B.M.S. COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB RECORD

Object Oriented Analysis and Design

Submitted in partial fulfillment for the 6th Semester Laboratory

Bachelor of Technology

in

Computer Science and Engineering

Submitted By

VINAY Y
1BM21CS415

Department of Computer Science and Engineering

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

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

CERTIFICATE

This is to certify that the Object-Oriented Analysis and Design(20CS6PCOMD) laboratory has been carried out by **VINAY Y (1BM21CS415)** duringthe 6th Semester April-July 2023.

Signature

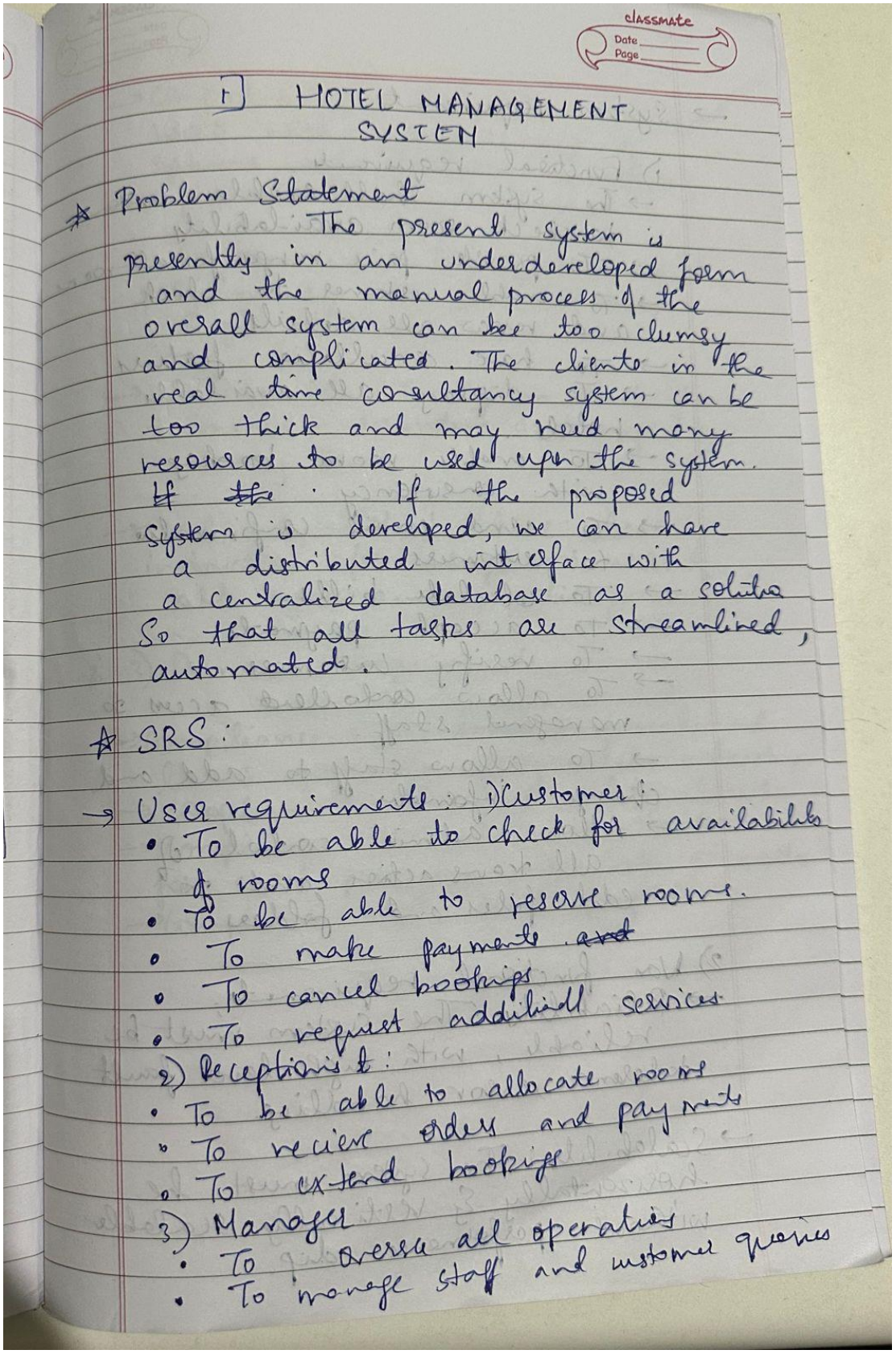
Dr. Pallavi G B

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

TABLE OF CONTENTS

1. Hotel Management System

1.1 SRS



1] HOTEL MANAGEMENT SYSTEM

★ Problem Statement

The present system is presently in an underdeveloped form and the manual process of the overall system can bee too clumsy and complicated. The clients in the real time consultancy system can be too thick and may need many resources to be used upon the system. If the proposed system is developed, we can have a distributed interface with a centralized database as a solution So that all tasks are streamlined, automated.

★ SRS:

→ User requirements: 1) Customer:
- To be able to check for availability of rooms
- To be able to reserve rooms.
- To make payments
- To cancel bookings
- To request additional services.

2) Receptionist:
- To be able to allocate rooms
- To recieve orders and payments
- To extend bookings

3) Manager
- To oversee all operations
- To manage staff and customer queries

→ System requirements

1) Functional requirements
→ The system shall enable customers to check room availability
→ To provide price ranges for all rooms
→ To enable customers to check and review all details.
→ To have cancellation features
→ To display all available hotels.
→ To enable room booking with consistency.
→ To send email confirmation to customers.
→ To calculate bills.
→ To accept payments.
→ To verify customers
→ To allow controlled access to managerial staff.
→ To allow staff to add and change information.
→ Allow admin control of all transactions and to edit files and folders.

2) Non functional requirements.
→ Reliability: The system must be reliable, with high level fault tolerance, error handling.

→ Scalability: The system must be horizontally & vertically scalable without performance dip.

→ Compatibility : The system must work with a wide range of systems with hardware & software configurations.

→ Performance : The system must be performant with fast responses, high throughput.

→ Availability : The system must have minimal downtime and rapid recovery time.

→ Usability : Must have clear instructions for usage for users of all ages and expertise.

3) Domain requirements

→ Compliance with local and national laws governing hospitality industry.
→ Compliance with food safety laws.
→ Compliance with industry standars for card processing and data security.
→ Compliance with ACID properties.

