

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT

on

Object Oriented Modelling & Design

Submitted by

Mazin Salim (1BM19CS201)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

April-2022 to July-2022

**B. M. S. College of Engineering,
Bull Temple Road, Bangalore 560019**
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Modelling & Design” carried out by **Mazin Salim(1BM19CS201)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the academic year 2021-2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Object Oriented Modelling & Design - (20CS6PCOMD)** work prescribed for the said degree.

Name of the Lab-In charge: Madhavi R.P
Designation: Associate Professor
Department of CSE
BMSCE, Bengaluru

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

,

Index Sheet

Sl. No.	Experiment Title	Page No.
1	College Information system	
2	Hostel Management System	
3	Stock Maintenance System	
4	Coffee Vending Machine	
5	Online Shopping System	
6	Railway Reservation System	
7	Graphic Editor	

Course Outcome

CO4	Ability to conduct practical experiment to solve a given problem using Unified Modeling language.
------------	--

1. College Information System

Problem statement:

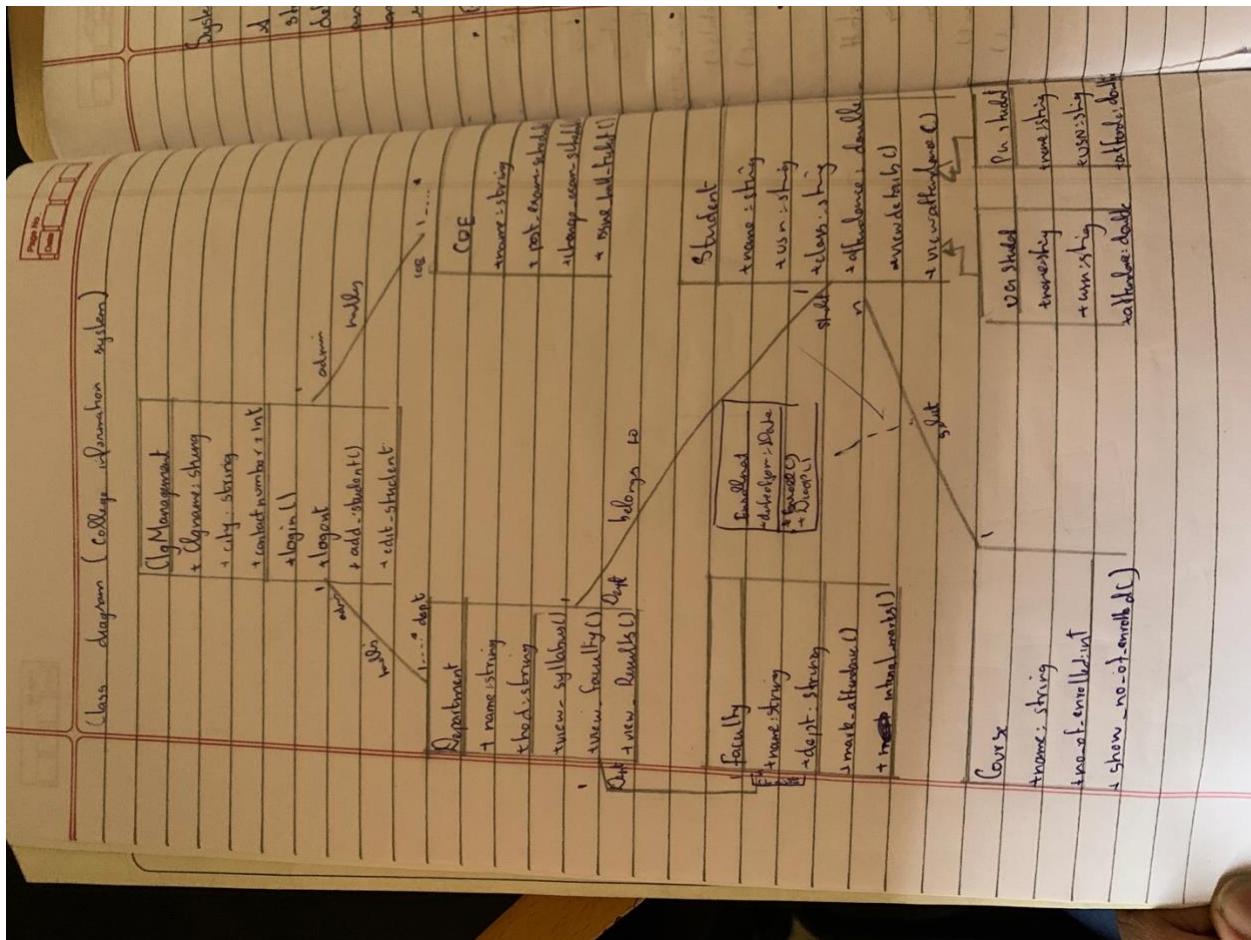
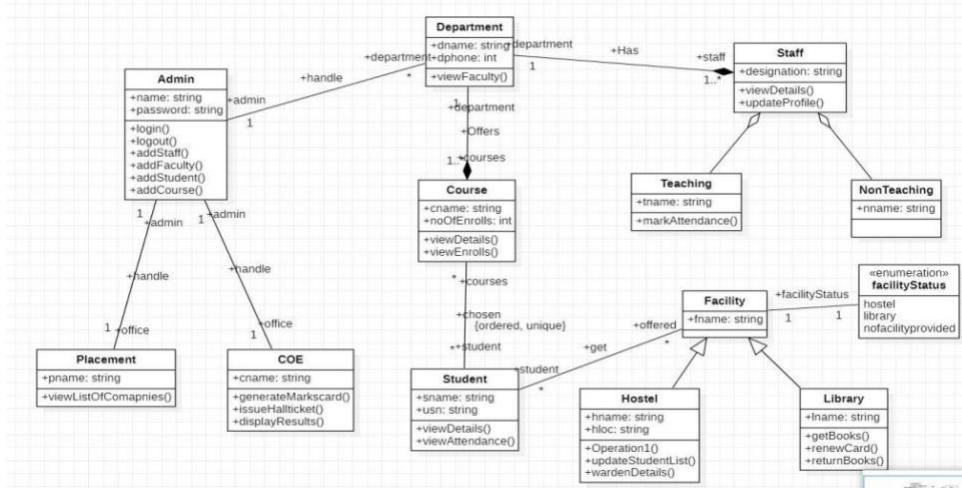
Design UML diagrams for College Information System with system requirements specification.

Software Requirements Specification (SRS):

A centralized approach and system for managing, storing, accessing and updating all the information and details present in relevance to students, and teaching and non-teaching faculty, increasing efficiency and convenience of information management in educational institutions.

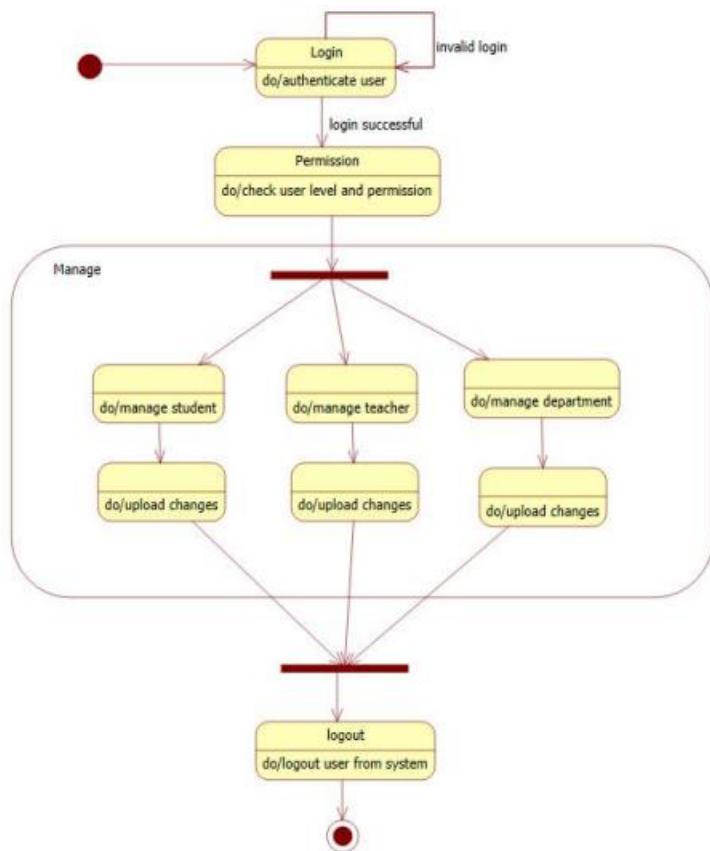
- Educational institutions should be able to add, edit and view student personal details, like name, age, gender, email, phone number, address and so on.
- Educational institutions should be able to add, edit and view student academic details, like USN, department, semester and registered courses.
- Faculty should be able to view all student personal details, and should be able to view and edit internal evaluation marks and attendance of students.
- The COE office should be able to view all student details, and view and edit internal examination marks, and publish results.
- Placement section should be able to view all student details, and add companies coming to the campus for placements.
- Management section should be able to view, add and edit teaching and non-teaching staff details.
- Students should not be allowed to edit their personal or academic details.
- The system should be convenient and easy to use by students, management and faculty.
- The system should be a reliable source of information viewing (most importantly, academic grades) for students, COE and faculty.

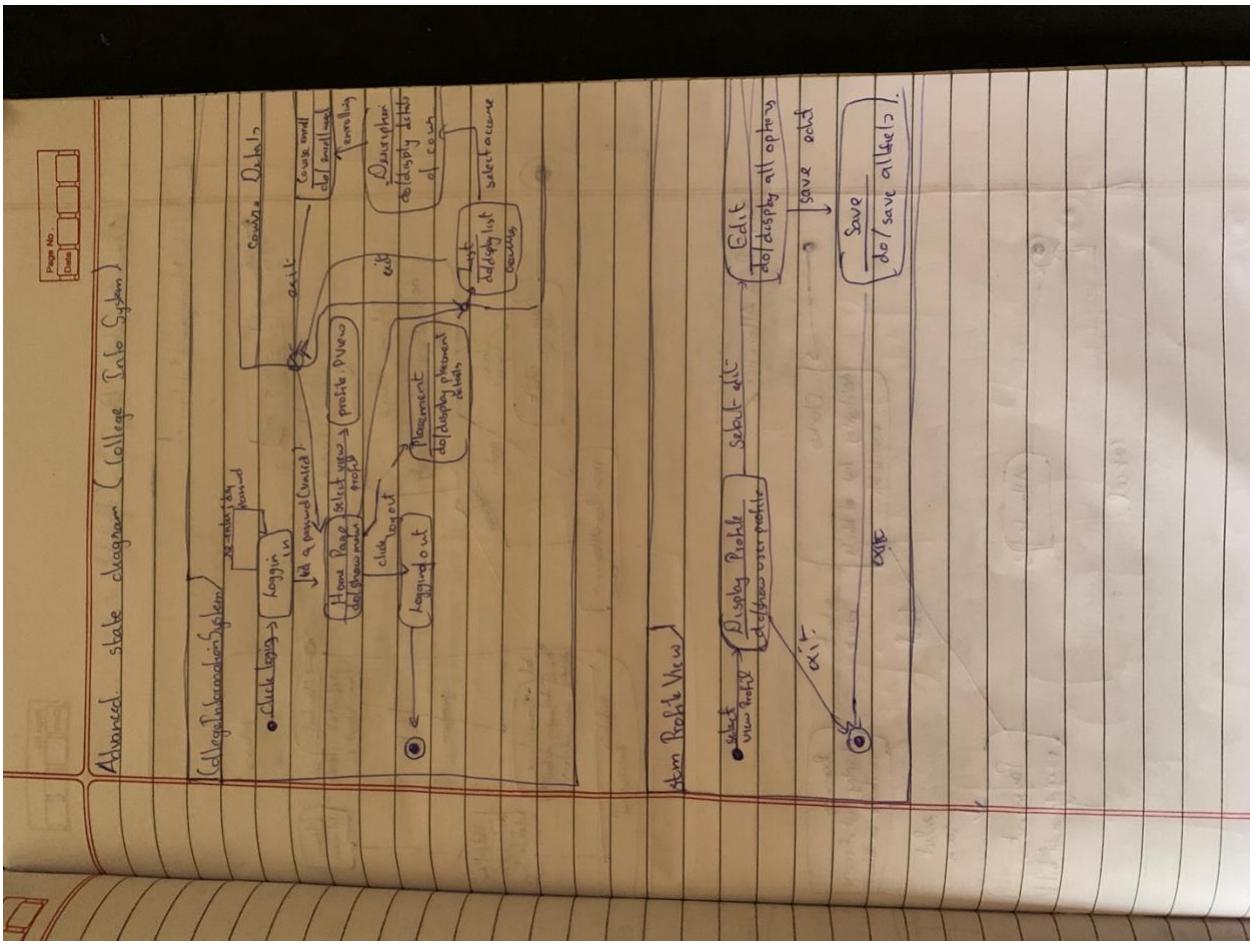
Class Diagram:



State Diagram:

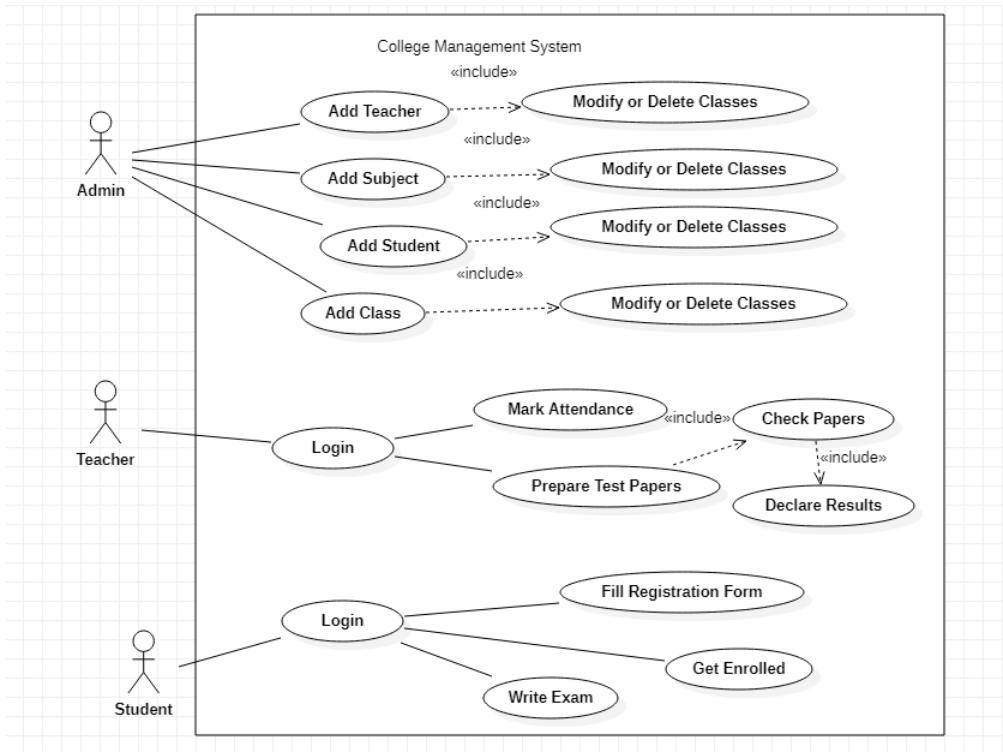
The advanced state diagram depicted below contains one nested state and one submachine, which on successful login shows the course details and profile View procedure of student. It contains initial state and termination state with Courses as a nested state including the required simple states. It also has a submachine state named Profile View with initial, termination state along with simple states; Display profile, Edit, Save.

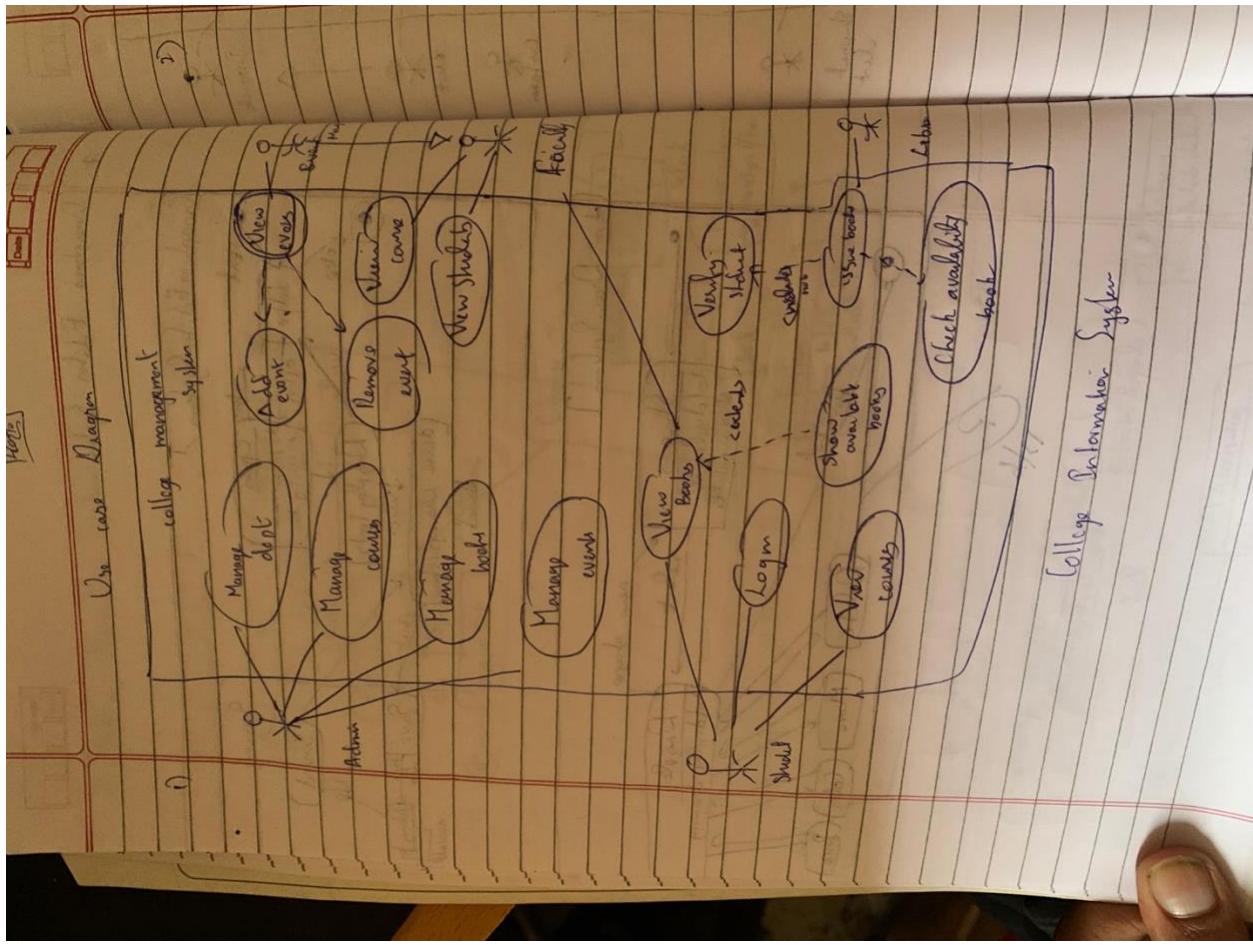




Use Case Diagram:

The advanced use case diagram has extra functionalities which includes extends, includes and generalization. The show available books use case extends view books use case, view events use case includes add events and remove events use case includes verify student.



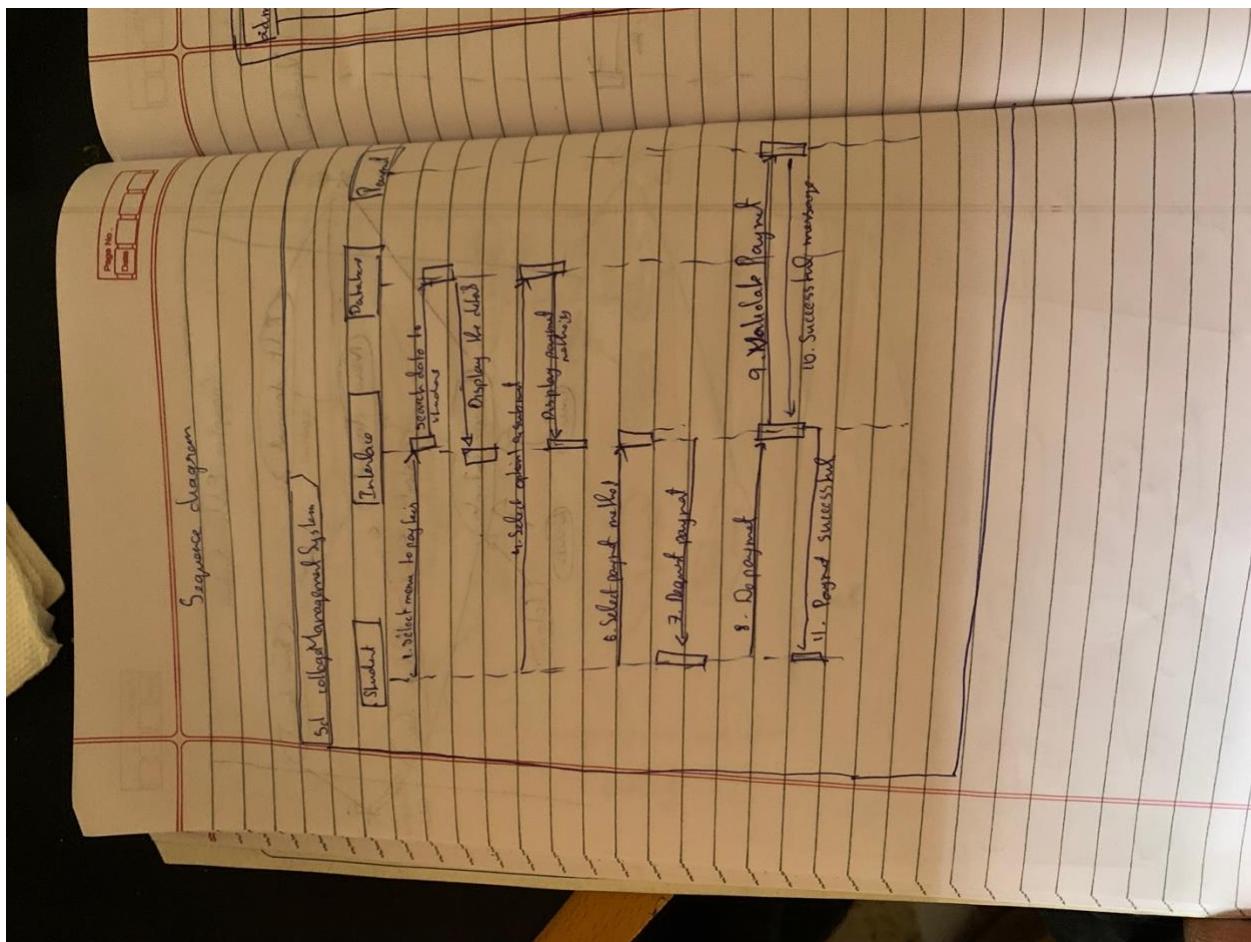
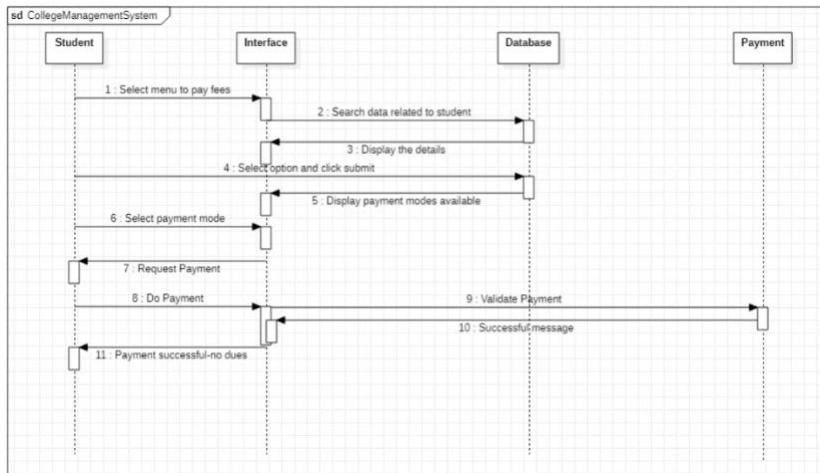


Sequence Diagram:

The lifeline is the dotted line and the rectangles represent the period of time the object is executing and is hence called activation.

