VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

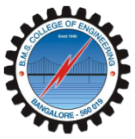
LAB REPORT
on

Object Oriented Modeling and Design

Submitted by

Dhruv Dubey (1BM19CS048)

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 "LAB COURSE **OBJECT ORIENTED MODELING AND DESIGN**" carried out by **DHRUV DUBEY (1BM19CS048),** 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 OBJECT **ORIENTED MODELING AND DESIGN - (20CS6PCOMD)** work prescribed for the said degree.

Dr. Latha NR
Assistant Professor
Department of CSE
BMSCE, Bengaluru

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

`

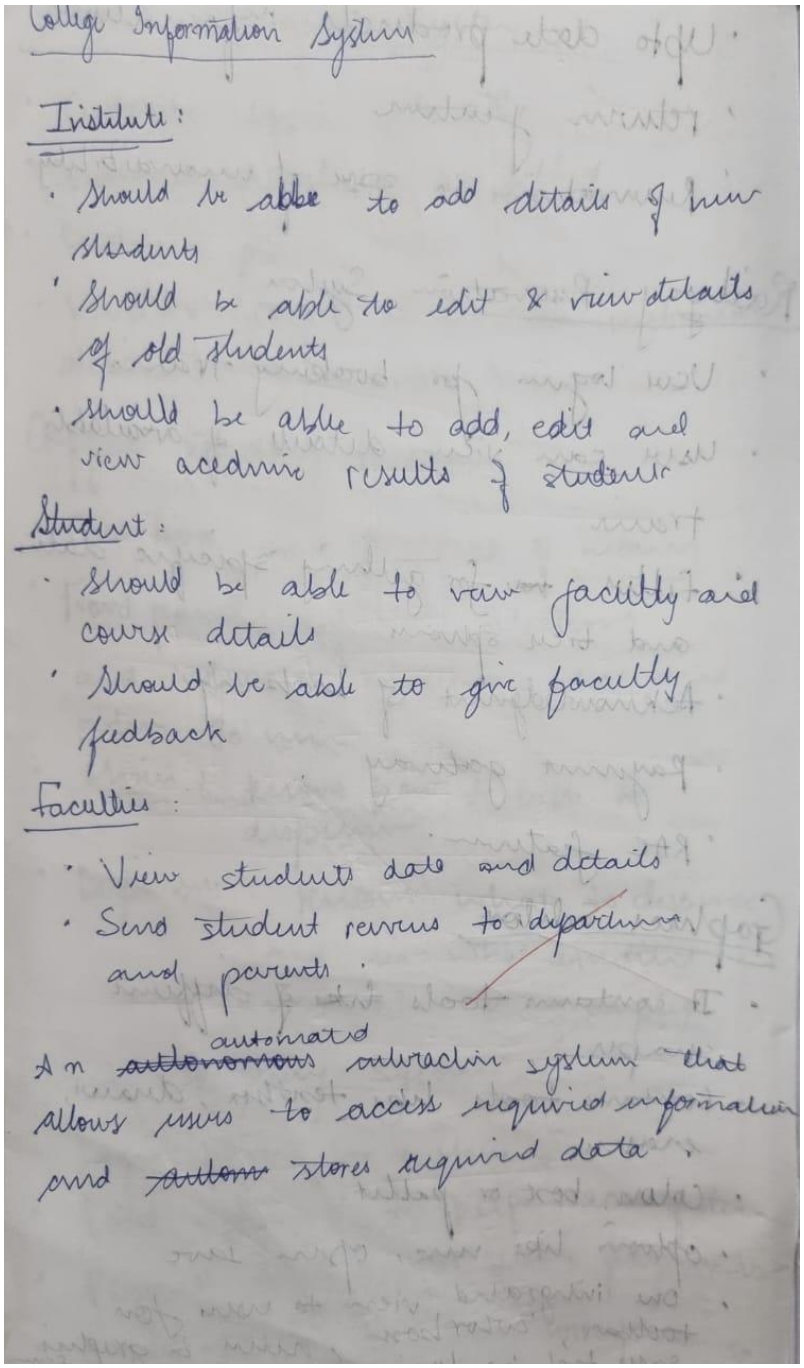
Index Sheet

Course Outcome

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

Exercise 1: College Information System

1. SRS



College Information System

Institute:

- Should be able to add details of new students
- Should be able to edit & view details of old students
- Should be able to add, edit and view academic results of students

Student:

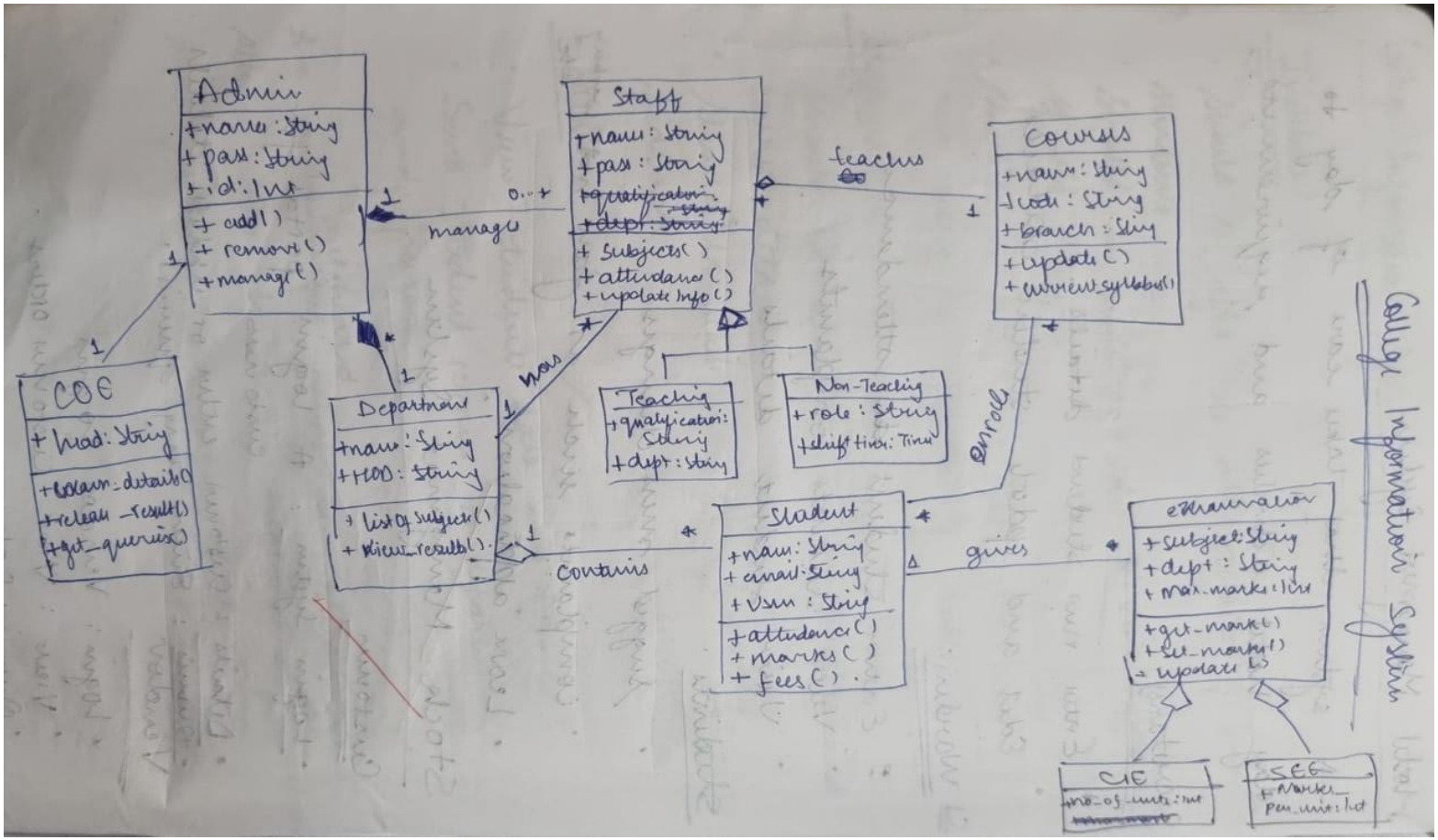
- Should be able to view faculty and course details
- Should be able to give faculty feedback