1.2 Class Diagram

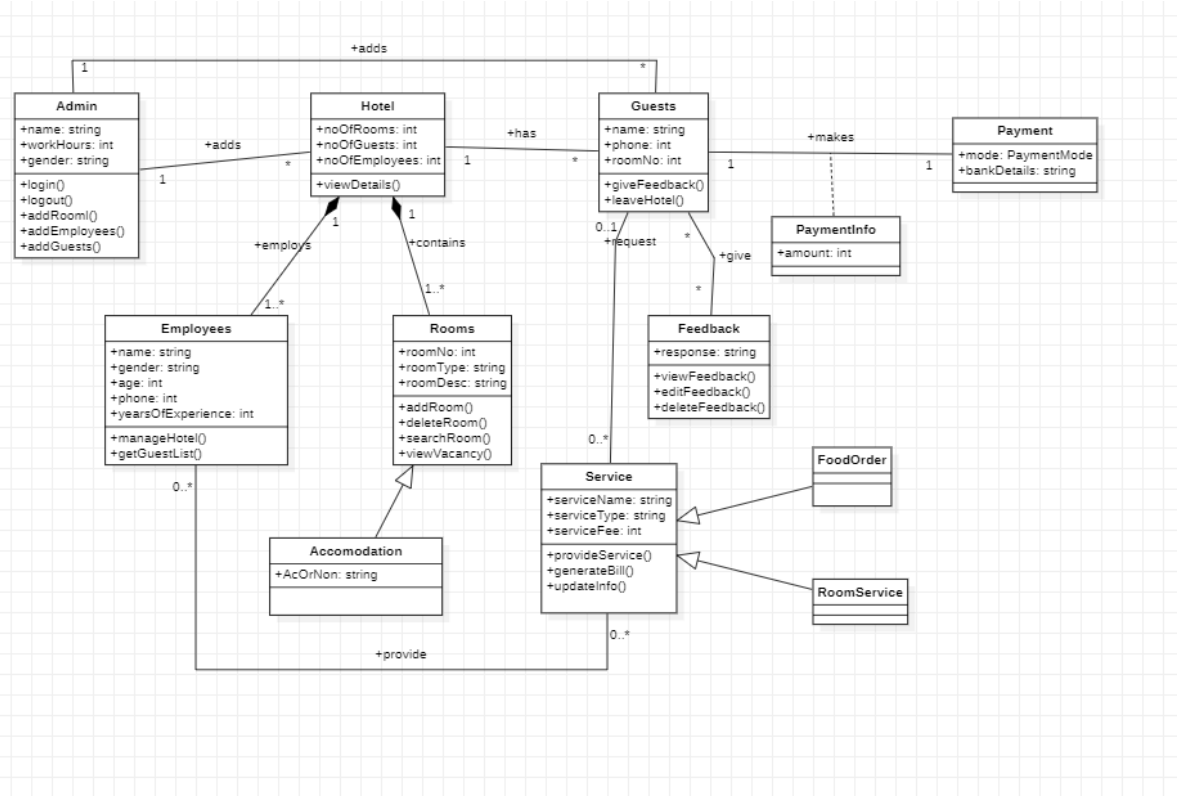


Fig 1.1 - Class Diagram for Hotel Management System

1.3 State Diagram

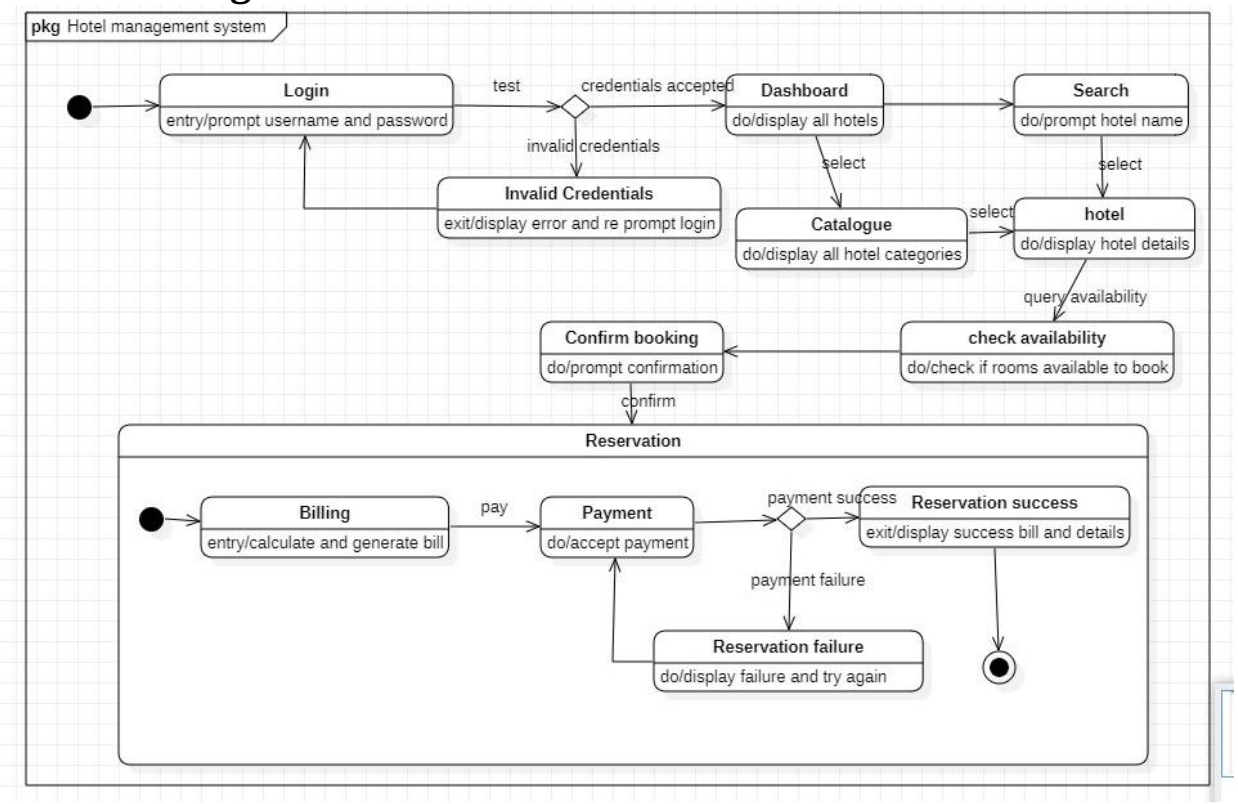


Fig 1.2 – State Diagram for Hotel Management System

1.4 Interaction Diagrams

1.4.1Use Case Diagram

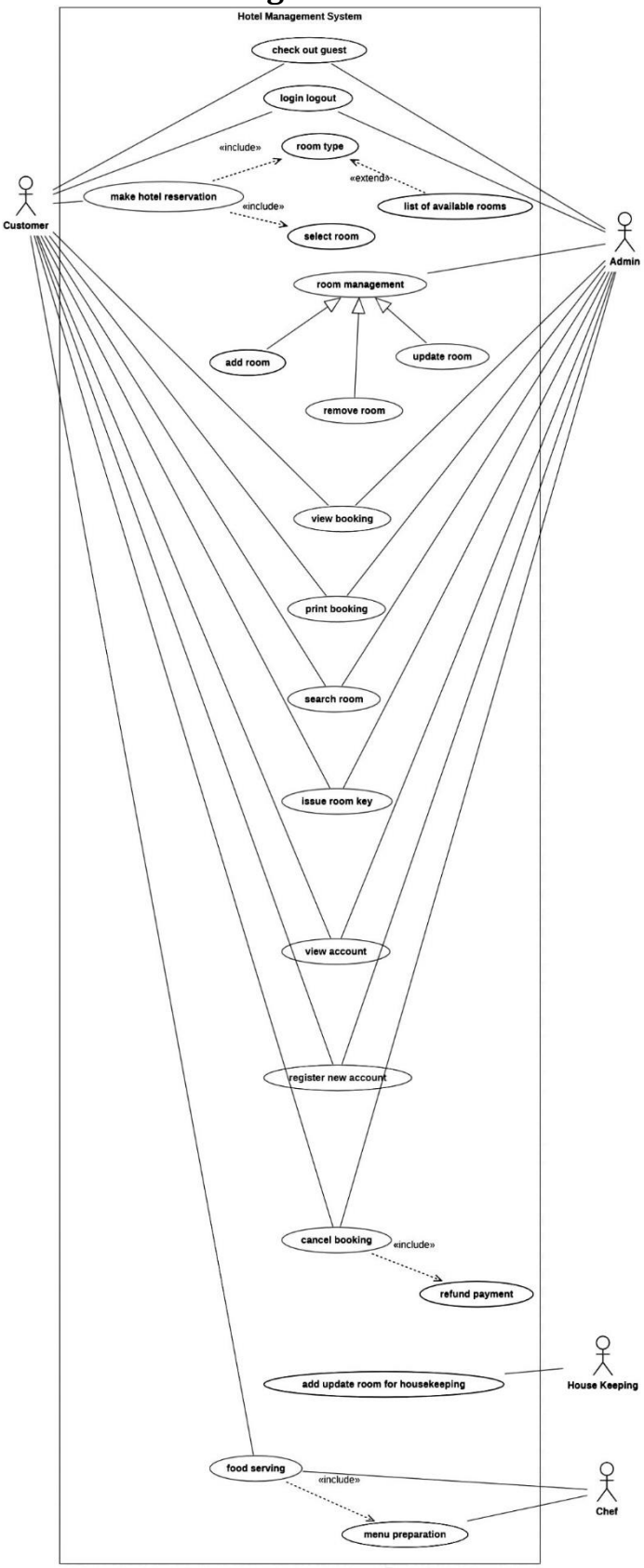


Fig 1.3 – Use Case Diagram for Hotel Management System

1.4.2Sequence Diagram

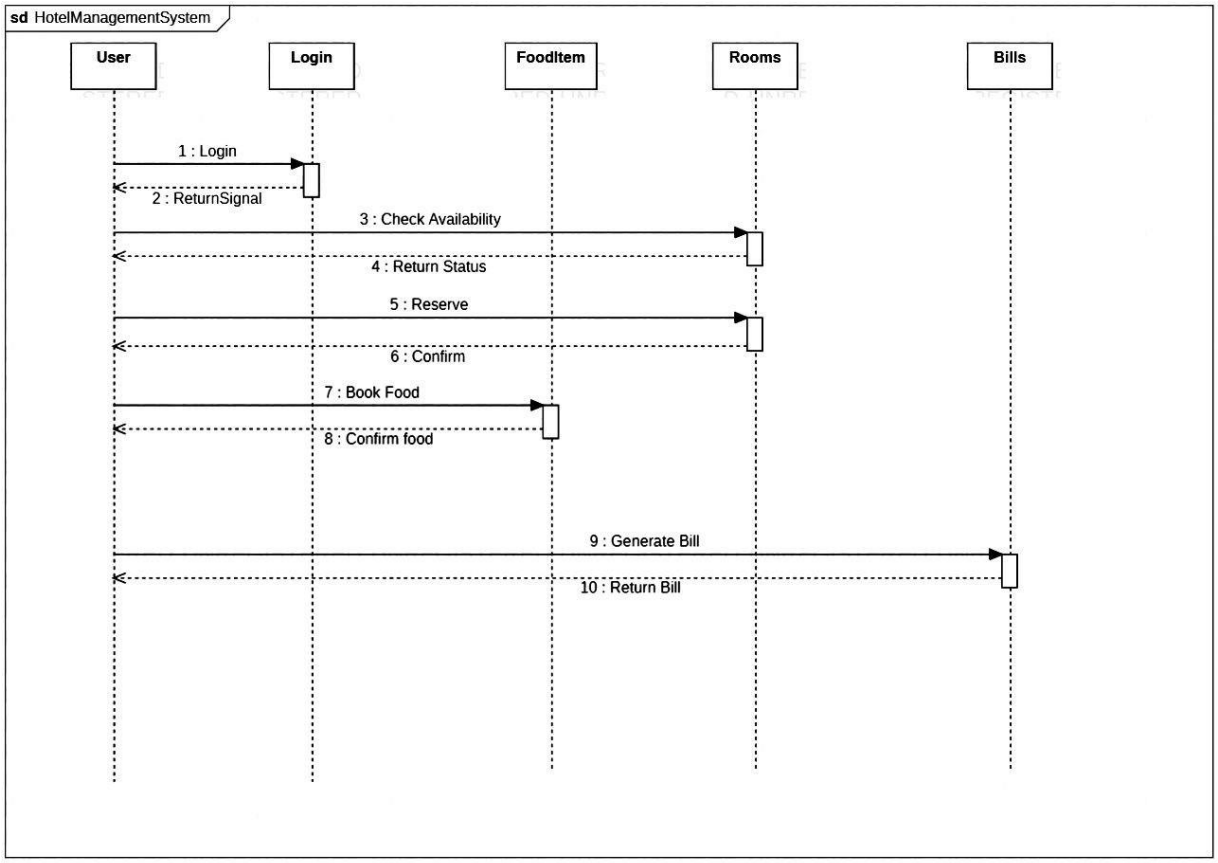


Fig 1.4 – Sequence Diagram for Hotel Management System

1.4.3Activity Diagram

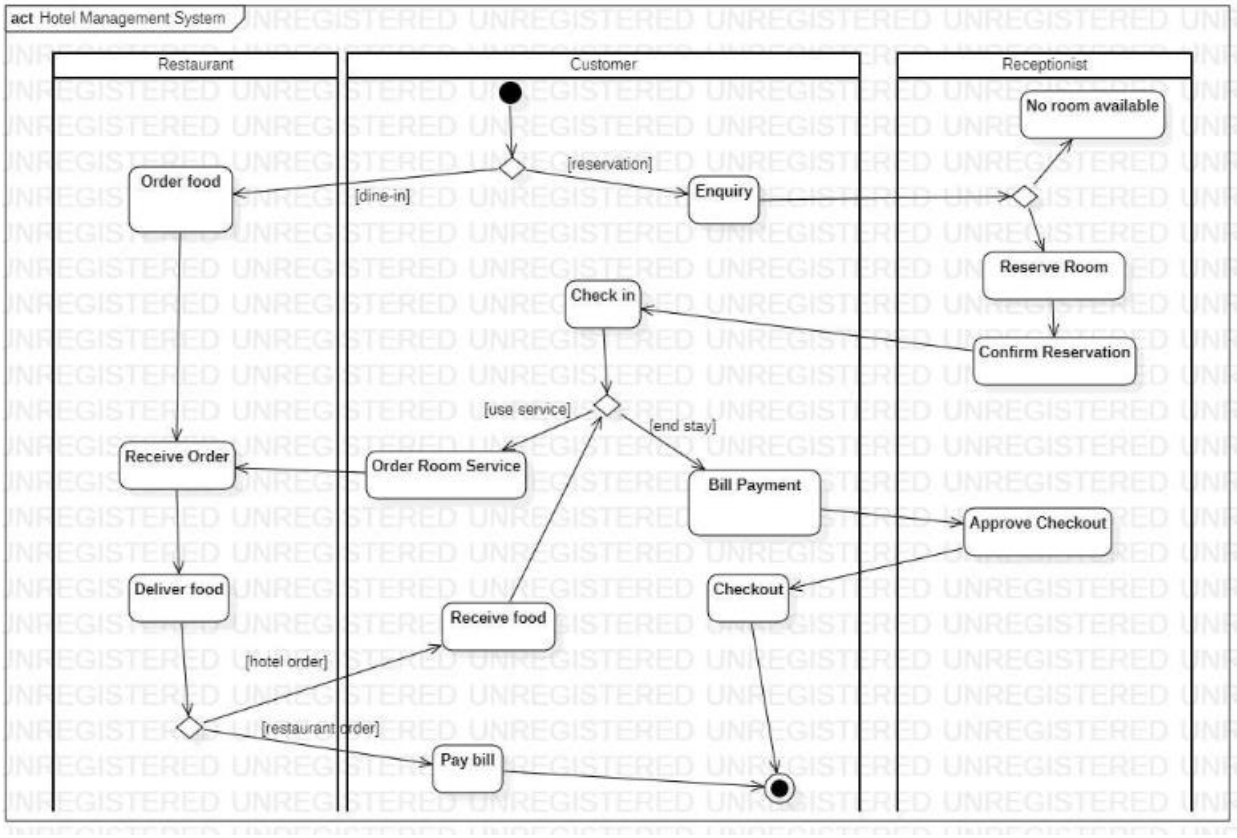
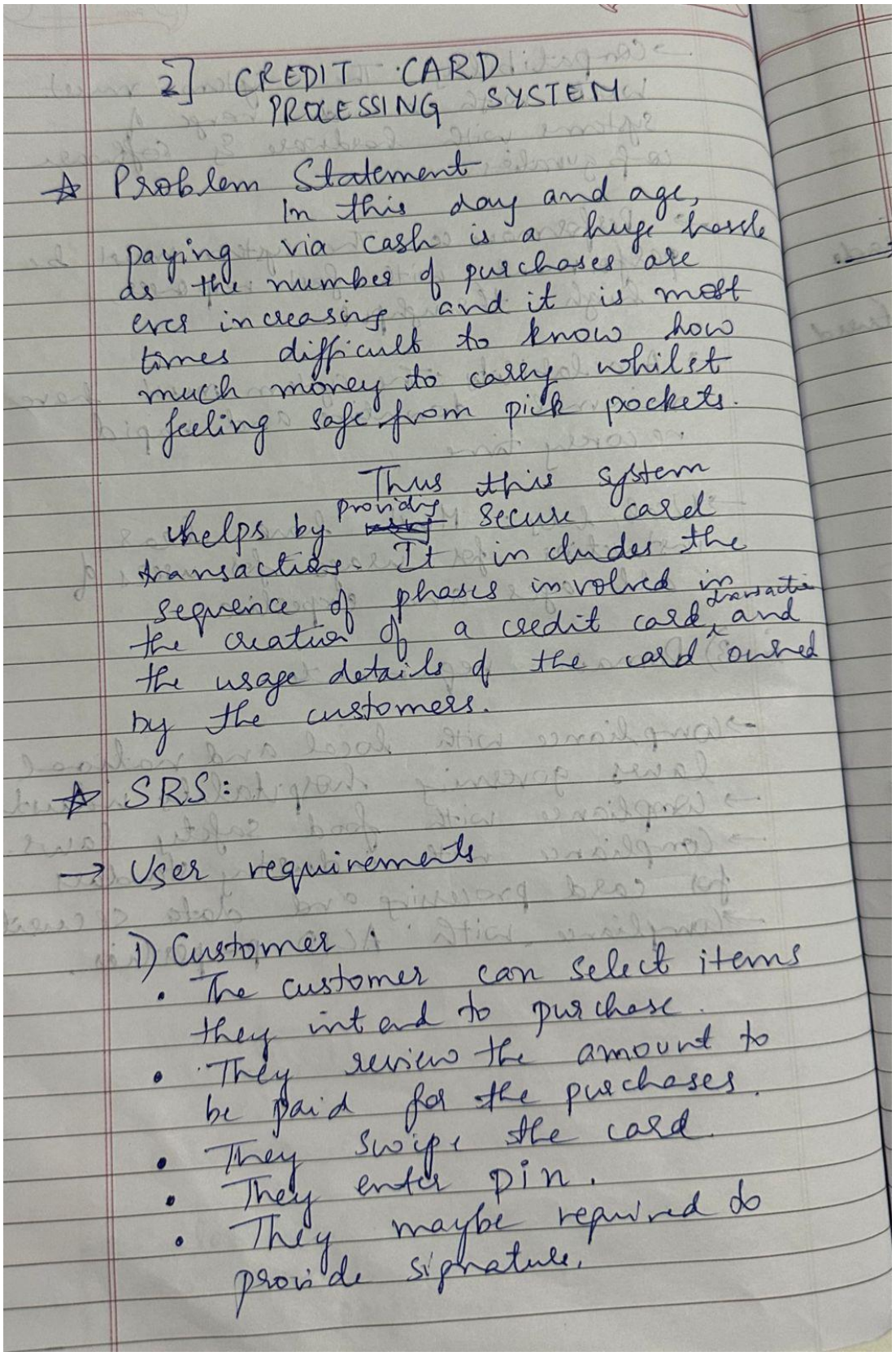


Fig 1.5 – Activity Diagram for Hotel Management System

2. Credit Card Processing System

2.1 SRS



2] CREDIT CARD PROCESSING SYSTEM

☆ Problem Statement

In this day and age, paying via cash is a huge hassle as the number of purchases are ever increasing and it is most times difficult to know how much money to carry whilst feeling safe from pick pockets.

Thus this system helps by providing secure card transactions. It includes the sequence of phases involved in the creation of a credit card, and the usage details of the card owned by the customers.

☆ SRS:

→ User requirements

1) Customer
- The customer can select items they intend to purchase.
- They review the amount to be paid for the purchases.
- They swipe the card.
- They enter pin.
- They maybe required do provide signature.

2) Vendor
- The vendor can set the amount to be debited
- They can generate the bill if payment successful.

→ System Requirements

1) Functional requirements

→ The System must be configurable as to deposit the money to the vendor's account.
→ It must accept the card swipe input.
→ It must accept amount to be debited.
→ It must accept PIN.
→ It must perform verification with bank databases.
→ It must ~~genera~~ capture exceptions.
→ It must generate reciepts for customer and vendor.
→ Ability to handle credentials with utmost security.

2) Non-functional requirements.

→ Reliability: The system must be reliable, with backup and recovery mechanisms with fault tolerance.

→ Security: The system must be secure, with strong authentication and encryption mechanisms.

→ Availability: The system must be highly available, with minimal downtime and rapid recovery from failures.

→ Maintainability: The system must be easy to maintain and upgrade update, with clear documentation, modular design and well-structured code.

3) Domain requirements

→ It must be compliant with industry standard security regulations such as PCI-DSS.
→ It must be compliant with Online transaction laws.
→ It must practice highest level encryption and data security.
→ It must be properly configured for interfacing and authenticating with bank ~~database and~~ systems.

