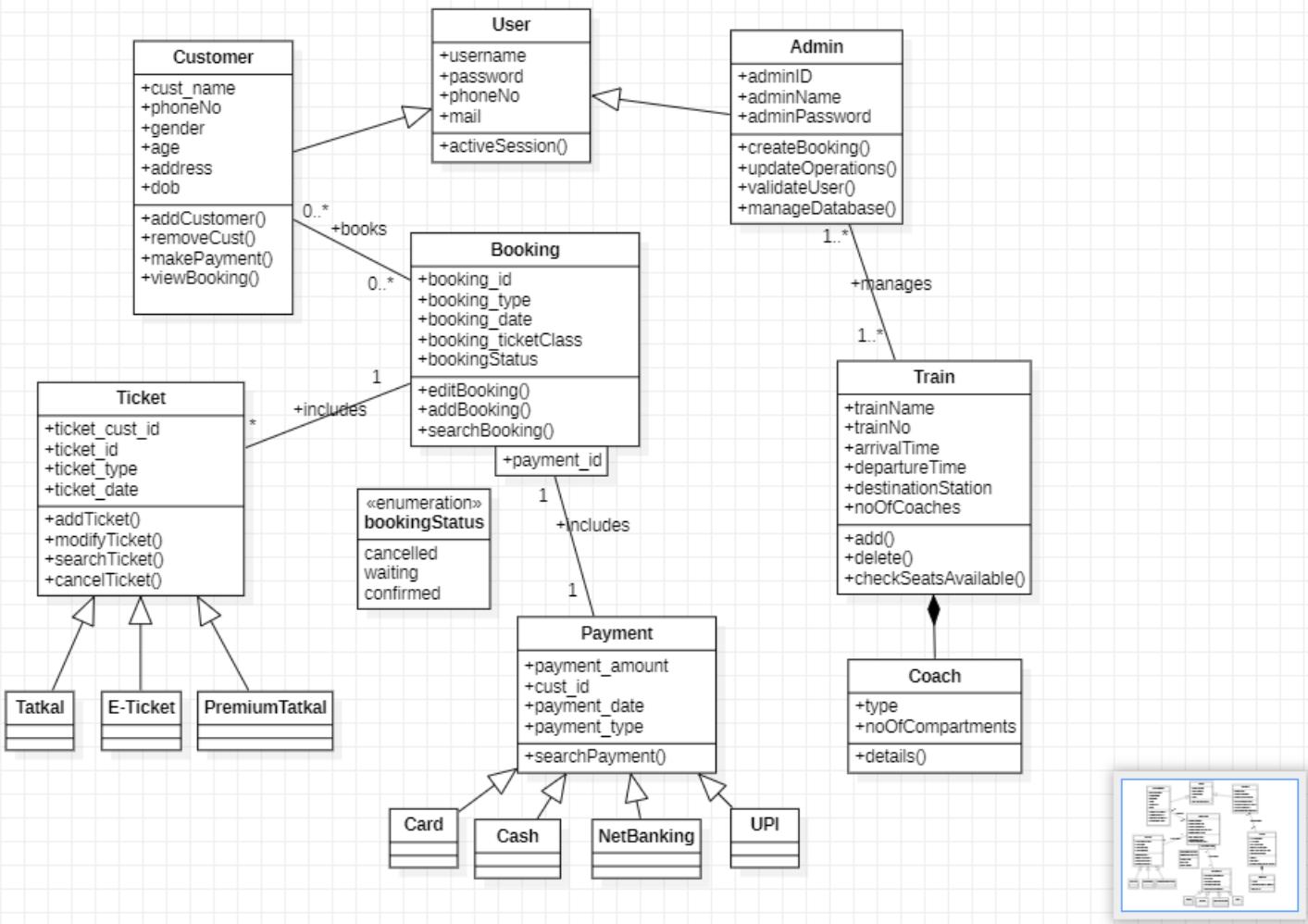


Railway Reservation System

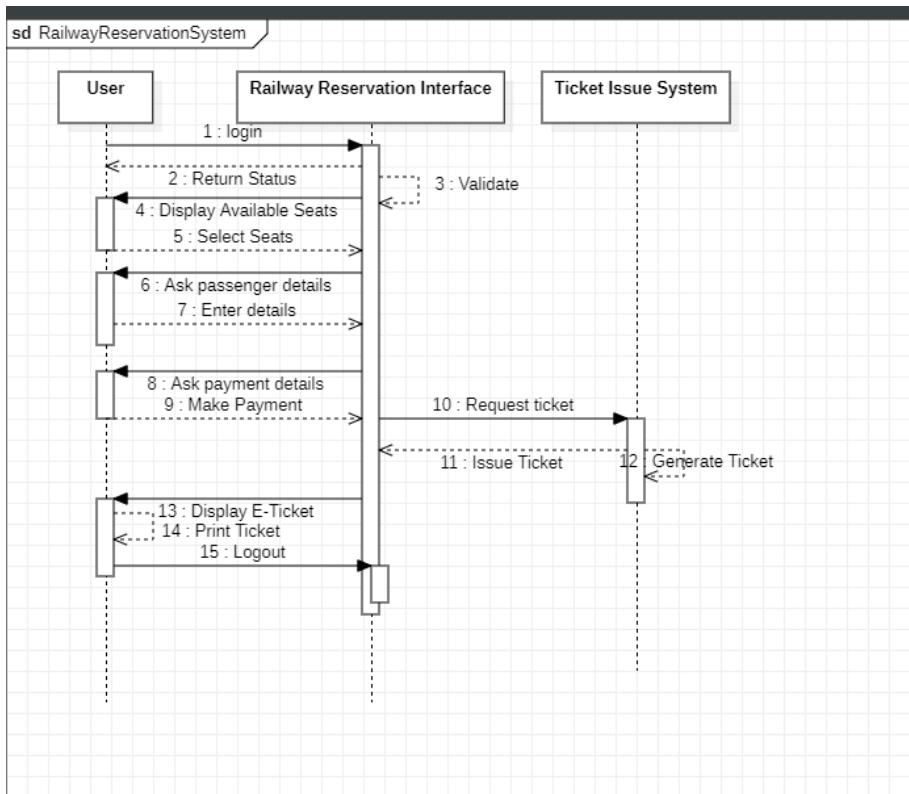