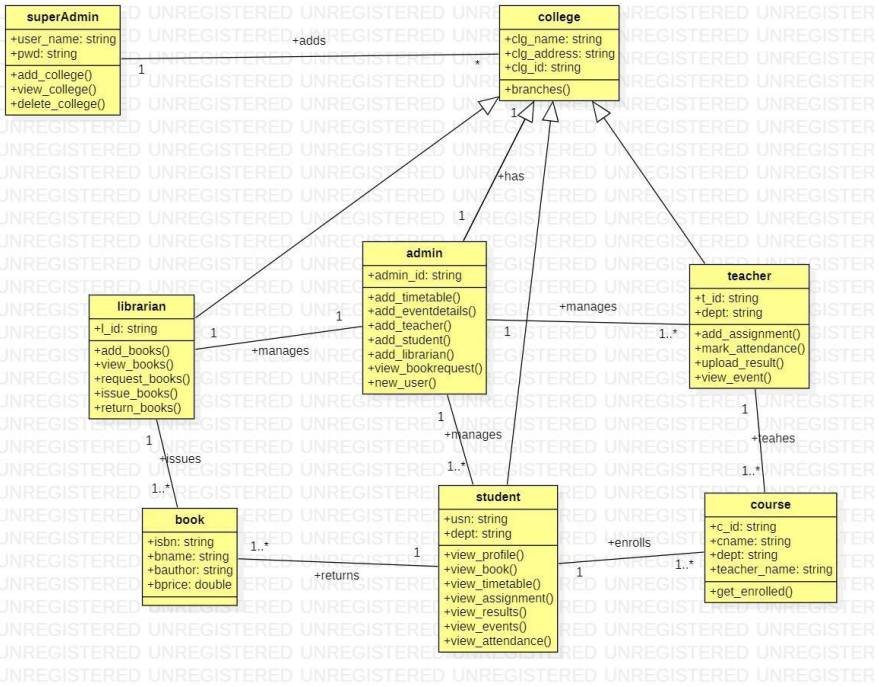
Faculties:

- View students data and details
- Send student reports to department and parents

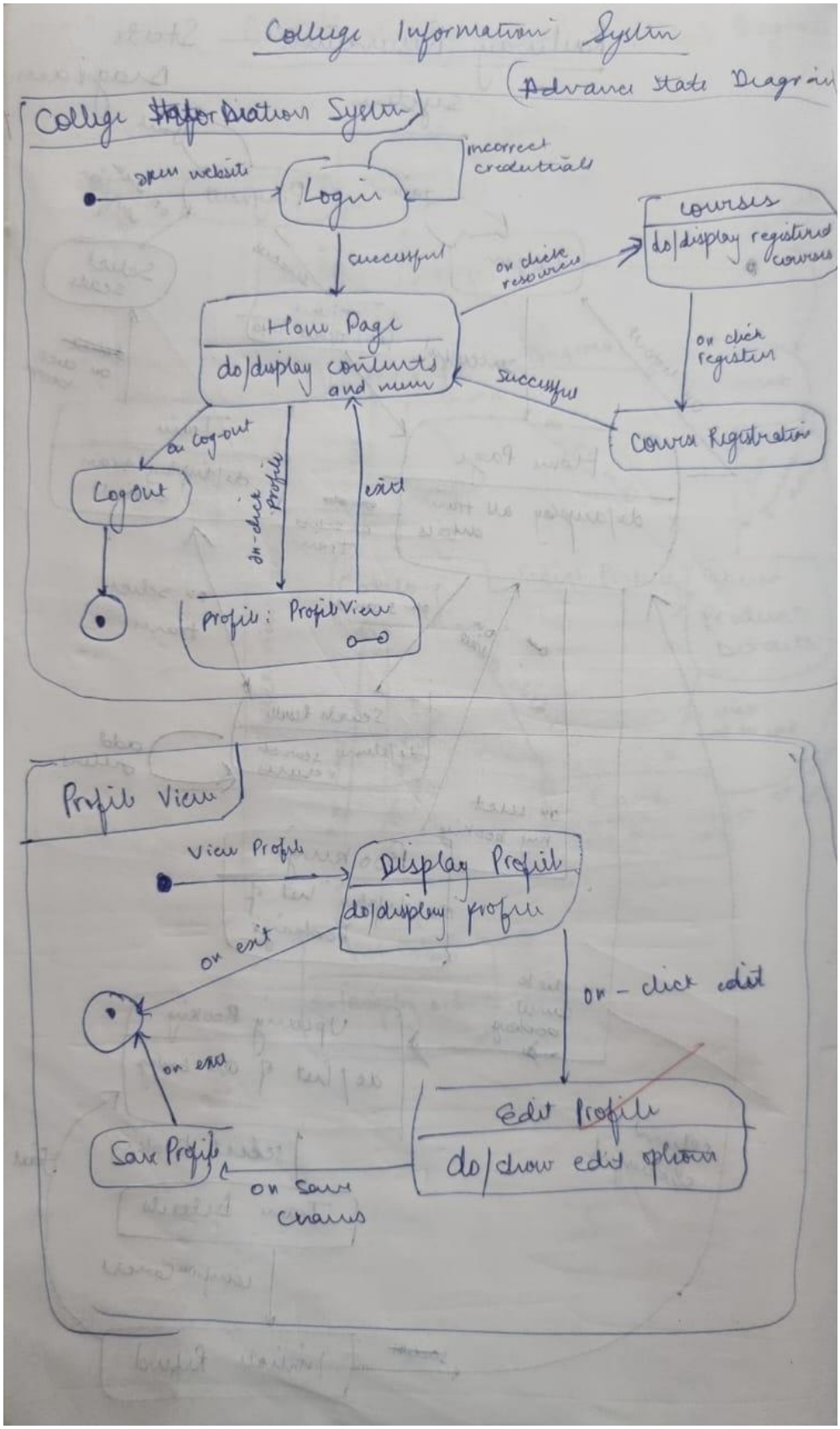
An ~~autonomous~~ automated attendance system that allows users to access required information and ~~autom~~ stores required data.

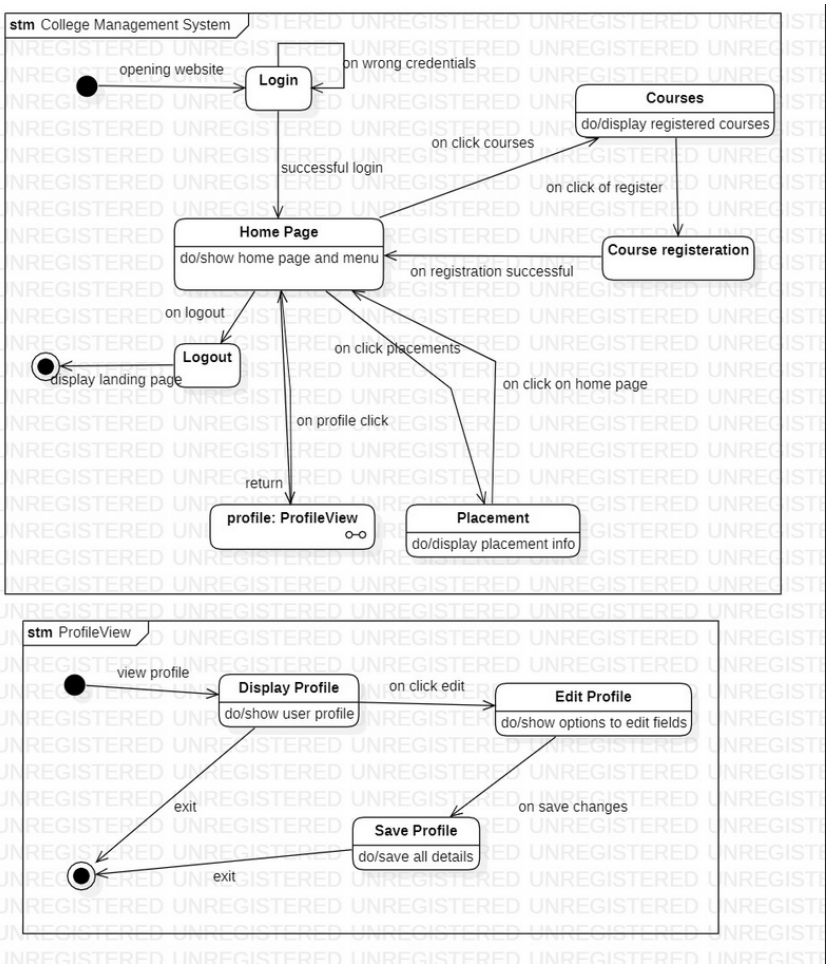
2. advanced class diagram


Admin
+name: String
+pass: String
+id: Int
+ add()
+ remove()
+ manage()
manage
0..*
1
Staff
+name: String
+pass: String
+qualification: String
+dept: String
+ subjects()
+ attendance()
+ update info()
teaches
Courses
+name: String
+code: String
+branch: String
+ update()
+ review_syllabus()
COE
+ head: String
+ exam_details()
+ release_result()
+ get_queries()
Department
+name: String
+HOD: String
+ list of subjects()
+ view_results()
has
Teaching
+qualification: String
+dept: String
Non-Teaching
+ role: String
+ shift time: Time
enrolls
Student
+name: String
+ email: String
+ USN: String
+ attendance()
+ marks()
+ fees()
contains
gives
Examination
+ subject: String
+ dept: String
+ max_marks: Int
+ get_mark()
+ set_marks()
+ update()
CIE
no_of_units: Int
assignment
SEE
+ marks_per_unit: Int
College Information System