2.2 Class Diagram

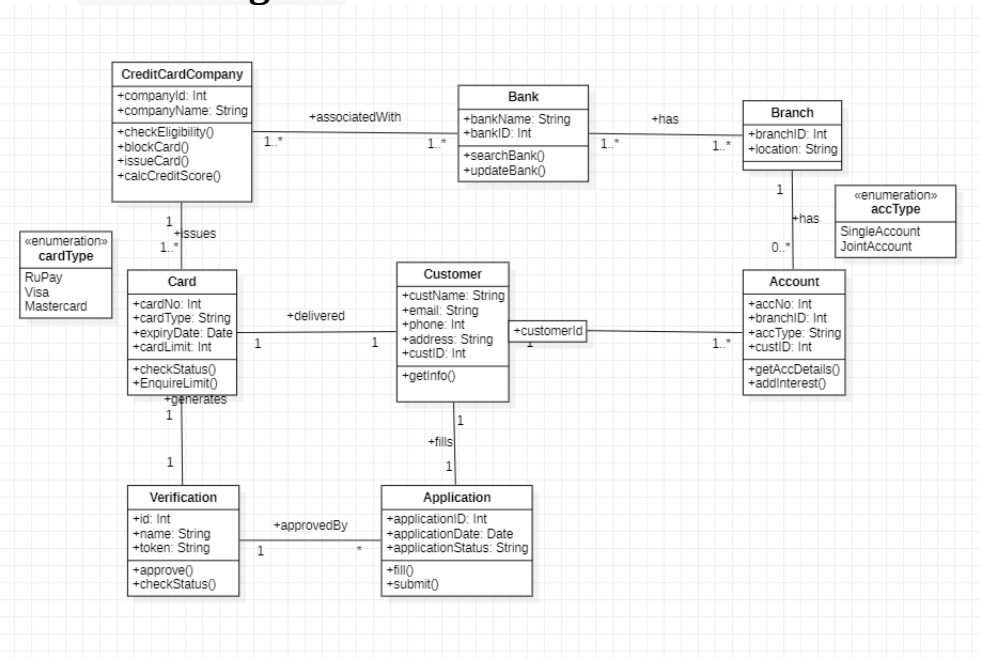


Fig 2.1 – Class Diagram for Credit Card Processing System

2.3 State Diagram

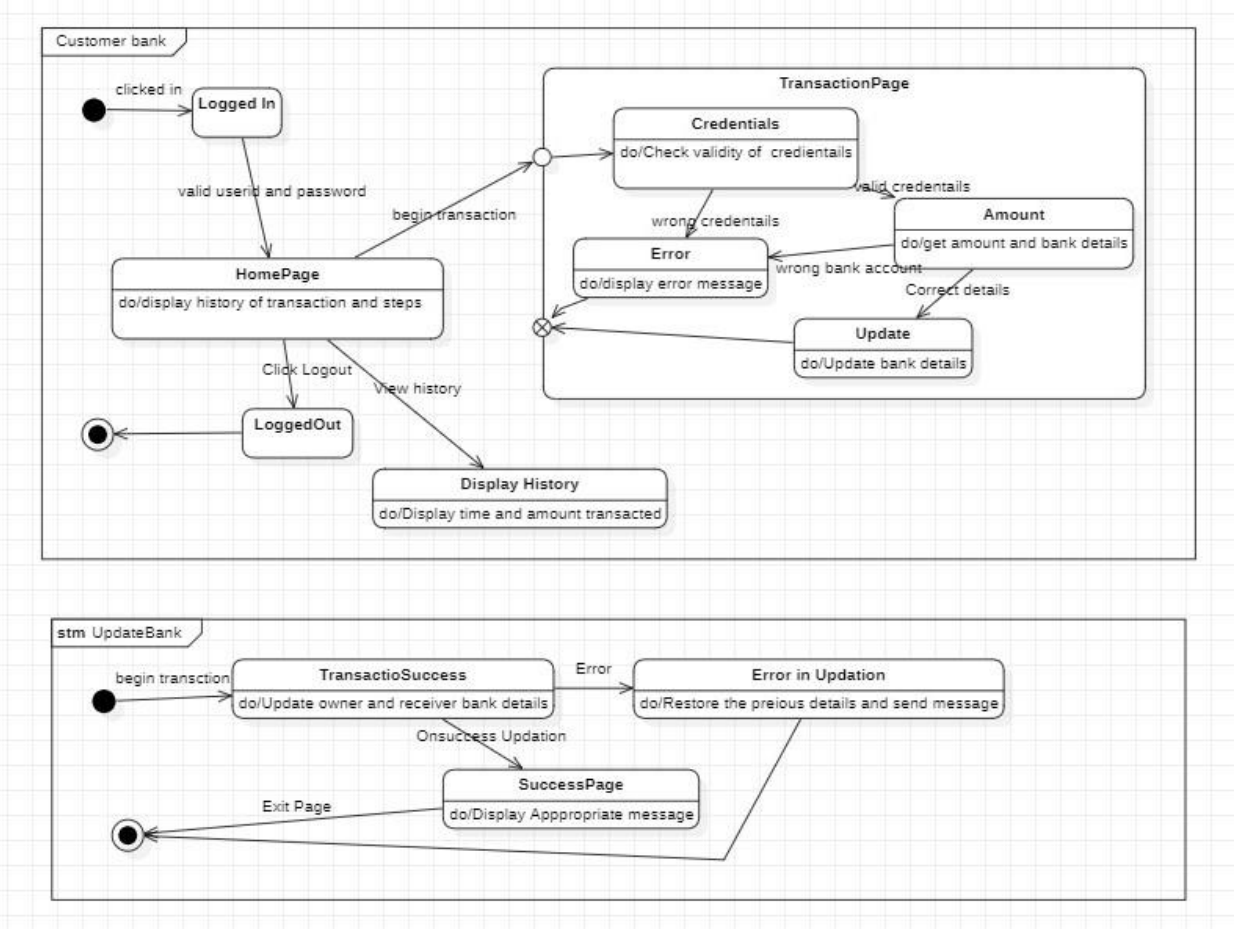


Fig 2.2 – State Diagram for Credit Card Processing System

2.4 Interaction Diagrams

2.4.1 Use Case Diagram

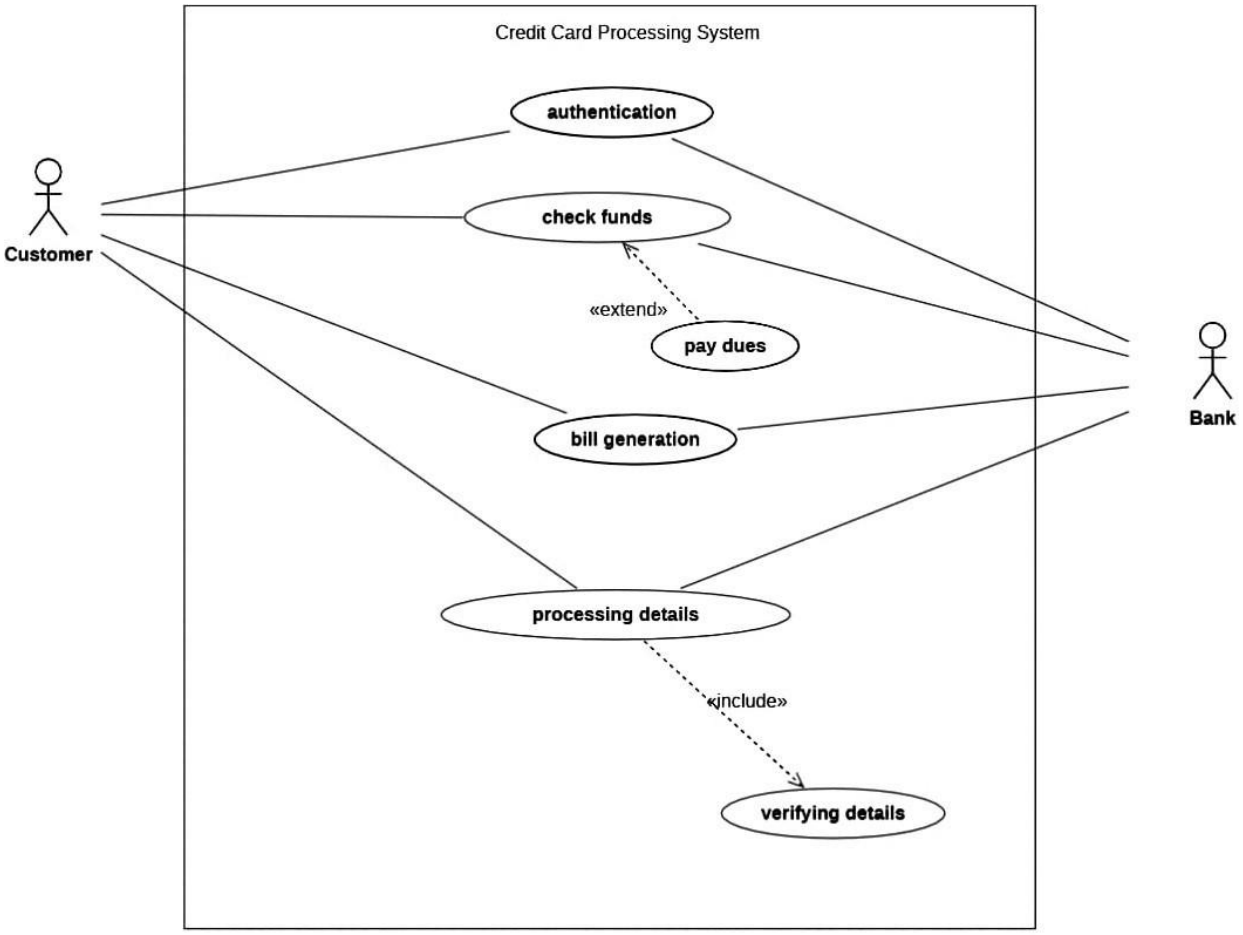


Fig 2.3 – Use Case diagram for Credit Card Processing System

2.4.2 Sequence Diagram

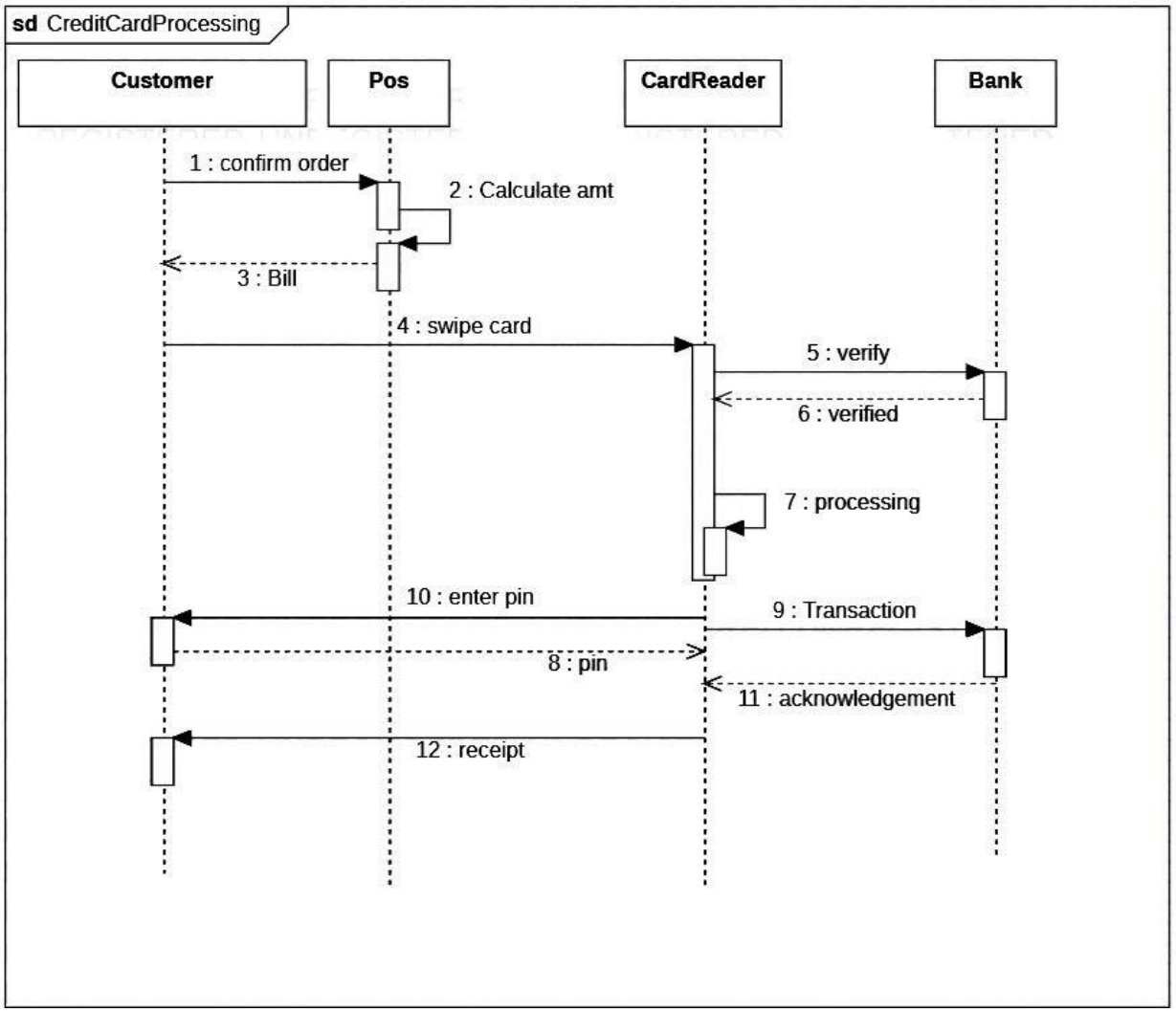


Fig 2.4 – Sequence Diagram for Credit Card Processing System

2.4.3 Activity Diagram

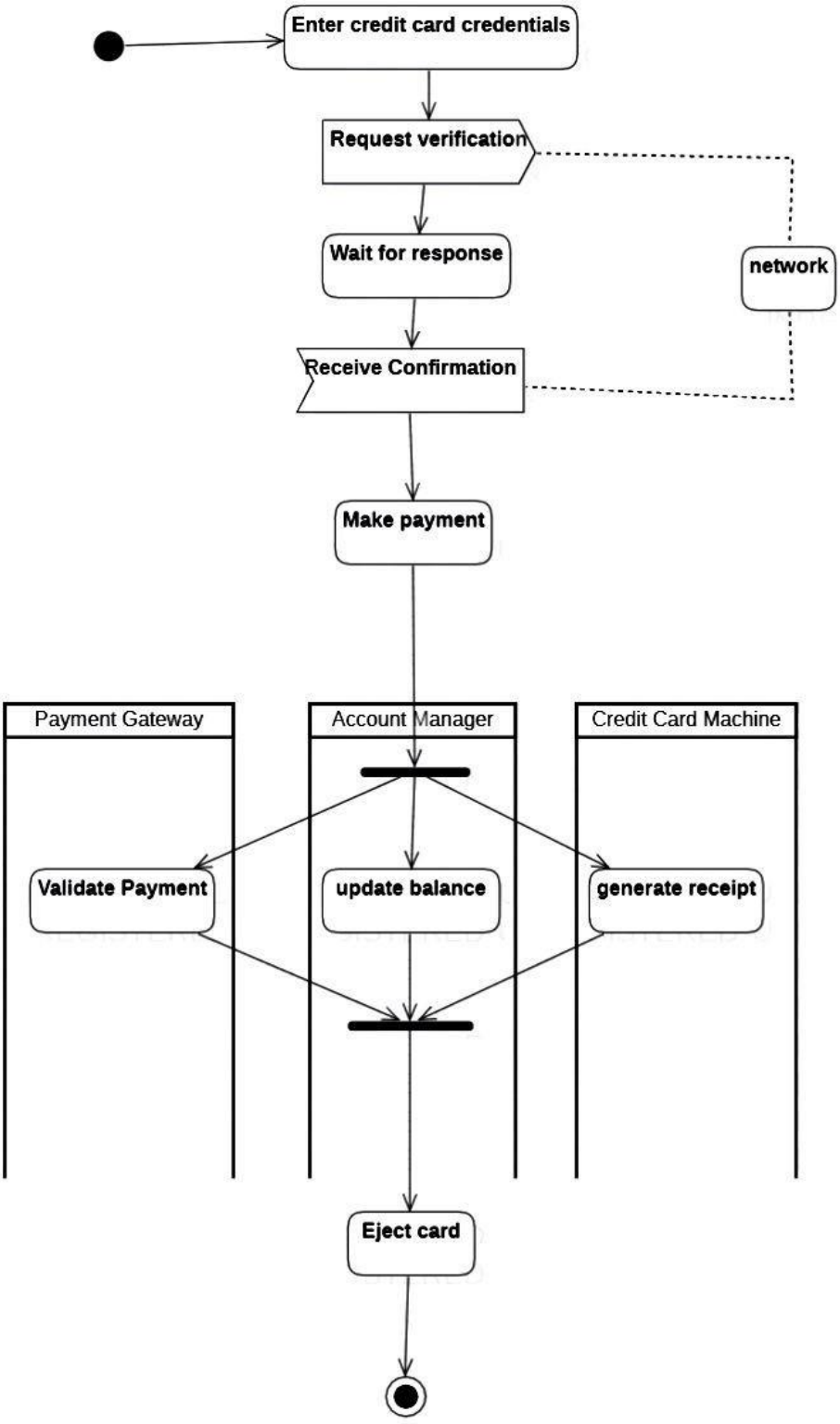
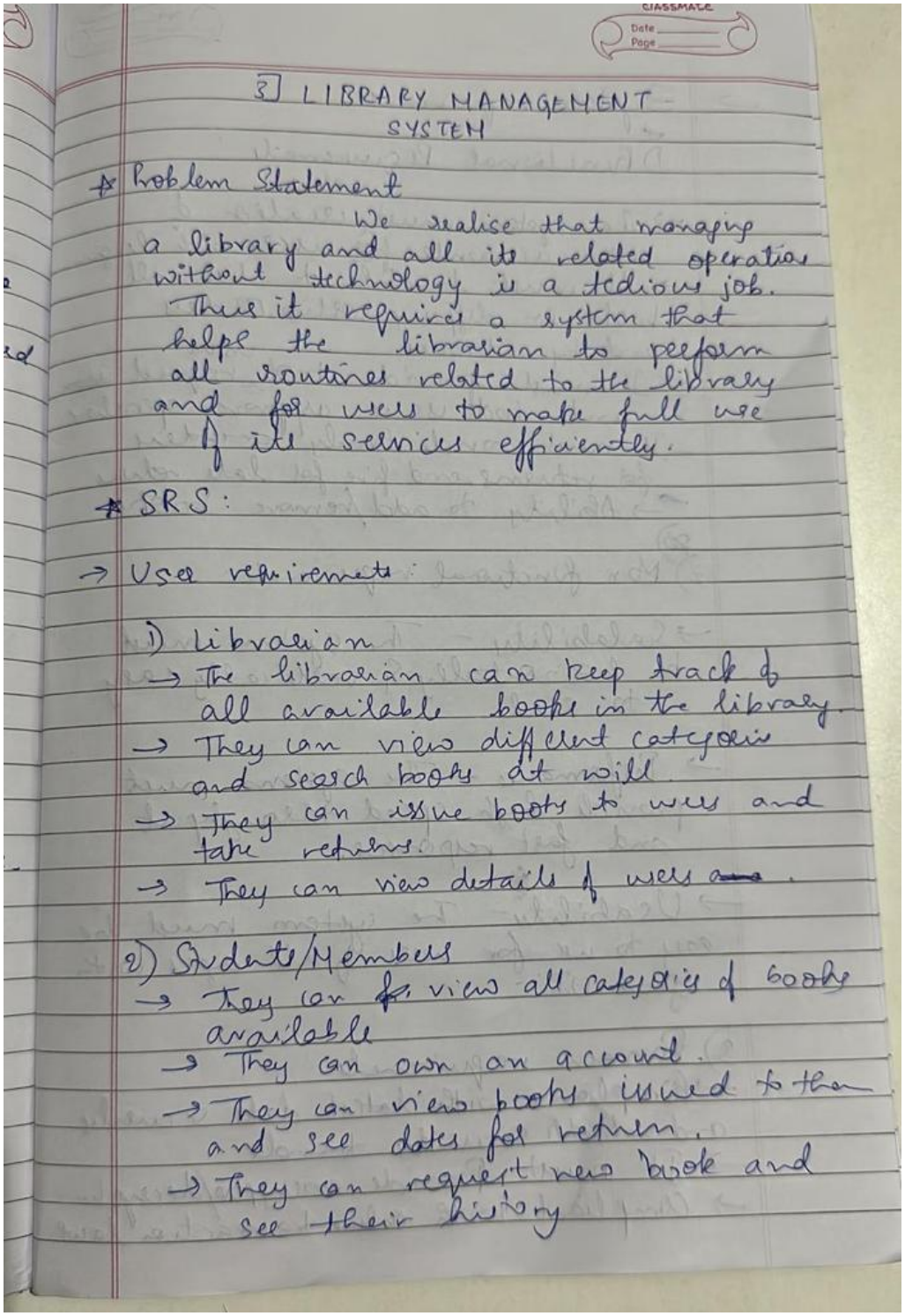


Fig 2.5 – Activity Diagram for Credit Card Processing System

3. Library Management System

3.1 Problem Statement



3] LIBRARY MANAGEMENT SYSTEM

* Problem Statement

We realise that managing a library and all its related operation without technology is a tedious job. Thus it requires a system that helps the librarian to perform all routines related to the library and for users to make full use of its services efficiently.

* SRS:

→ User requirements:

1) Librarian
→ The librarian can keep track of all available books in the library.
→ They can view different categories and search books at will.
→ They can issue books to users and take returns.
→ They can view details of users.

2) Students/Members
→ They can view all categories of books available
→ They can own an account.
→ They can view books issued to them and see dates for return.
→ They can request new book and see their history

→ System requirements

1) Functional Requirements

→ The system must allow creation of new accounts for users with login

→ It must be able to list all books in differing sorting Orders and by categories.

→ It must provide book issue request and book issue approve operations

→ It must provide timely reminders for returns and fine for late returns

→ Ability to add/remove books

2) Non functional requirements:

→ Scalability - The system must be horizontally & vertically easy to scale.

→ Performance - The systems must provide high speed throughput and fast responses.

→ Usability - The system must be easy to use for users of all ages with good UI.

3) Domain requirements.

→ Compliance with latest data security and network security trends.

→ It must Provide encrypt/decrypt.

→ Compliant with online transaction laws.