Activity Diagram -



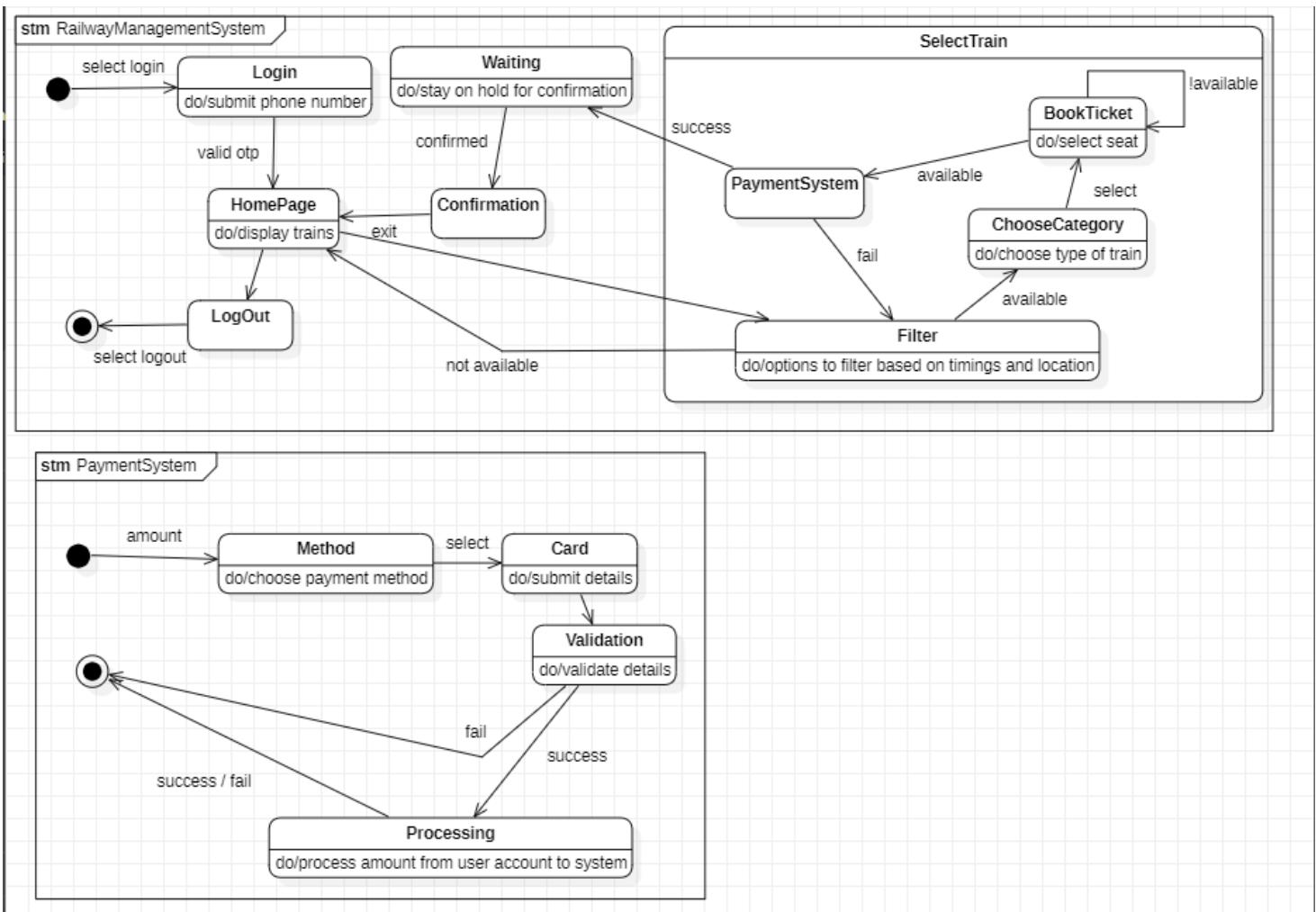
Class Model -