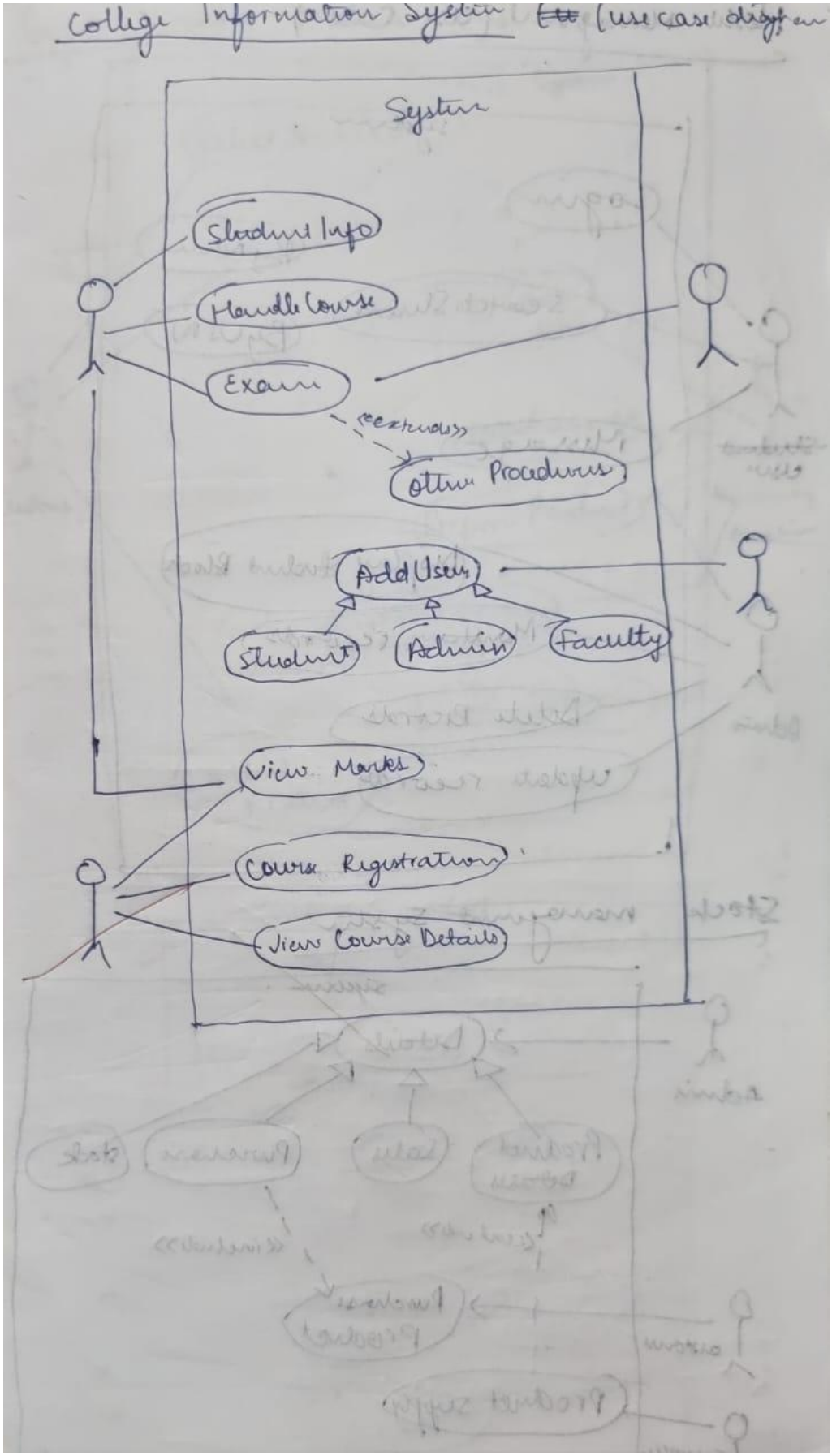
3. advanced state diagram

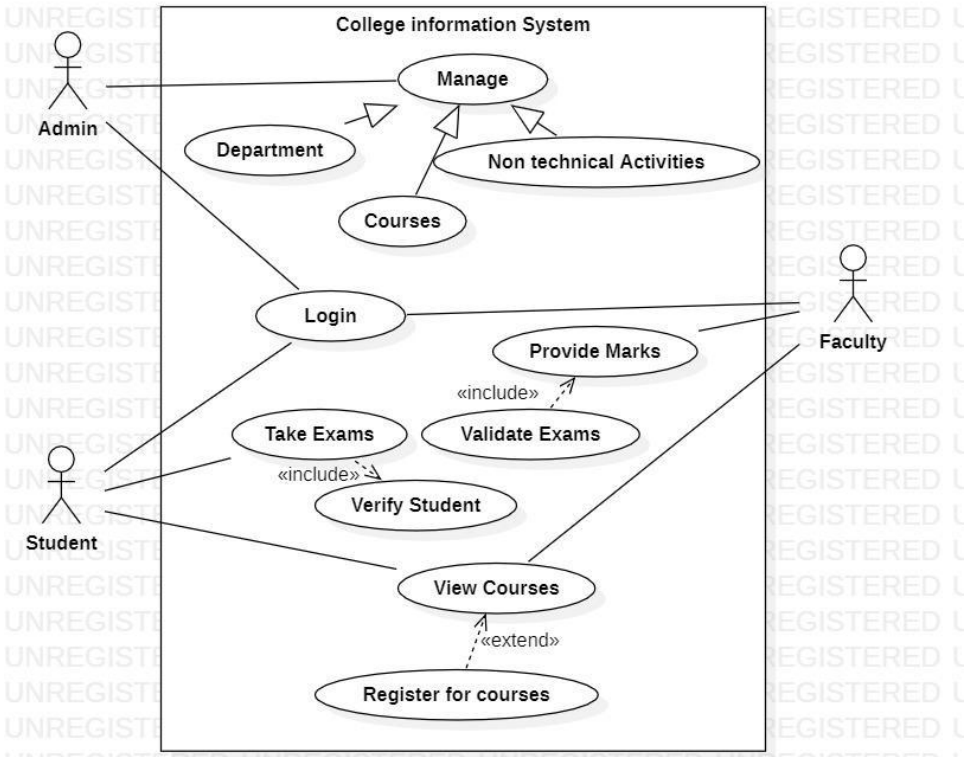

College Information System
(Advance State Diagram)
College Information System
open website
Login
incorrect credentials
successful
courses
do/display registered courses
on click resources
on click register
Home Page
do/display contents and menu
successful
Course Registration
on log-out
Log out
on-click Profile
exit
Profile : ProfileView
Profile View
View Profile
Display Profile
do/display profile
on exit
on-click edit
on exit
Save Profile
on save changes
Edit Profile
do/show edit options