3.2 Class Diagram

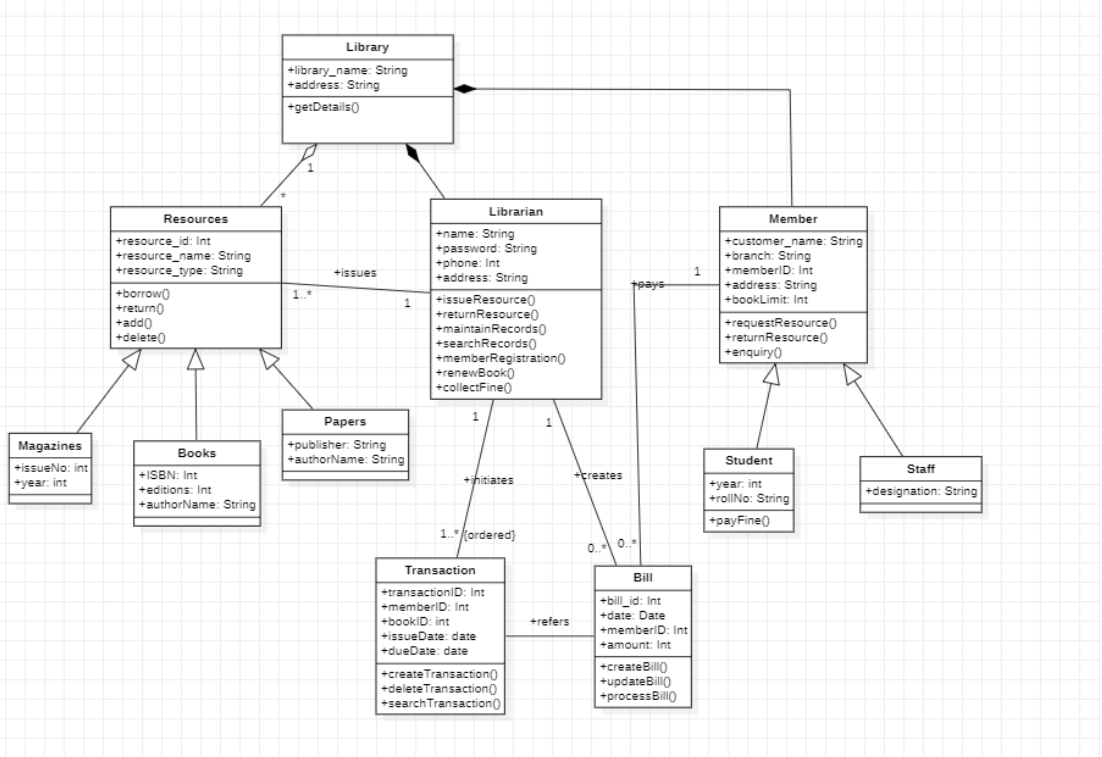


Fig 3.1 – Class Diagram for library management

3.3 State Diagram

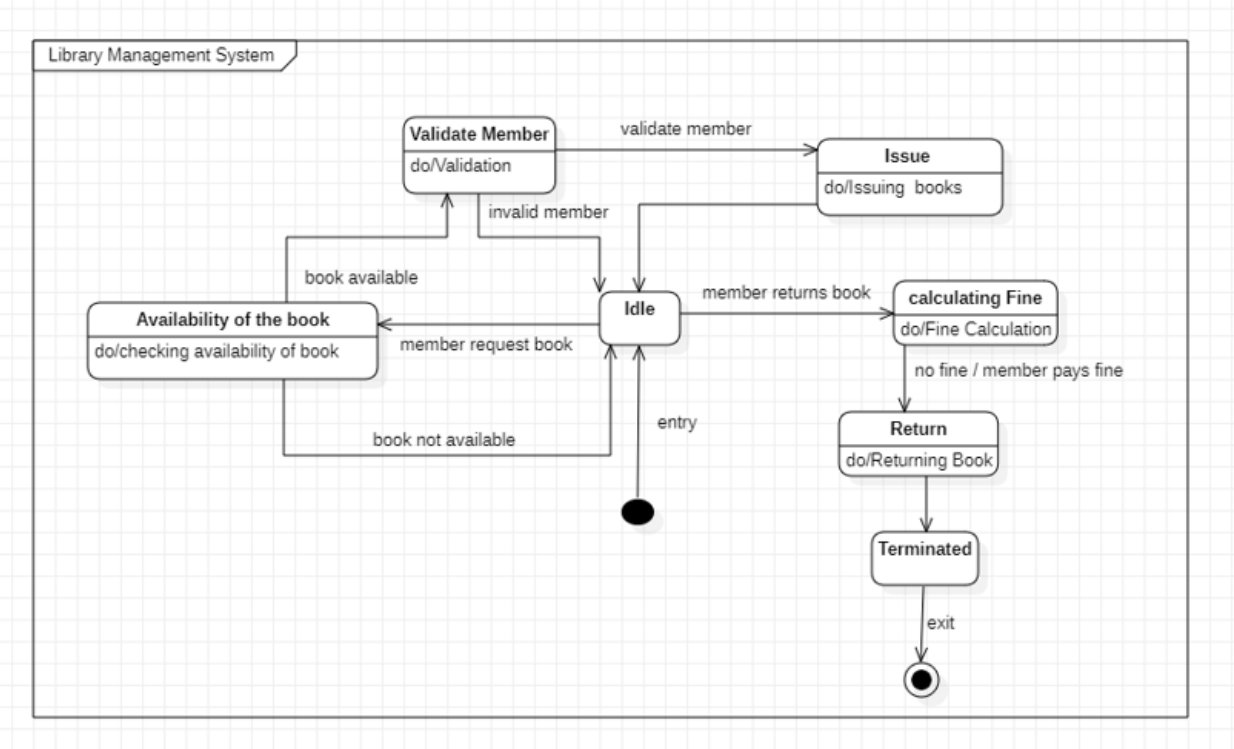


Fig 3.2 – State Diagram for library management

3.4 Interaction Diagram

3.4.1 Use Case Diagram

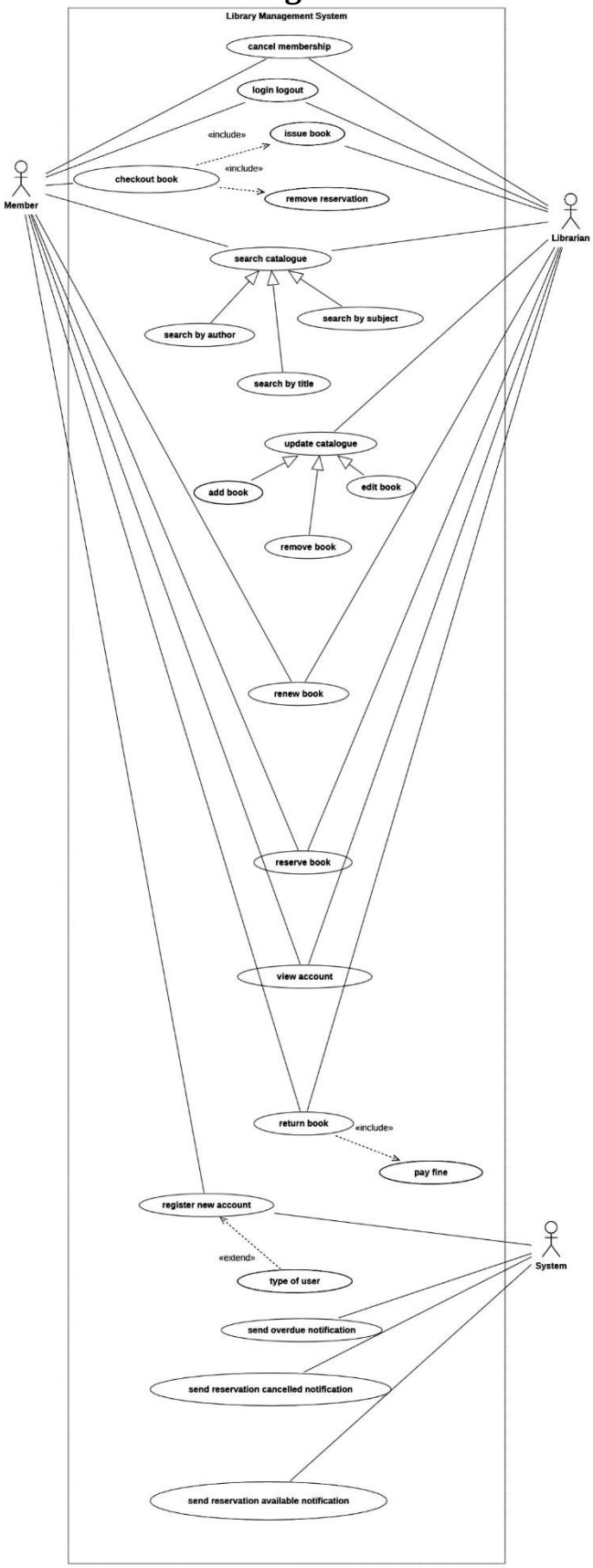


Fig 3.3 – Use Case Diagram for library management

3.4.2 Sequence Diagram

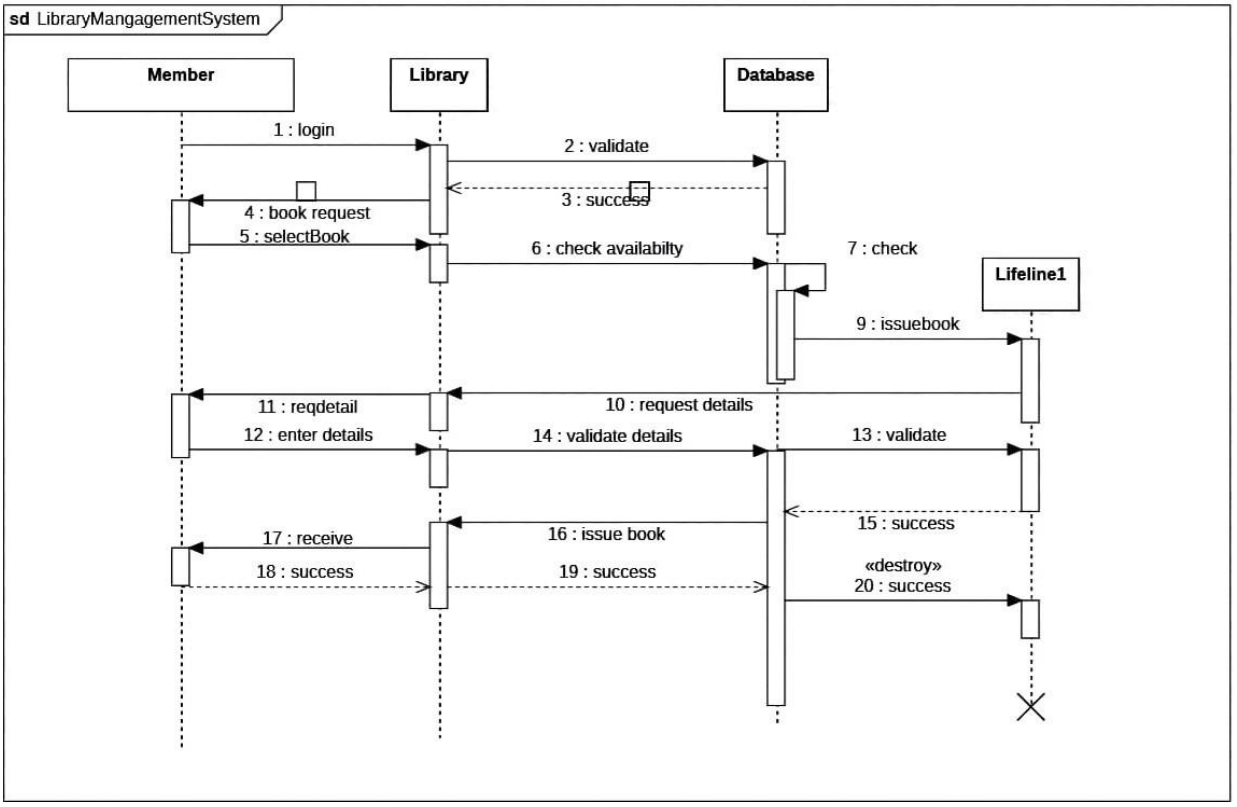


Fig 3.4 - Sequence Diagram for library management

3.4.3 Activity Diagram

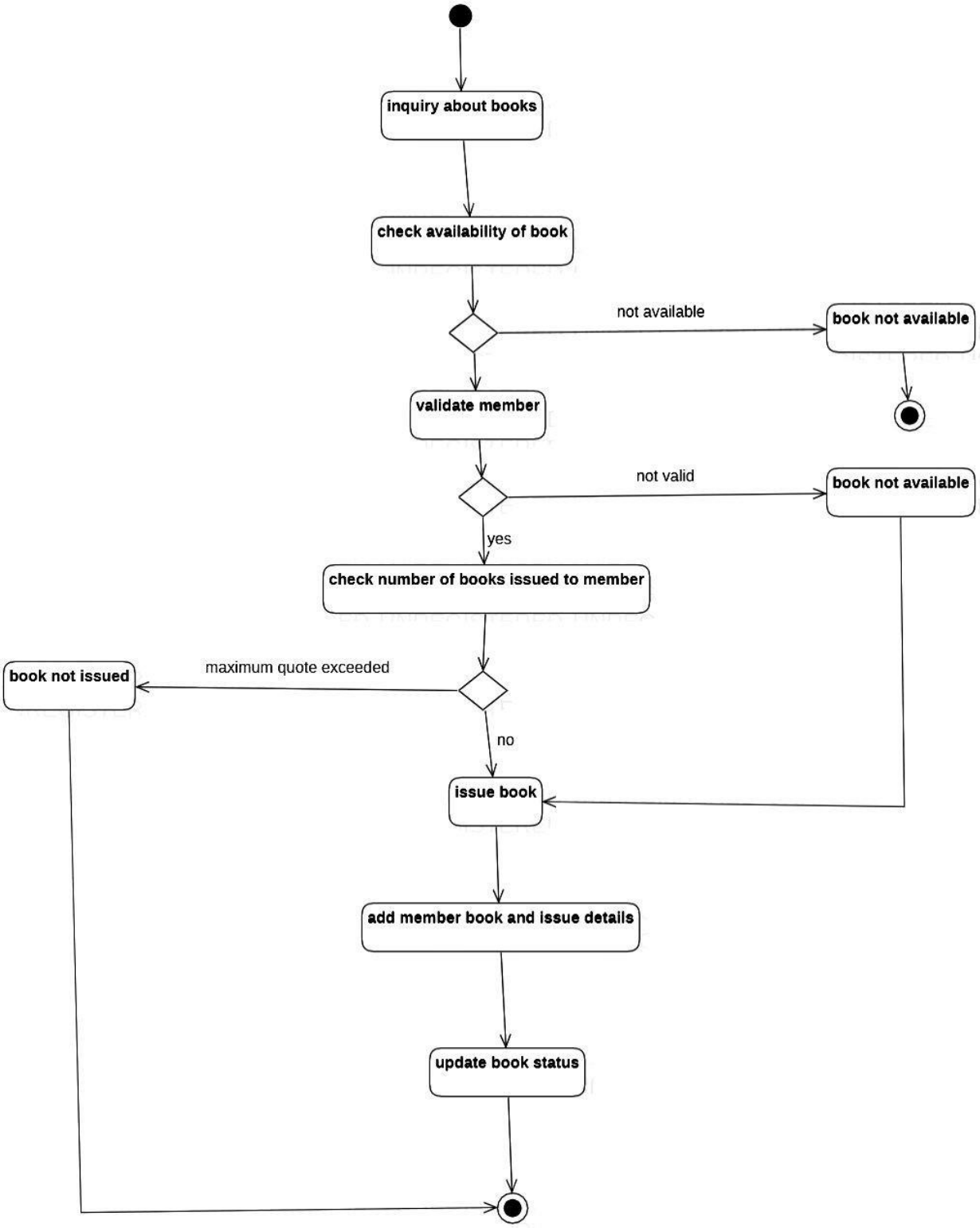
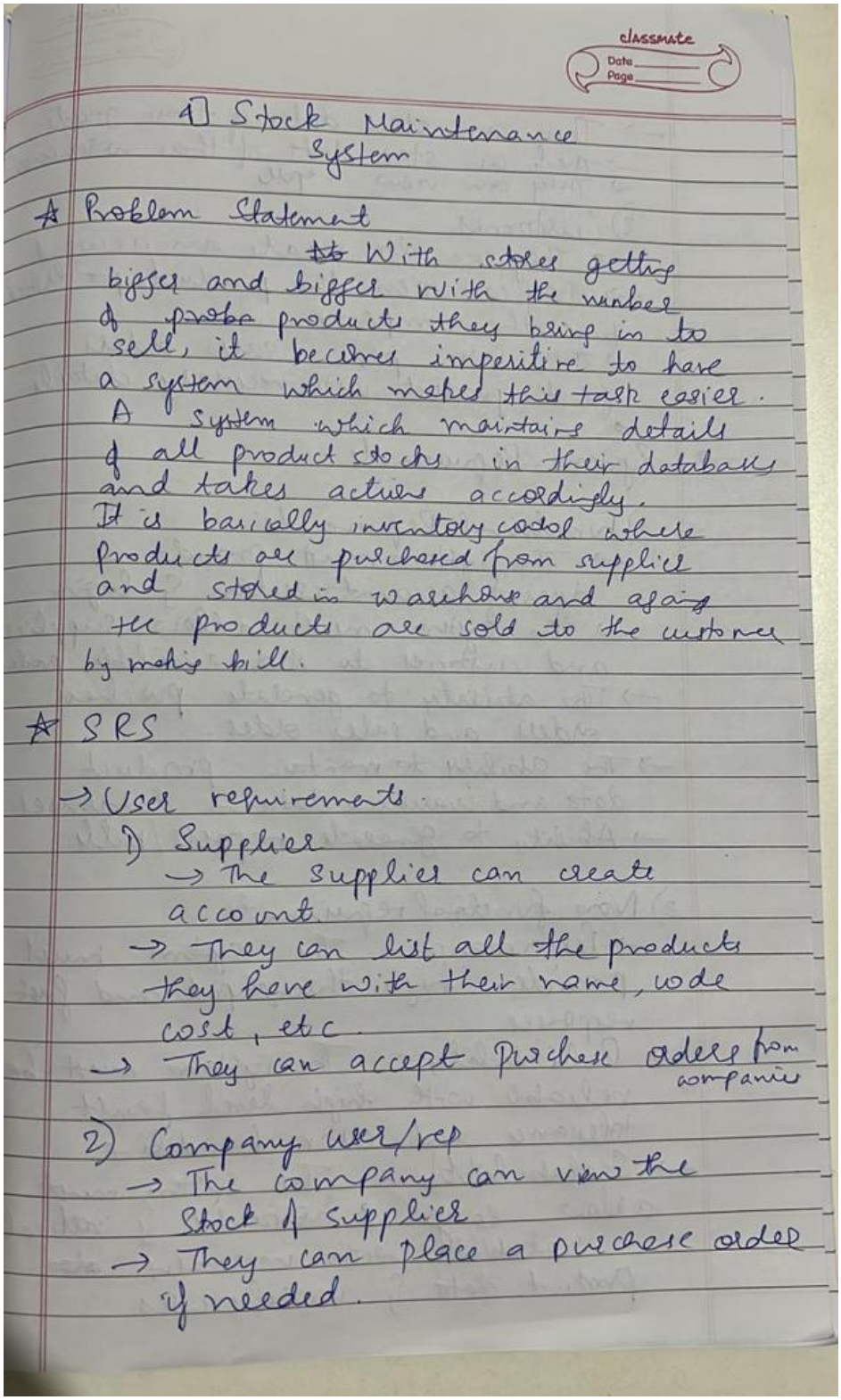


Fig 3.5 - Activity Diagram for library management

4. Stock Maintenance System

4.1 Problem Statement



4] Stock Maintenance System

★ Problem Statement

With stores getting bigger and bigger with the number of products they bring in to sell, it becomes imperitive to have a system which makes this task easier. A system which maintains details of all product stocks in their database and takes actions accordingly. It is basically inventory control where products are purchased from supplier and stored in warehouse and again the products are sold to the customer by making bill.

★ SRS

→ User requirements

1) Supplier
   → The supplier can create account.
   → They can list all the products they have with their name, code cost, etc.
   → They can accept purchase orders from companies

2) Company user/rep
   → The company can view the stock of supplier
   → They can place a purchase order if needed.

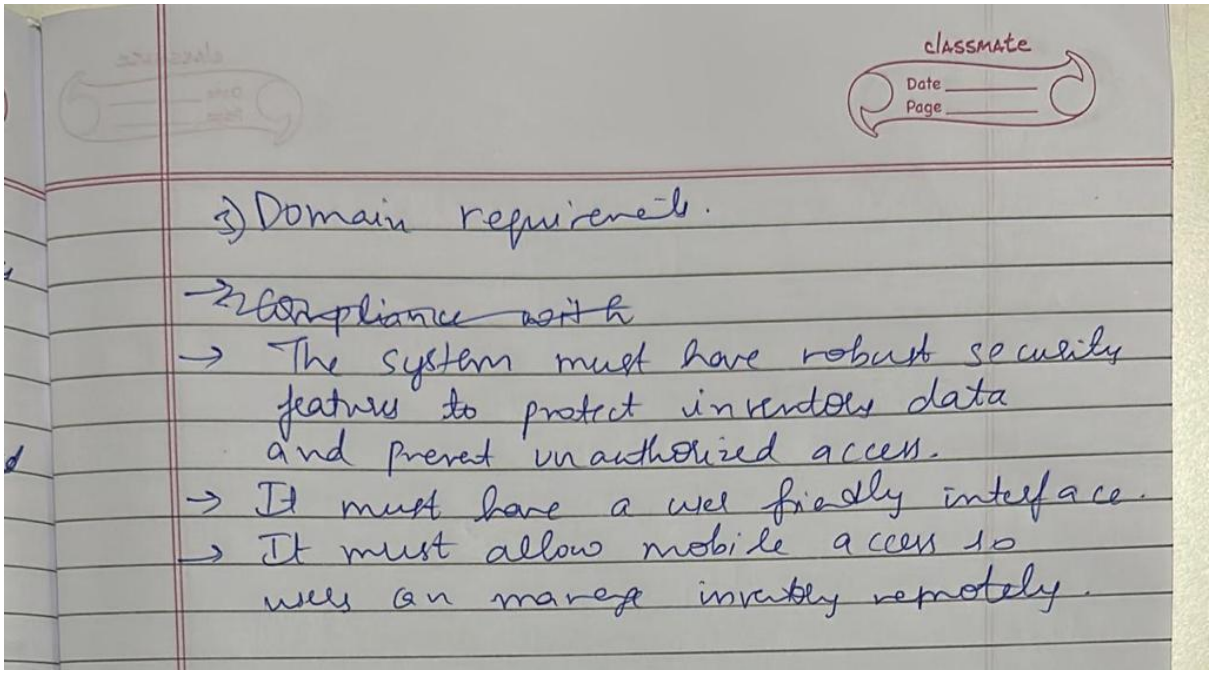
→ They can store data of their goods
→ They can store data of their warehouse
→ They can view reports

3) Customer
→ They can also create an account
→ They can view the products listed by the company.
→ They can place sales orders for the product with necessary details.

→ System Requirements

1) Functional Requirements
→ The system must provide 3 types of account creation & login.
→ The system must allow supplier and customer to show available products
→ The ability to generate purchase orders and sales order.
→ The ability to maintain product data and warehouse data in databases.
→ Ability to generate reports/bills.

2) Non functional requirements
→ Performance - The system must provide high throughput and fast responses.
→ Reliability - The system must be reliable with high level fault tolerance and error handling.
→ Scalability - The system must allow easy horizontal & vertical scalability with increasing product data & warehouses