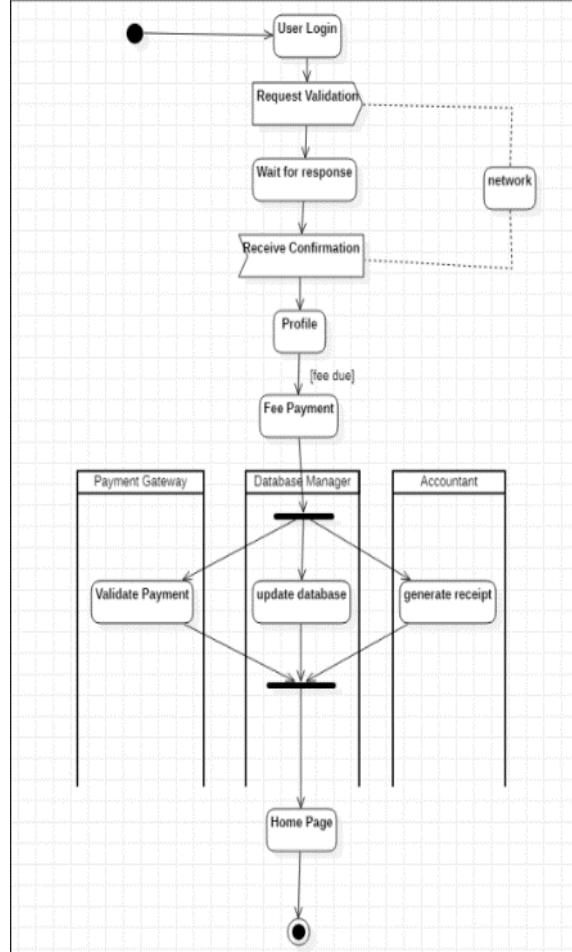
The recursive function of verify is shown by double activation rectangle of verify payment and

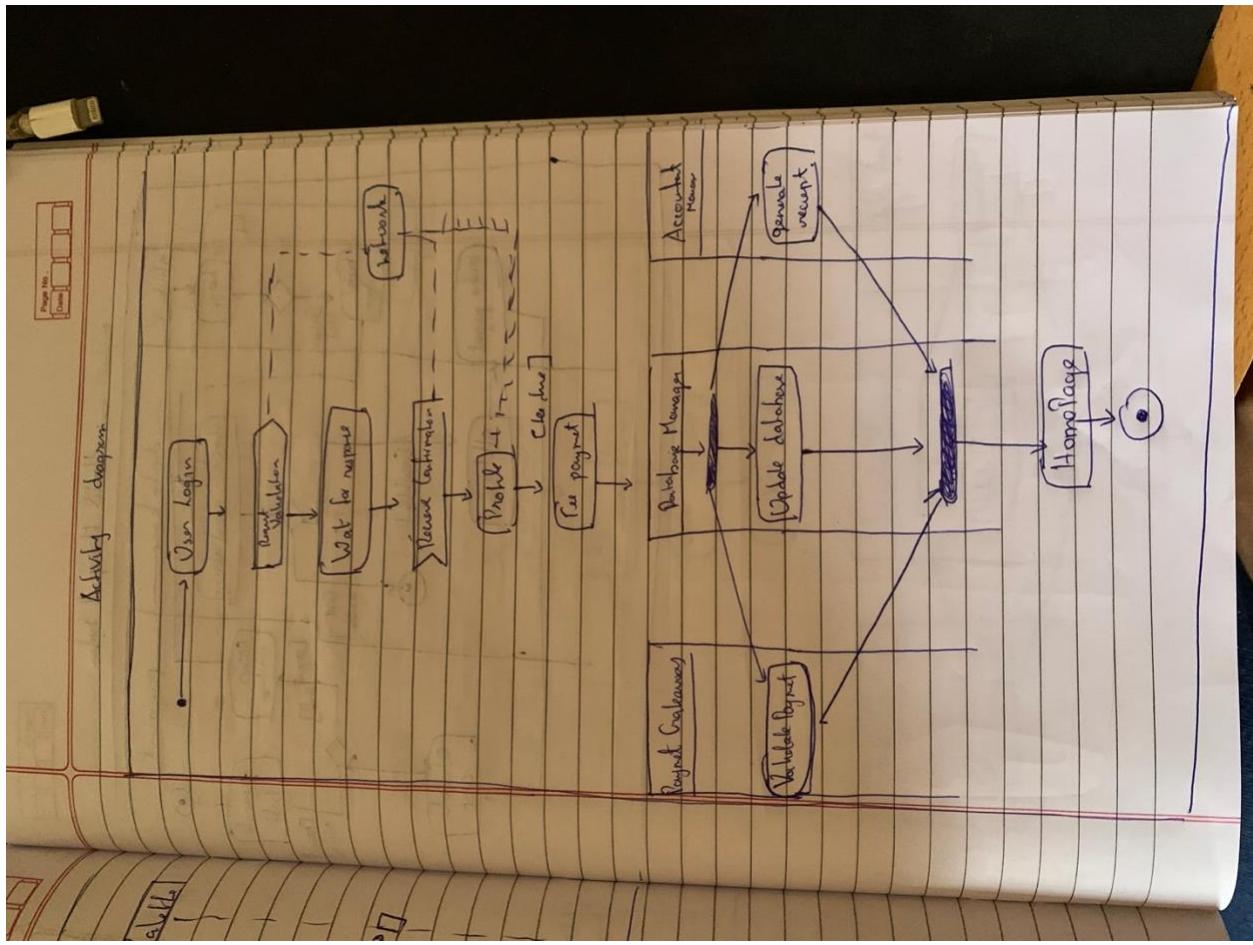
successful message.



Activity Diagram:

The advanced activity diagram starts from initiation and then user login activity where a signal is sent to the network for request validation and upon confirmation the control flows to profile and then fee payment activity. There are three swim lanes for Payment gateway, Database manager and accountant where validate payment, update database and generate receipt respectively. Then the control flows to the home page and then termination activities.





2.Hostel Management System

Problem statement:

Design UML diagrams for Hostel Management System with system requirements specification.

Software Requirements Specification (SRS):

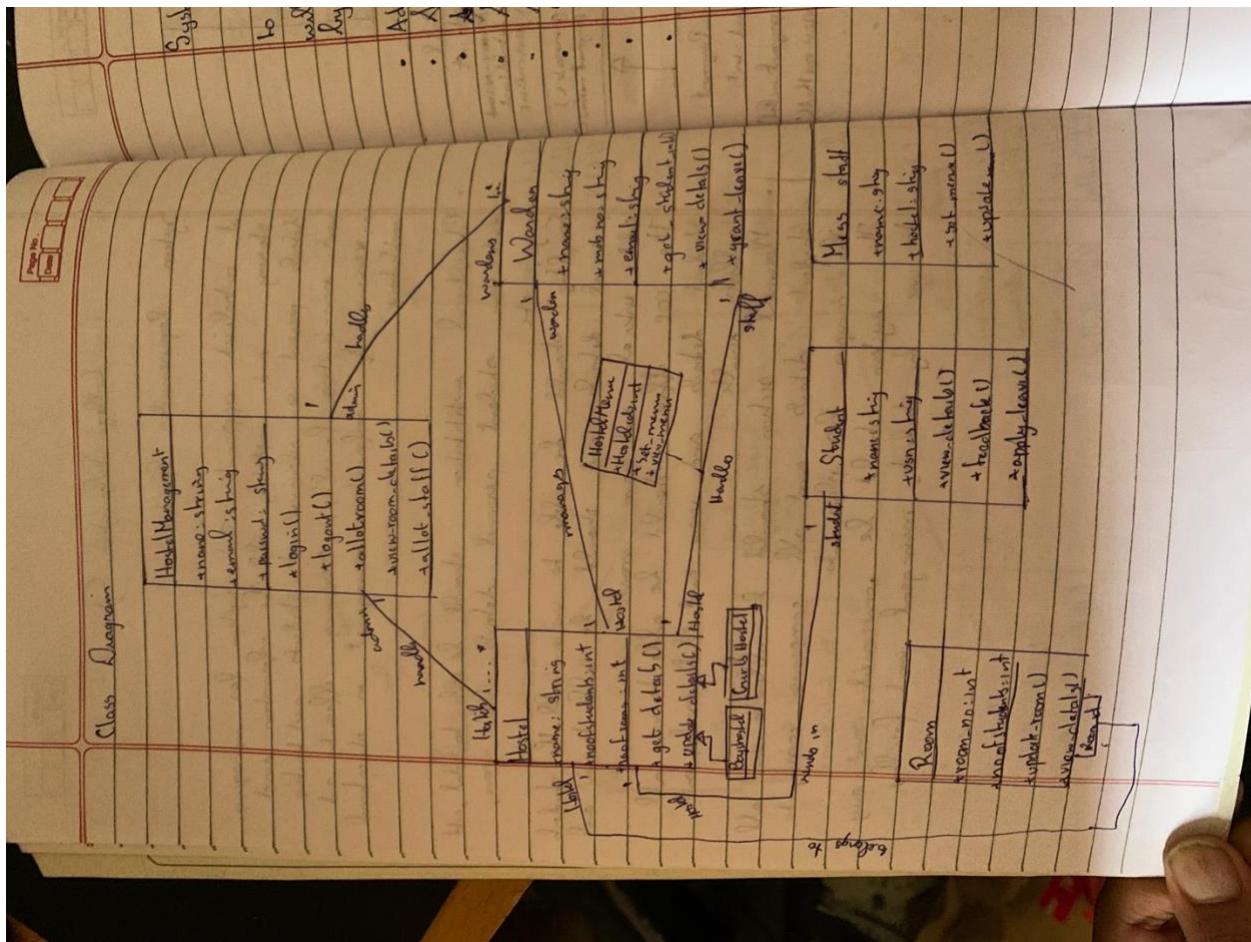
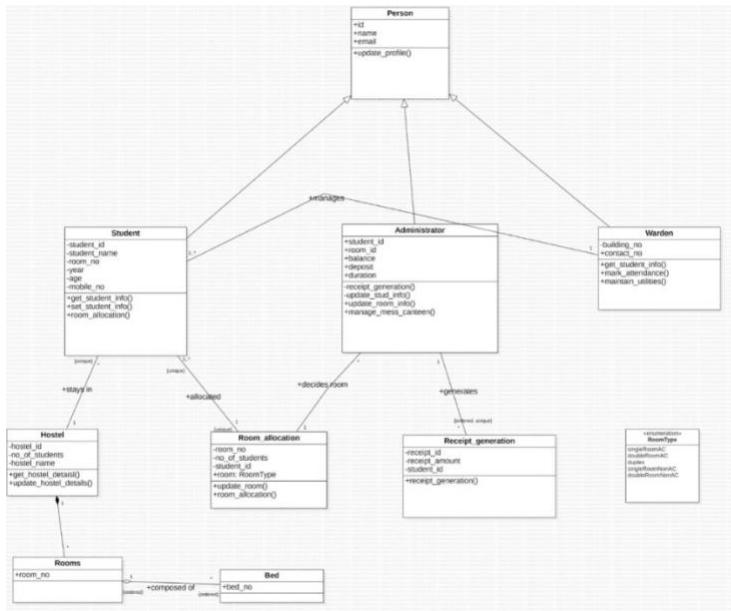
The purpose of the Hostel Management System is to carry out different operations of a hostel.

This system will provide ease of use to the staff of the hostel by performing all work on

computers. It helps to manage student and staff records.

- Admin can login using credentials provided to him.
- Admin can allot room to students.
- Students can login using the credential provided and can give feedback about staff.
- Admin can review the feedback provided by students.
- Admin can appoint staff.
- Students can provide message feedback.
- Mess manager can update the menu list.
- Admin can assign work to staff members.
- The system should be easy to handle.
- System should give expected performance results.
- The response time should be small.

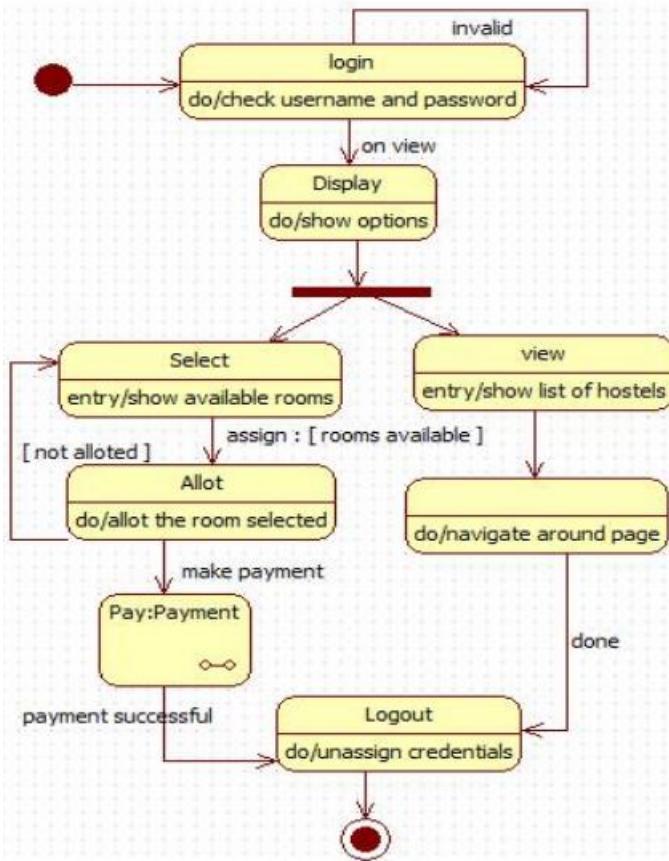
Class Diagram:

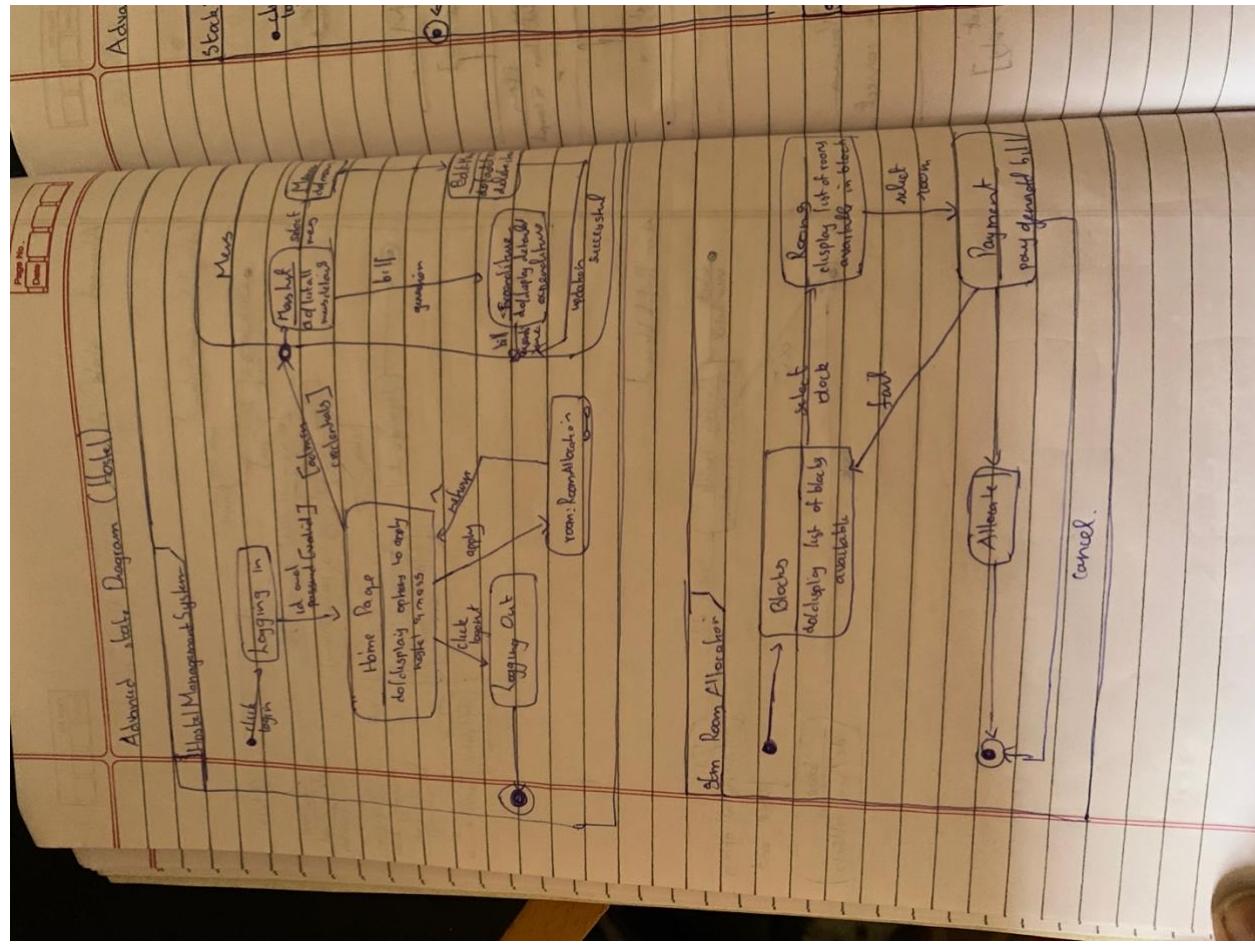


State Diagram:

The advanced state diagram depicted below contains one nested state and one submachine,

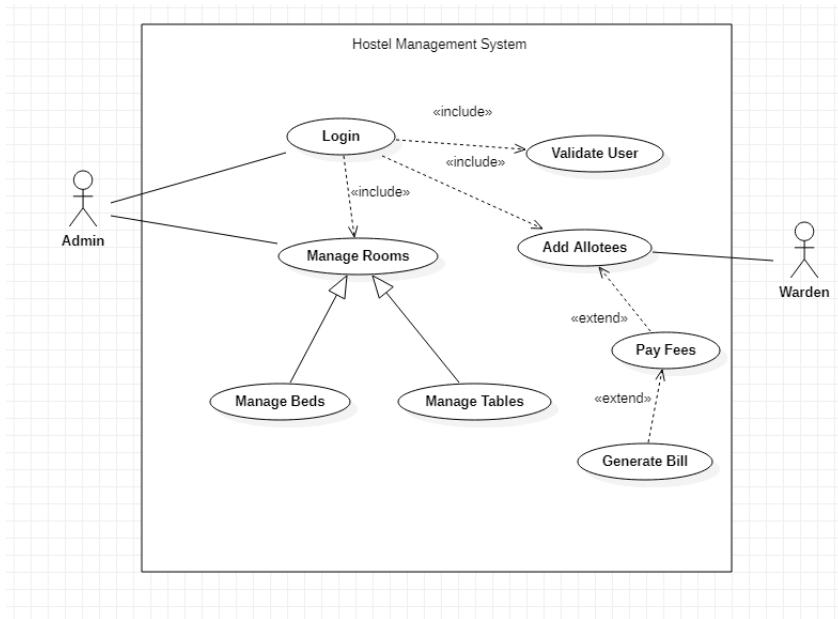
which on successful login shows the mess details and room allocation procedure. It contains initial state and termination state with Mess as a nested state including the required simple states. It also has a submachine state named Room Allocation with initial, termination state along with simple states; Blocks, Rooms, Allocate, Payment.

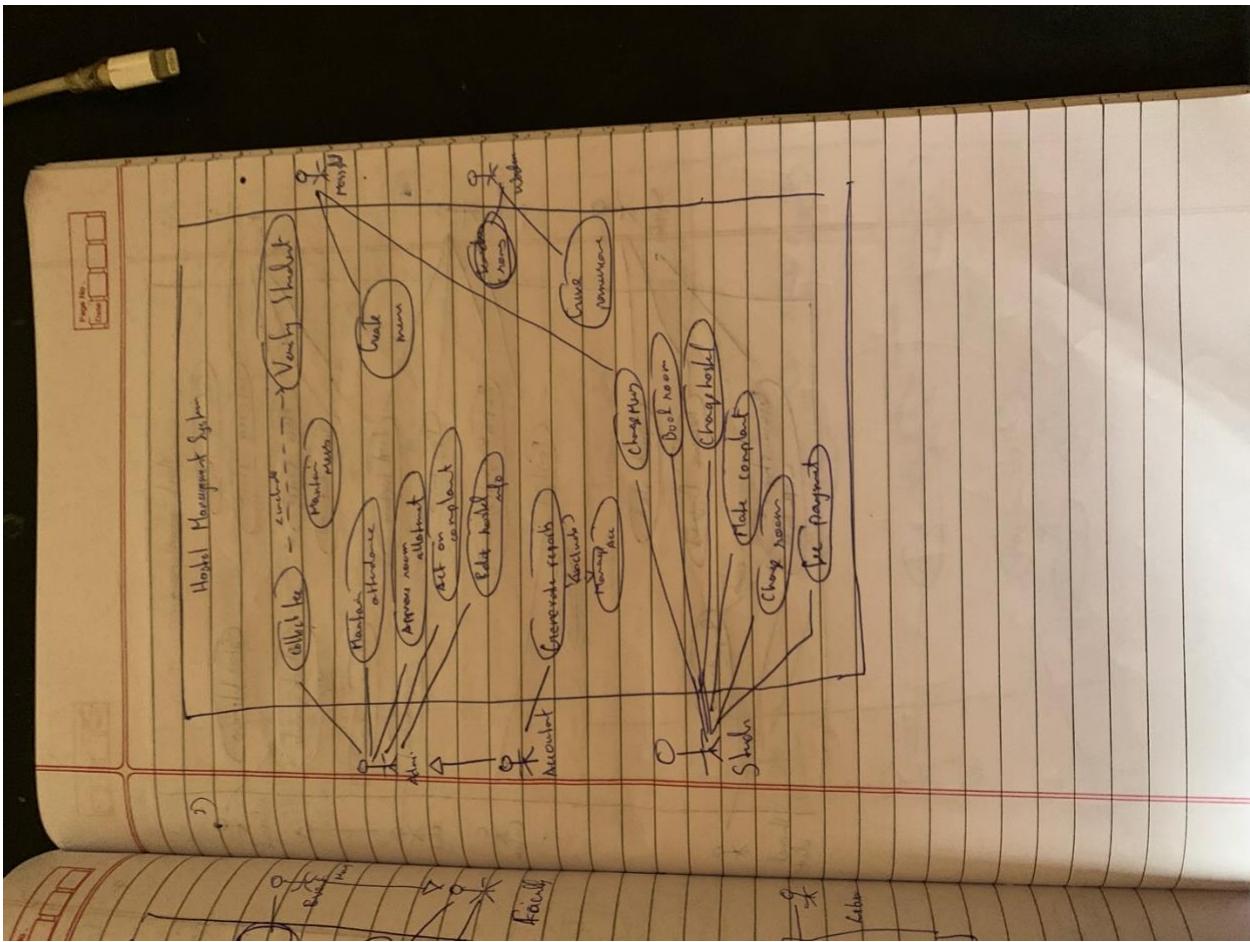




Use Case Diagram:

The advanced use case diagram has extra functionalities which includes extends, includes and generalization. The edit hostel info use case extends add room use case, collect fee use case includes verify student, add room use case includes delete room use case.