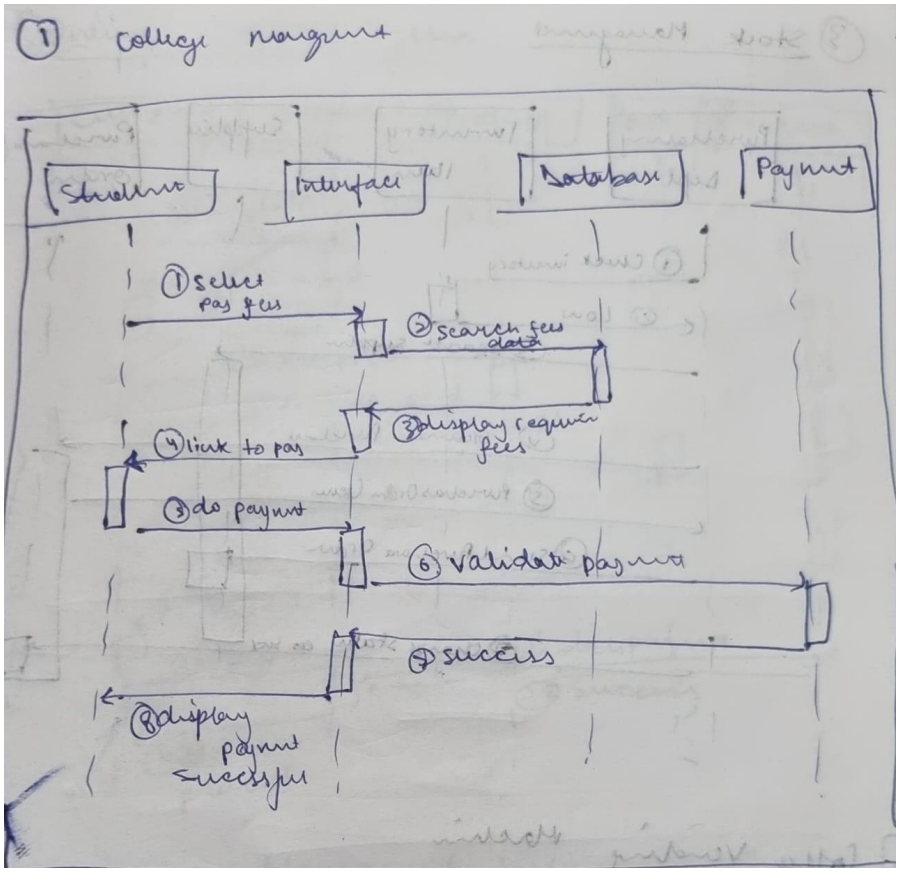
4. advanced use case diagram

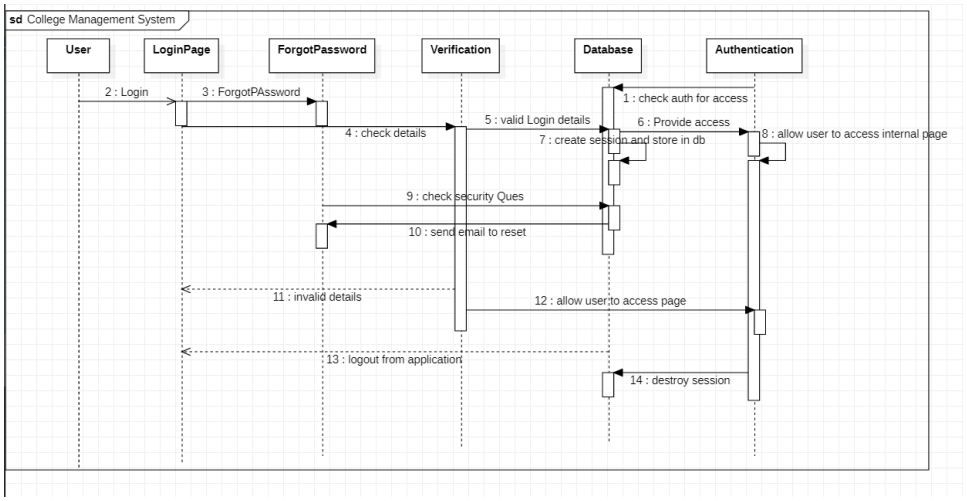
College Information System (use case diagram