4.2 Class Diagram

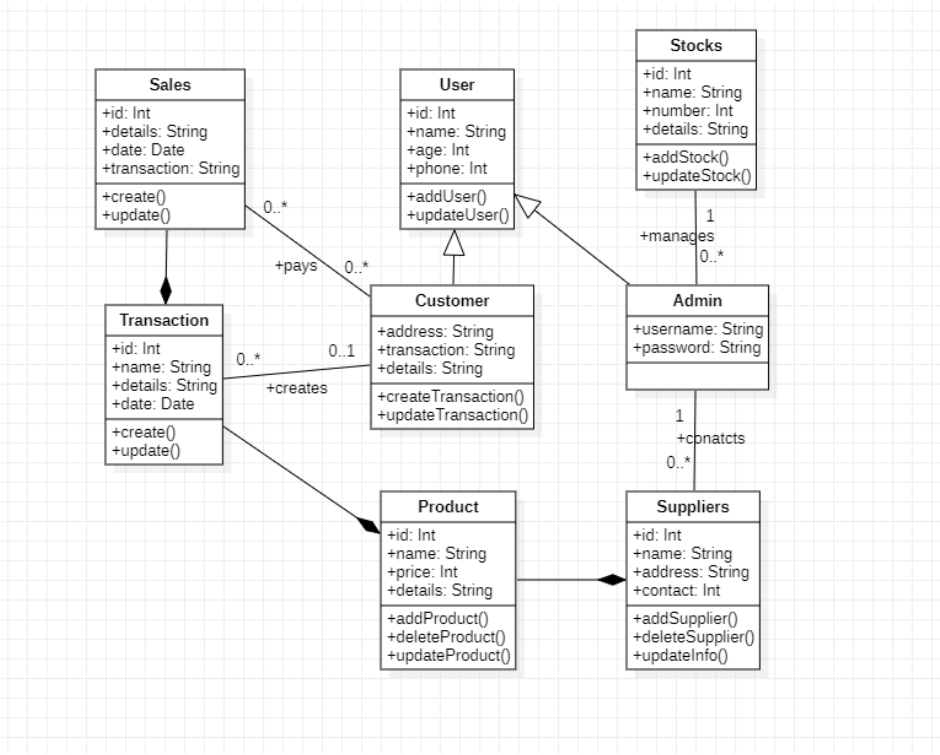


Fig 4.1 - Class Diagram for stock maintenance

4.3 State Diagram

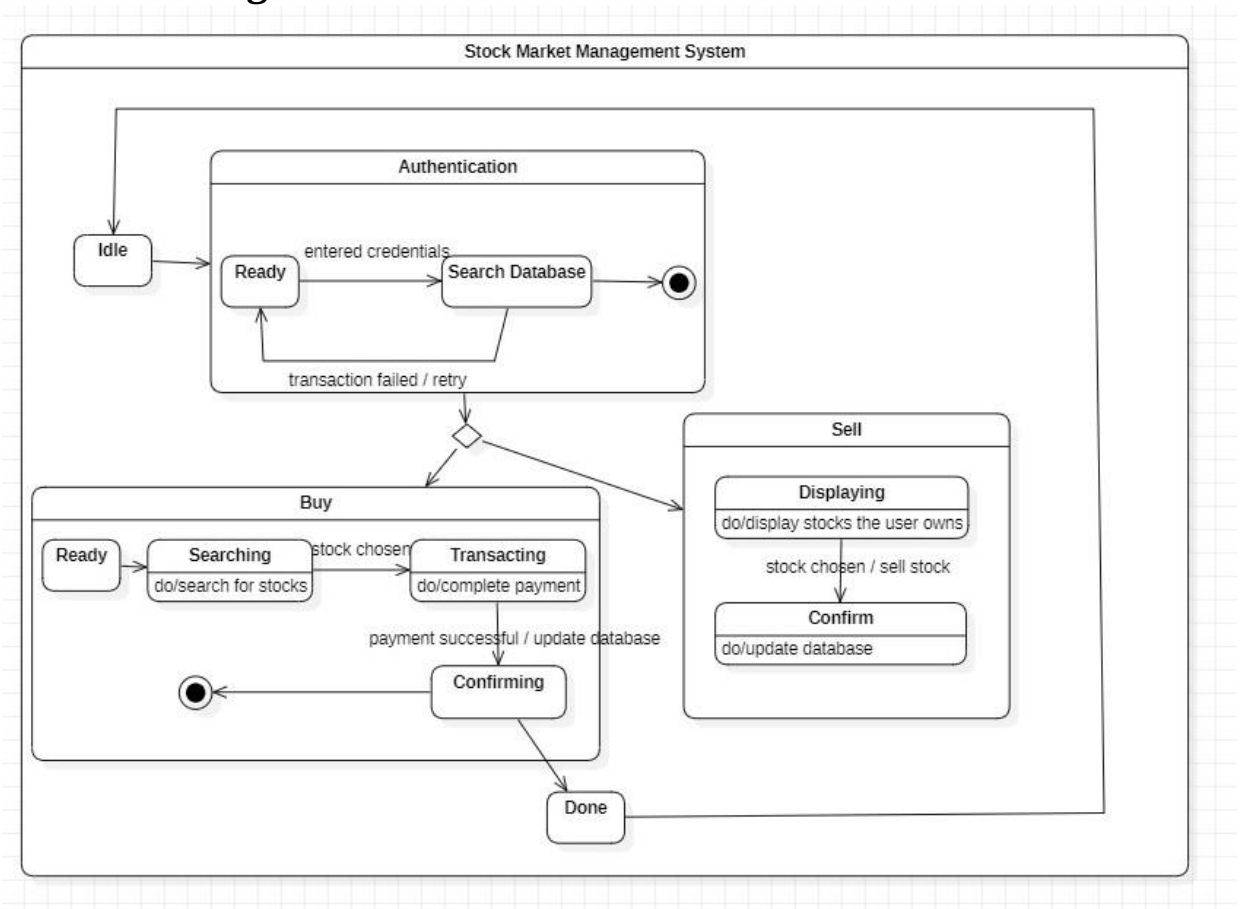


Fig 4.2 - State Diagram for stock maintenance

4.4 Interaction Diagram

4.4.1 Use Case Diagram

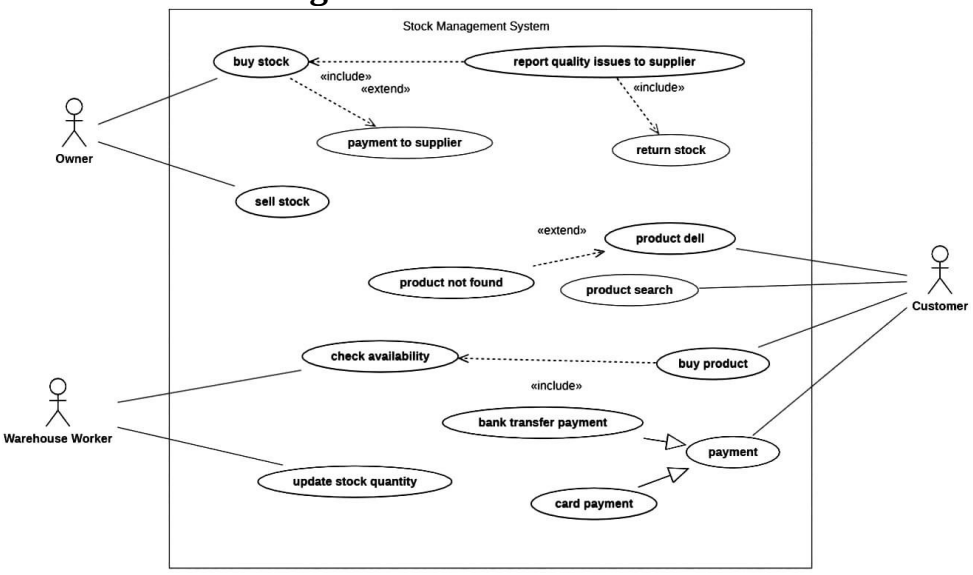


Fig 4.3 – Use Case Diagram for stock maintenance

4.4.2 Sequence Diagram

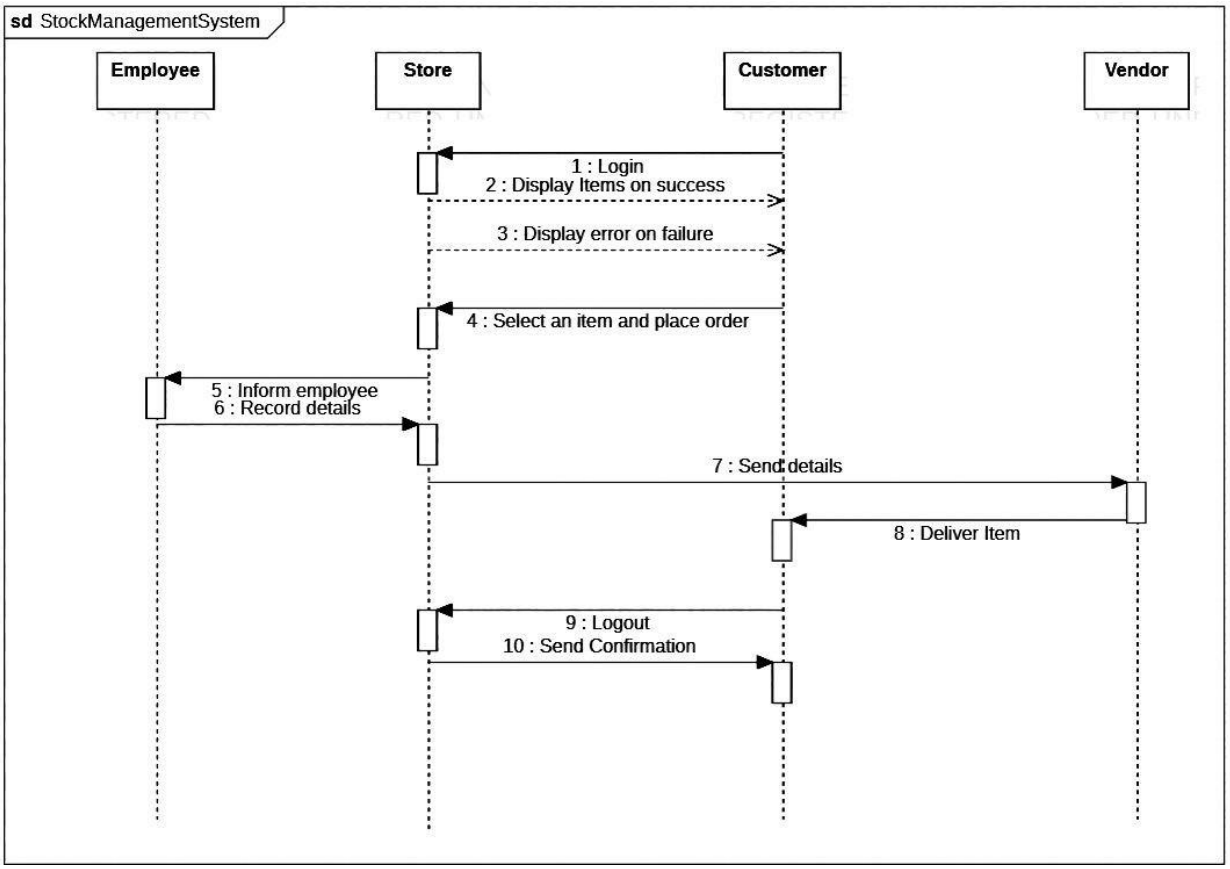


Fig 4.4 – Sequence Diagram for stock maintenance

4.4.3 Activity Diagram

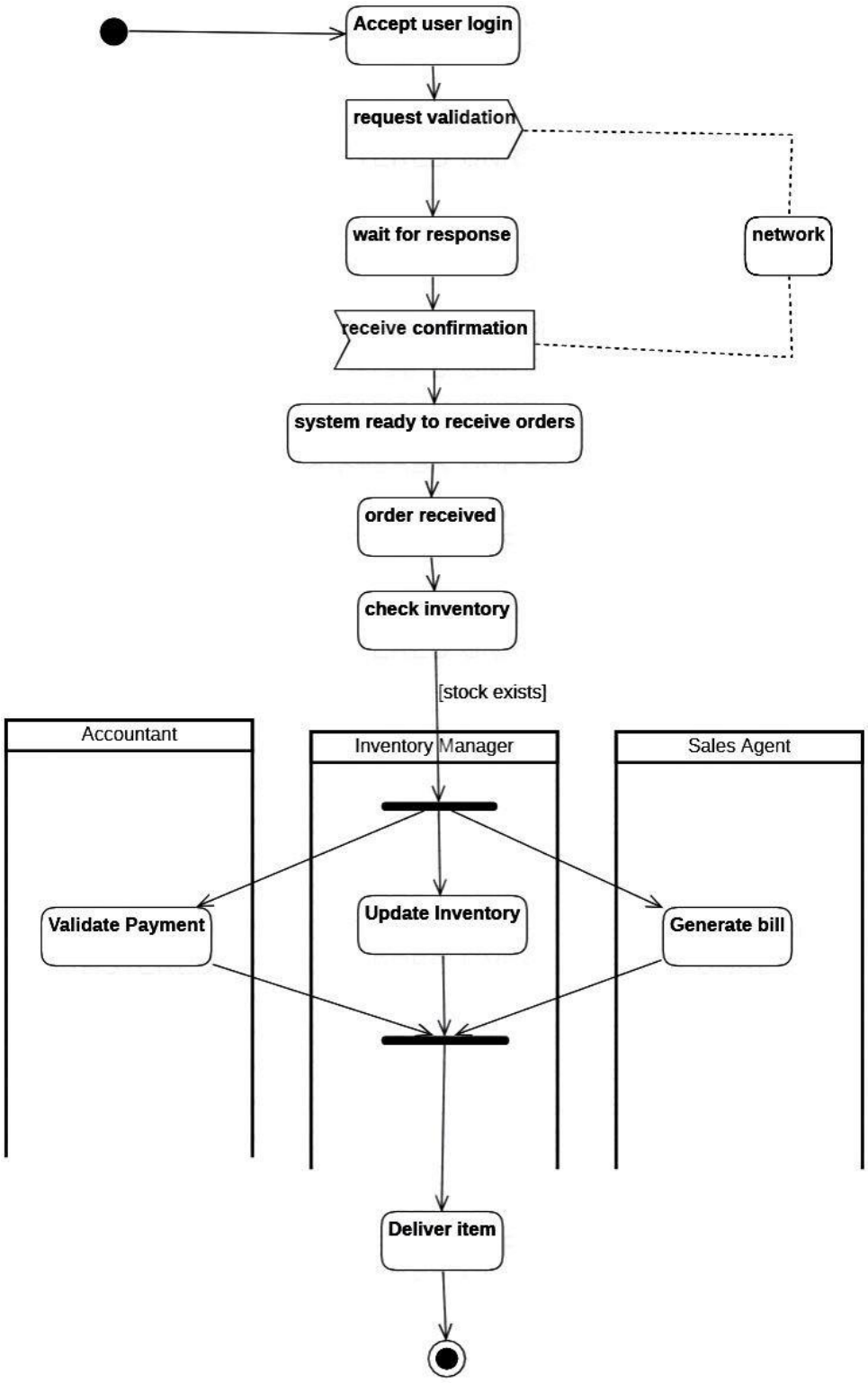
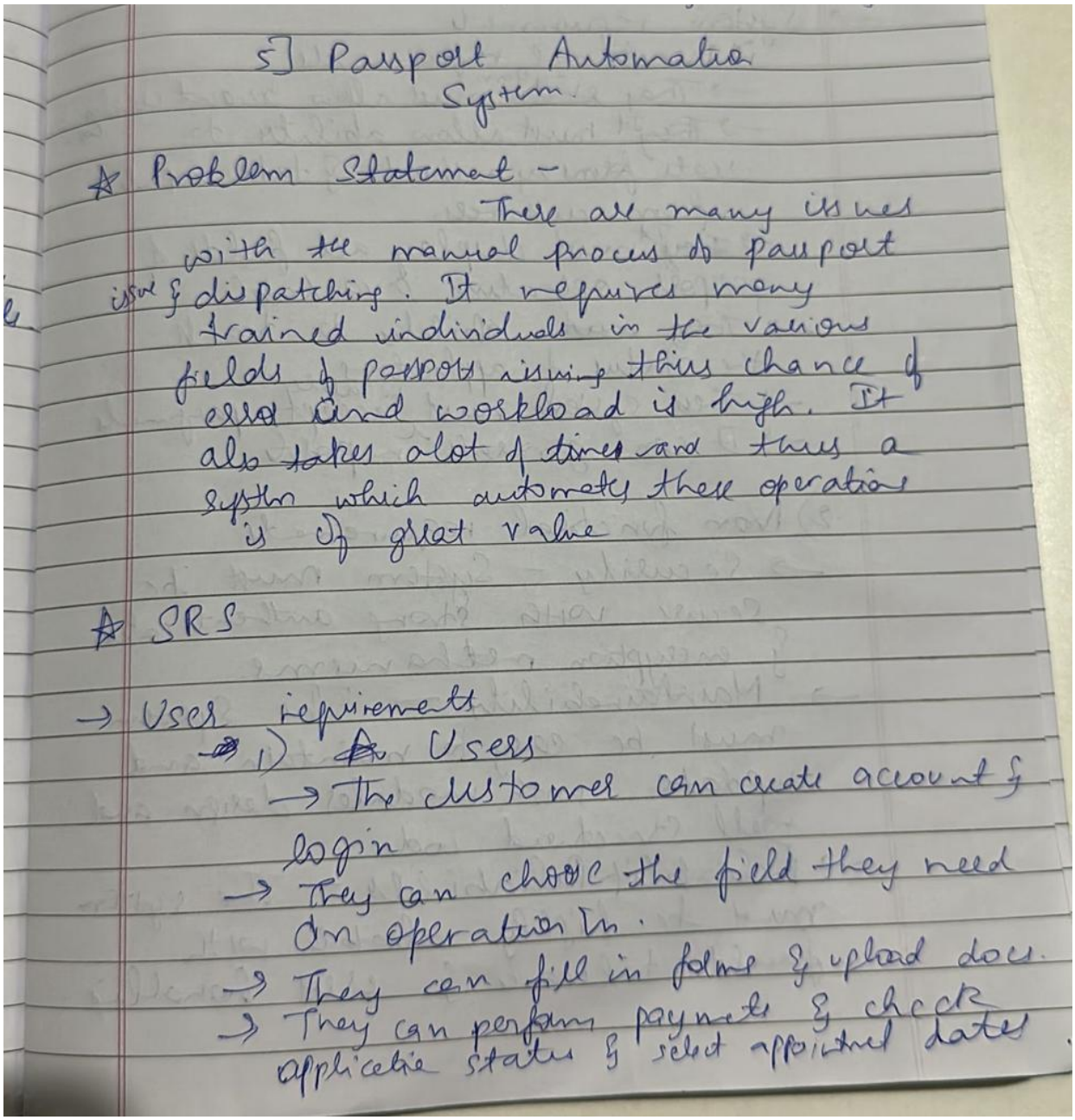


Fig 4.5 – Activity Diagram for stock maintenance

5. Passport Authentication System

5.1 Problem Statement



5] Passport Automation System.

☆ Problem Statement -

There are many issues with the manual process of passport issue & dispatching. It requires many trained individuals in the various fields of passport issuing this chance of error and workload is high. It also takes alot of time and thus a system which automate these operation is of great value

☆ SRS

→ User requirements

1) Users

→ The customer can create account & login

→ They can choose the field they need an operation in.

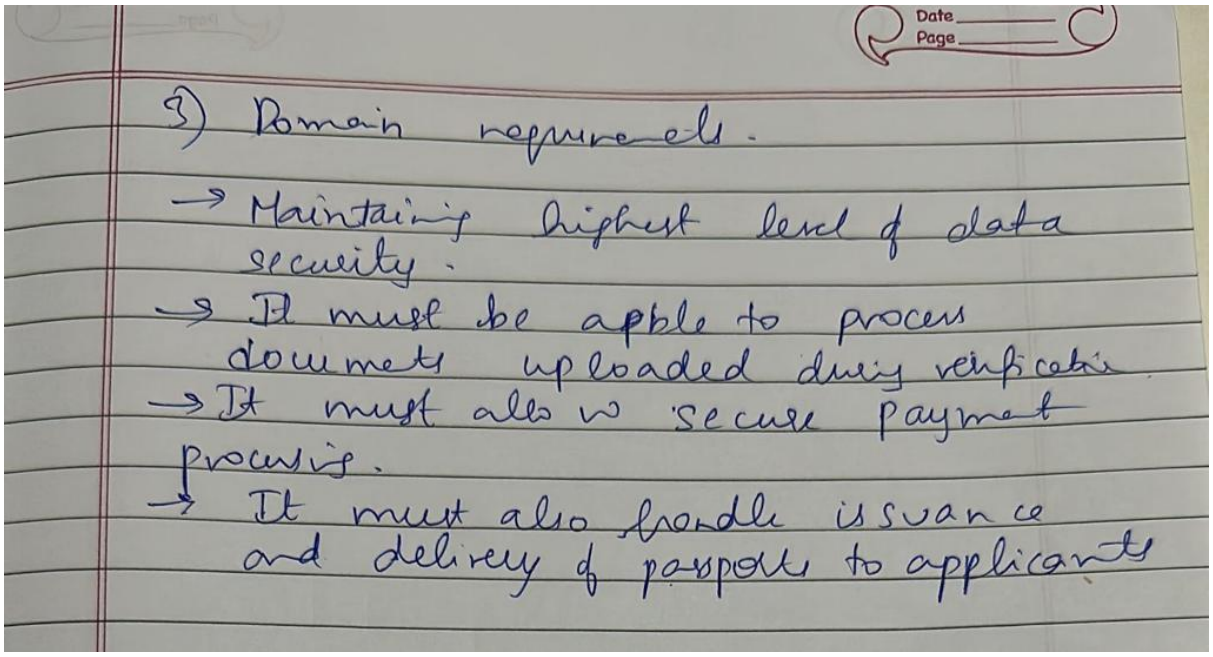
→ They can fill in forms & upload docs.

→ They can perform payment & check application status & select appointed dates.

2) Authority
→ They can view all customer requests & service them.
→ They can see history of visa applications & which one are in dispute
→ They can view customer form & perform validation
→ They can accept payment & generate reports

→ System requirements
1) Functional requirements
→ The system must allow account creation & login.
→ It must allow ability to create forms by authority & take input from customer.
→ It must display all fields of passport operation available
→ The ability to track the status of application
→ The system must be able to accept payments
→ It can generate reports

2) Non functional requirements.
→ Security — System must be secure with strong authentication & encryption methanisms.
→ Maintainability — The system must be easy to maintain and update with modular design and well structured code.
→ Reliability — The system must be highly reliable with fault tolerance & error handling.