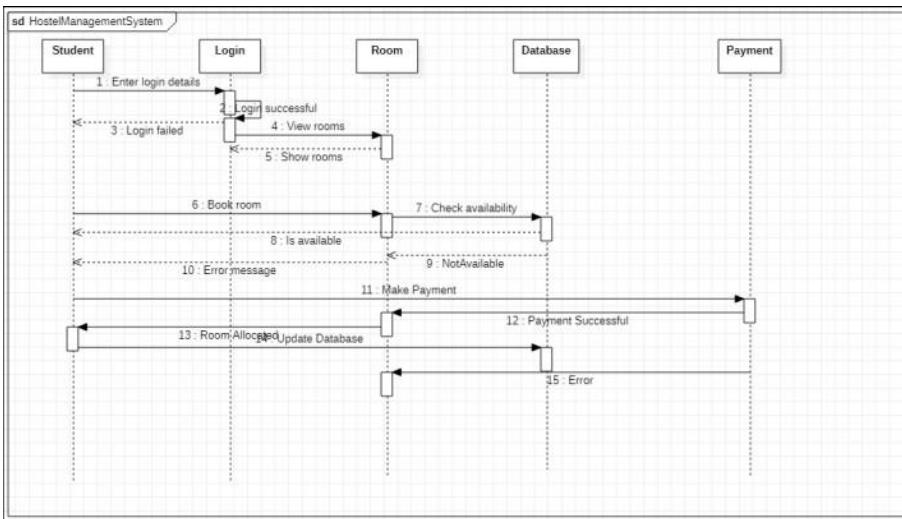


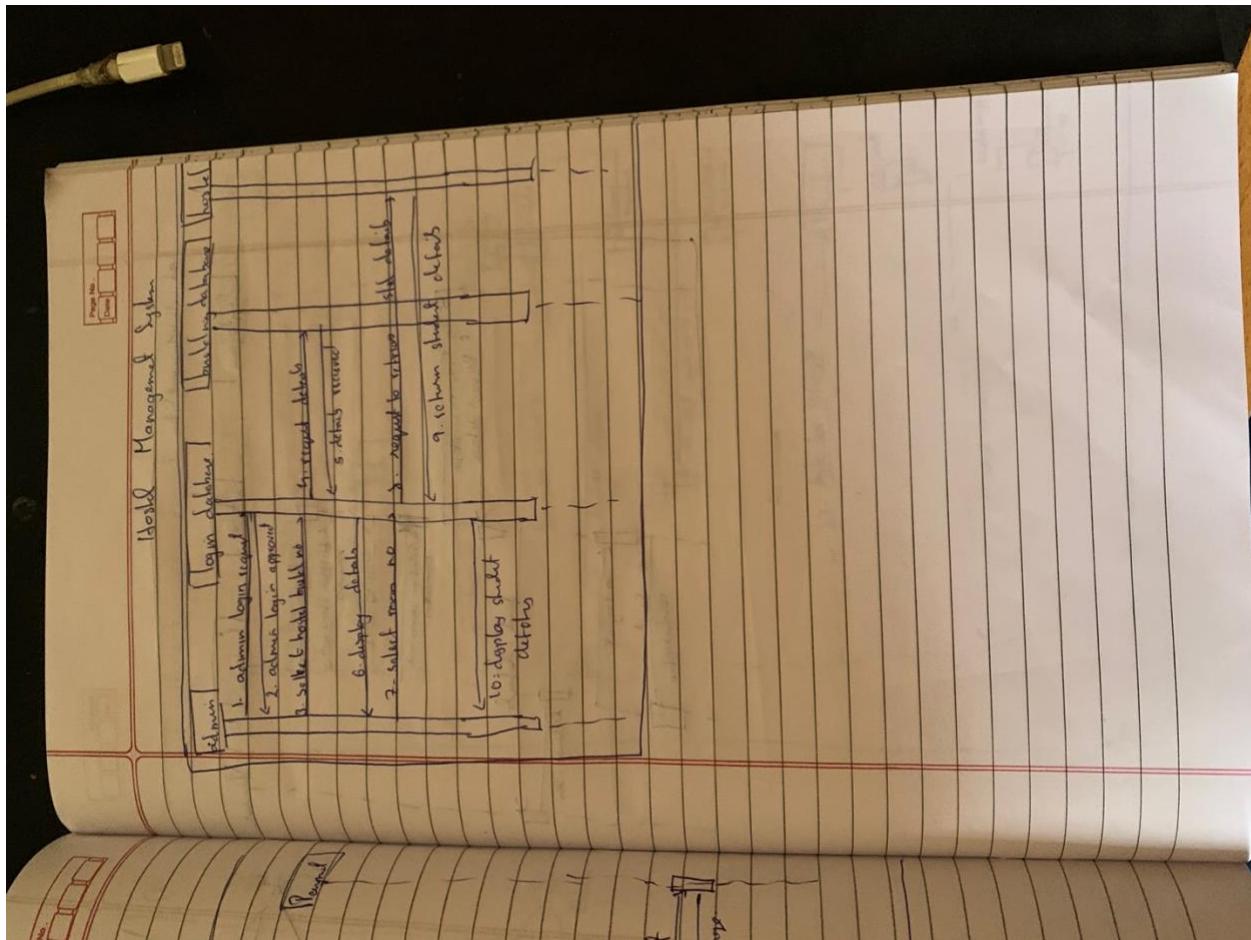
Sequence Diagram:

The lifeline is the dotted line and the rectangles represent the period of time the object is executing and is hence called activation.

The Login actor has self-message to check with the registration of the student.

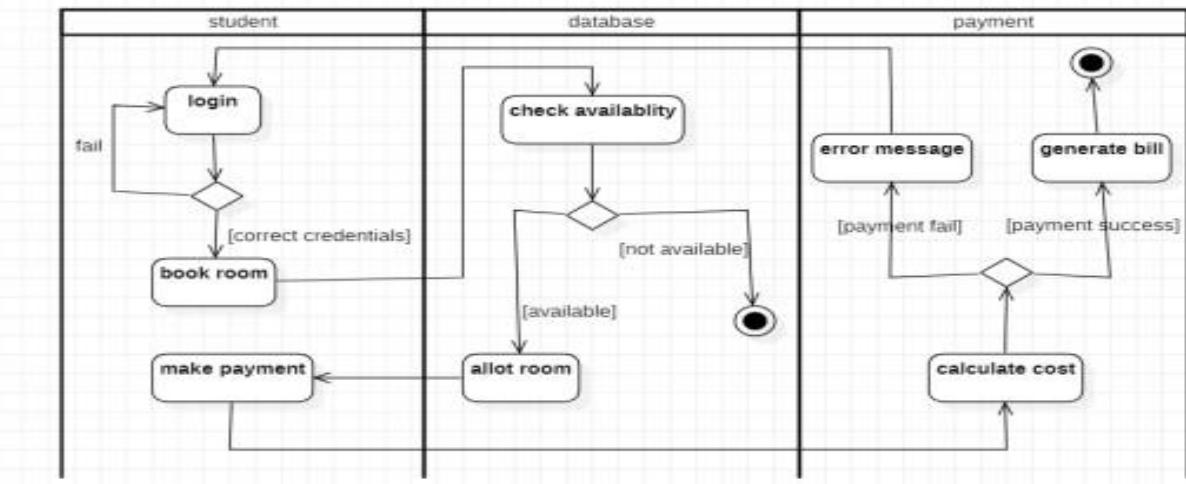
Async and sync signal replies (dotted line) are used to reply back with specificity to the object

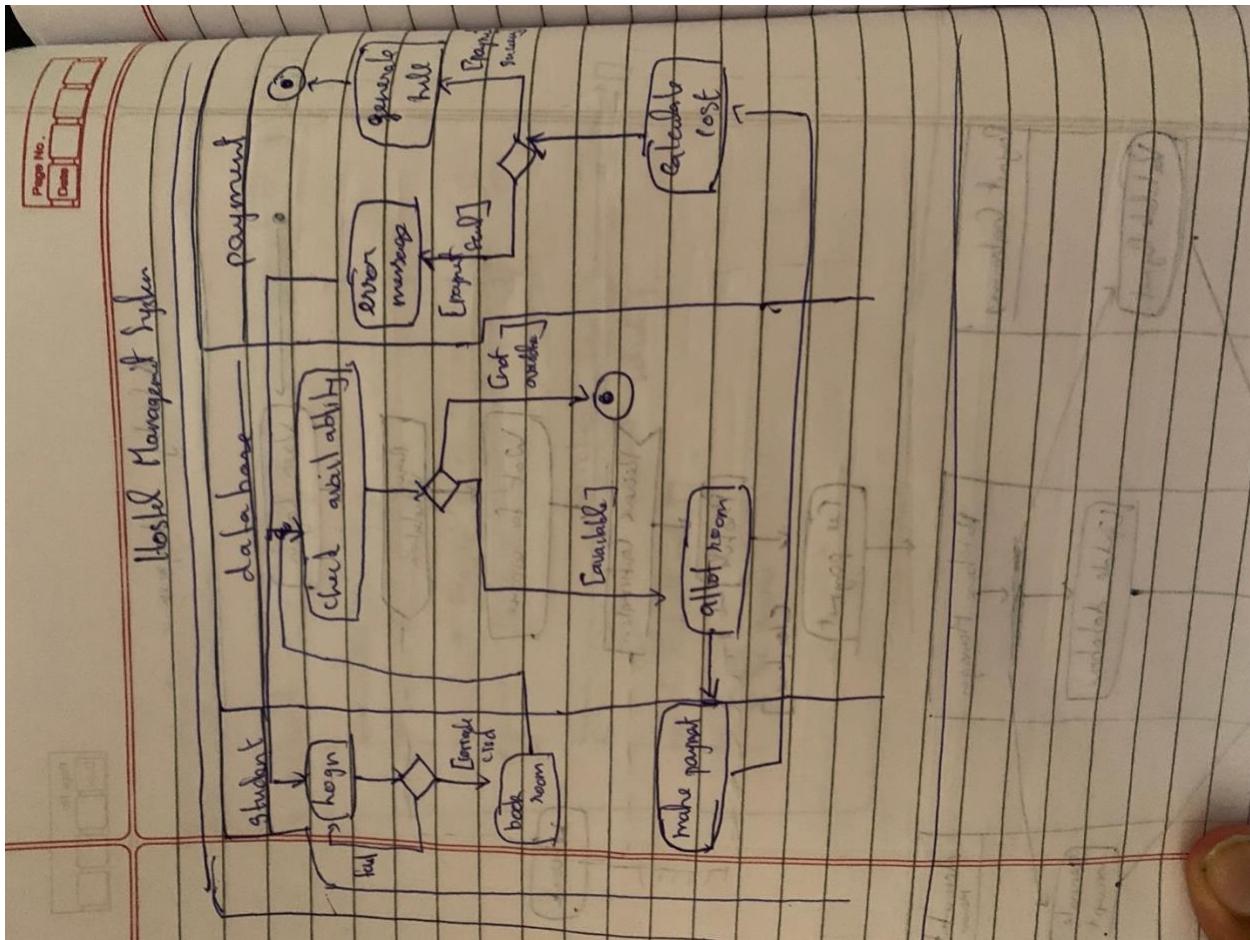




Activity Diagram:

The advanced activity diagram starts from initiation and then in the student swim lane, student login activity where a signal is sent to the network for request validation and upon confirmation the control flows to profile and then book room activity. There are three swim lanes namely student, database, payment where validate student, update database and confirm payment respectively. Then the control flows to the home page and then termination activities.,





3. Stock Maintenance System

Problem statement:

Design UML diagrams for Stock Maintenance System provided with system requirements specification.

Software Requirements Specification (SRS):

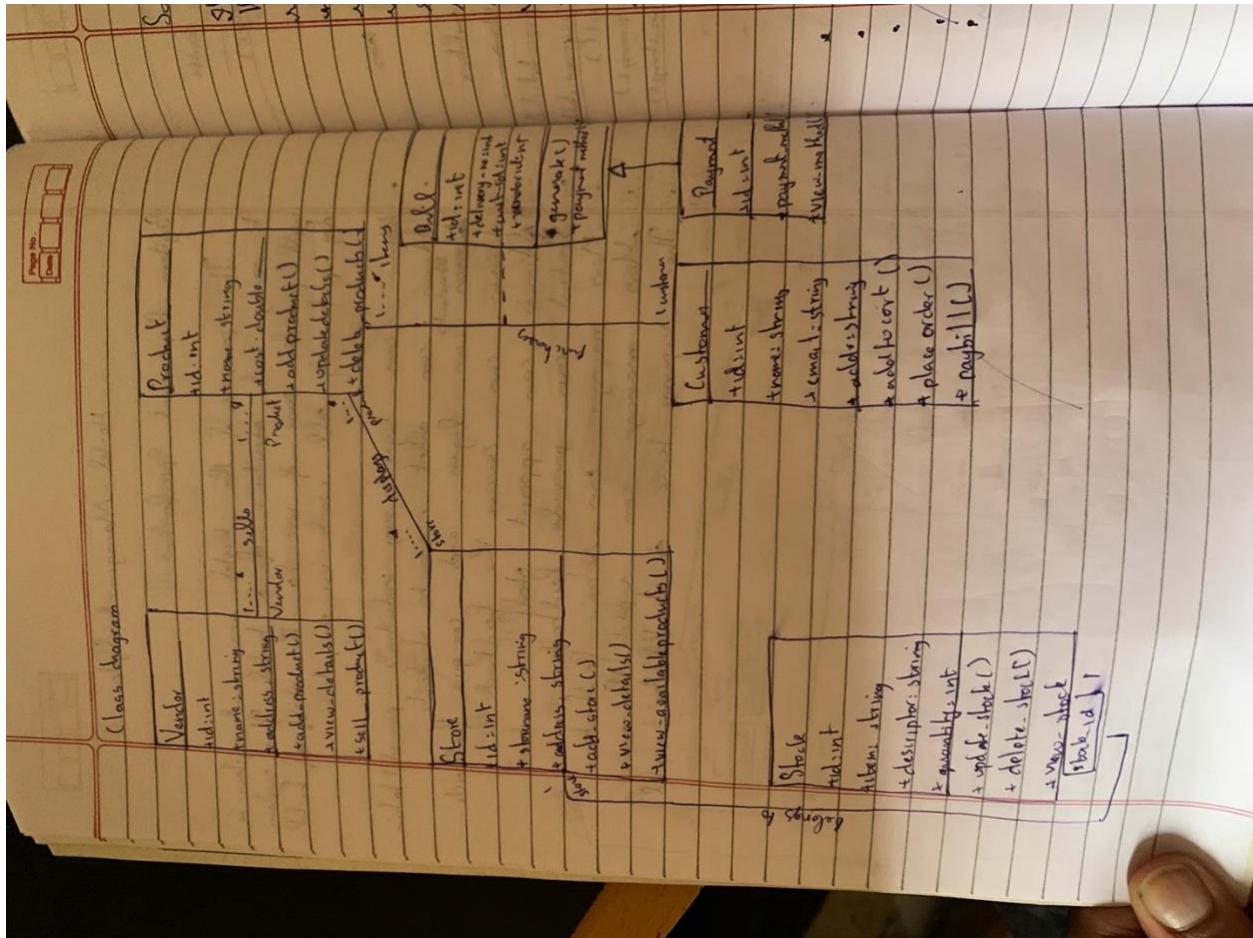
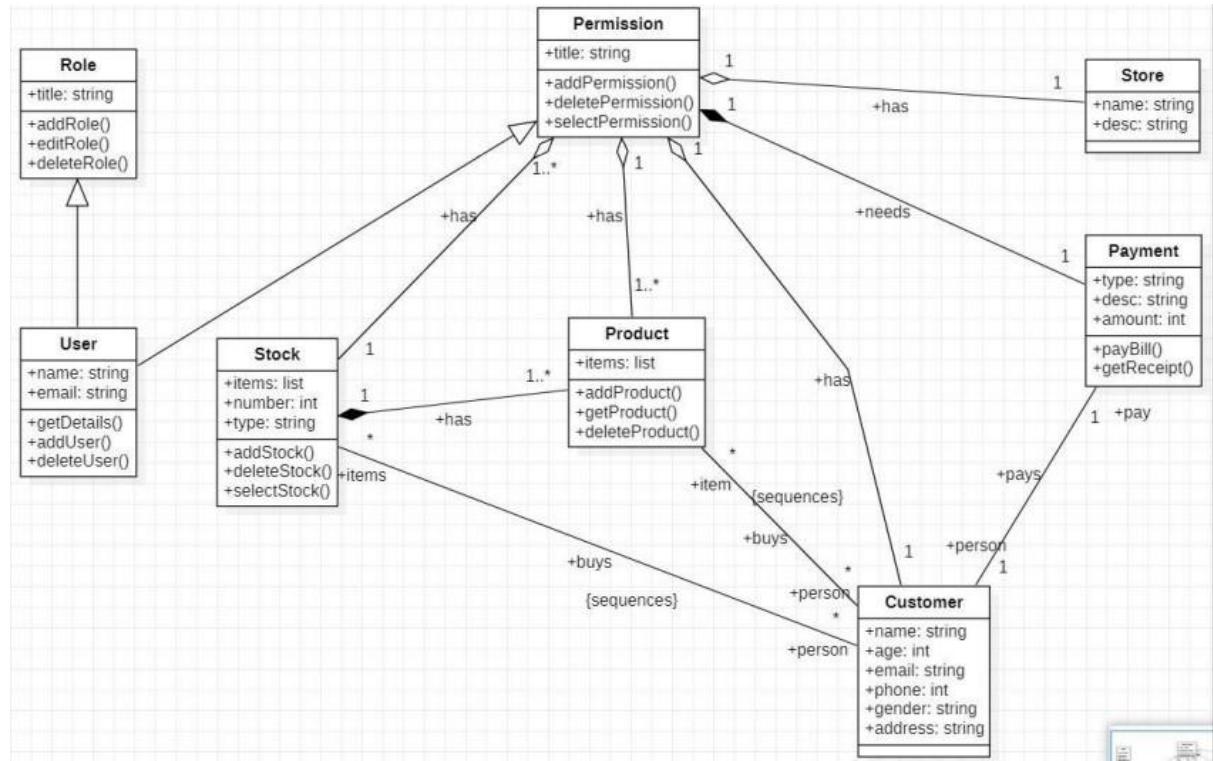
The stock maintenance system will allow the employees to record information of the items available in the store and generate reports based on the total amount of sales. The new system will have a windows-based desktop interface to allow employees to enter the information of sales, purchase orders, change employee preferences and create reports. The system retains information on all the items in the shop. The system retains the records of the cost, expiry date, vendor details, discount, quantity. The employee maintains the information of the sale of the

item. He can add the items at the right time and update the database. The customer can view the availability of the required items and the price of the items. The customer can just view them but cannot make any changes.

The process of the stock maintenance system is that the customer logs in to the particular site to place the order for the customer product. The stock maintenance system is described sequentially through steps

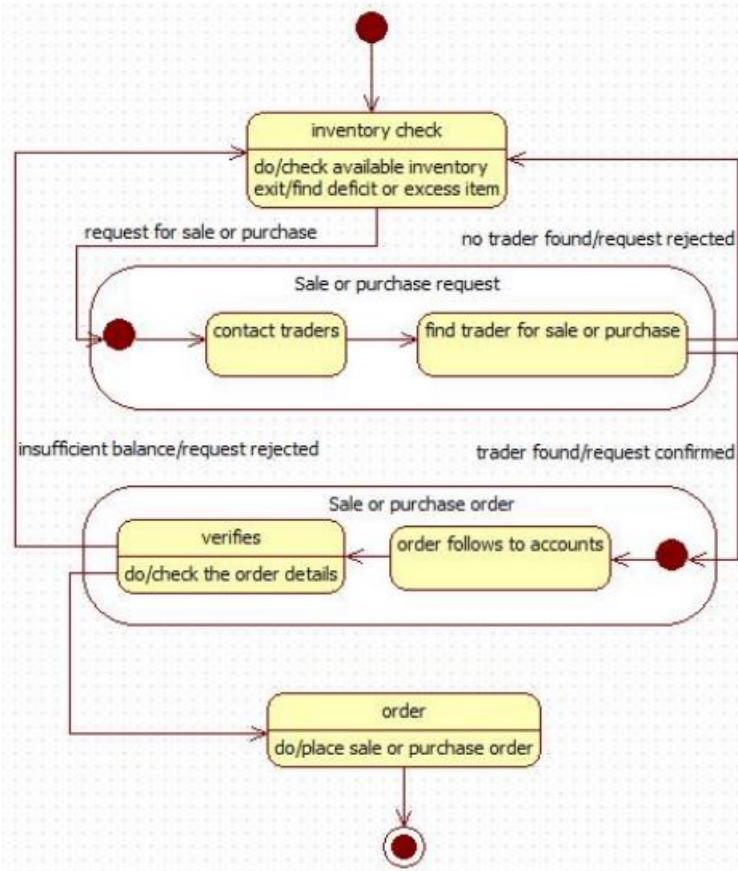
- The customer logs in to the particular site.
- They fill the customer details.
- They place the orders for their product.
- The vendor logs in and views the customer details and orders

Class Diagram:

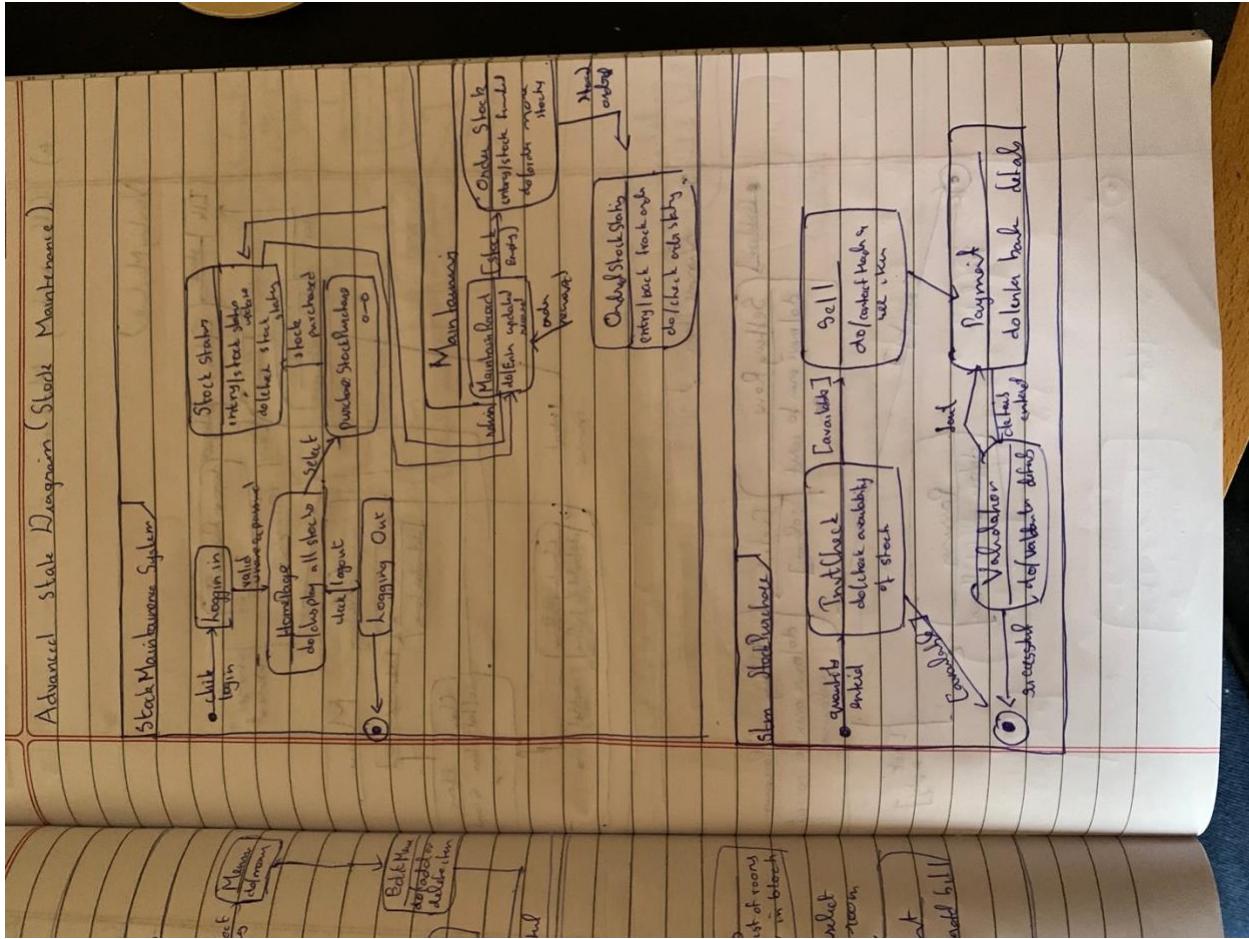


State Diagram:

The advanced state diagram depicted below contains one nested state and one submachine, which on successful login shows the Stock Status details and Stock Purchase procedure. It contains initial state and termination state with Maintaining as a nested state including the required simple states. It also has a submachine state named Stock Purchase with initial, termination state along with simple states; Inventory check, Sell, Payment, Validation.