5. advanced sequence diagram

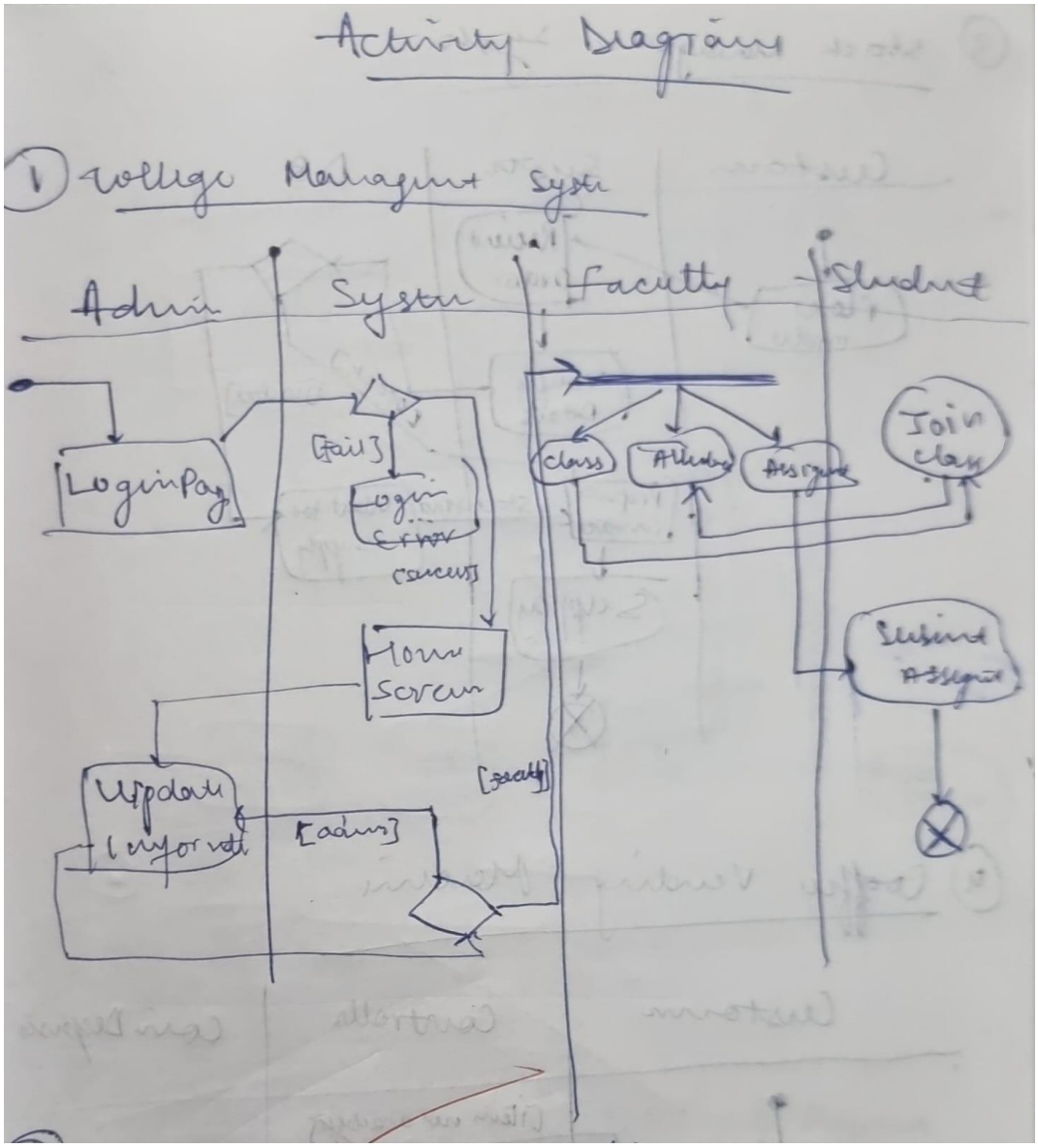


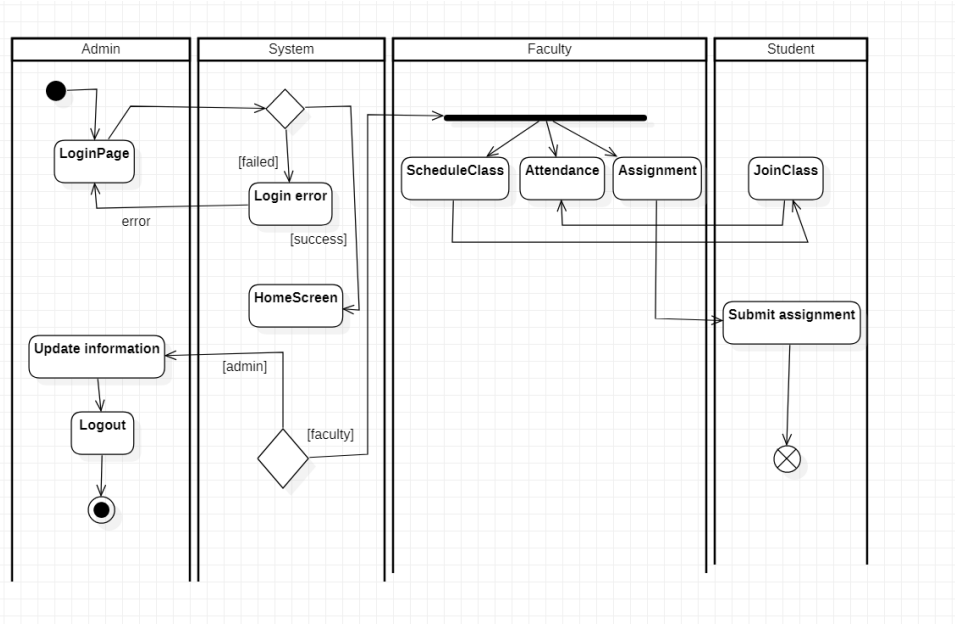


6. advanced activity diagram

Activity Diagram

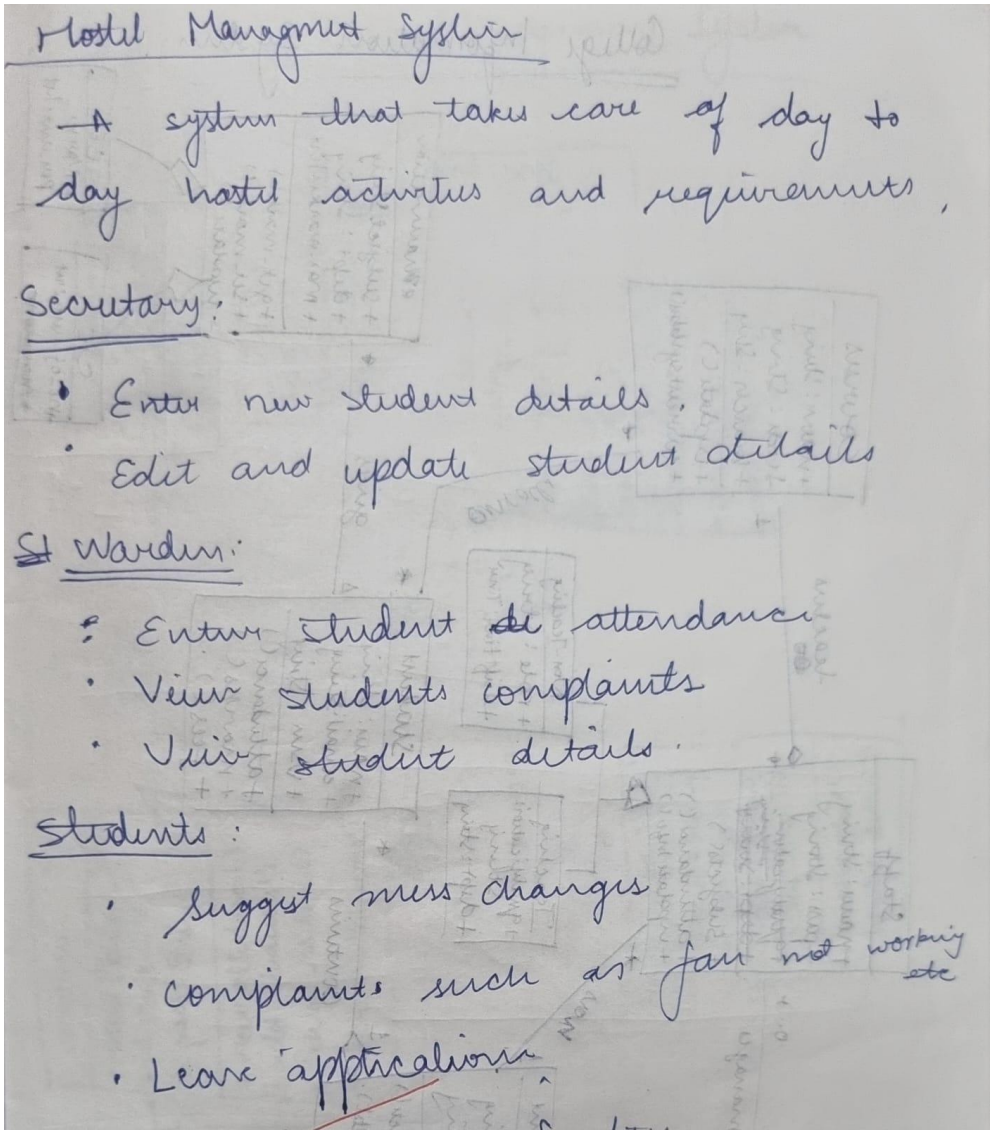
1) College Management Syst



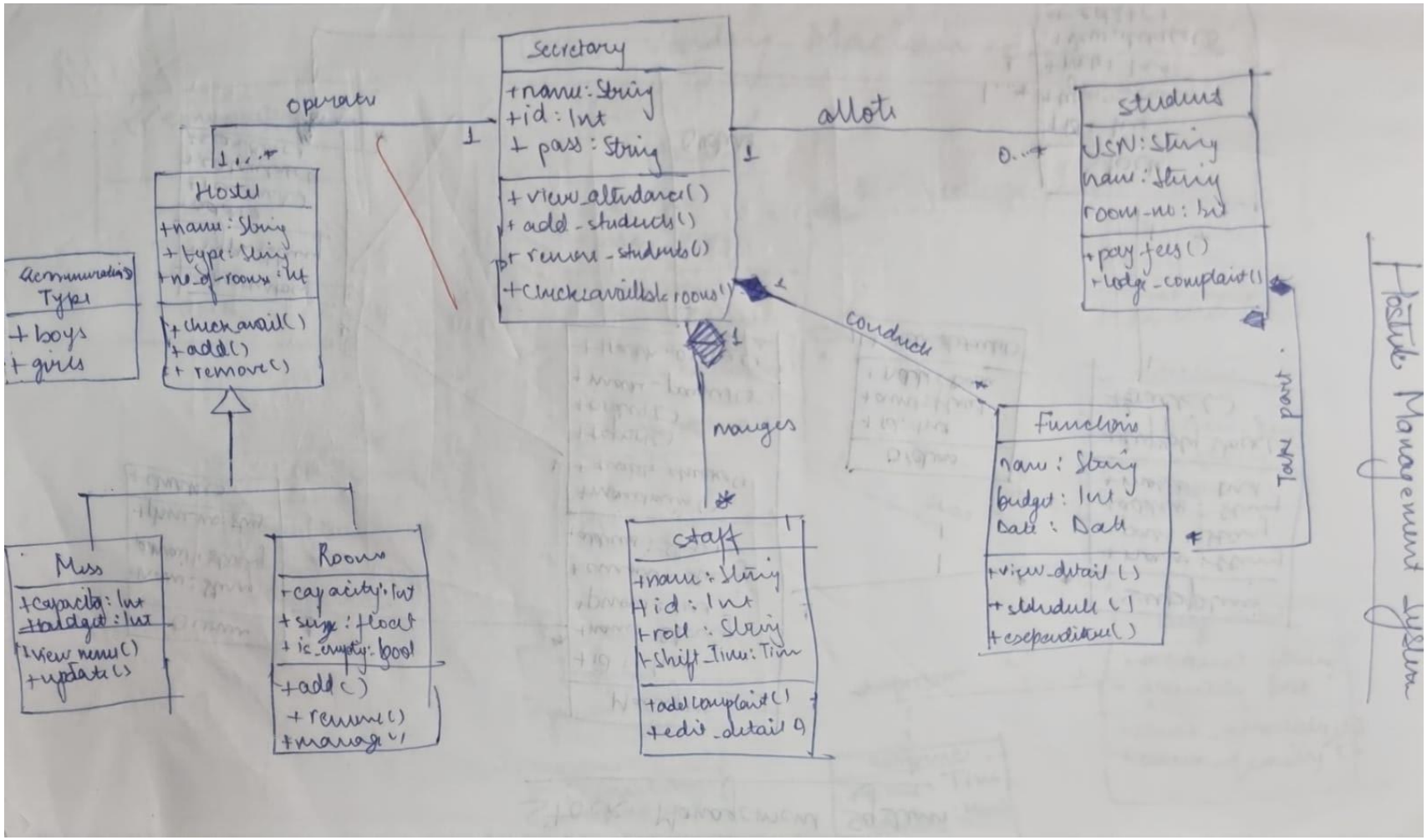


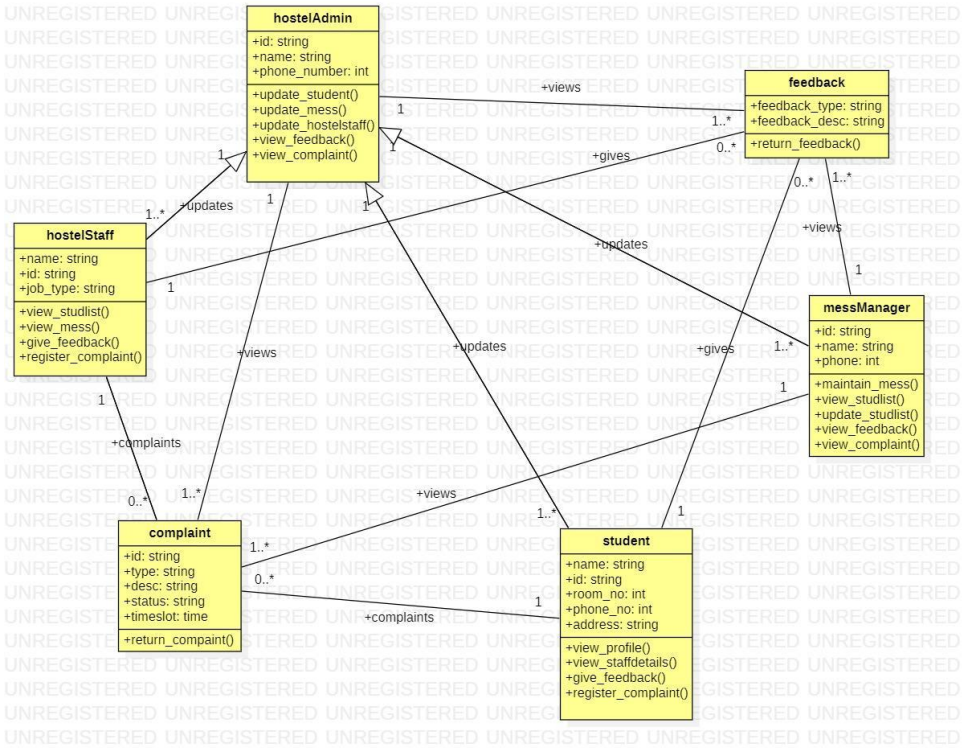
Exercise 2: Hostel Management System

1. SRS



2. advanced class diagram

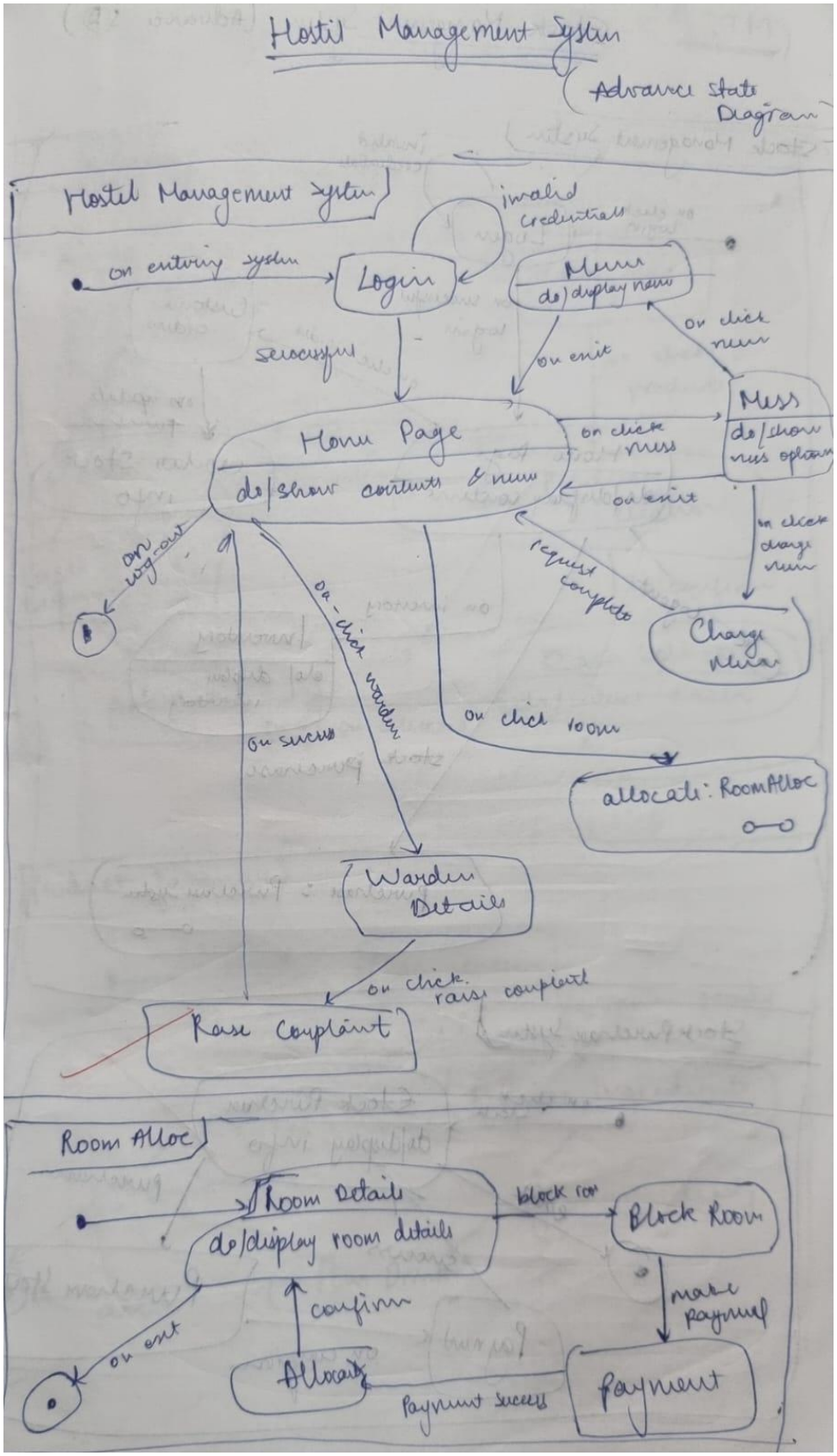

Hostel Management System
Secretary
+name: String
+id: Int
+pass: String
+view_attendance()
+add_students()
+remove_students()
+checkavailableroom()
operates
allots
conducts
manages
Hostel
+name: String
+type: String
+no_of_rooms: Int
+checkavail()
+add()
+remove()
Enumeration Type
+boys
+girls
Mess
+capacity: Int
+budget: Int
+view menu()
+update()
Room
+capacity: Int
+size: float
+is_empty: bool
+add()
+remove()
+manage()
staff
+name: String
+id: Int
+roll: String
+Shift_Time: Time
+add complaint()
+edit_detail()
Student
USN: String
name: String
room_no: Int
+pay_fees()
+lodge_complaint()
Functions
name: String
budget: Int
Date: Date
+view_detail()
+schedule()
+expenditure()
have part