5.2 Class Diagram

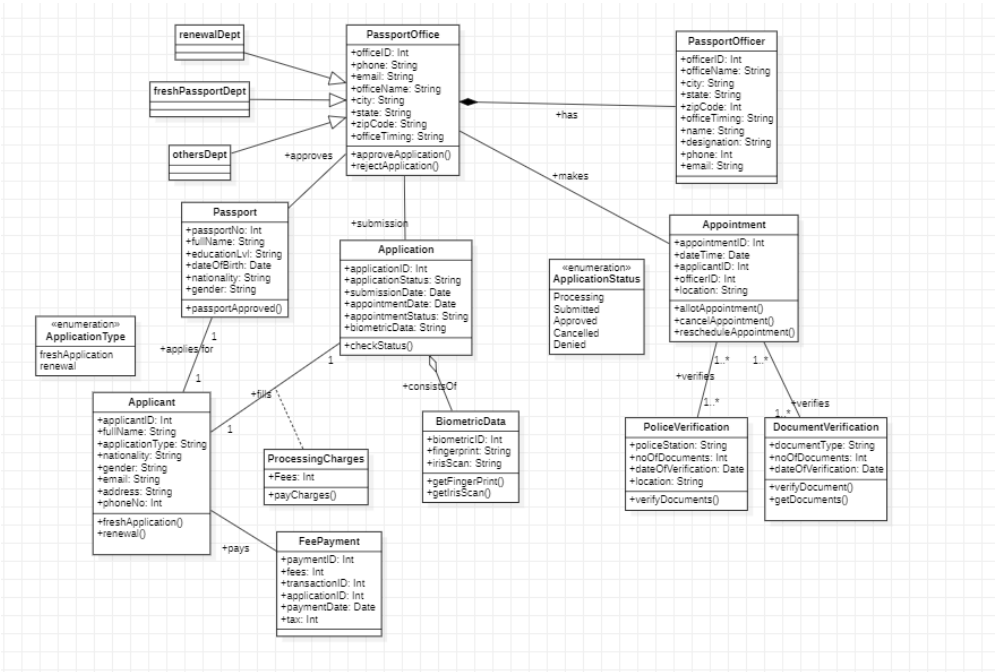


Fig 5.1 – Class Diagram for passport authentication

5.3 State Diagram

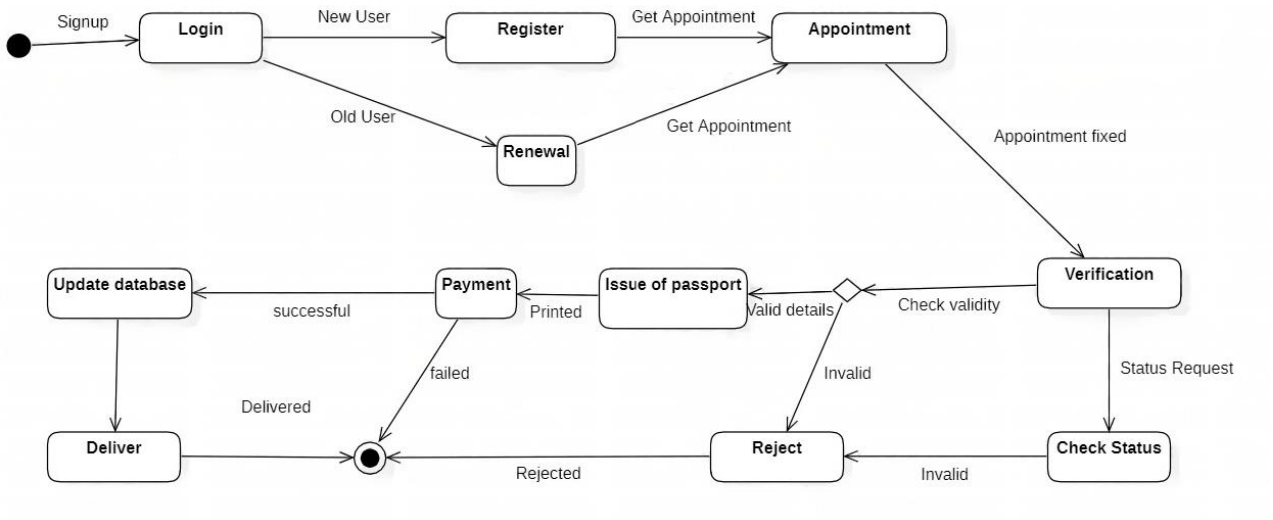


Fig 5.2 – State Diagram for passport authentication

5.4 Interaction Diagram

5.4.1 Use Case Diagram

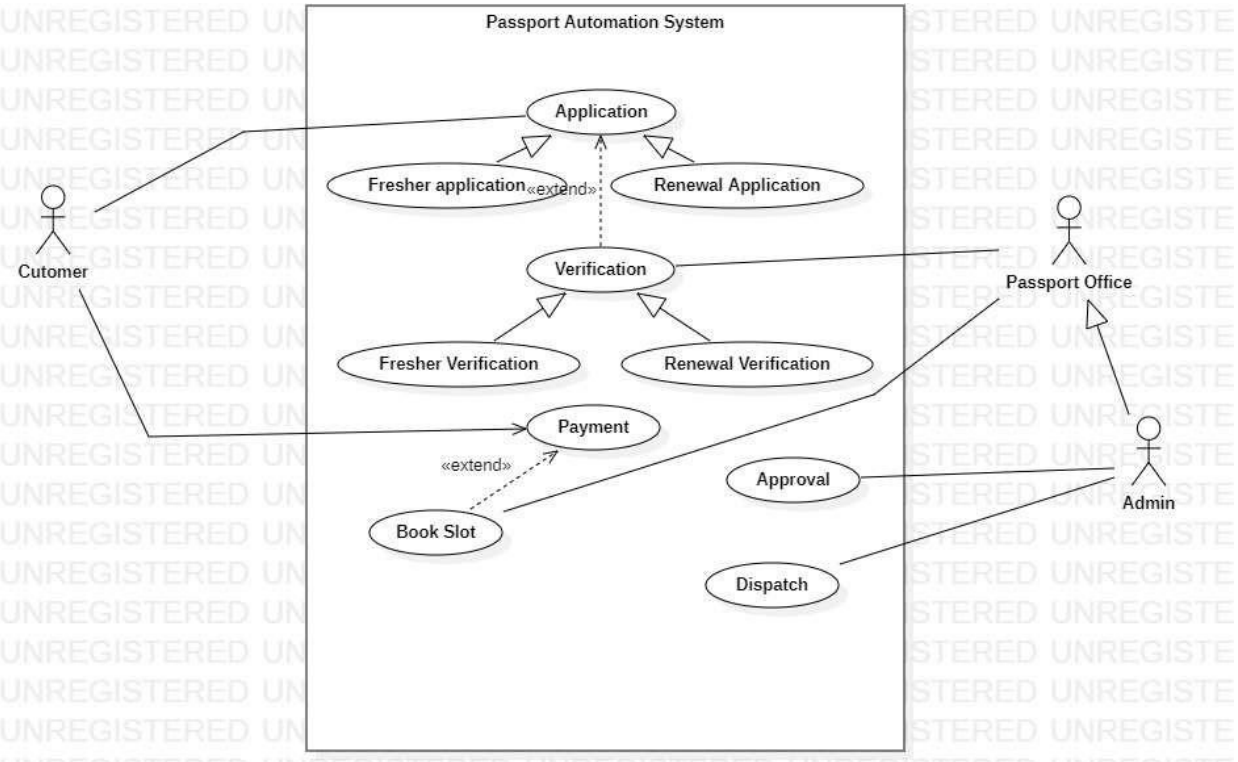


Fig 5.3 – Use Case Diagram for passport authentication

5.4.2 Sequence Diagram

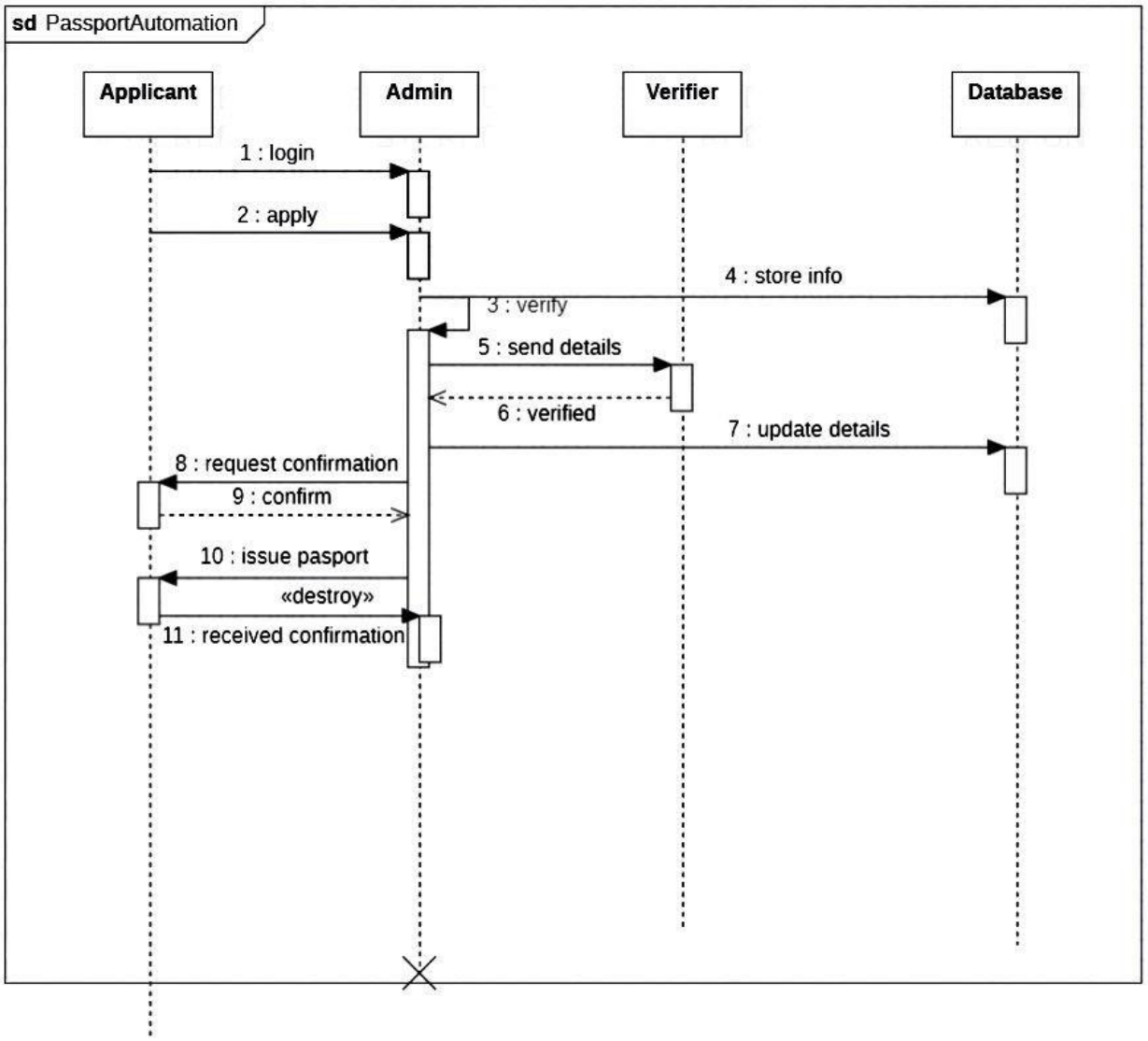


Fig 5.4 – Sequence Diagram for passport authentication

5.4.3 Activity Diagram

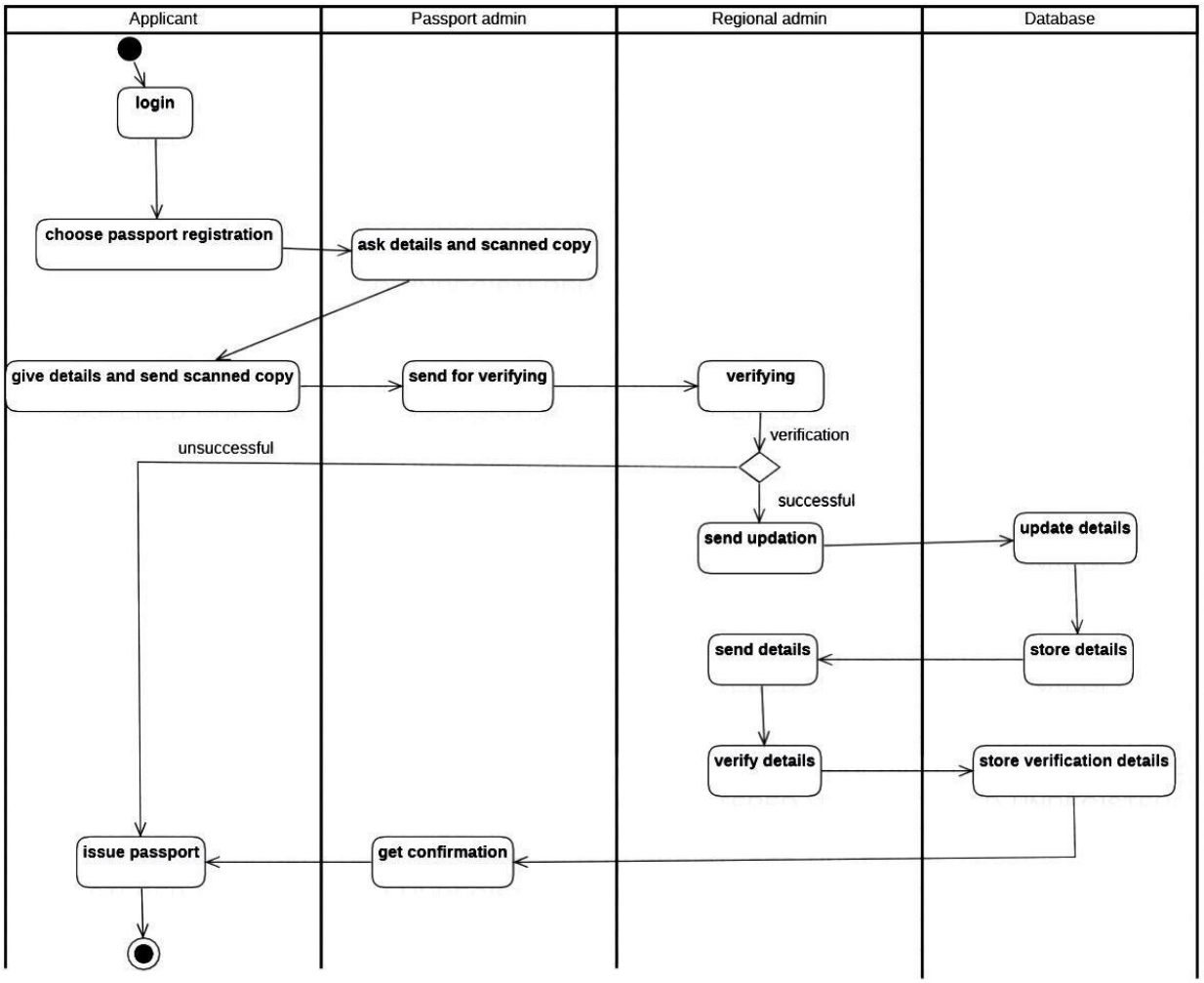
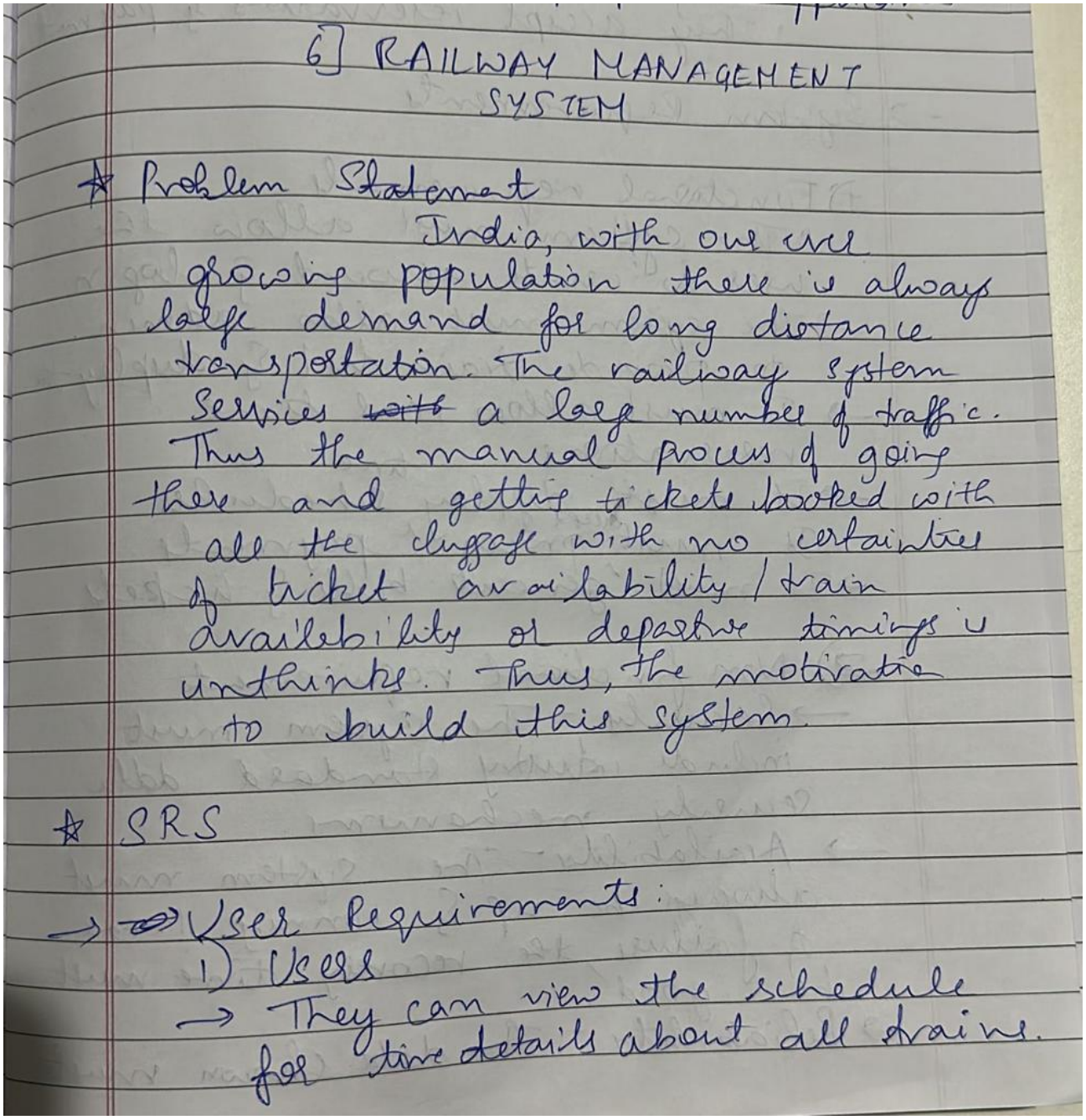


Fig 5.5 – Activity Diagram for passport authentication

6. Railway Reservation System

6.1 Problem Statement



6] RAILWAY MANAGEMENT SYSTEM

★ Problem Statement

India, with our ever growing population there is always large demand for long distance transportation. The railway system services ~~with~~ a large number of traffic. Thus the manual process of going there and getting tickets booked with all the luggage with no certainties of ticket availability / train availability or departure timings is unthinkable. Thus, the motivation to build this system.

★ SRS

→ ⇒ User Requirements:

1) Users

→ They can view the schedule for time details about all trains.

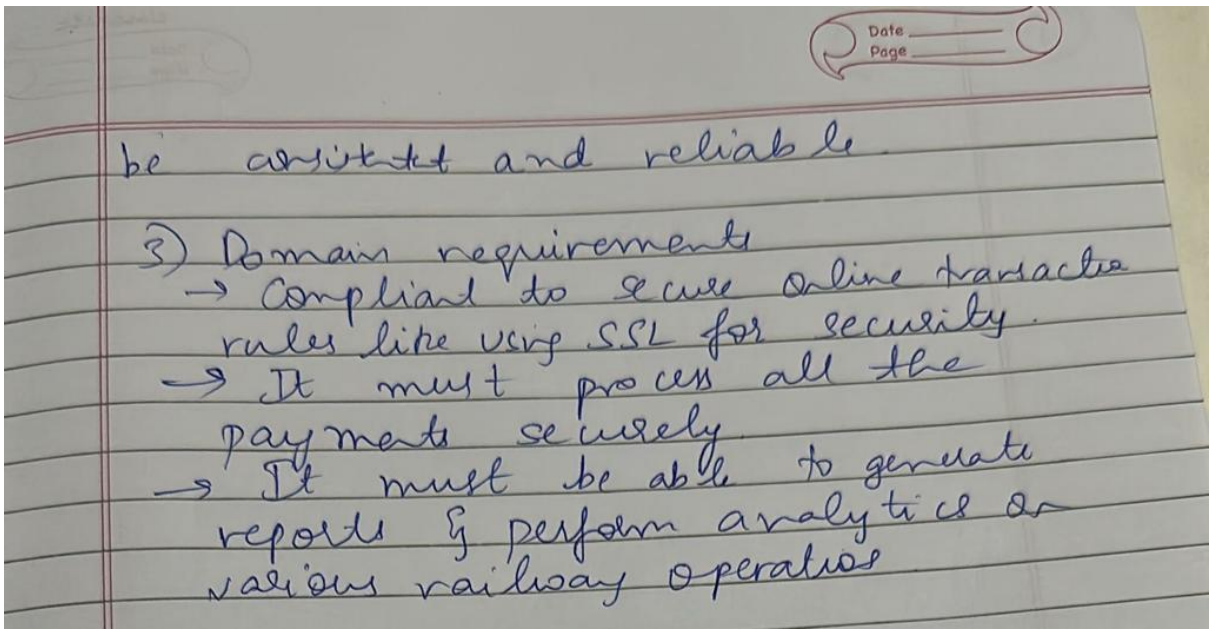
→ They can select trains with the desired compartment & to reserve seats
→ They can then pay for the tickets.
→ They can track the tickets or if possible cancel them before a certain time limit

2) Administrator
→ They provide list of all trains.
→ They set train schedules
→ They accept reservations & payment

→ System Requirements

1) Functional requirements.
→ The system must allow the user to create account & login.
→ The system must allow admin to provide train list & display to
→ It must allow real time reservations.
→ It must display time schedules
→ It must accept & procus payments.
→ It must generate bill & tickets.