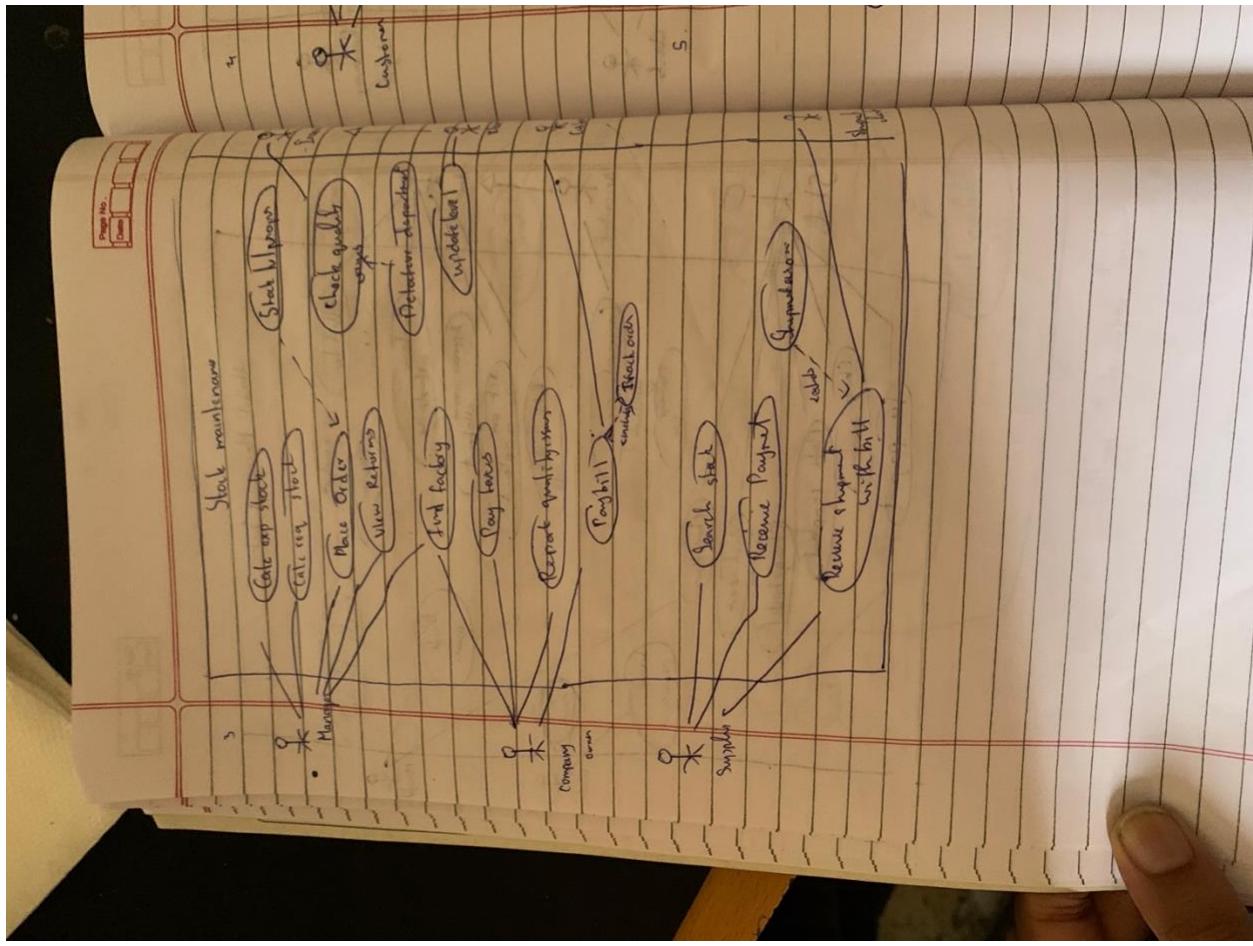
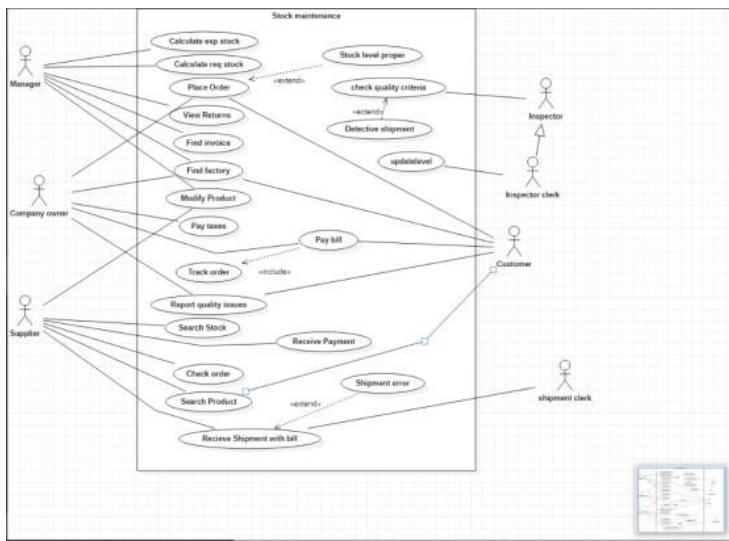


Advanced State Diagram (Stock Maintenance)



Use Case Diagram:

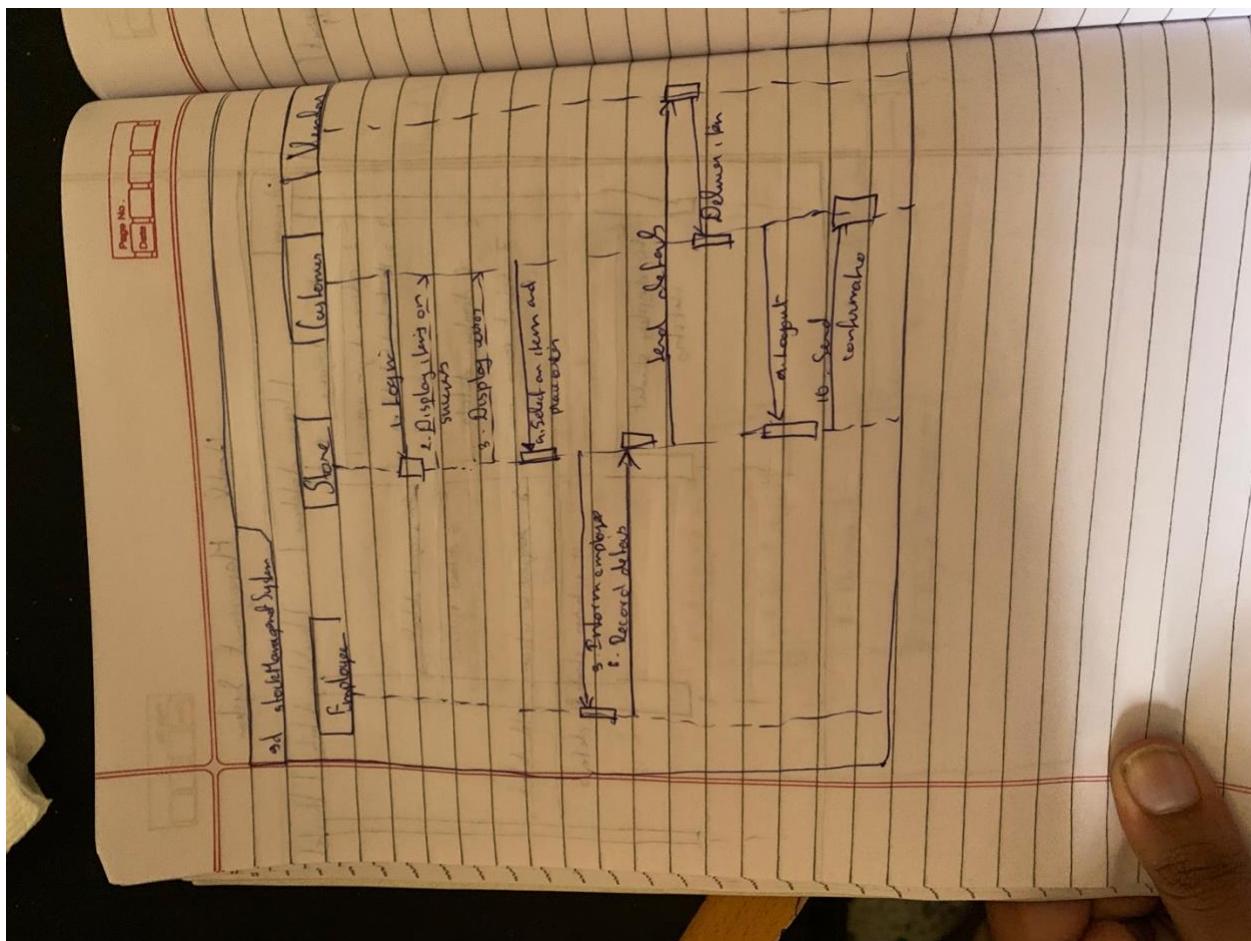
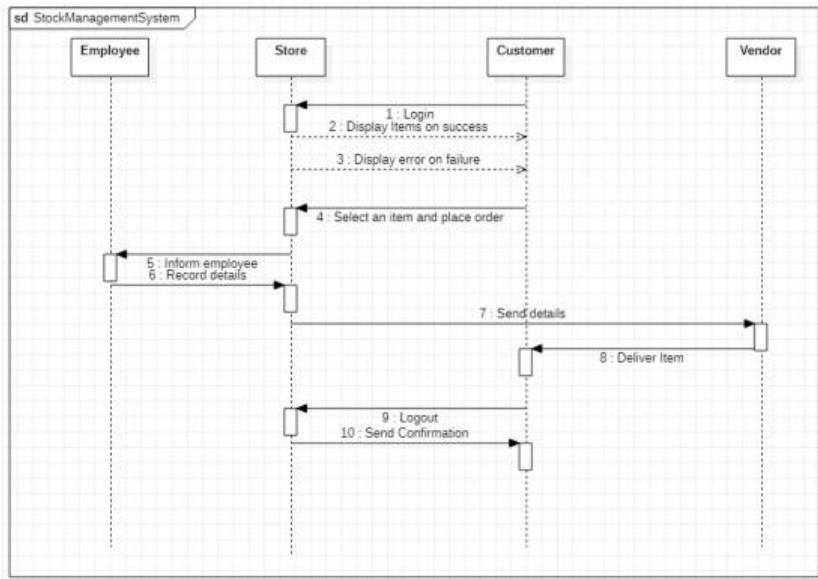
The advanced use case diagram has extra functionalities which includes extends, includes and generalization. The stock level use case extends place order use case, detective shipment use case extends check quality criteria use case , shipment error use case extends receive shipment with bill use case, pay bill use case includes track order use case.



Sequence Diagram:

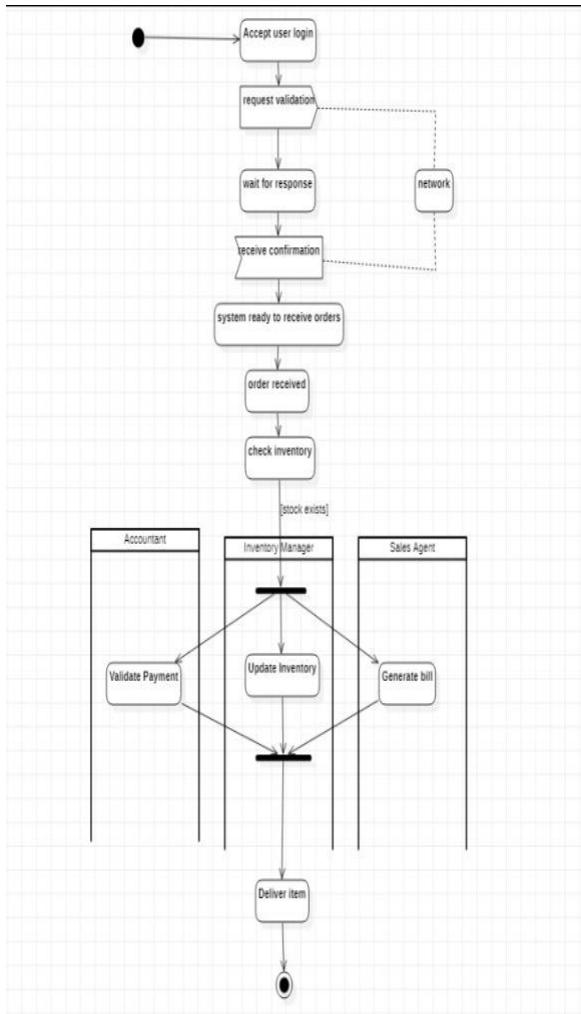
The lifeline is the dotted line and the rectangles represent the period of time the object is executing and is hence called activation.

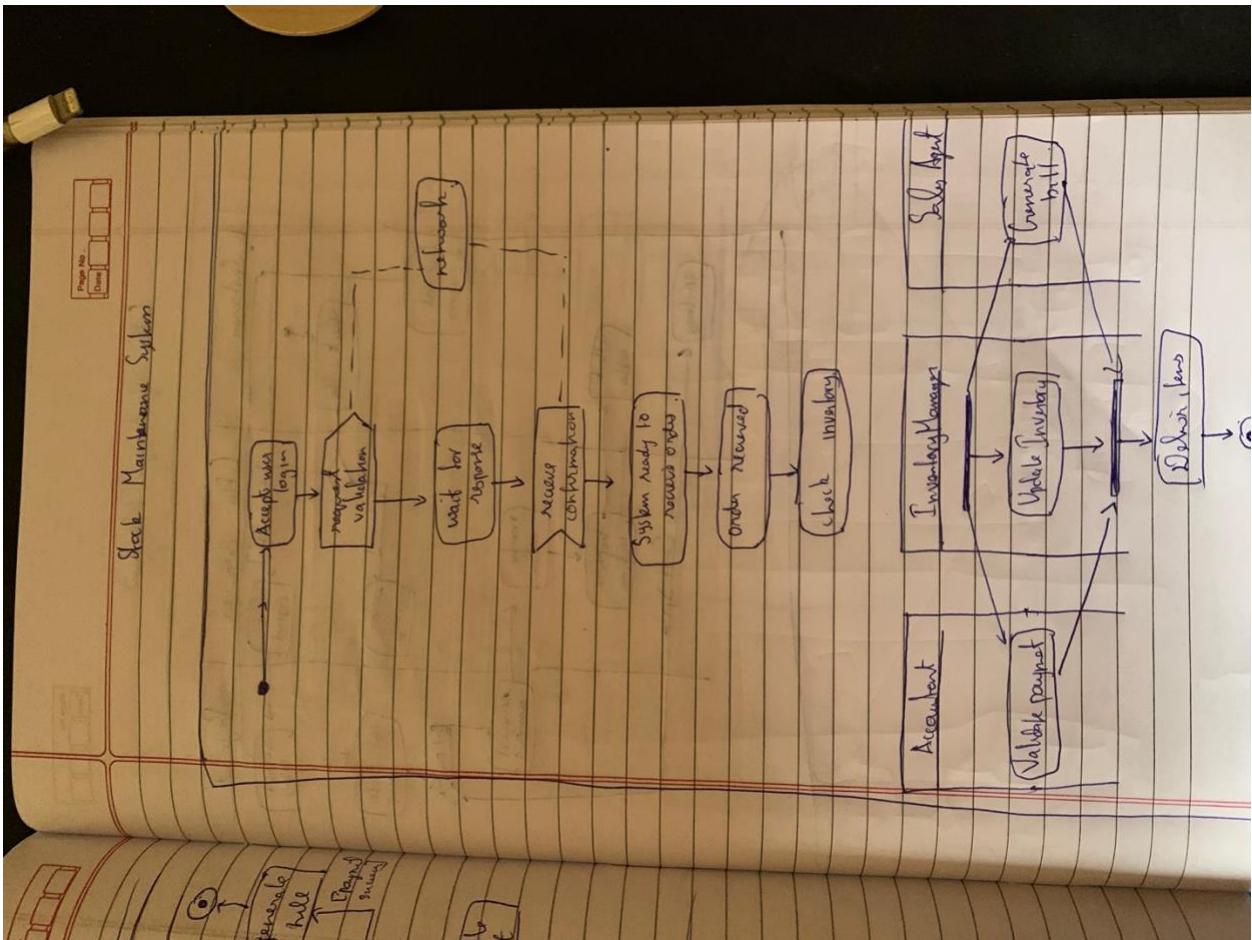
Create message signal is used to indicate the display of failure in any failure situation.



Activity Diagram:

The advanced activity diagram starts from initiation and then user login activity where a signal is sent to the network for request validation and upon confirmation the control flows to order received and then check inventory activity. There are three swim lanes namely inventory manager, accountant and sale agent where update inventory, update payment and generate bill respectively. Then the control flows to the home page and then termination activities.





4. Coffee Vending Machine

Problem statement:

Design UML diagrams for Stock Maintenance System provided with system requirements specification.

Software Requirements Specification (SRS):

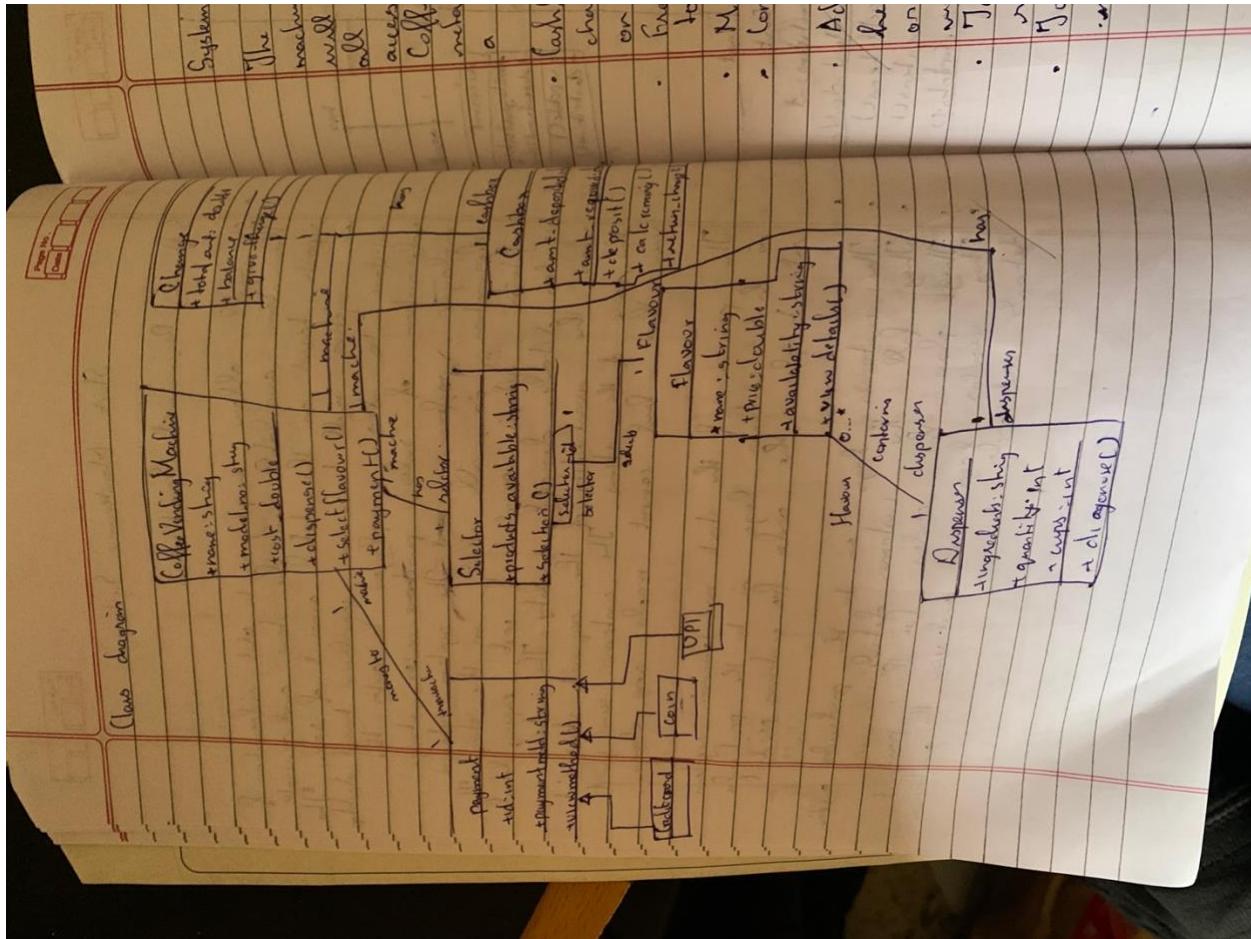
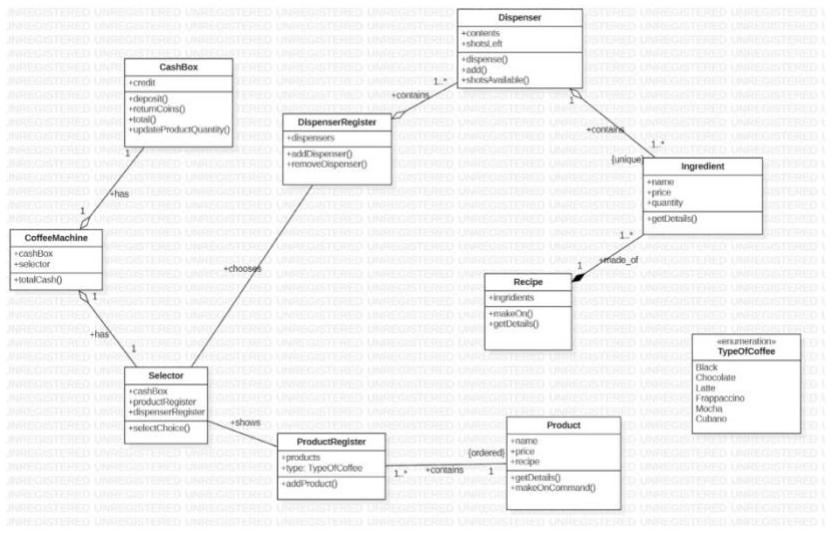
The stock maintenance system will allow the employees to record information of the items available in the store and generate reports based on the total amount of sales. The new system

will have a windows-based desktop interface to allow employees to enter the information of sales, purchase orders, change employee preferences and create reports. The system retains information on all the items in the shop. The system retains the records of the cost, expiry date, vendor details, discount, quantity. The employee maintains the information of the sale of the item. He can add the items at the right time and update the database. The customer can view the availability of the required items and the price of the items. The customer can just view them but cannot make any changes.