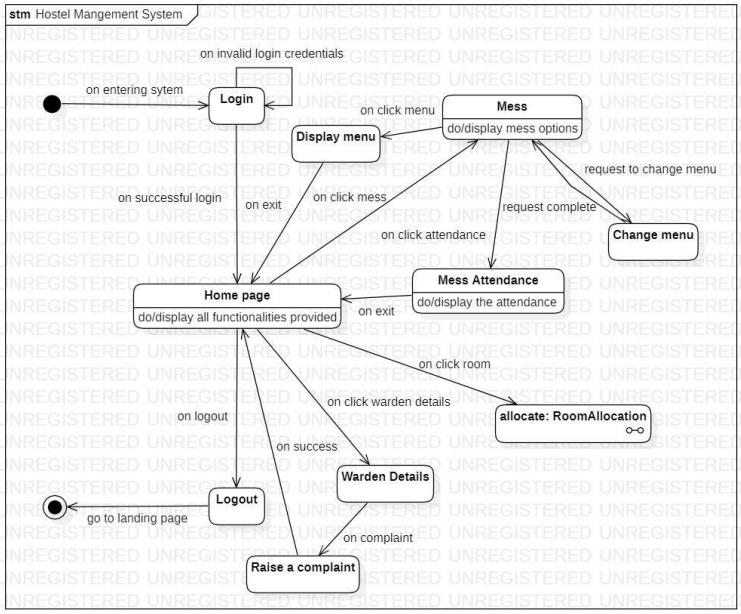


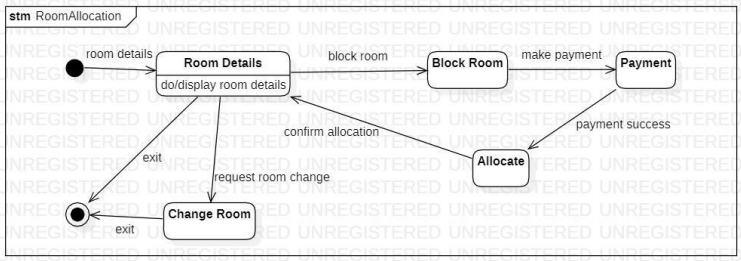
3. advanced state diagram

Hostel Management System

Advance State Diagram

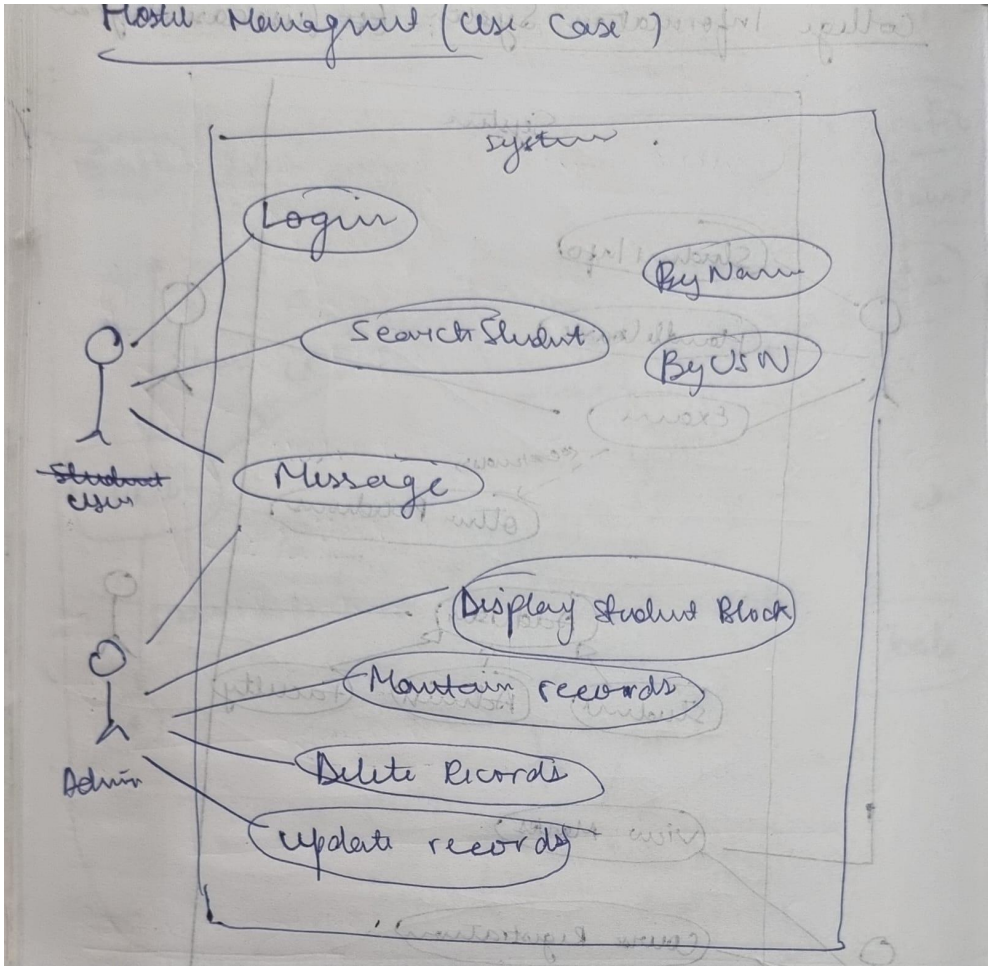


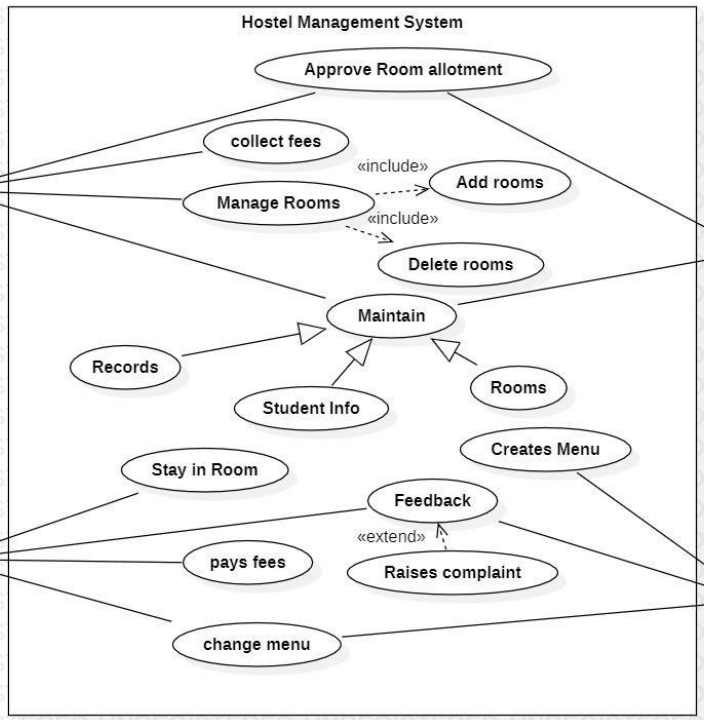