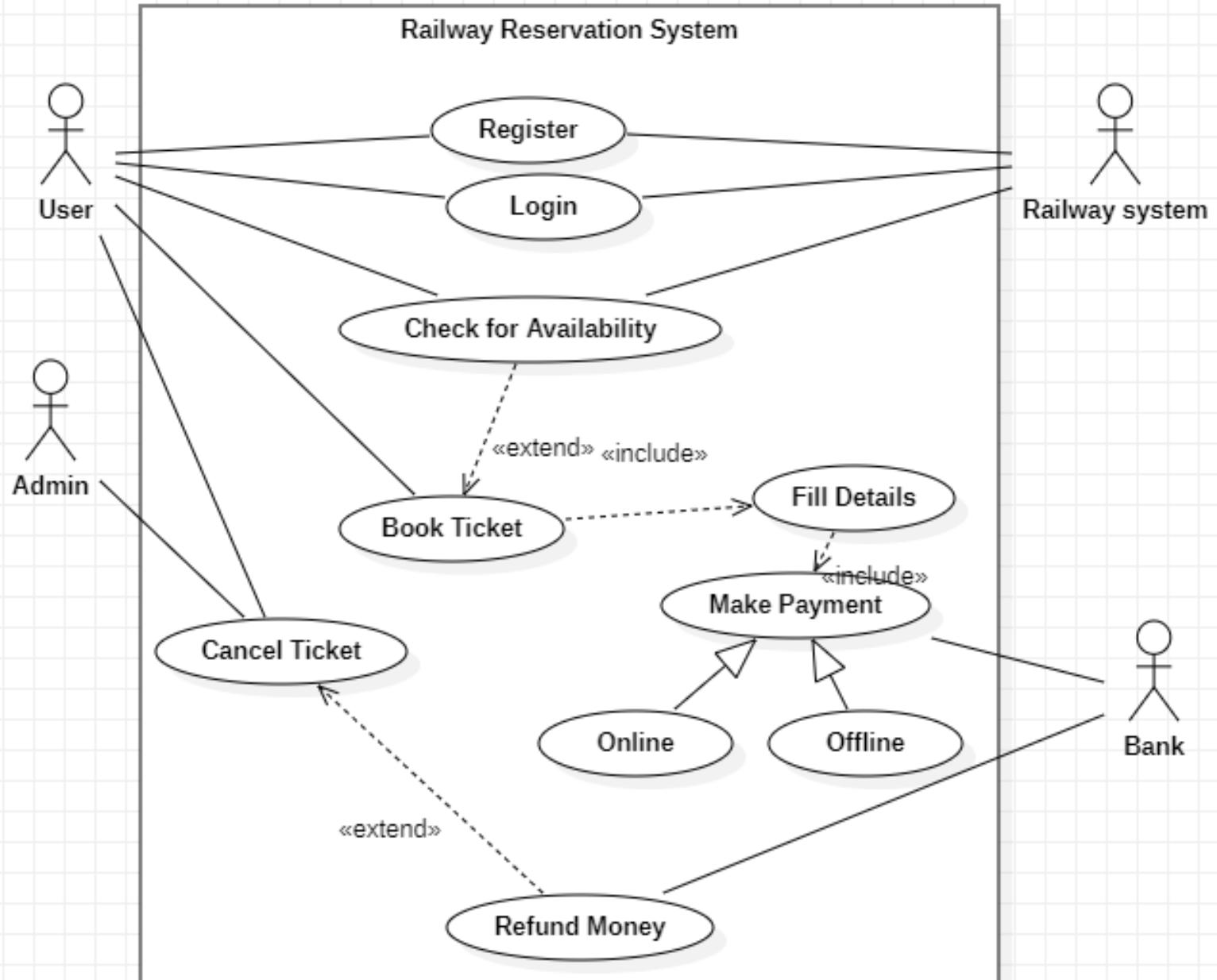
Sequence Diagram -



State Diagram -