The process of the stock maintenance system is that the customer logs in to the particular site to place the order for the customer product. The stock maintenance system is described sequentially through steps

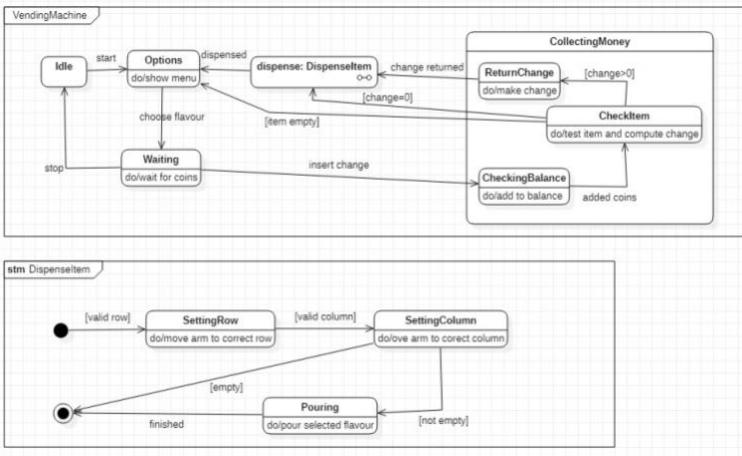
- The customer logs in to the particular site.
- They fill the customer details.
- They place the orders for their product.
- The vendor logs in and views the customer details and orders

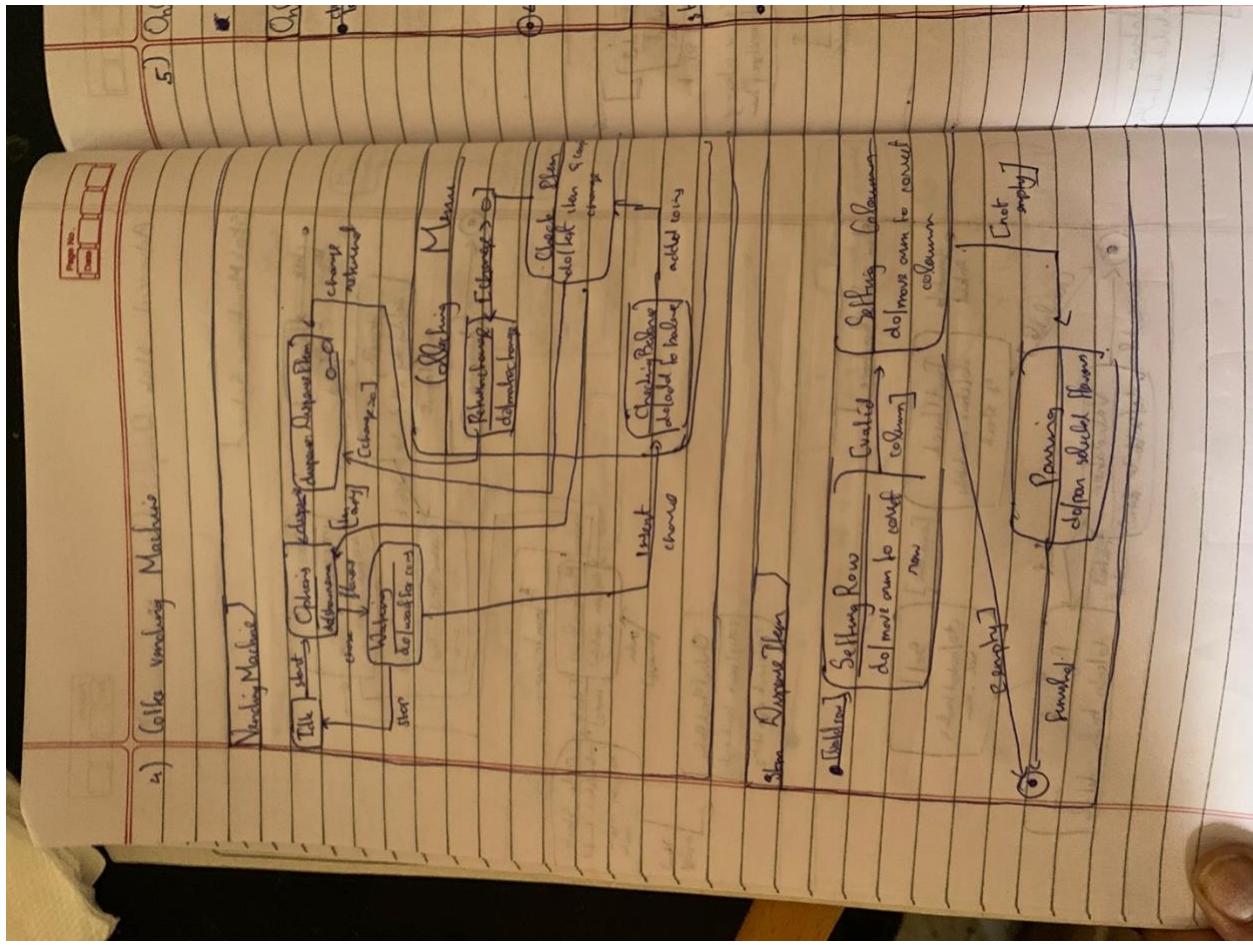
Class Diagram:



State Diagram:

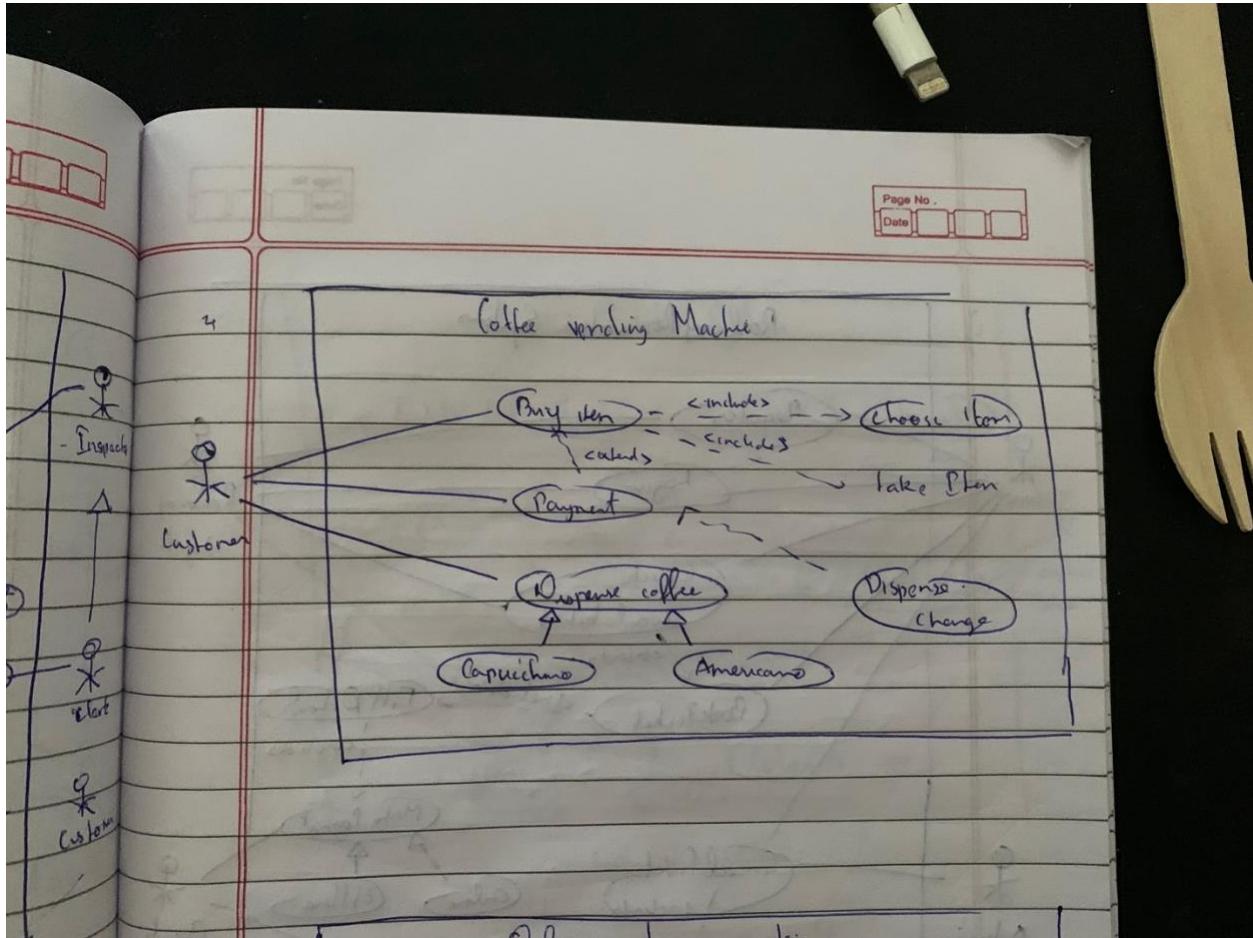
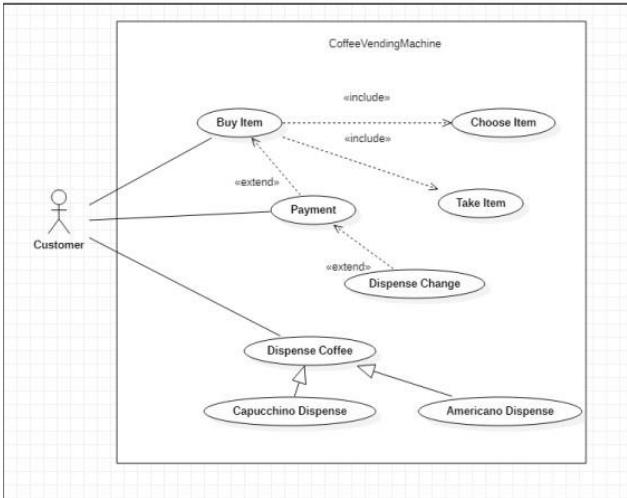
The advanced state diagram depicted below contains one nested state and one submachine, which on successful login shows the Collecting Money procedure and DispenseItem procedure. It contains initial state and termination state with Collecting Money as a nested state including the required simple states. It also has a submachine state named DispenseItem with initial, termination state along with simple states; SettingRow, SettingColumn, Pouring.





Use Case Diagram:

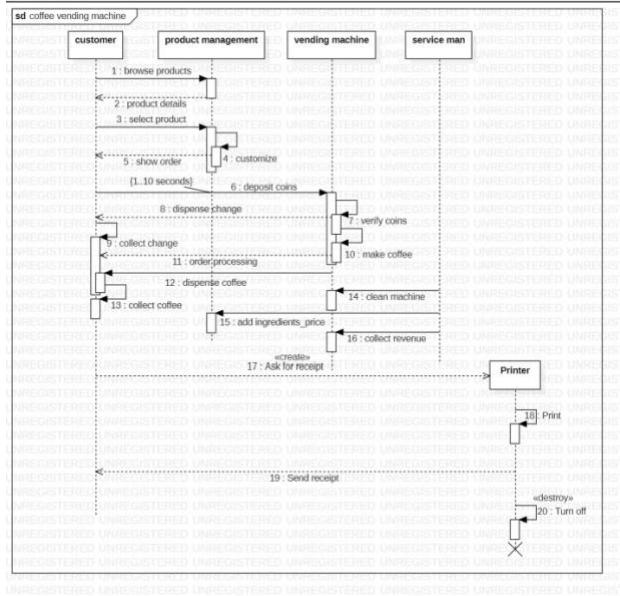
The advanced use case diagram has extra functionalities which includes extends, includes and generalization. The dispense change use case extends payment use case, payment use case extends buy item use case, buy item use case includes choose item and take item use case. Cappuccino dispense and American dispense is generalized to super class dispense

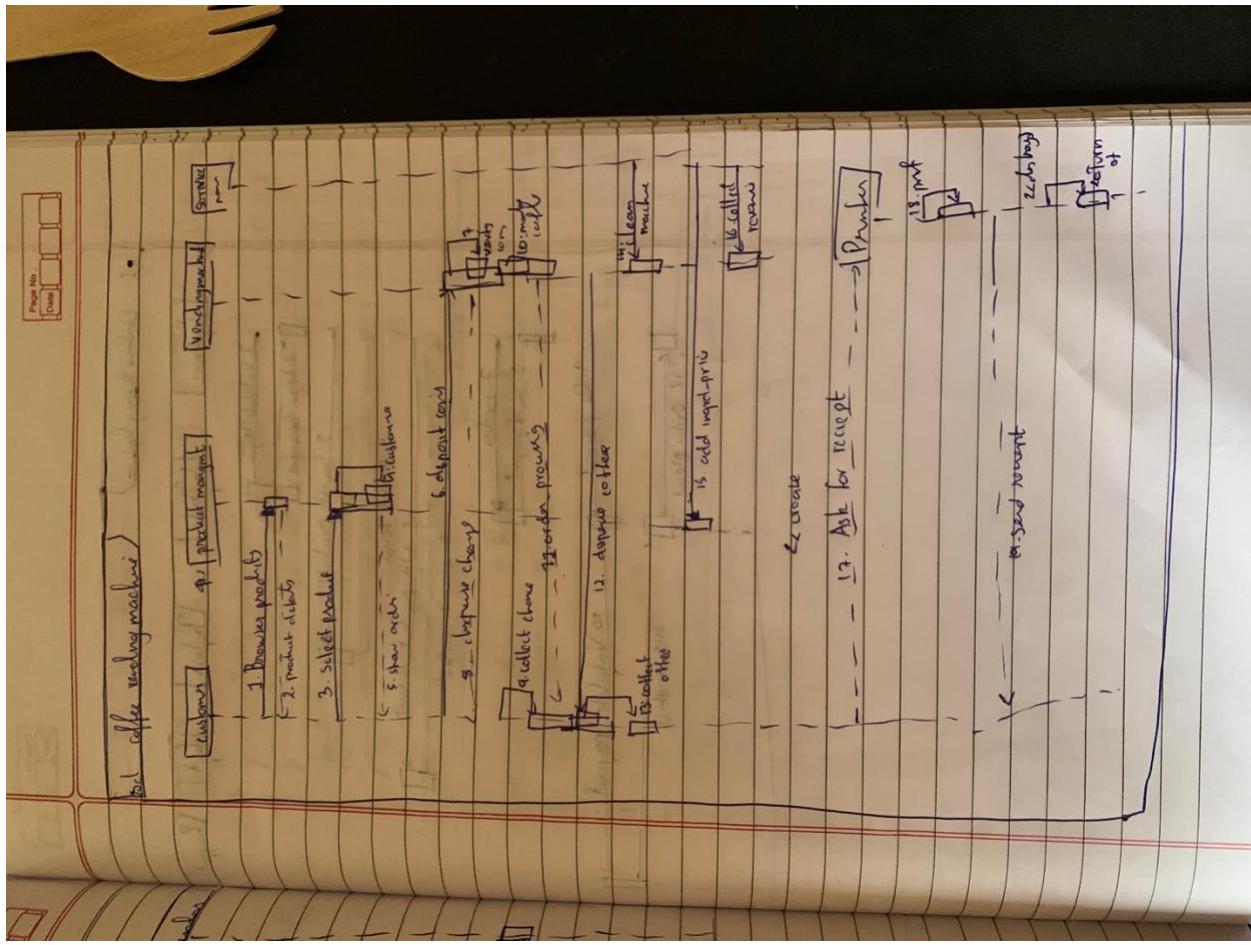


Sequence Diagram:

The lifeline is the dotted line and the rectangles represent the period of time the object is executing and is hence called activation.

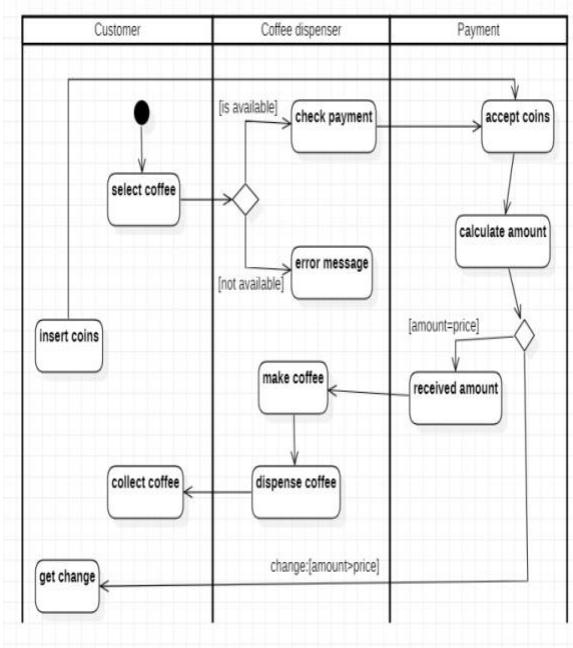
The recursive function of customize is shown by double activation rectangle of customize and verify coins.

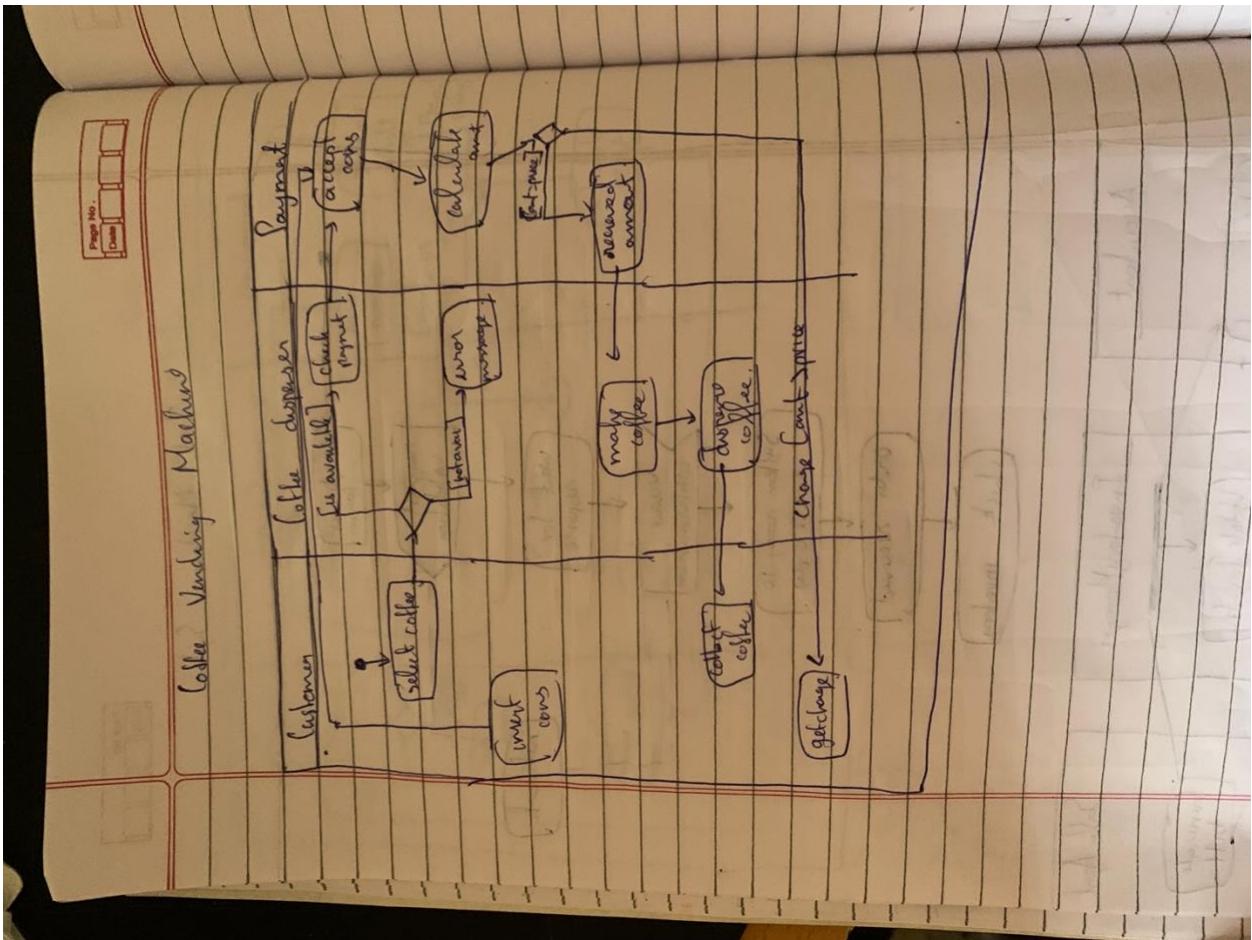




Activity Diagram:

The advanced activity diagram starts from initiation and in the customer swimlane, customer login activity where a signal is sent to the network for request validation and upon confirmation the control flows to order received and then check inventory activity. There are three swimlanes namely customer, coffee dispenser and payment where customer perform operations like order coffee, dispenses coffee and collect coins respectively. Then the control flows to the home page and then termination activities.





5. Online Shopping System

Problem statement:

Design UML diagrams for Online Shopping System with system requirements specification.

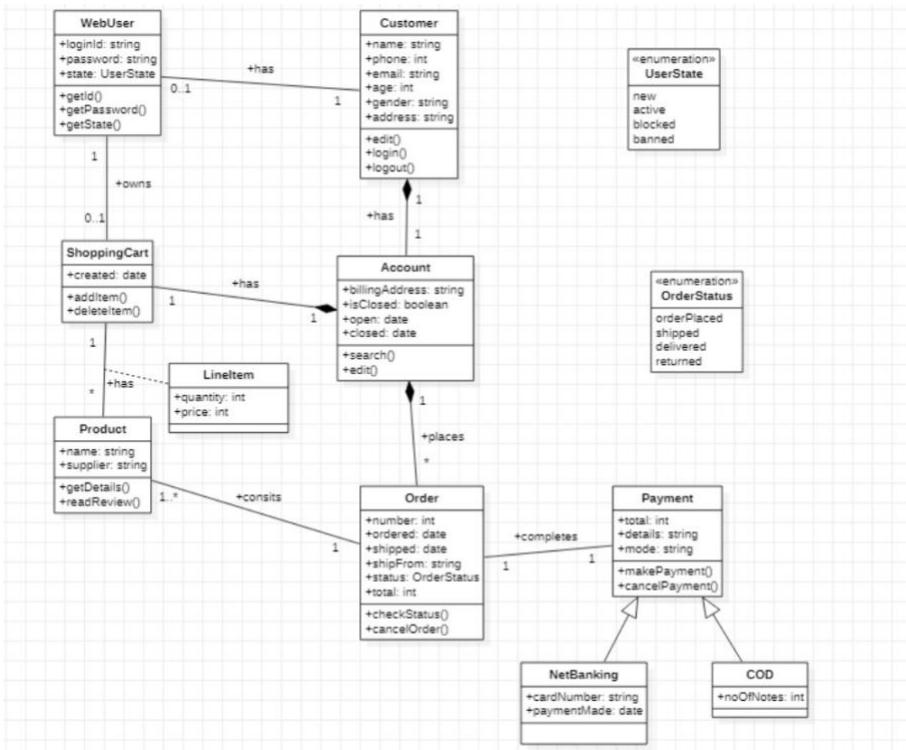
Software Requirements Specification (SRS):

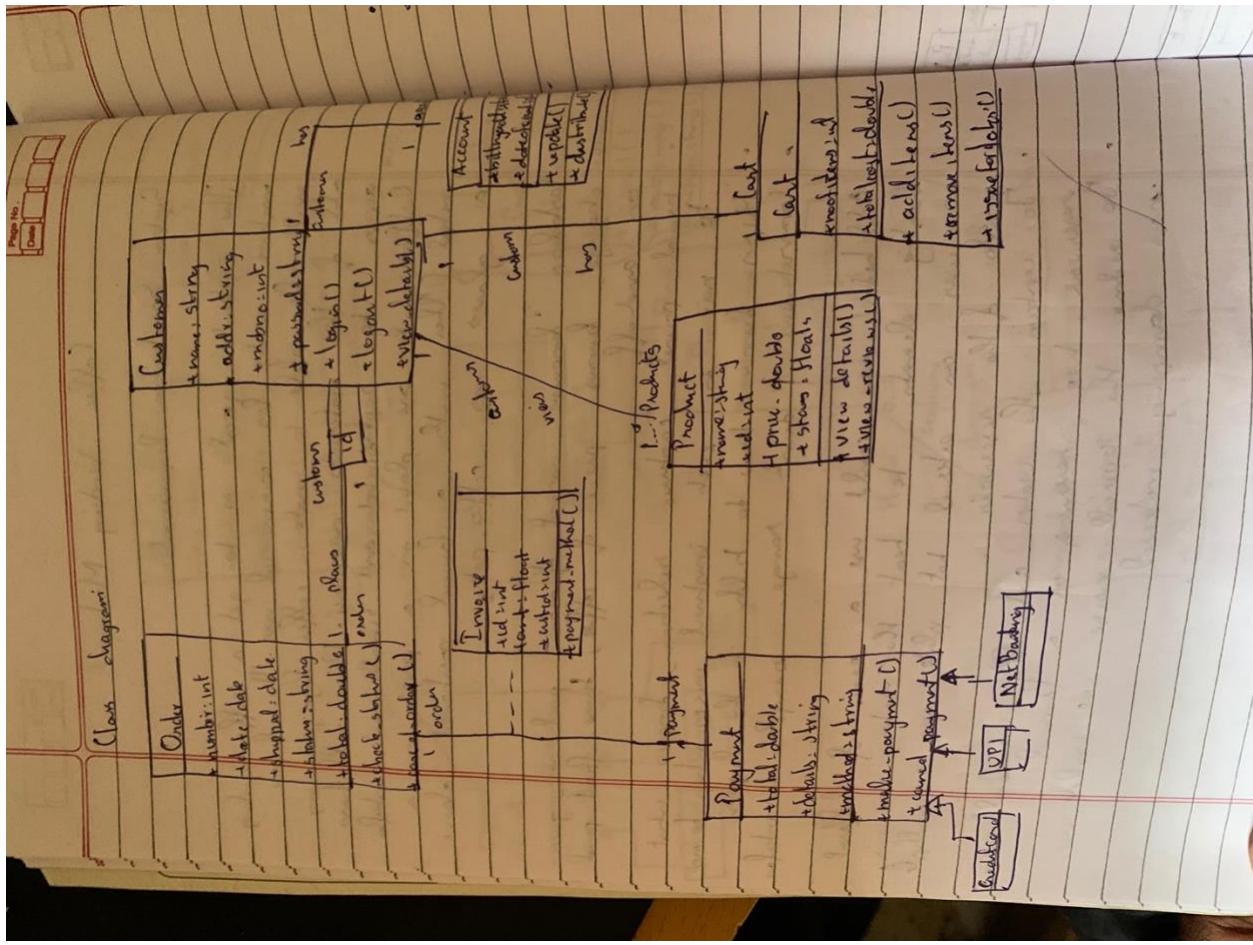
The online shopping system allows the users and vendors to exchange products remotely and

reduces the amount of cost and time substantially. The software provides the following facilities to the customers:

- Facilitates easy shopping online anywhere with free shipping (conditions apply).
- Provides information about the products in categories
- Can avail the facility of purchasing second hand products
- Can reserve if the particular product is not available
- Customers are provided with up-to-date information on the products available
- Provides email facility for future correspondence
- Can add nearly ten products to their shopping cart at a time.
- Cannot reserve the product for more than two days.
- Cannot reserve more than two products
- Responsibility of damages

Class Diagram:

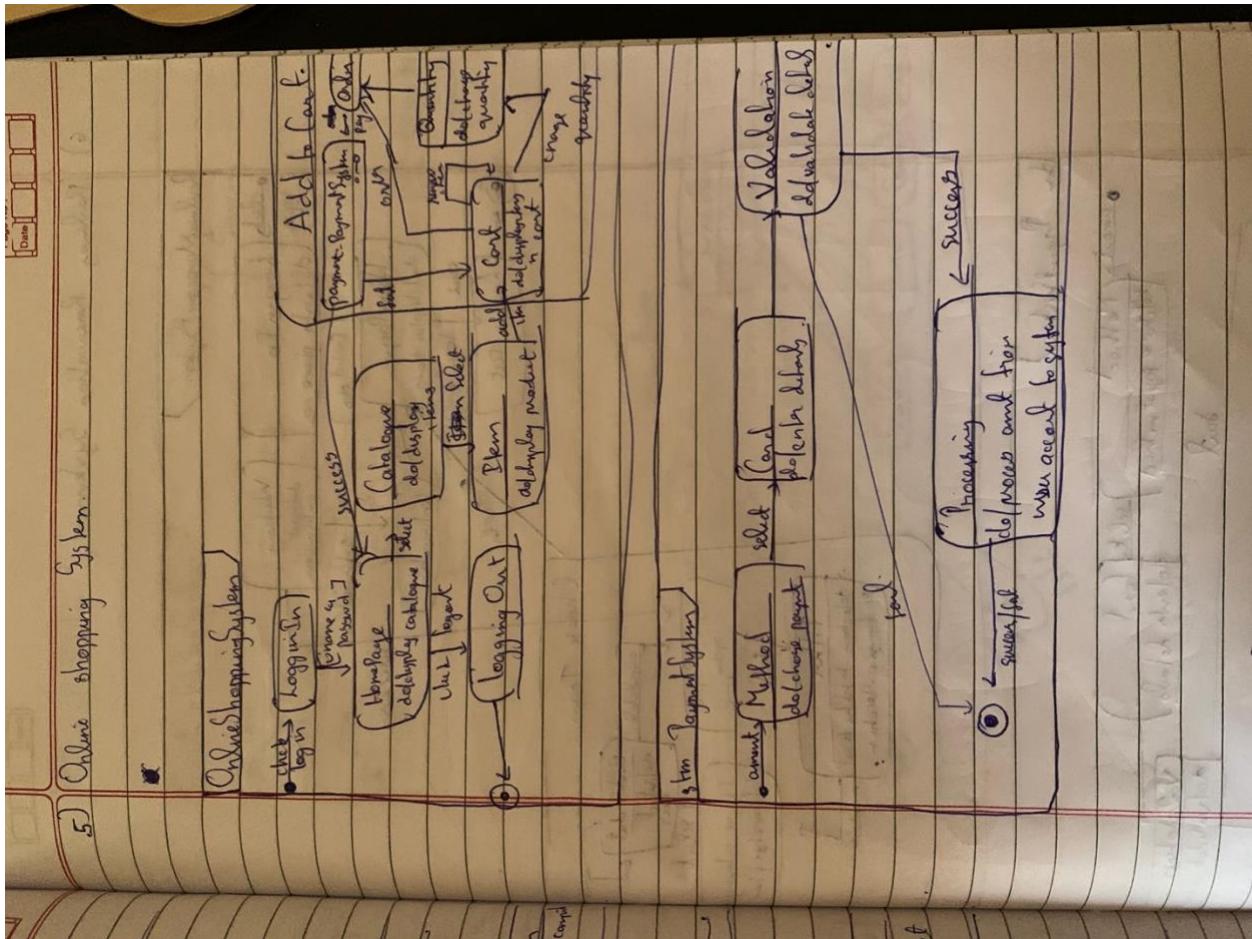
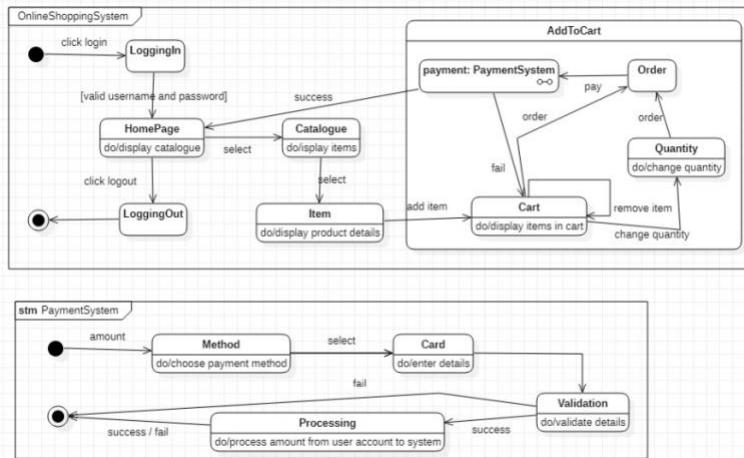




State Diagram:

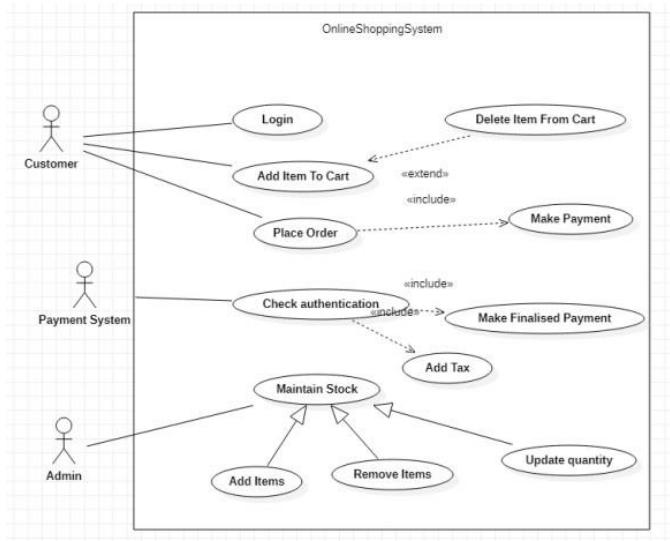
The advanced state diagram depicted below contains one nested state and one submachine, which on successful login shows the AddToCart procedure and PaymentSystem procedure. It contains initial state and termination state with AddToCart as a nested state including the

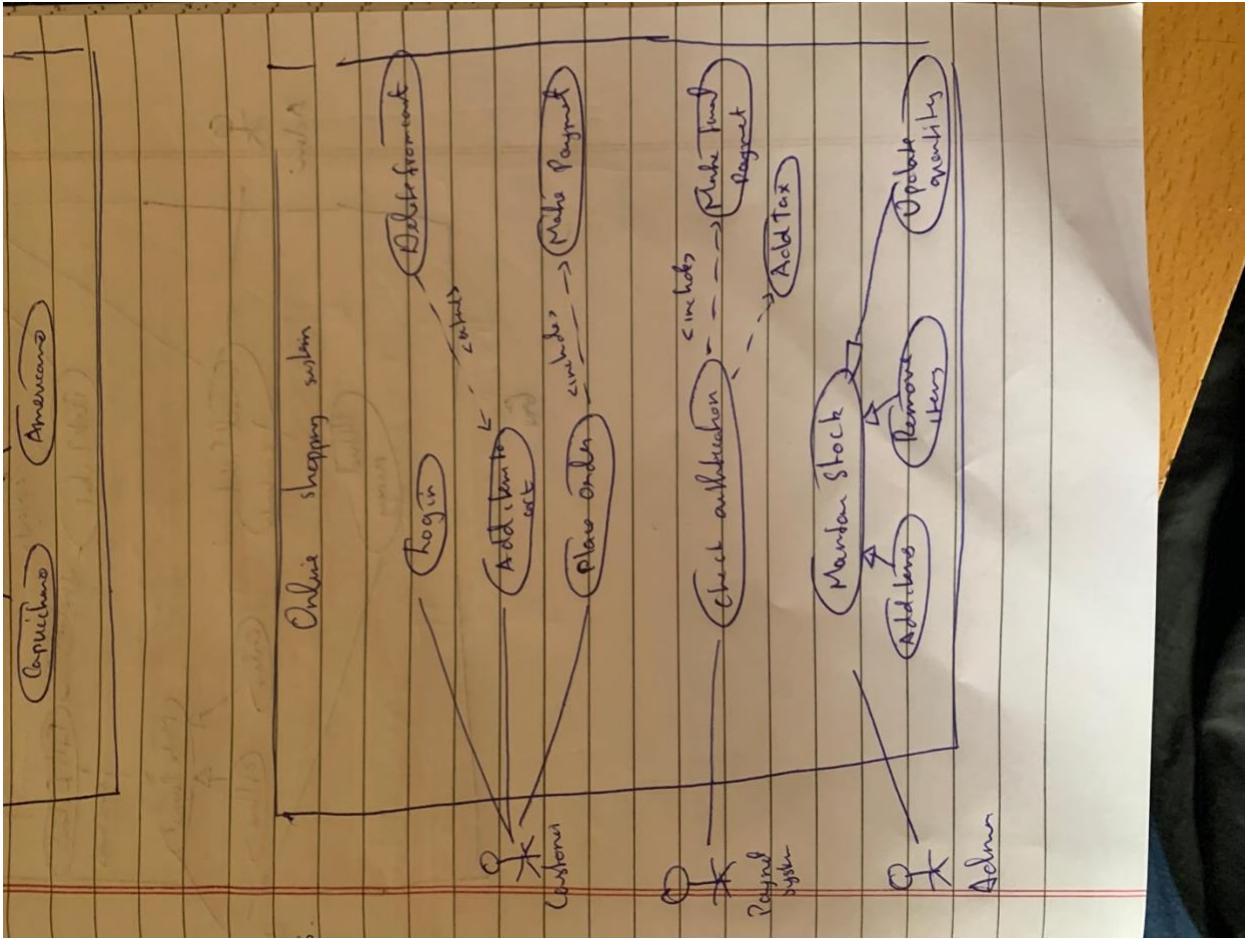
required simple states. It also has a submachine state named PaymentSystem with initial, termination state along with simple states; Method, Card, Validation, Processing.



Use Case Diagram:

The advanced use case diagram has extra functionalities which includes extends, includes and generalization. The delete from cart use case extends add item to cart use case, place order use case includes make payment use case, check authentication use case includes make finalized payment and add tax use case. Add item, remove item and update quantity is generalized to superclass maintain stock.

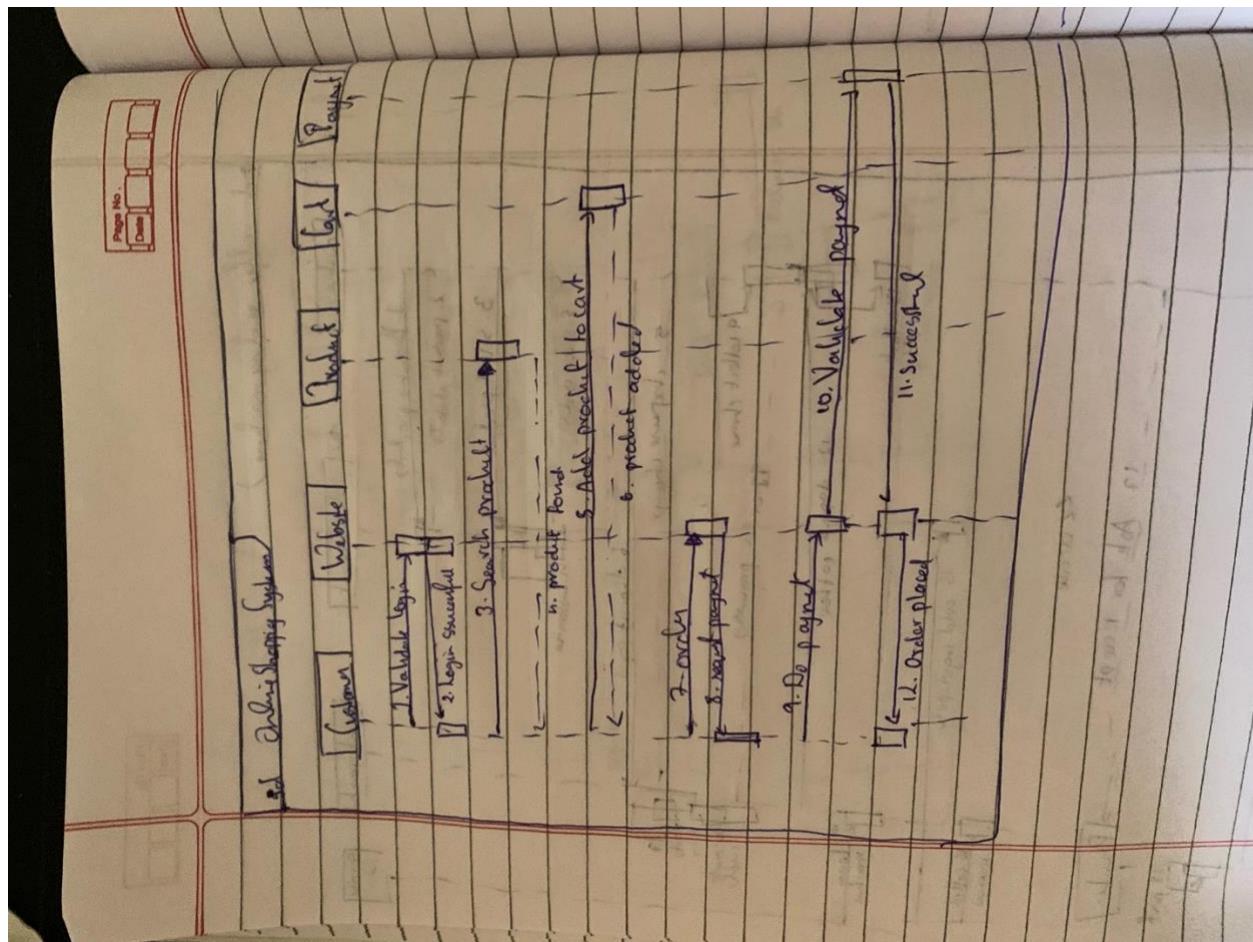
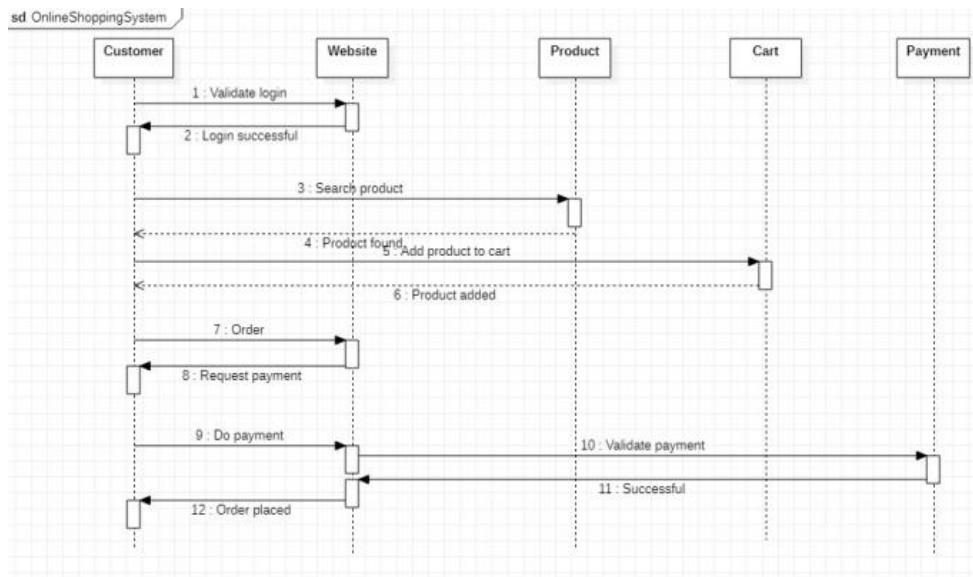




Sequence Diagram:

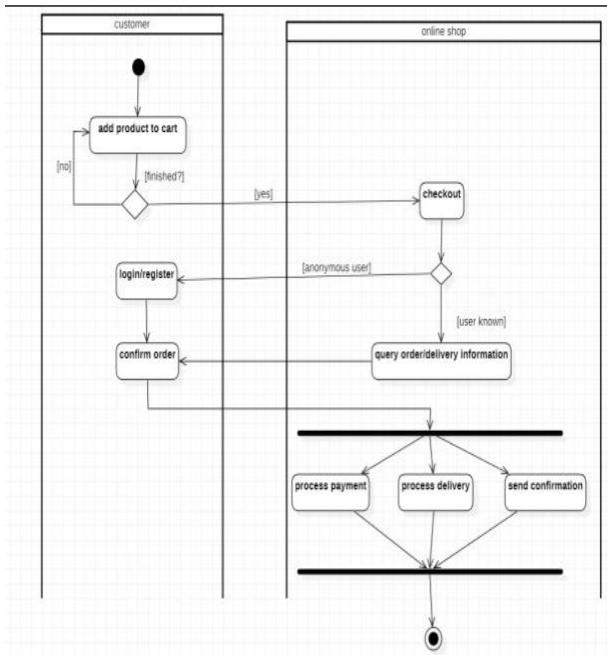
The lifeline is the dotted line and the rectangles represent the period of time the object is executing and is hence called activation.

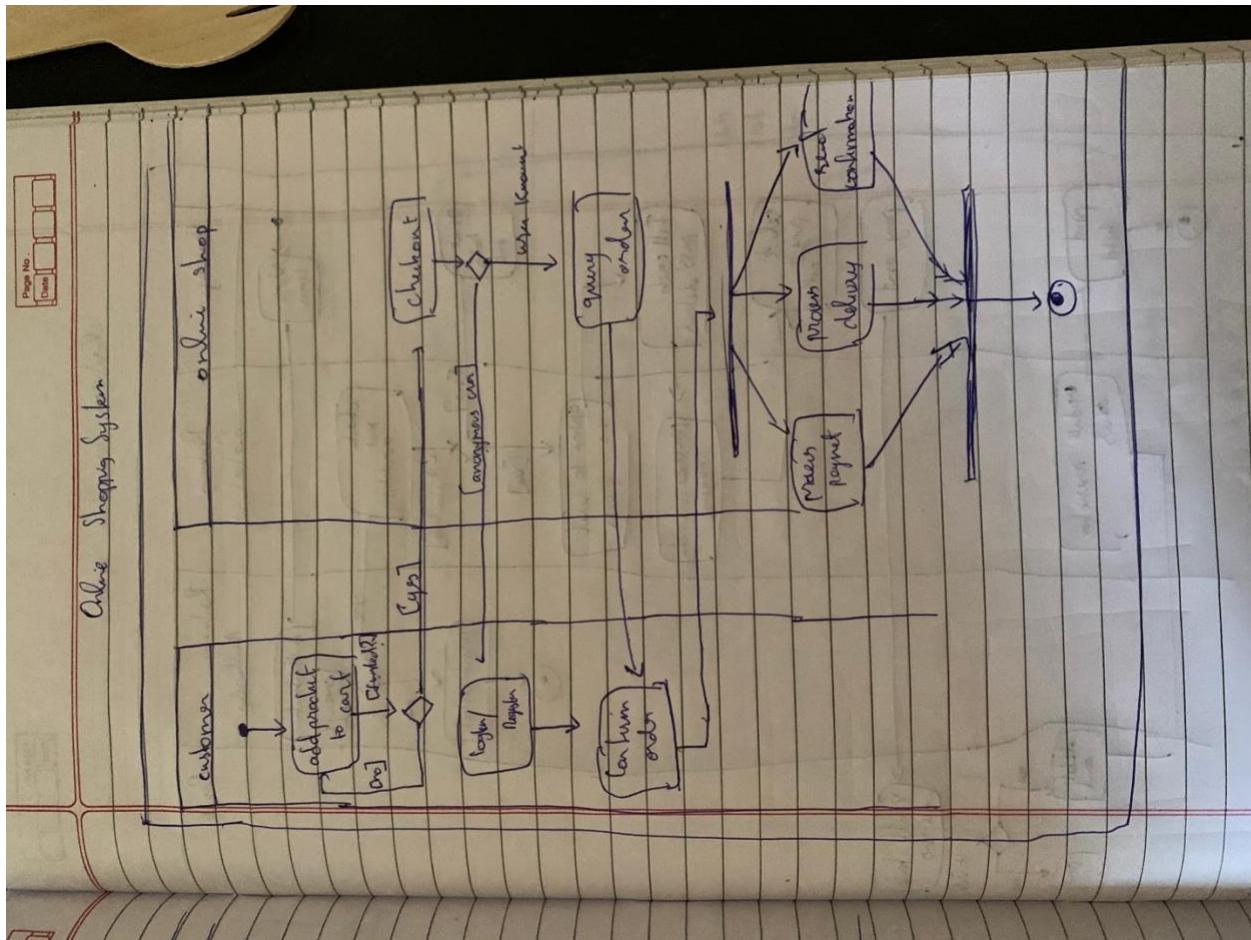
Reply message is used to return back to lifelines with the required message.



Activity Diagram:

The advanced activity diagram starts from initiation and in the customer swimlane, the customer login activity where a signal is sent to the network for request validation and upon confirmation the control flows to add product and checkout activity. There are two swimlanes namely customer and online shop where it confirms the order and delivery, payment process respectively. Then the control flows to the home page and then termination activities.





6.Railway Reservation System

Problem statement:

Design UML diagrams for Railway Reservation System with system requirements specification.