2) Non functional requirements
→ Security - The System must include industry standard ddta security mechanisms.
→ Availability - The system must always be up & running. Incase of failure, the recovery time must be good.
→ Reliability - The data shown must



6.2 Class Diagram

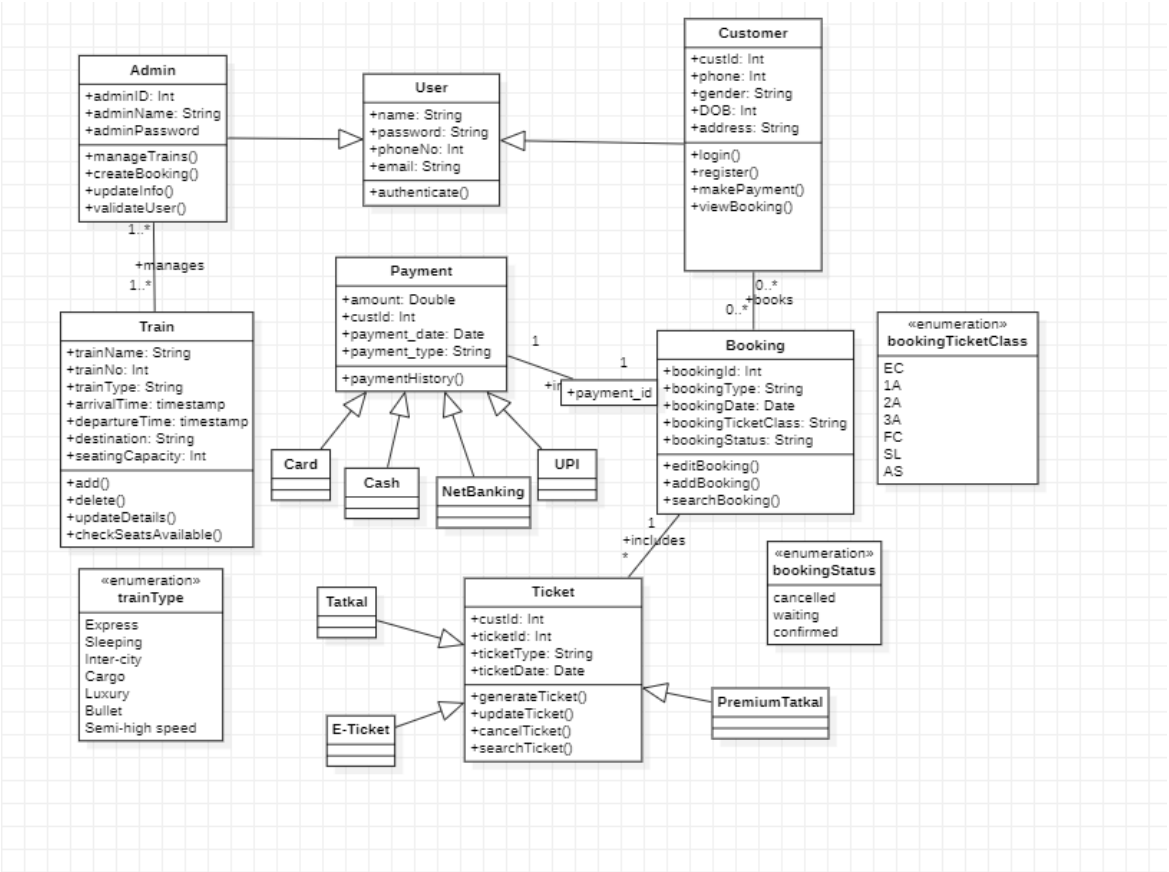


Fig 6.1 – Class Diagram for railway reservation

6.3 State Diagram

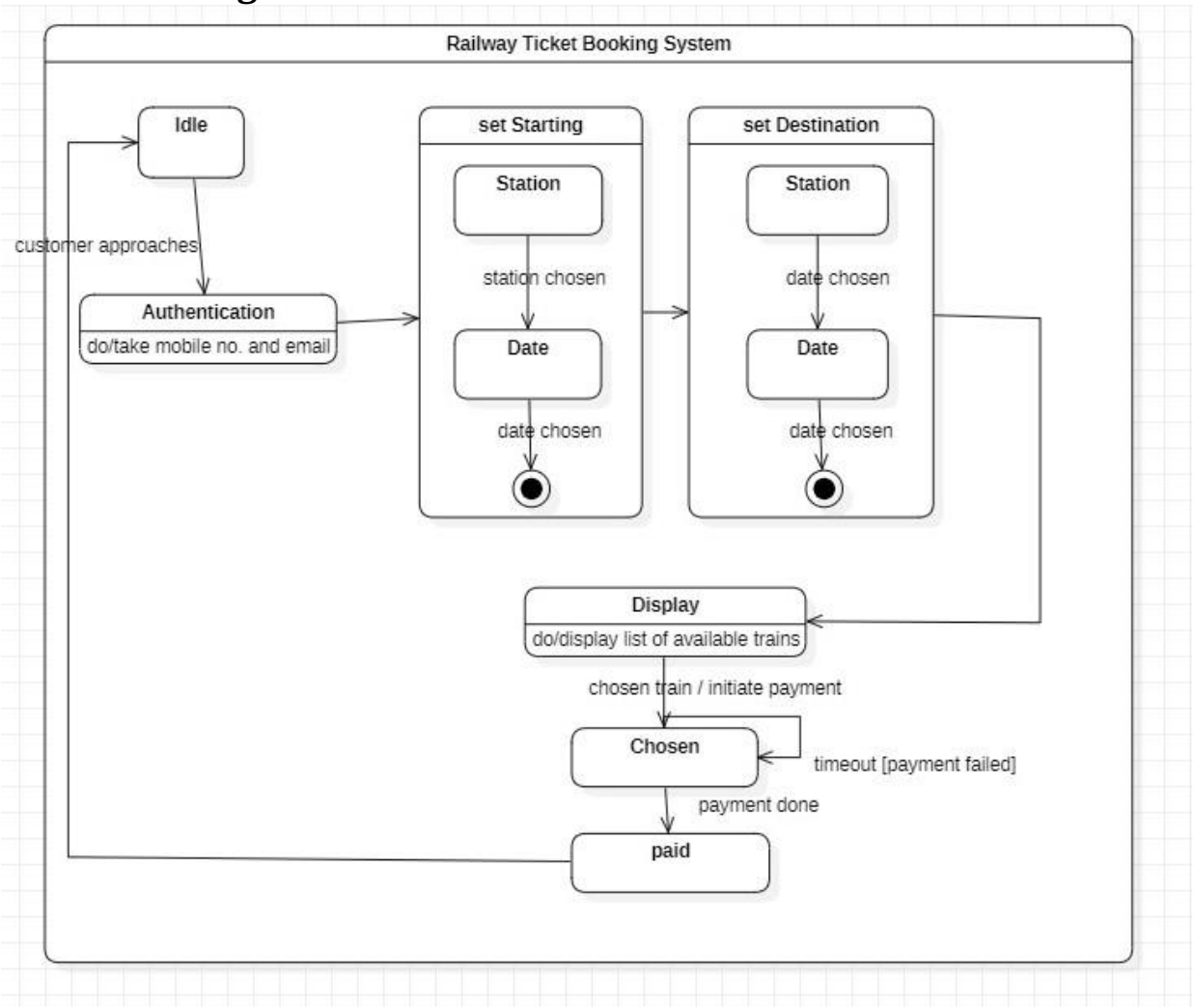


Fig 6.2 – State Diagram for railway reservation

6.4 Interaction Diagram

6.4.1 Use Case Diagram

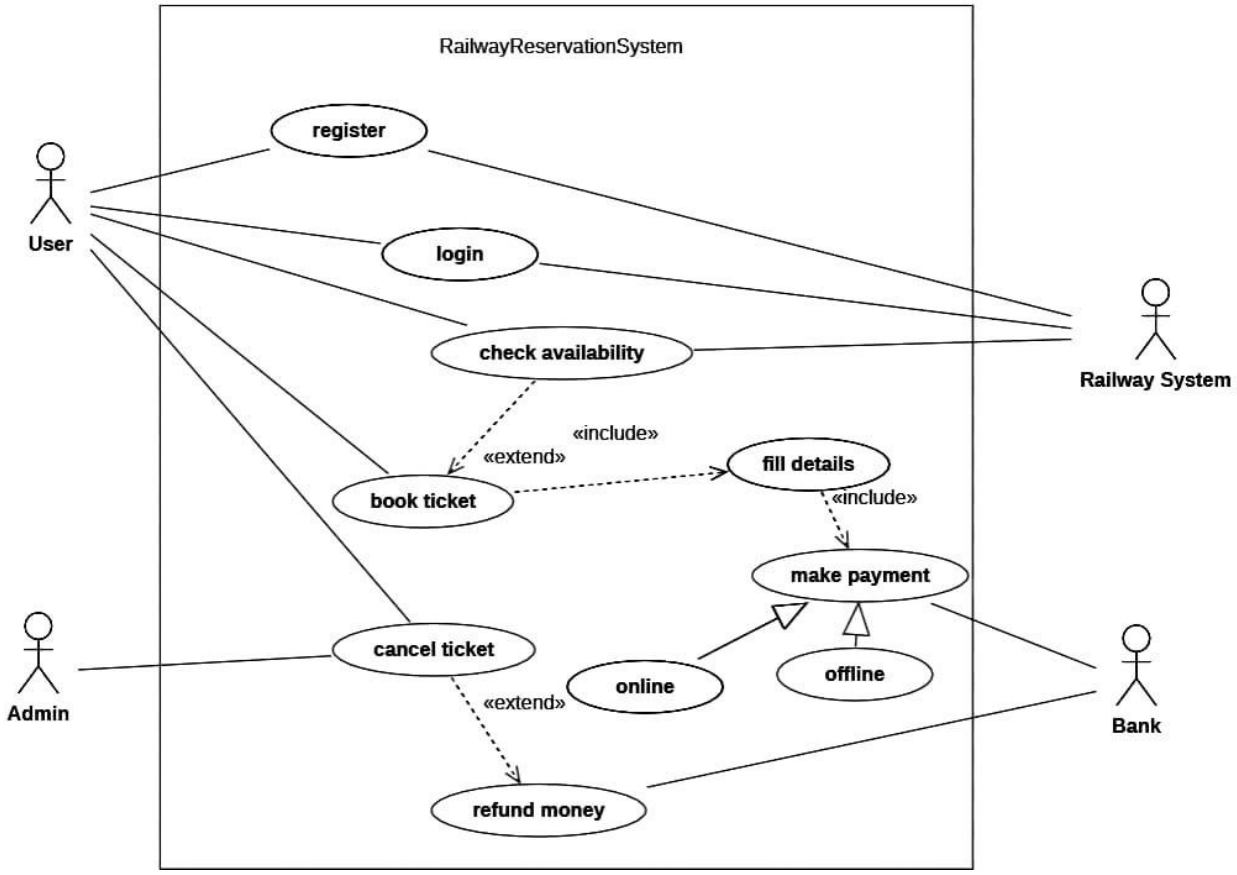


Fig 6.3 – Use Case Diagram for railway reservation

6.4.2 Sequence Diagram

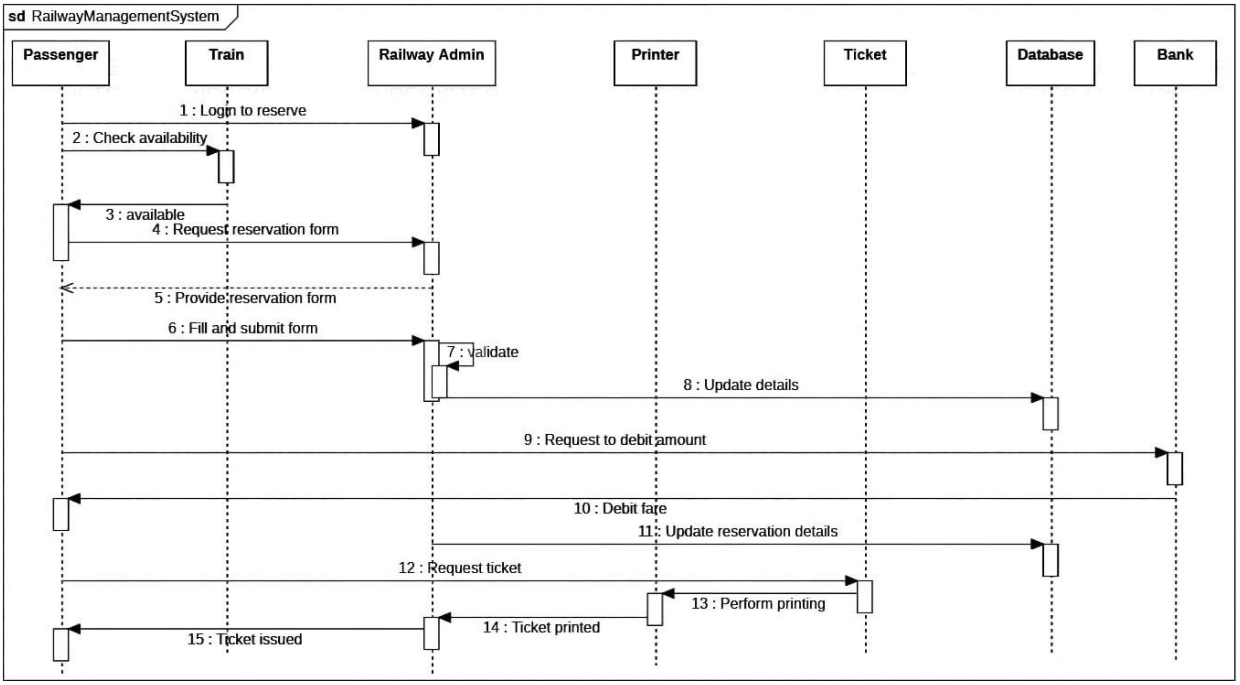


Fig 6.4 – Sequence Diagram for railway reservation

6.4.3 Activity Diagram

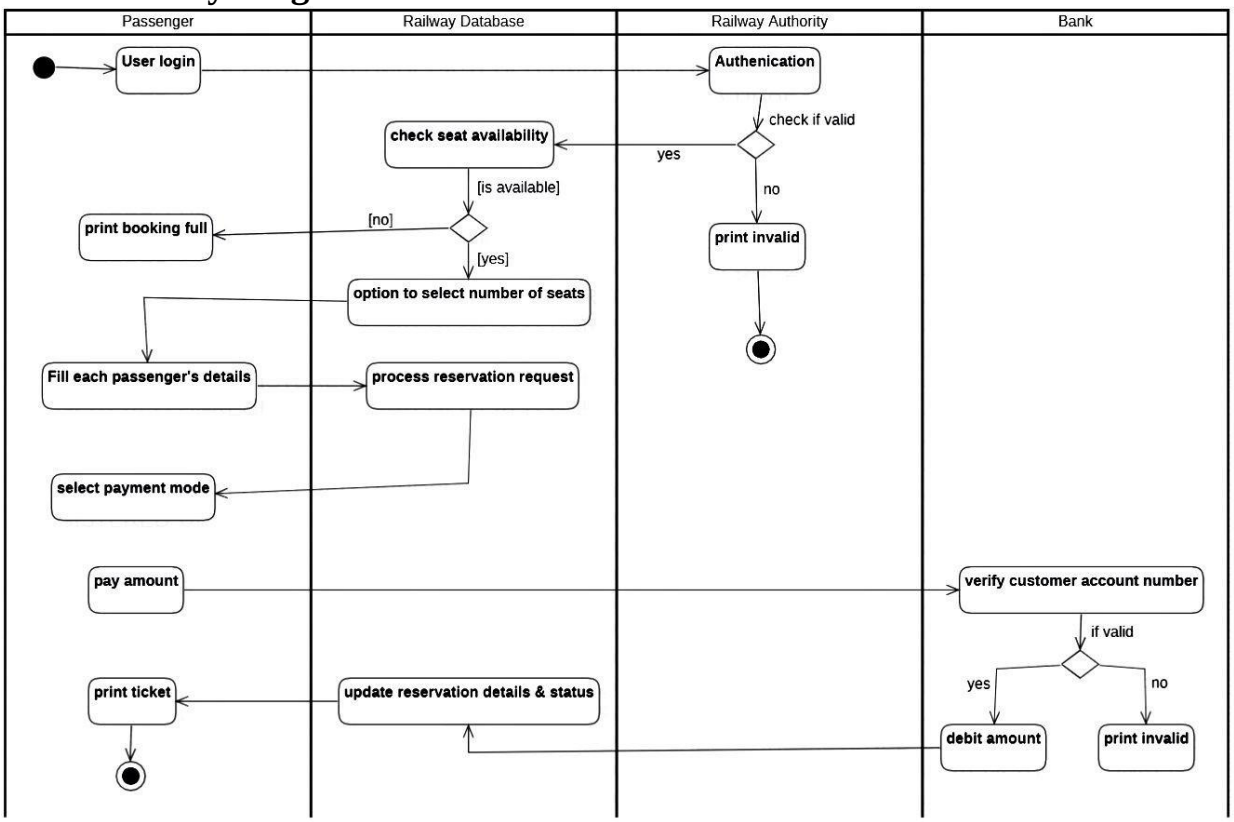
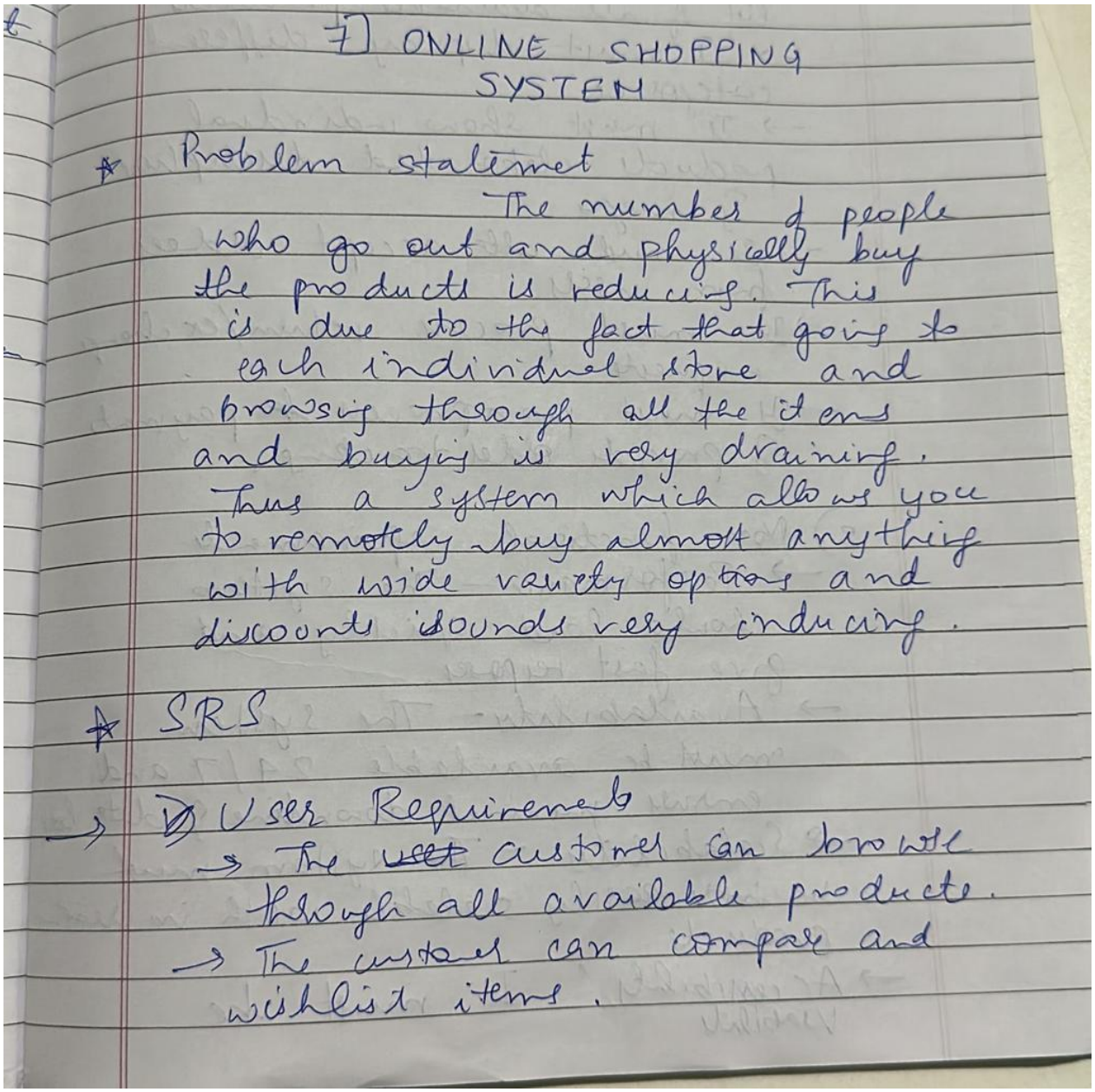


Fig 6.5 – Activity Diagram for railway reservation

7. Online Shopping System

7.1 Problem Statement



7] ONLINE SHOPPING SYSTEM

★ Problem statement

The number of people who go out and physically buy the products is reducing. This is due to the fact that going to each individual store and browsing through all the items and buying is very draining. Thus a system which allows you to remotely buy almost anything with wide variety options and discounts sounds very inducing.