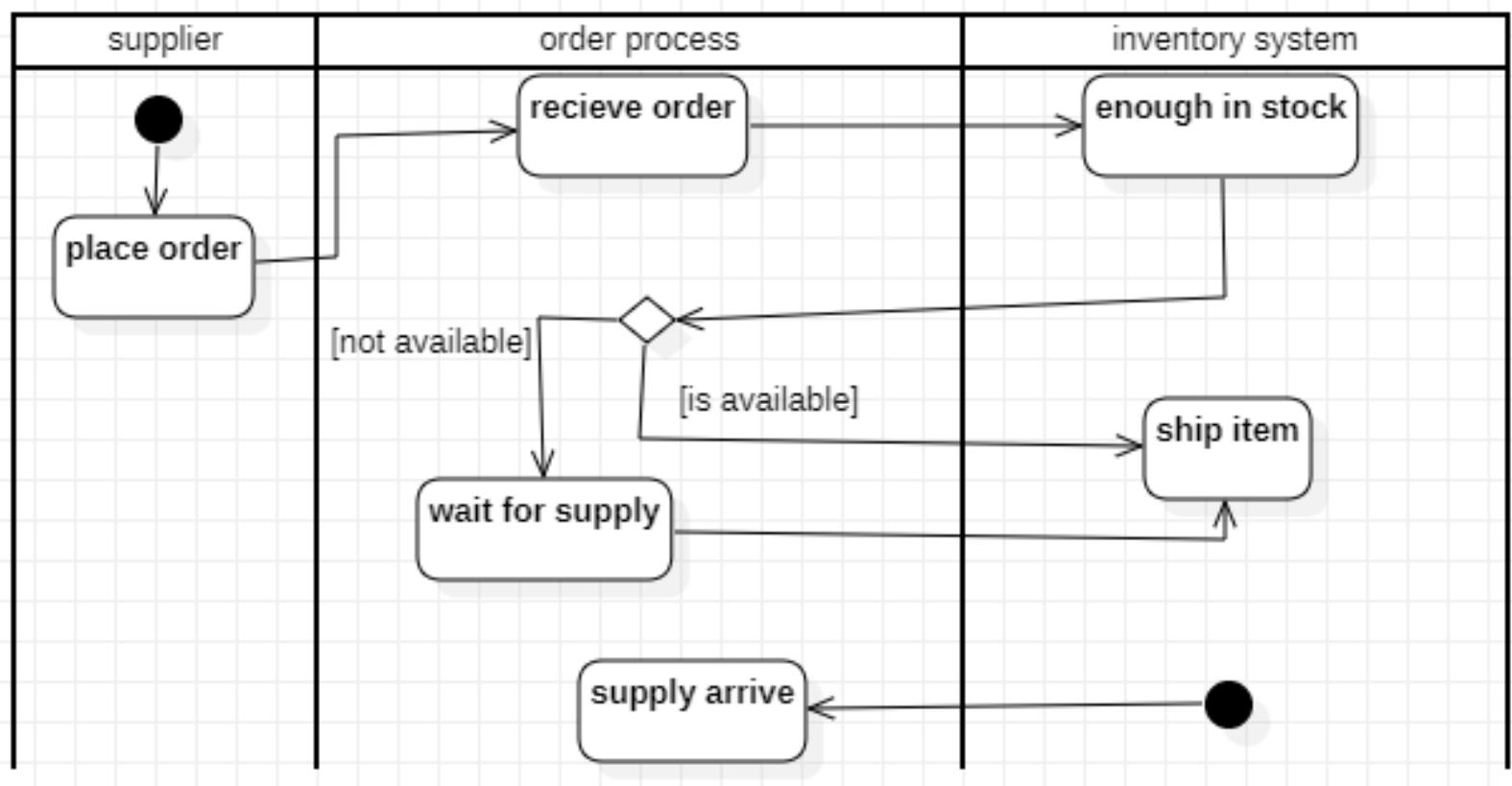


Use Case -

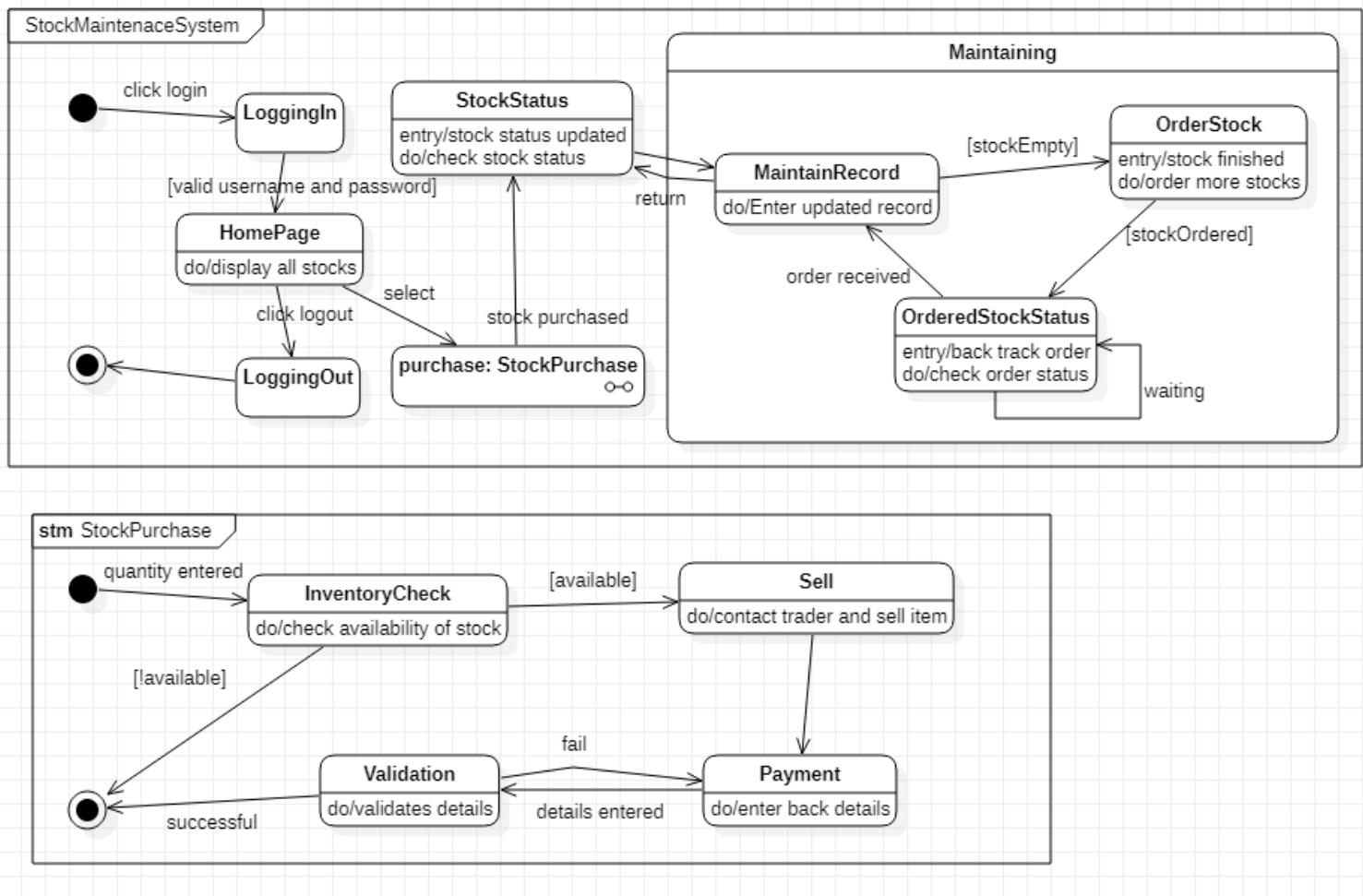


Stock Maintenance System