Software Requirements Specification (SRS):

To develop a user-friendly Railway Reservation System to enable passengers to book tickets online and make payment online as well. Railway reservation system project which provides

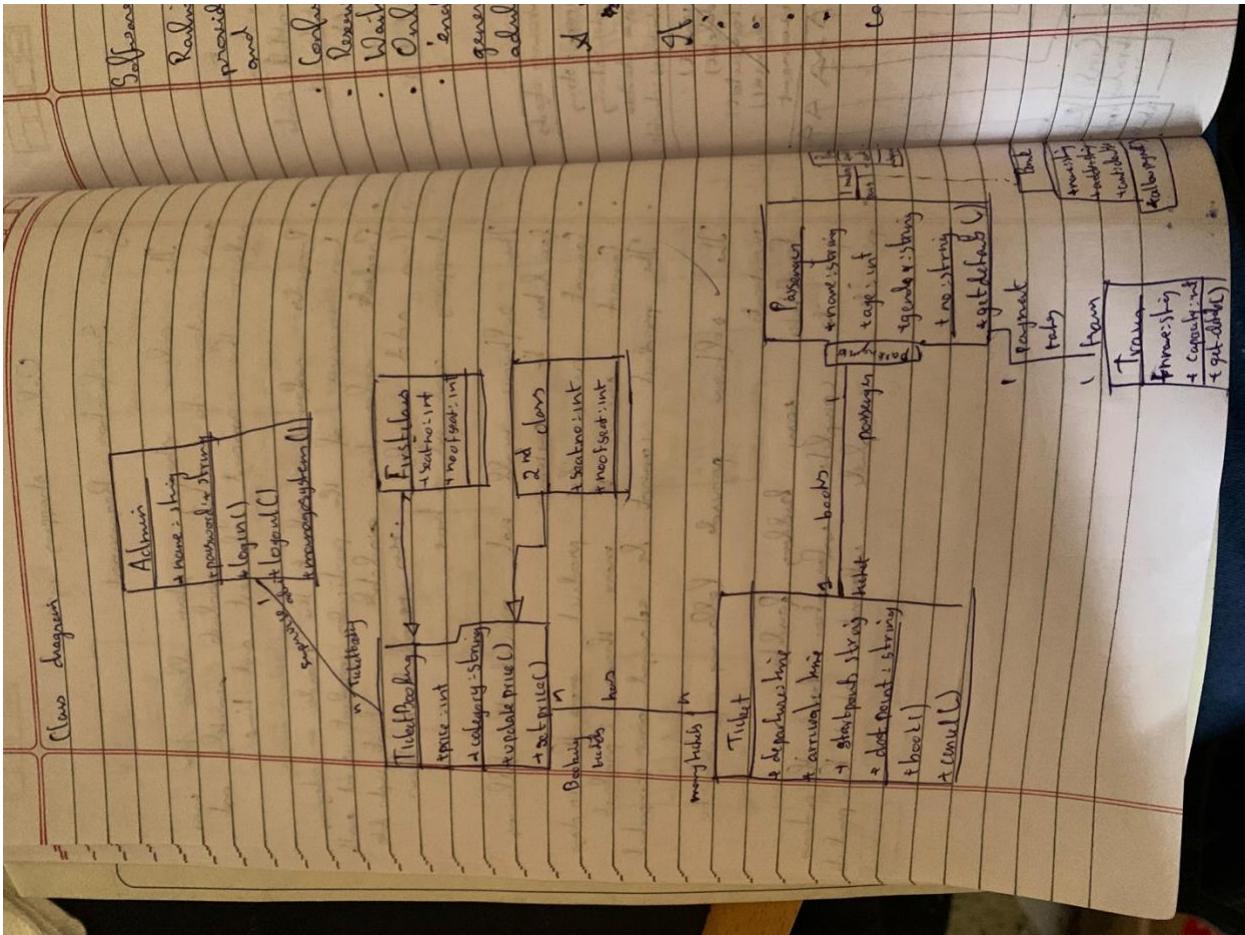
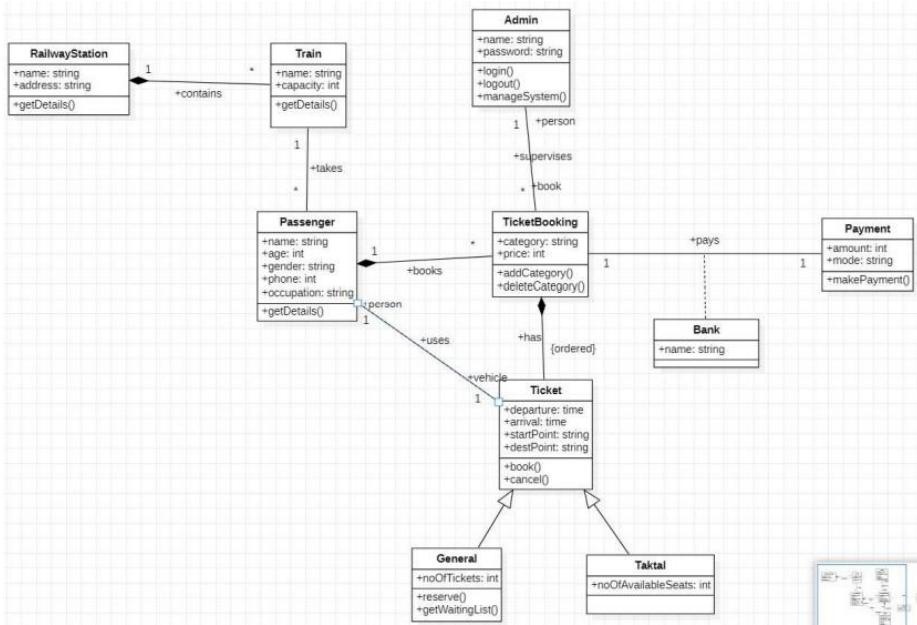
the train timing details, reservation, billing and cancellation on various types of reservation namely,

- Confirm Reservation for Seat.
- Reservation against Cancellation.
- Waiting list Reservation.
- Online Reservation.
- Tatkal Reservation

This system enables the Advance booking in any class, against general and lady's quota, on payment of fare in full for adults and children, a maximum of six berths/seats at a time, for journey between any two stations served by a train. It also provides details about

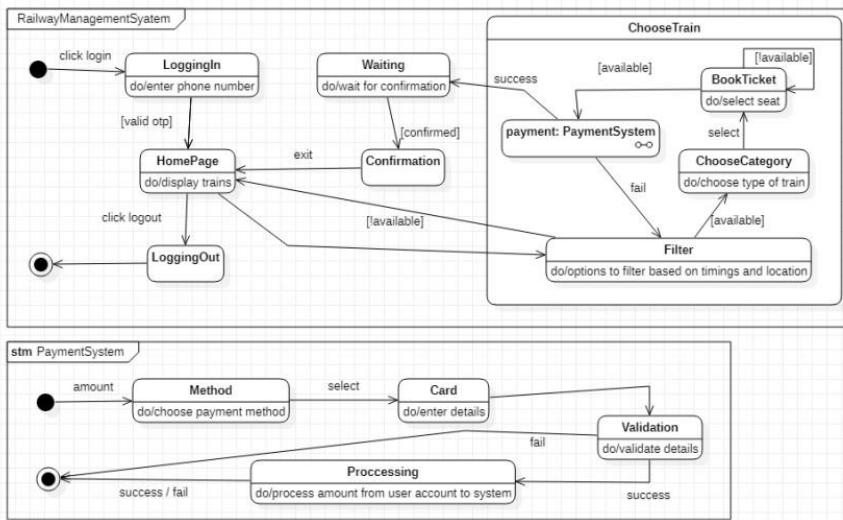
1. Timetable
2. Train Fares
3. Current status of reservation position
4. Train available between a pair of stations
5. Accommodation available for a train/date combination Types of tickets: General and Tatkal

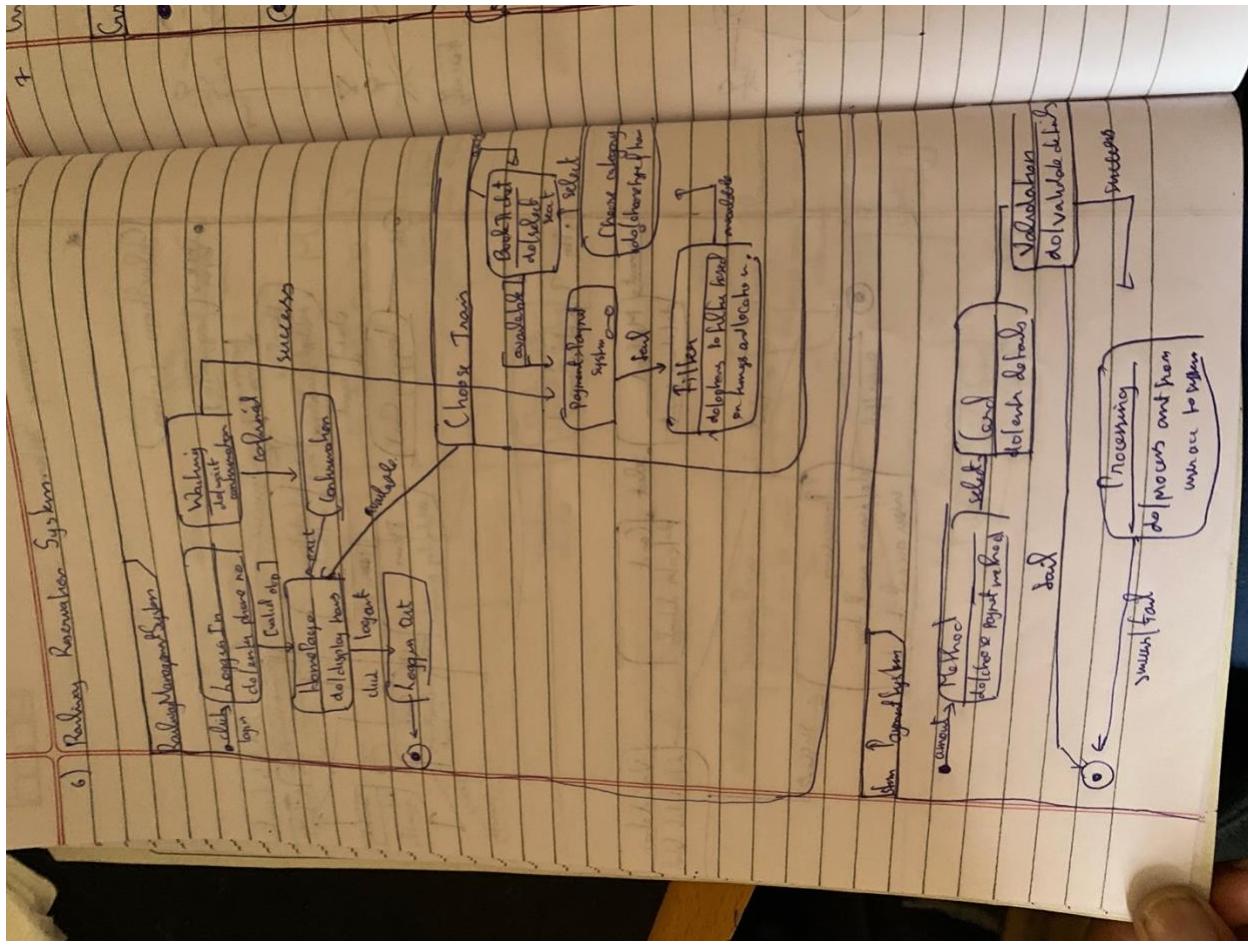
Class Diagram:



State Diagram:

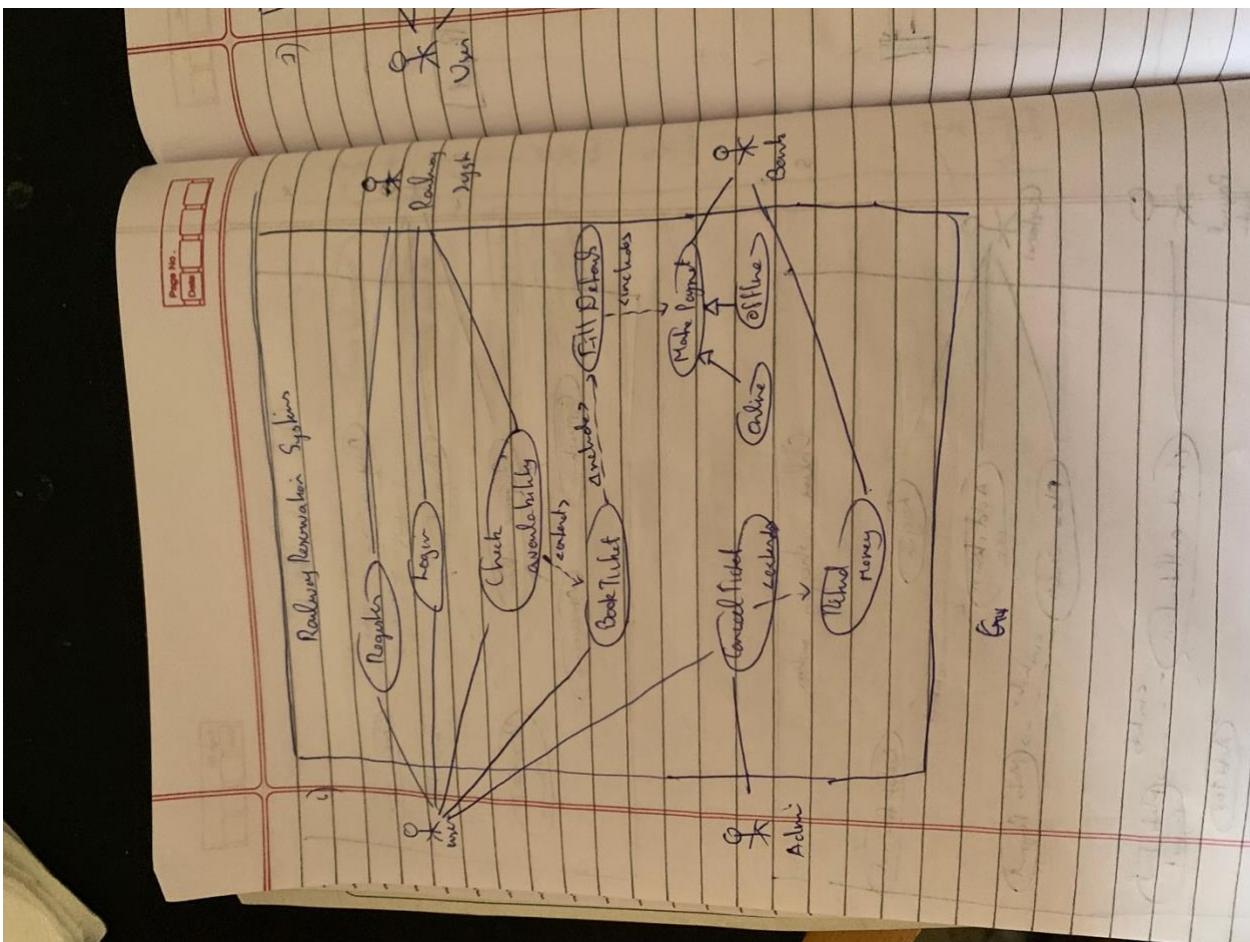
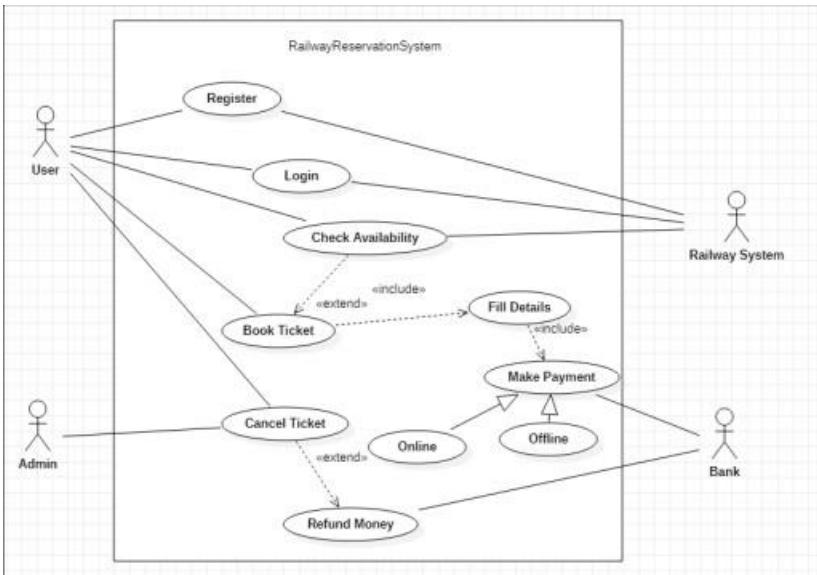
The advanced state diagram depicted below contains one nested state and one submachine, which on successful login shows the ChooseTrain details and PaymentSystem procedure. It contains initial state and termination state with ChooseTrain as a nested state including the required simple states. It also has a submachine state named PaymentSystem with initial, termination state along with simple states; Method, Card, Validation, Processing.





Use Case Diagram:

The advanced use case diagram has extra functionalities which includes extends, includes and generalization. The cancel ticket use case extends refund money use case, check availability use case extends book ticket use case, book ticket use case includes fill details use case, fill details use case includes make payment. Online and offline is generalized to super class make payment.

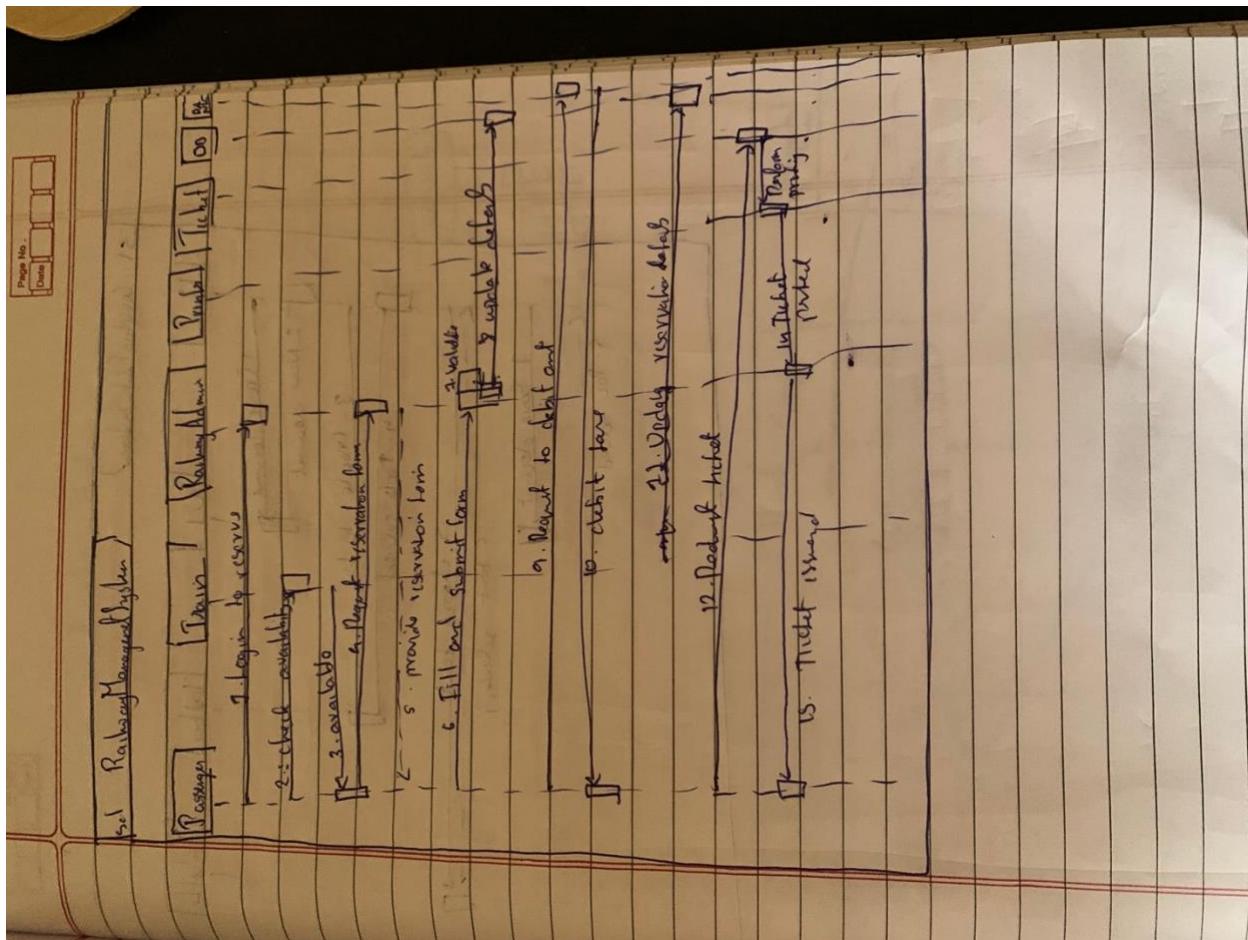
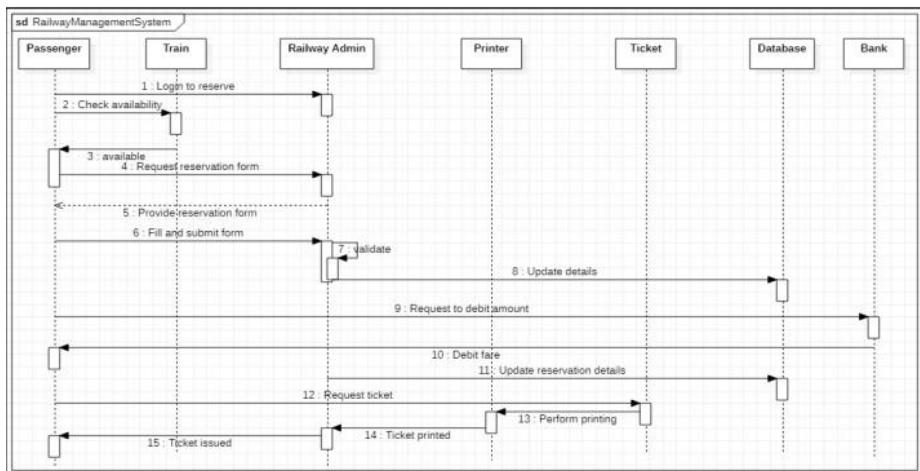


Sequence Diagram

The lifeline is the dotted line and the rectangles represent the period of time the object is

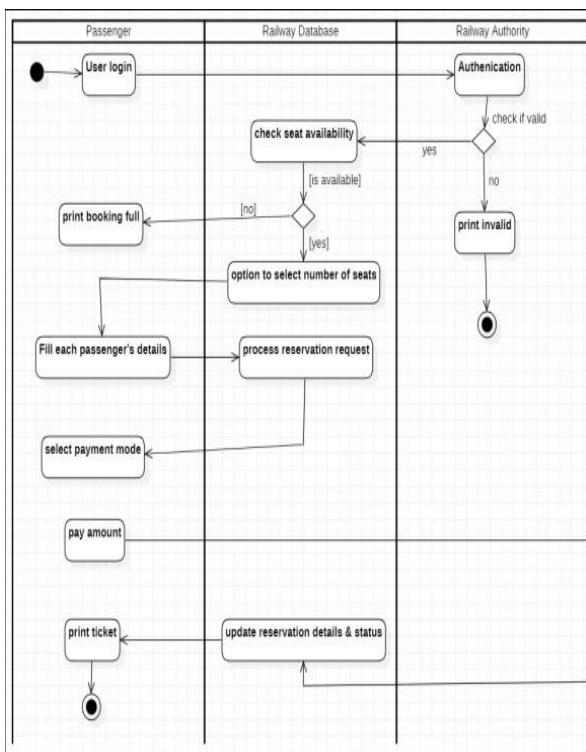
executing and is hence called activation.

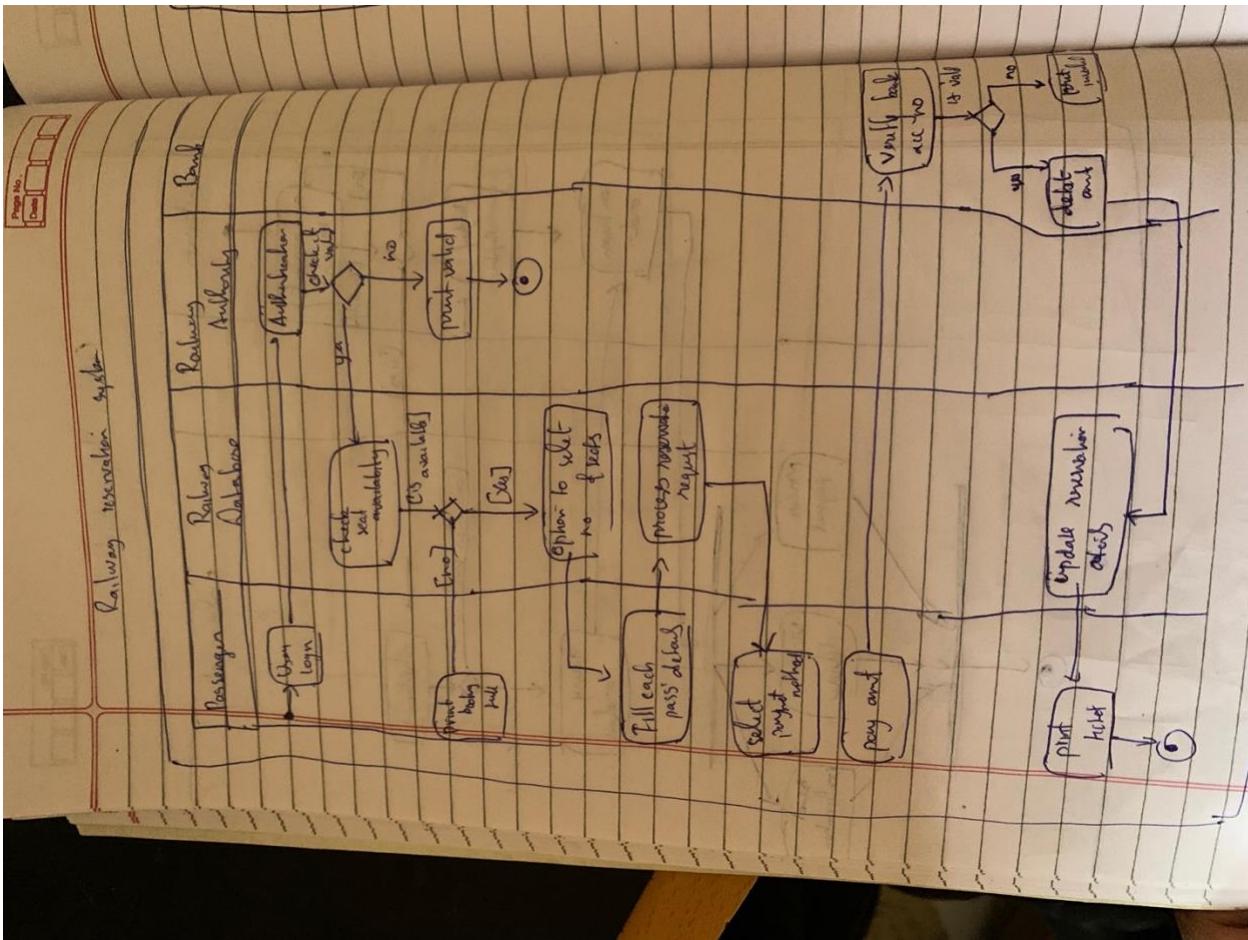
The recursive function of validation is shown by double activation rectangle of validation with self-transition and verify user. Reply message is used to return back to lifelines with the required message.



Activity Diagram:

The advanced activity diagram starts from initiation and in the passenger swimlane, the passenger login activity where a signal is sent to the network for request validation and upon confirmation the control flows to check seat availability activity. There are four swimlanes namely passenger, railway database, railway authority and bank where each one indicates the passenger operations, check seat availability, check validation, confirm payment respectively. Then the control flows to the home page and then termination activities.