★ SRS

→ ⓓ User Requirements

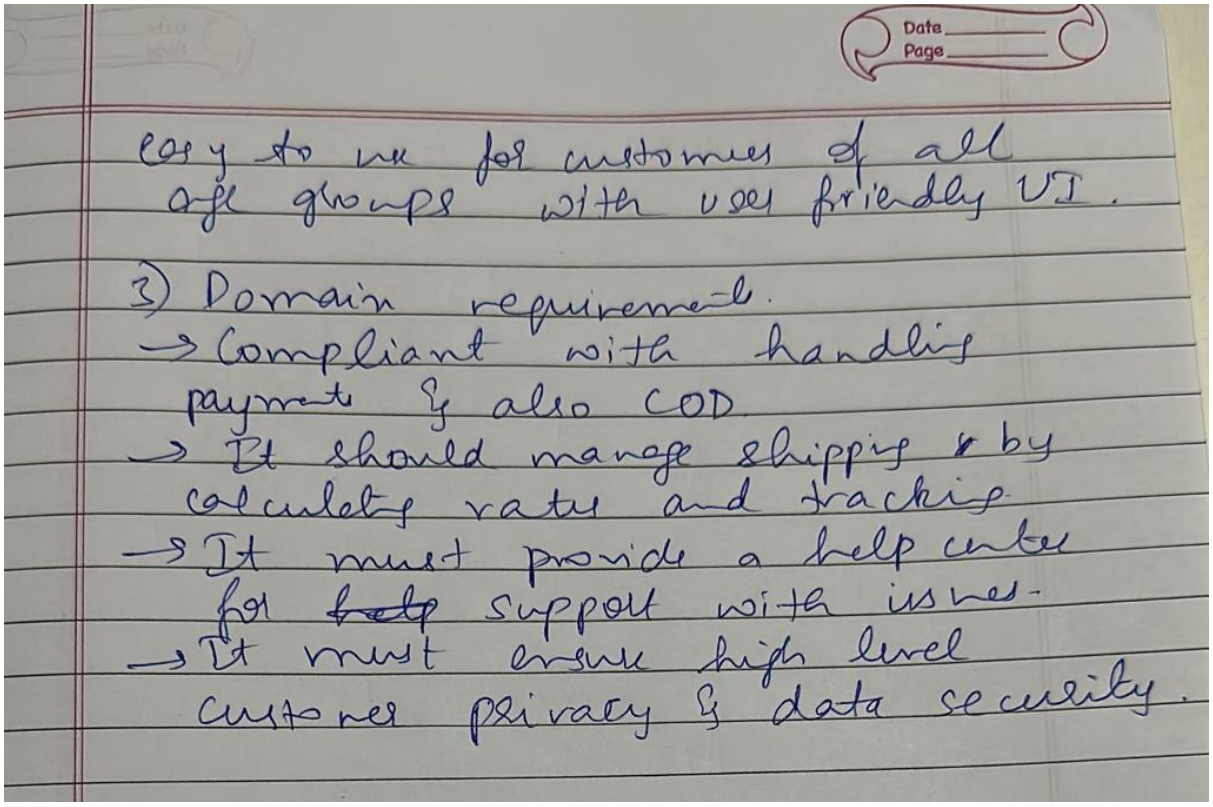
→ The ~~user~~ customer can browse through all available products.

→ The customer can compare and wishlist items.

→ They can select items to buy and finalize payment.
→ They can track orders and give ratings
→ They can also perform returns & exchanges if applicable

→ System requirements
1) Functional requirements
→ The system must display list of all available products.
→ It must display different categories.
→ It must show individual products elaborated descriptions & images.
→ It must ~~alt~~ accept orders from users.
→ It must process return/exchange requests
→ It must ~~process~~ process ~~to~~ payments & provide order tracking.

2) Non functional requirements
→ Performance - The system should load quickly and give fast responses.
→ Availability - The system must be available 24/7 and ensure minimal downtime & data loss
→ Scalability - The system must be ~~int~~ easily scalable with increasing products.
→ Accessibility/ - It must be
Usability



7.2 Class Diagram

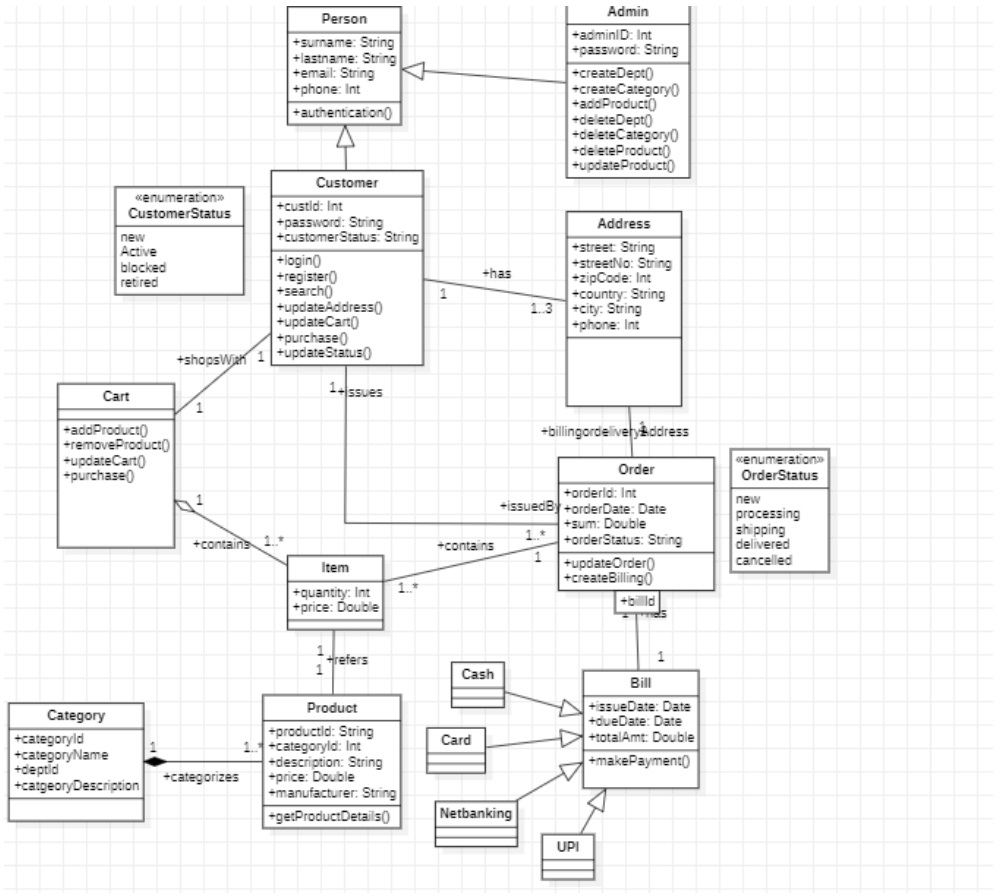


Fig 7.1 – Class Diagram for online shopping system

7.3 State Diagram

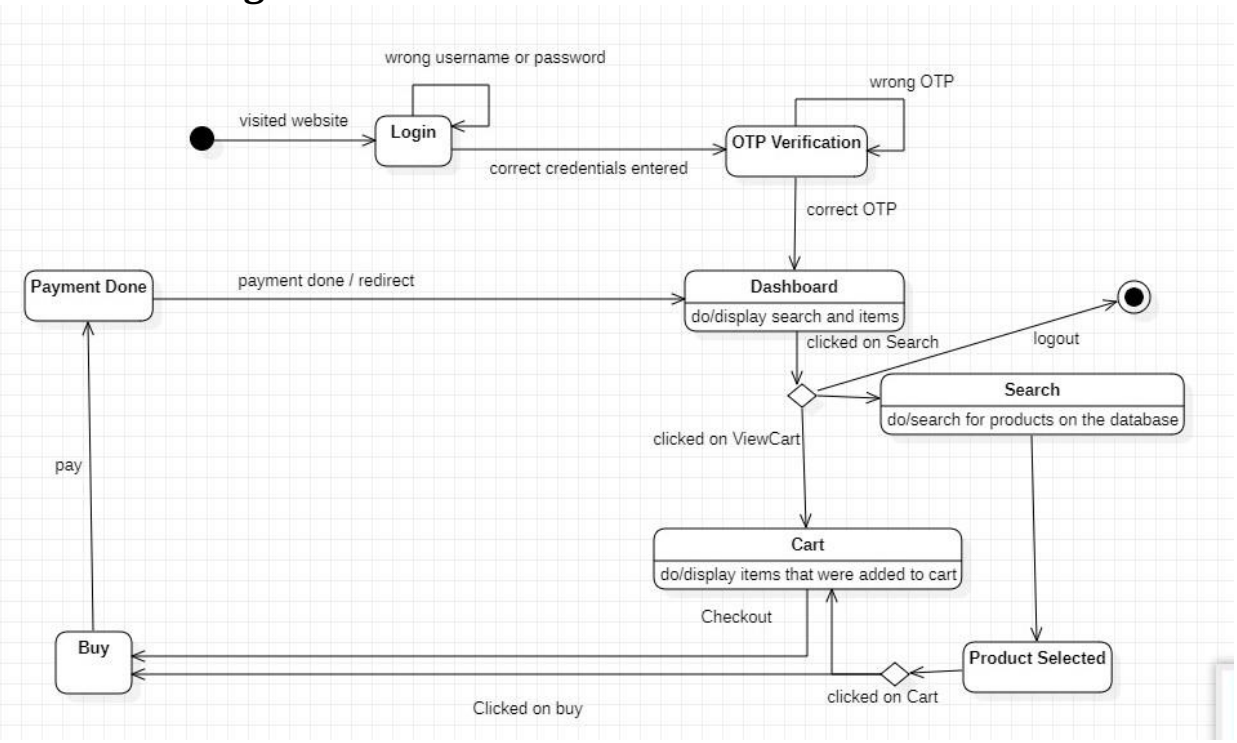


Fig 7.2 – State Diagram for online shopping system

7.4 Interaction Diagram

7.4.1 Use Case Diagram

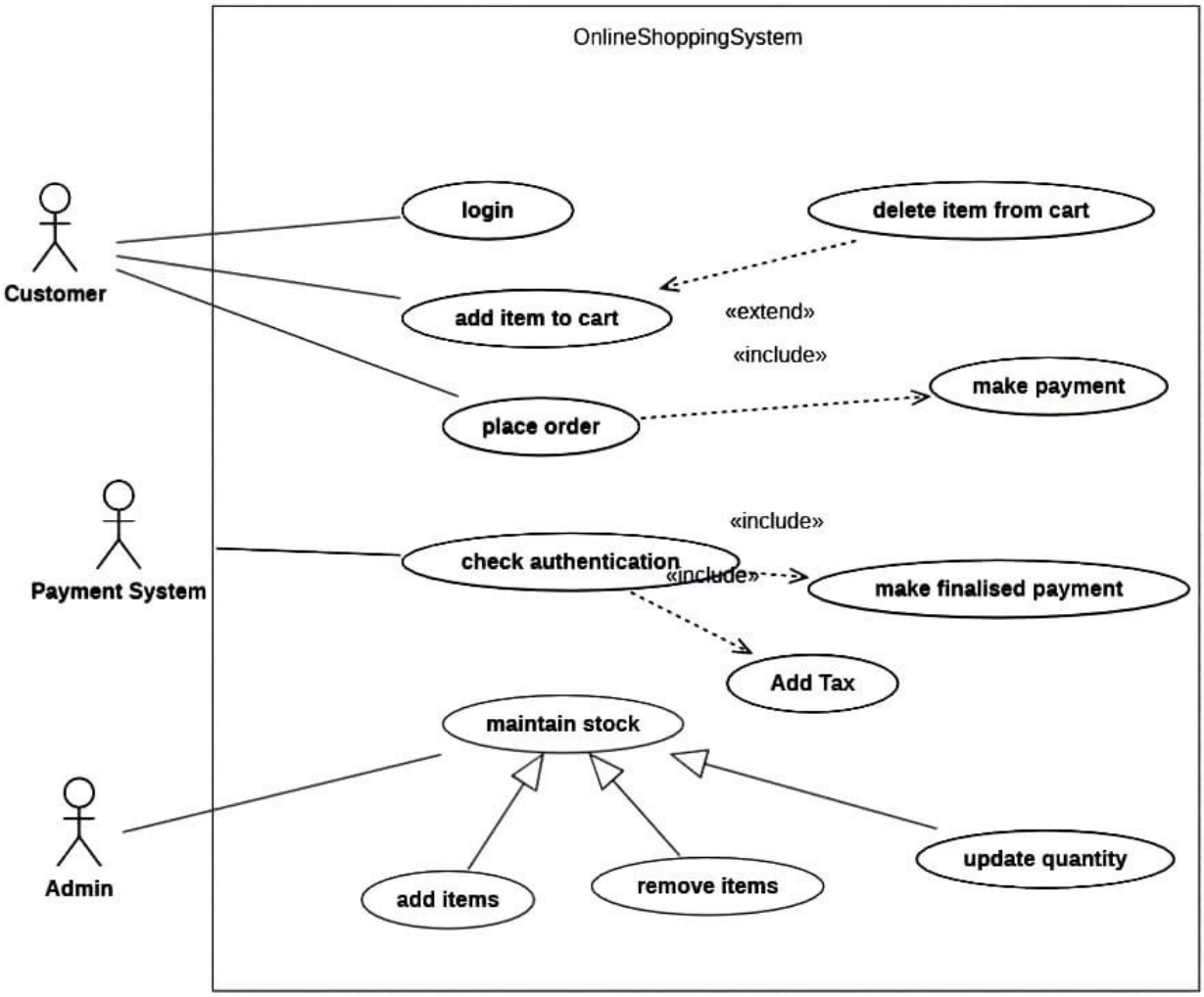


Fig 7.3 – Use Case Diagram for online shopping system

7.4.2 Sequence Diagram

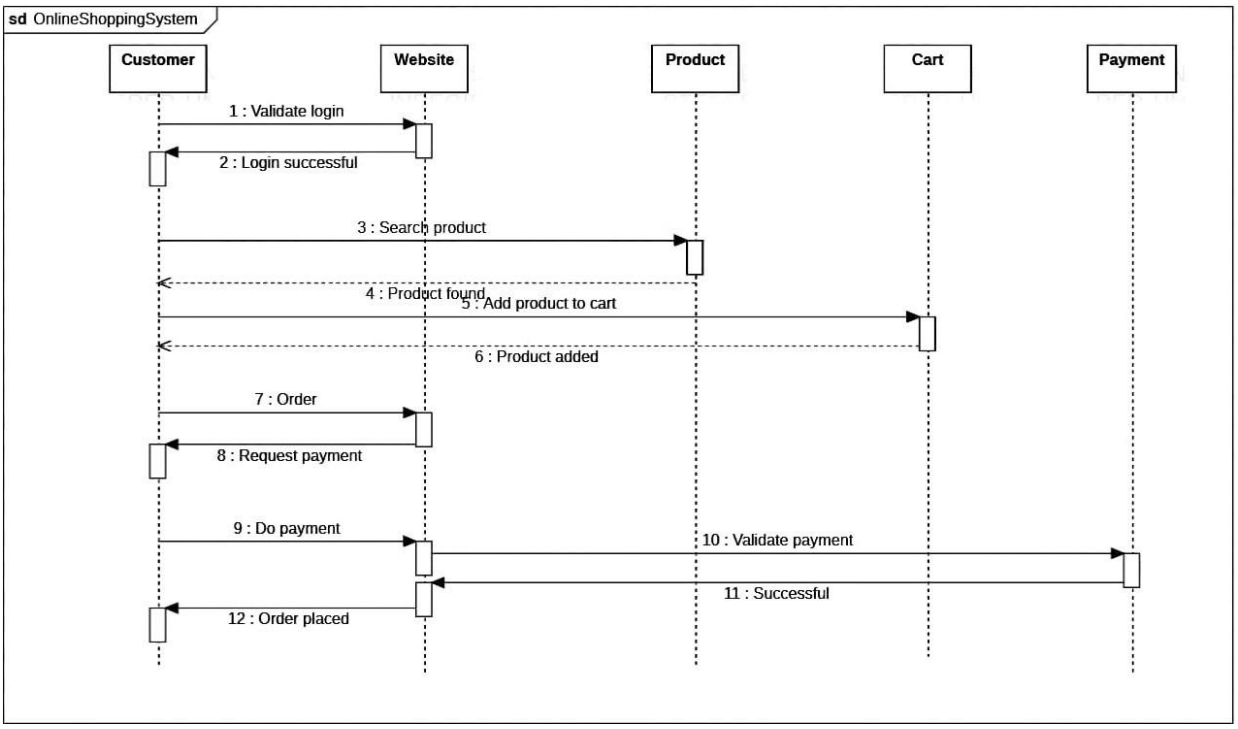


Fig 7.4 – Sequence Diagram for online shopping system

7.4.3 Activity Diagram

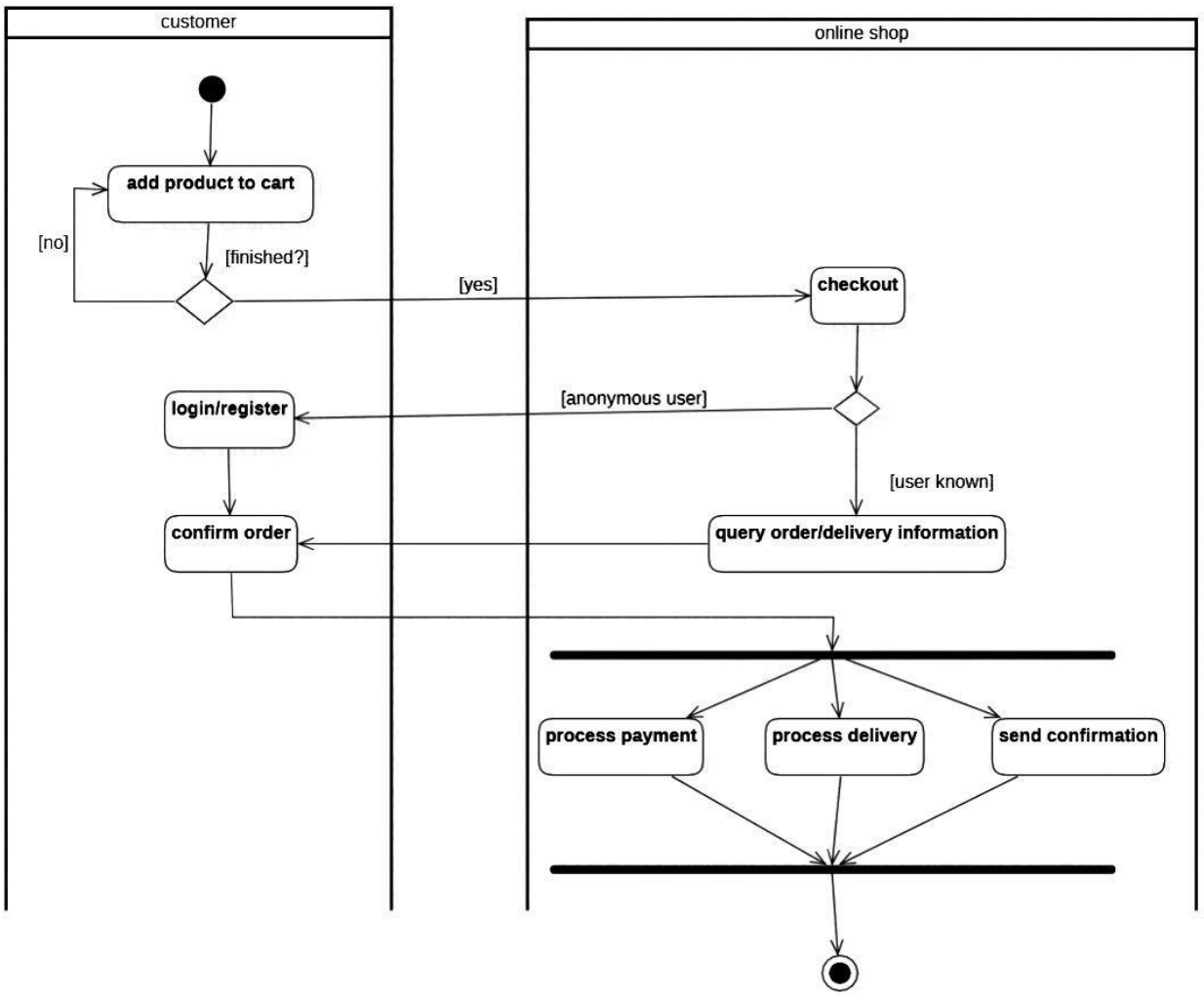


Fig 7.5 – Activity Diagram for online shopping system