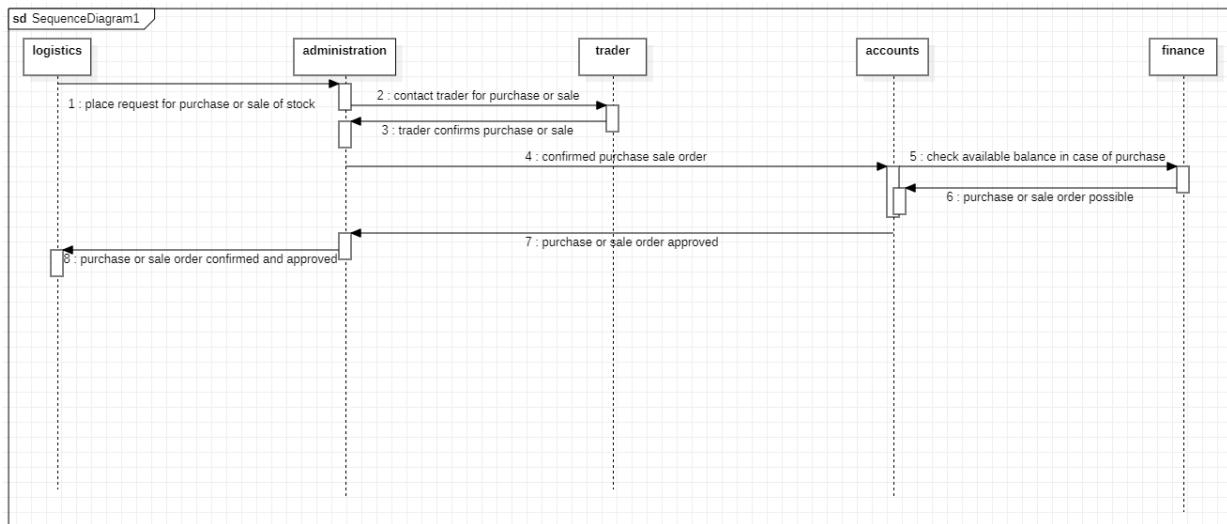
Activity Diagram -



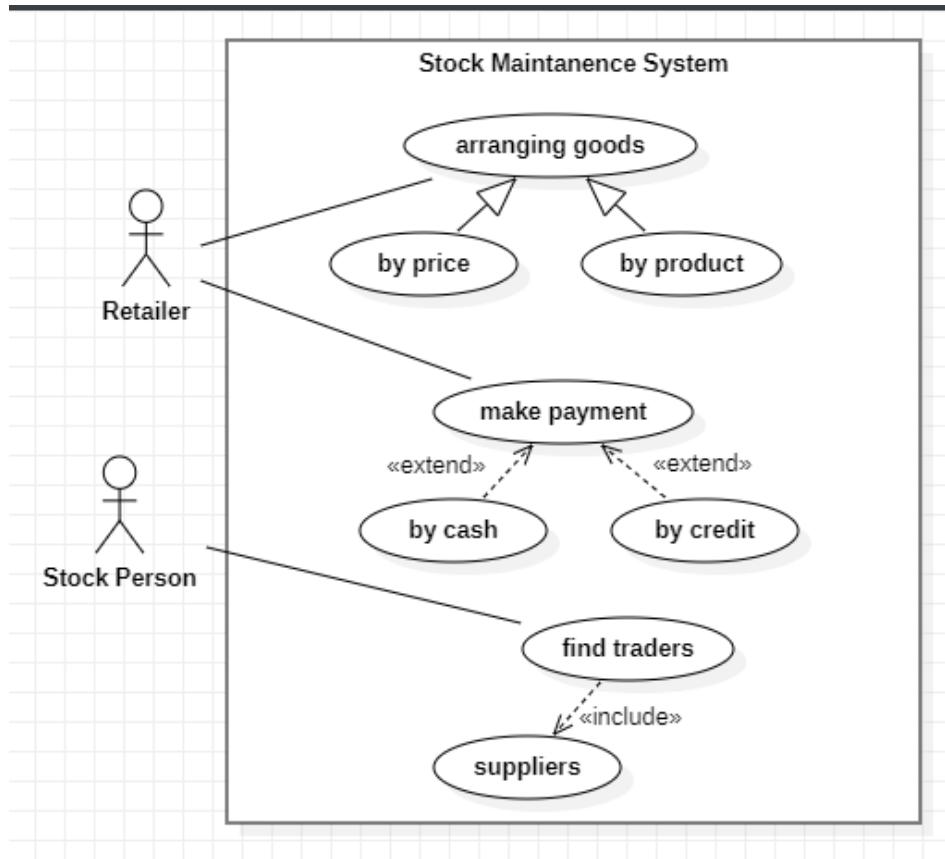
State Model



Sequence Diagram -



Use Case Diagram -



Class Diagram -