7. Graphics Editor

Problem statement:

Design UML diagrams for Graphics Editor with system requirements specification.

Software Requirements Specification (SRS):

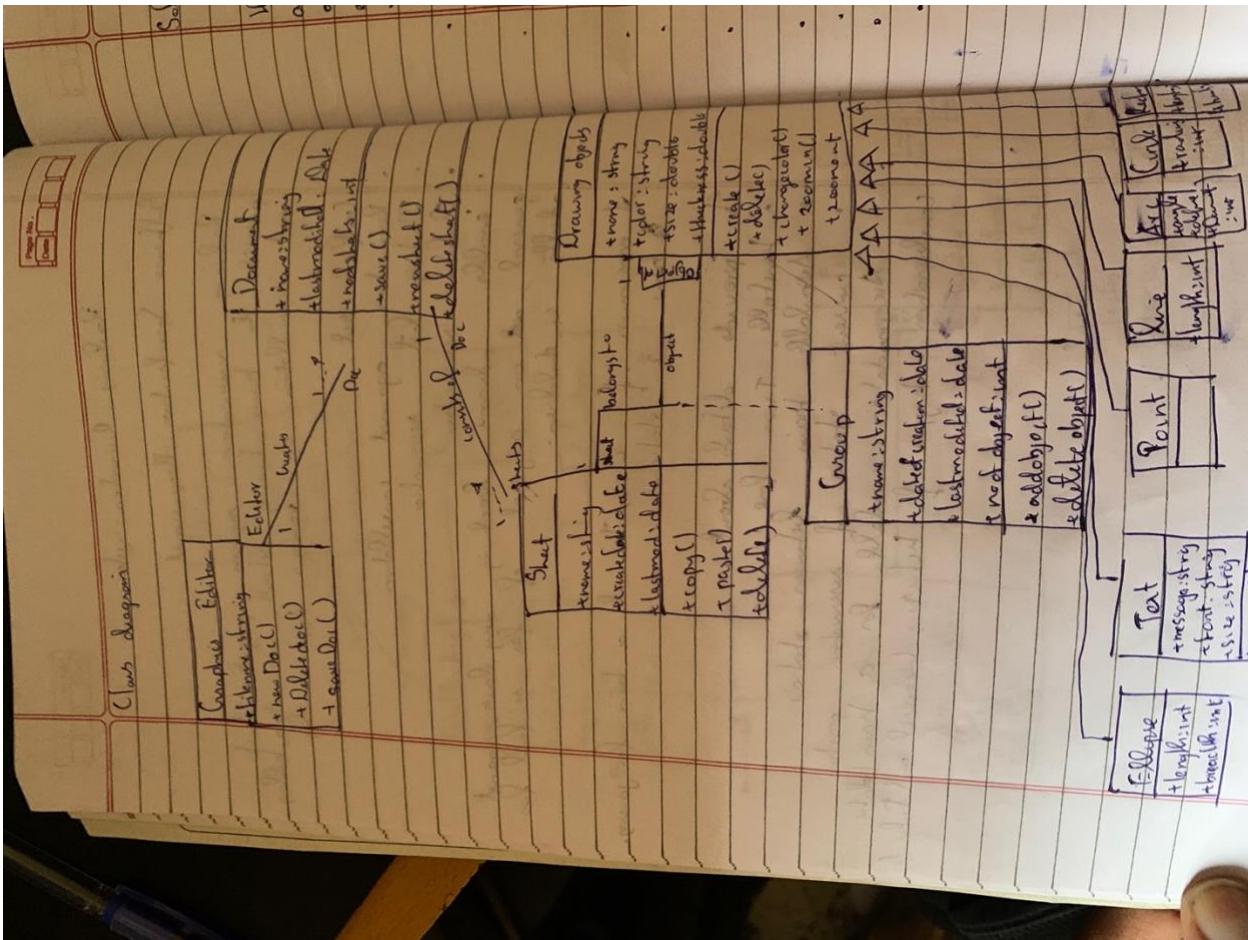
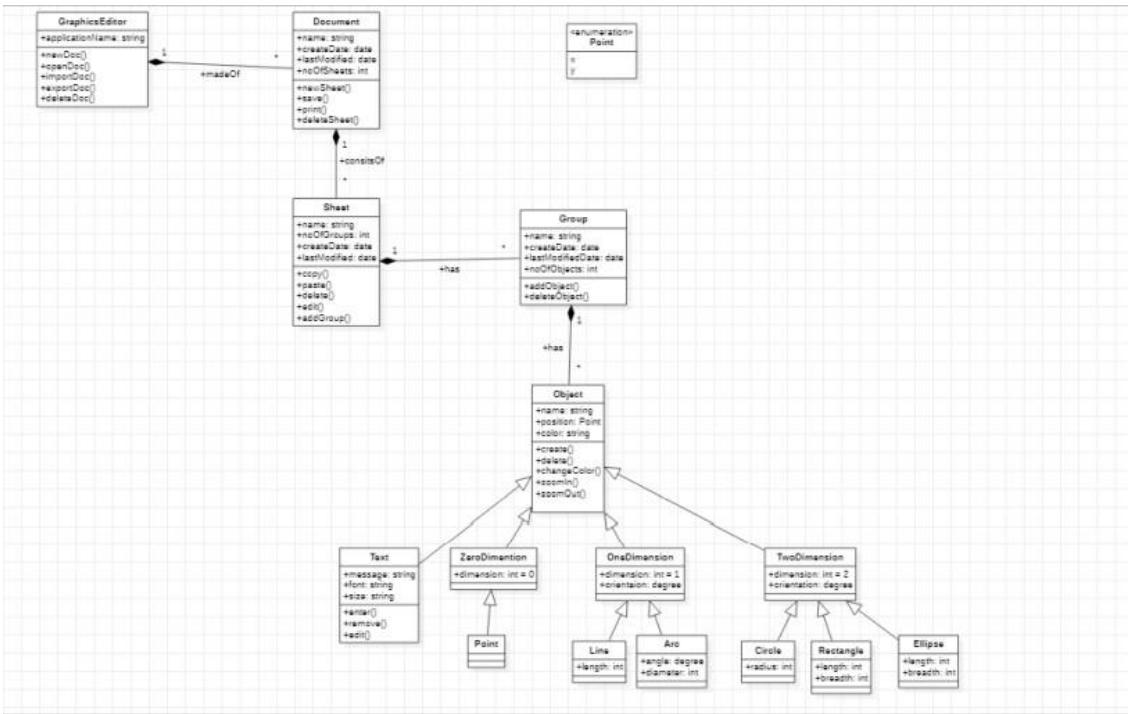
The graphics editor provides an Application Programmer's Interface that enables a programmer to develop their own graphical model editor for a specific type of model. This API in turn, relies

on extending the Eclipse Graphical Editing Framework to provide an environment in which the editor functions, and the programmer can create a graphical editor and palette of shapes in order to modify an underlying model. The graphical editor provides an interface with which the programmer implements said editor for a given underlying model. Such an instance of the graphical editor allows a user to drag objects from a specified model into a working graphical diagram.

It should support following functionalities:

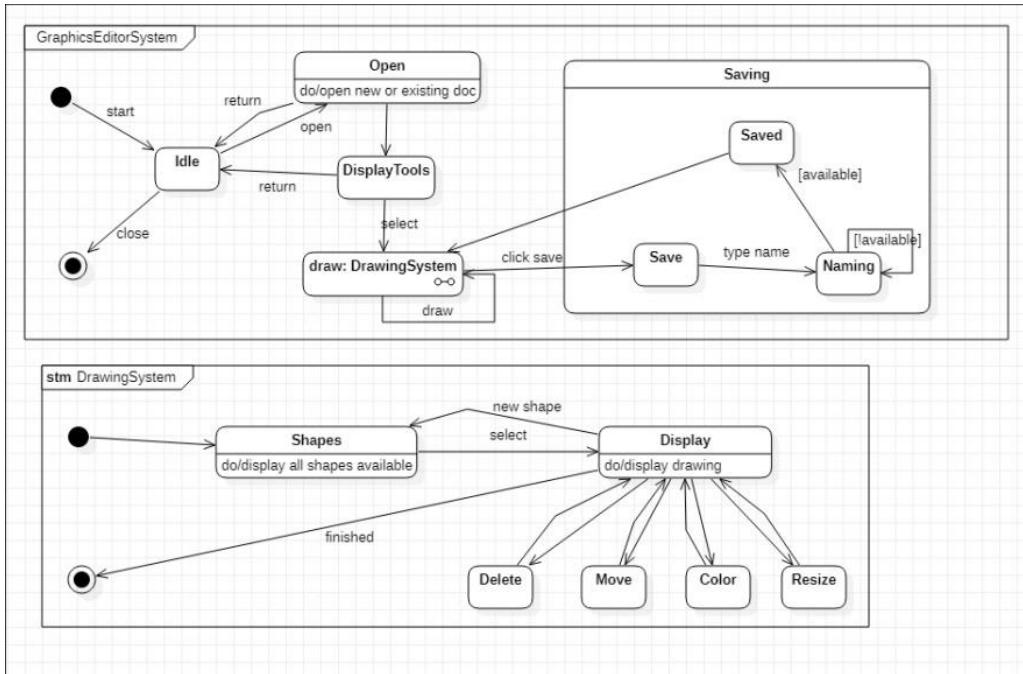
- It contains the toolbox which contains tools like: Line, Circle, Rectangle, Arc,Text, Draw,Eraser
- Color box or palette
- Standard toolbar with options for New, Open, Save, toolbox and Text Toolbox.
- One integrated view to users for toolbar, color box, menu, and graphic screen.
- Easy handling of tools for users.
- Ability to group several drawings into one i.e., complex drawing.
- Provision of zoom in and zoom out.
- Different shadings of line tool are provided

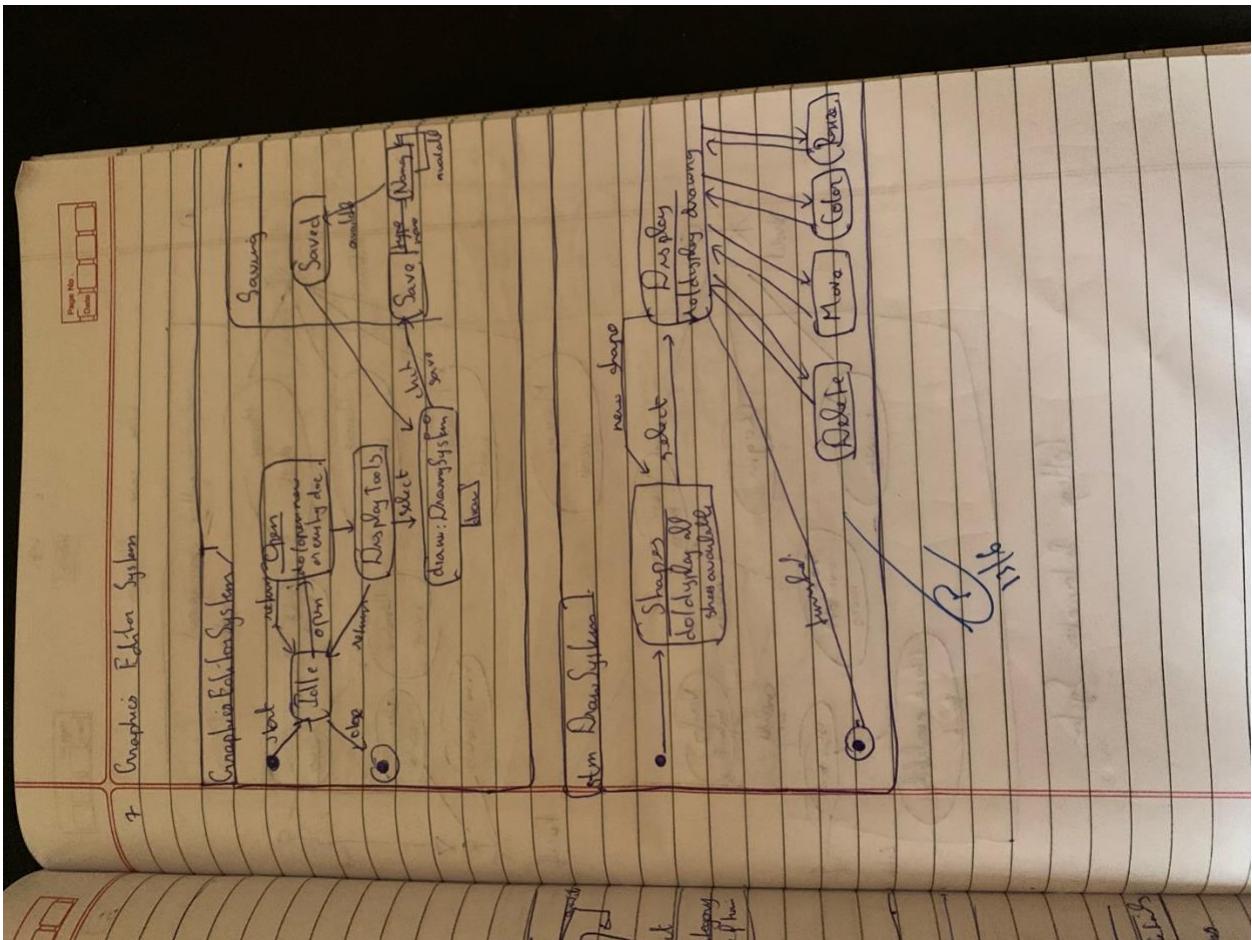
Class Diagram:



State Diagram:

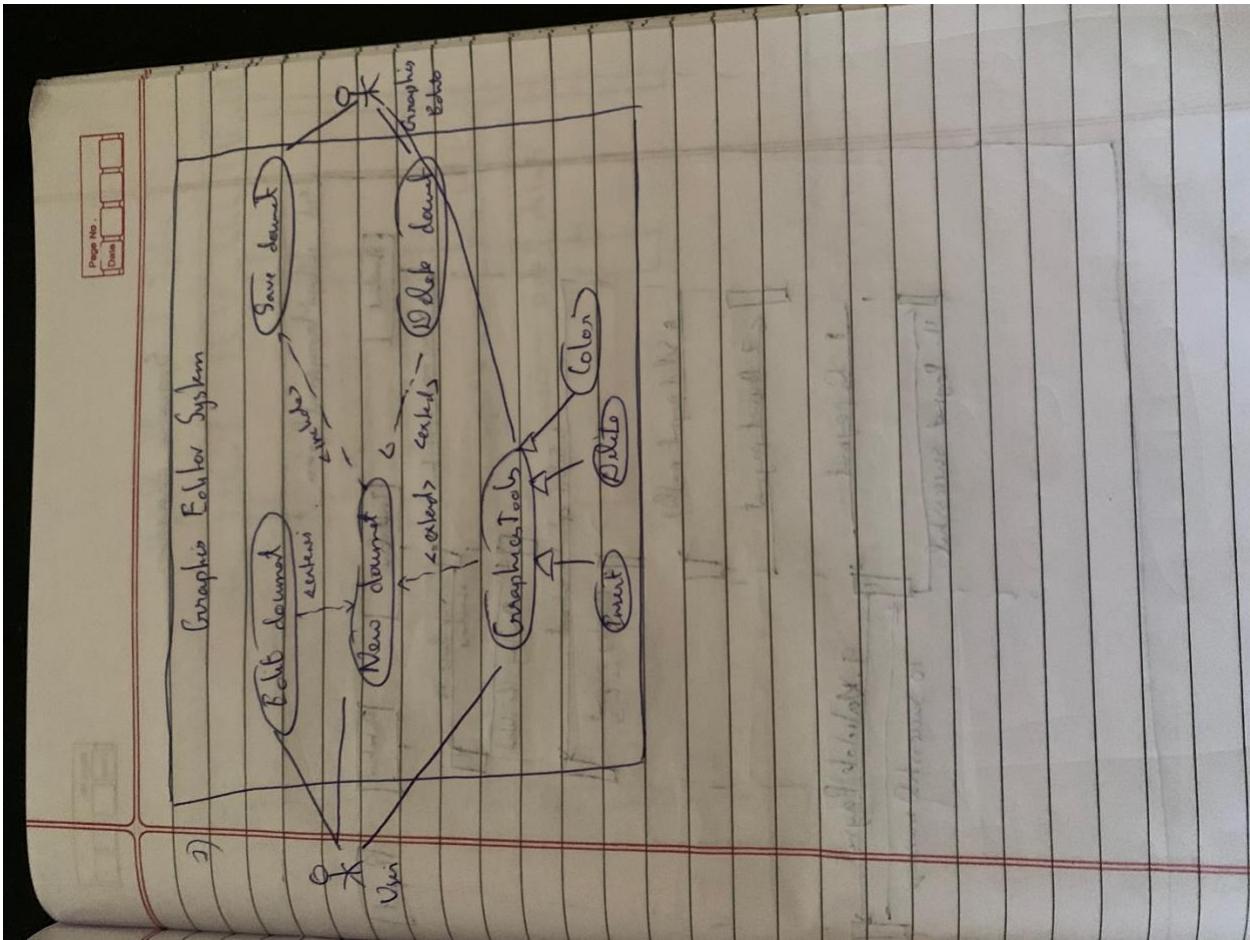
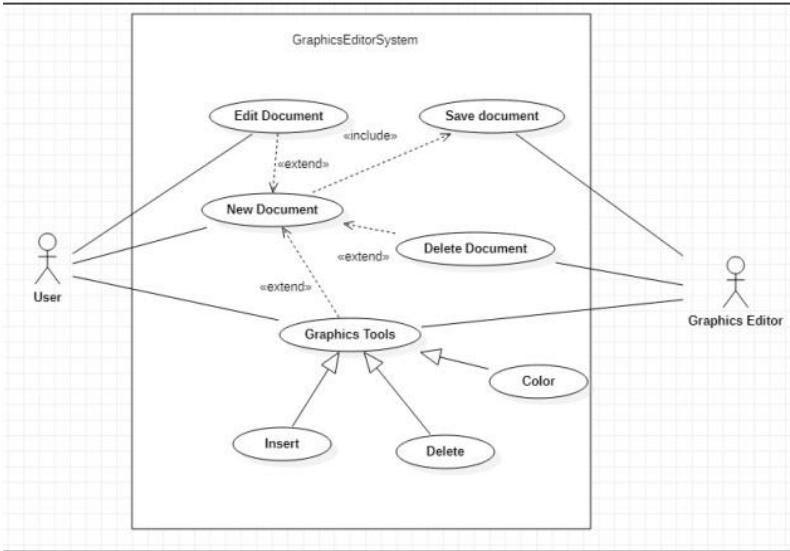
The advanced state diagram depicted below contains one nested state and one submachine, which on successful login shows the Saving procedure and Drawing System procedure. It also has a submachine state named Drawing System with initial, termination state along with simple states; Shapes, Display and format each shape.





Use Case Diagram:

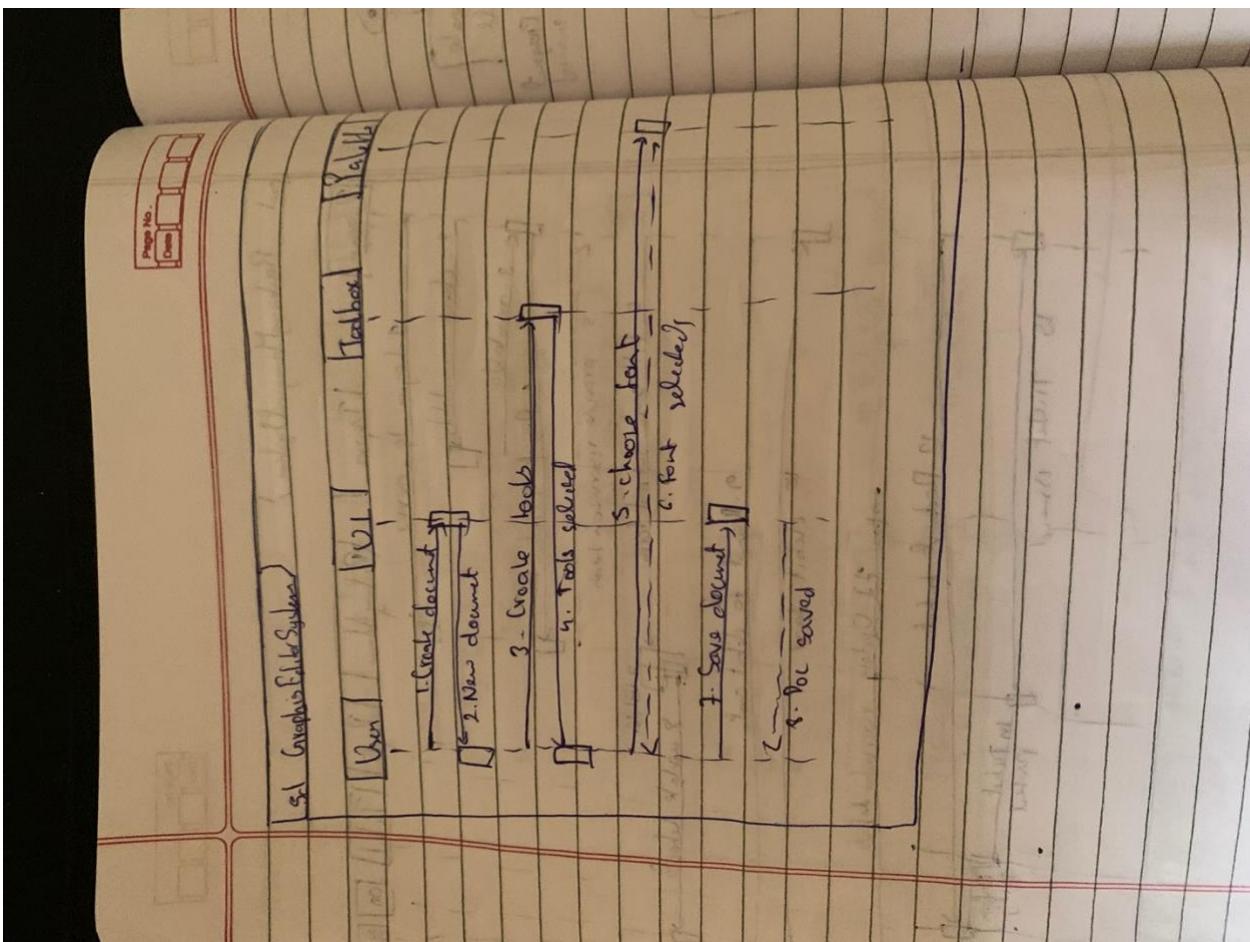
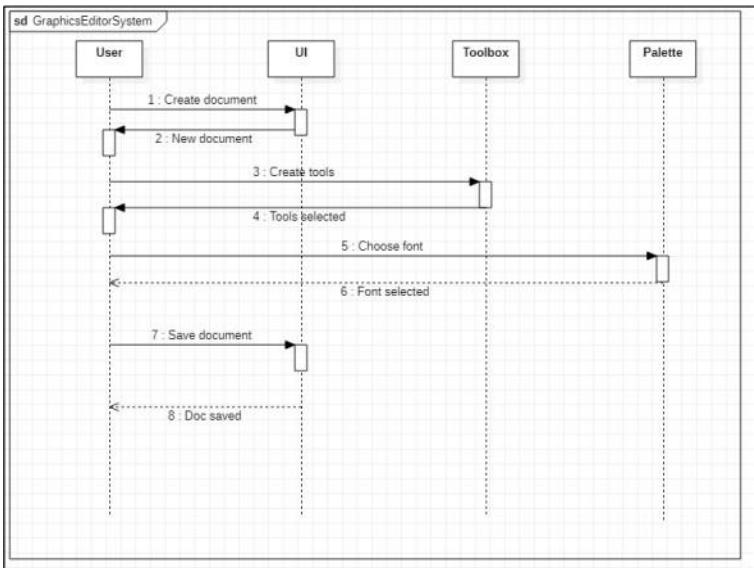
The advanced use case diagram has extra functionalities which includes extends, includes and generalization. The edit document use case extends new document use case, delete document usecase extends new document use case, graphic tools use case extends new document use case,new document use case includes save document use case. Insert, delete and color is generalizedto super class graphics tools.



Sequence Diagram:

The lifeline is the dotted line and the rectangles represent the period of time the object is executing and is hence called activation.

Reply message is used to return back to lifelines with the required message.



Activity Diagram:

The advanced activity diagram starts from initiation and in the user swimlane, the user login activity where a signal is sent to the network for request validation and upon confirmation the control flows to open file activity. There are two horizontal swimlanes namely user and editor where each one indicates the user operations and drawing a diagram respectively. Then the control flows to the close file activity and then termination activities.