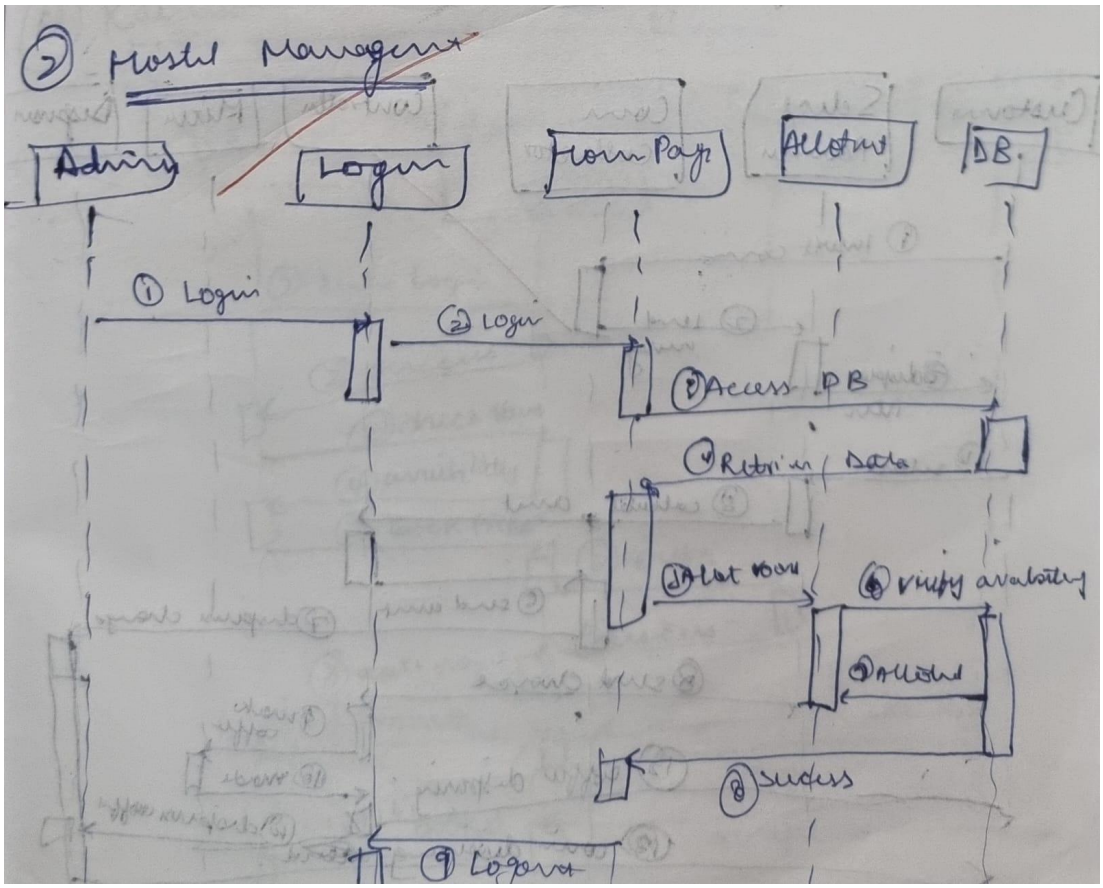
4. advanced use case diagram

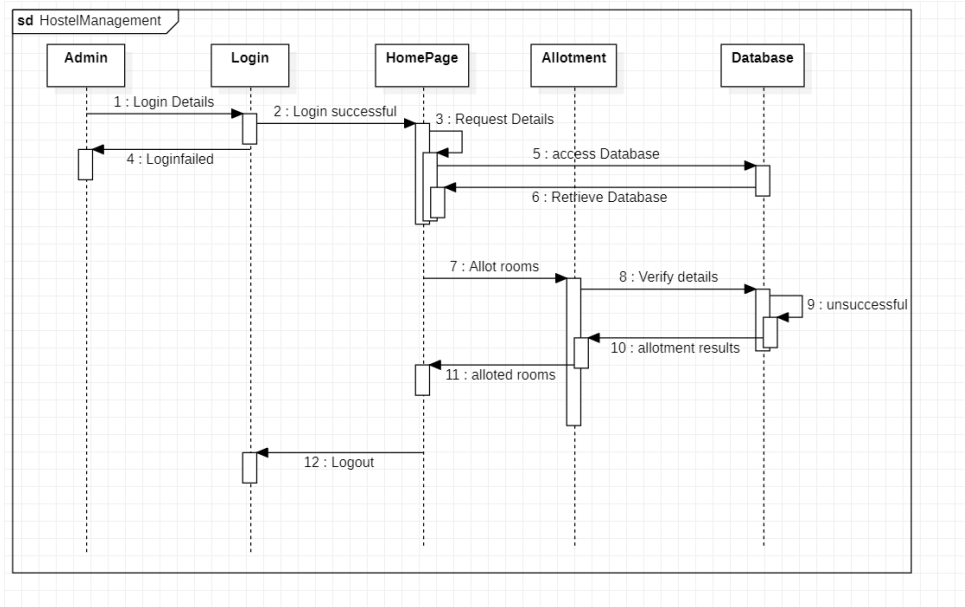
Hostel Management (use case)



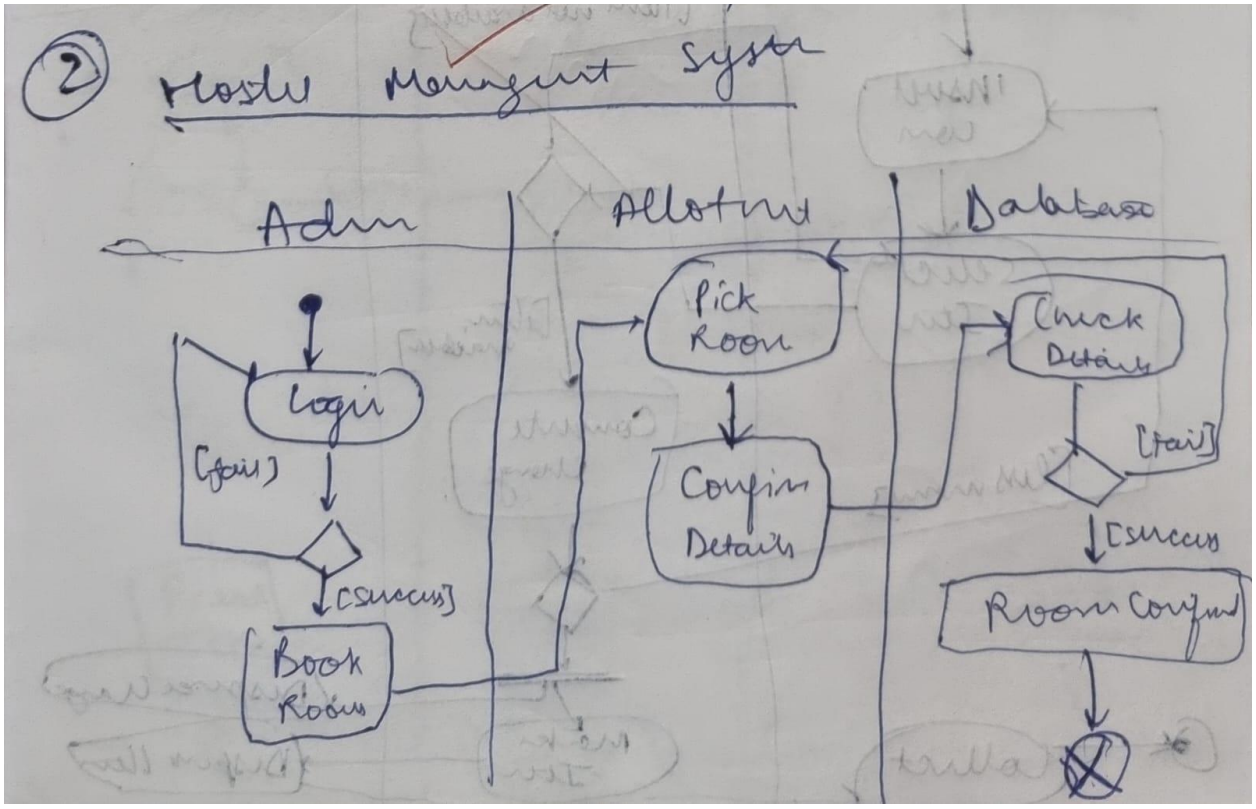


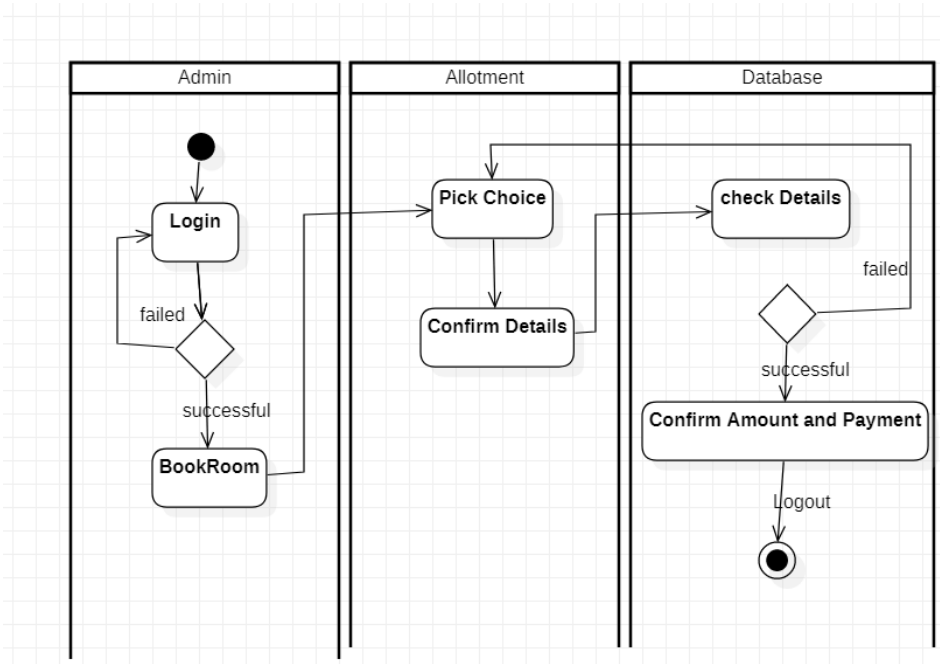
5. advanced sequence diagram





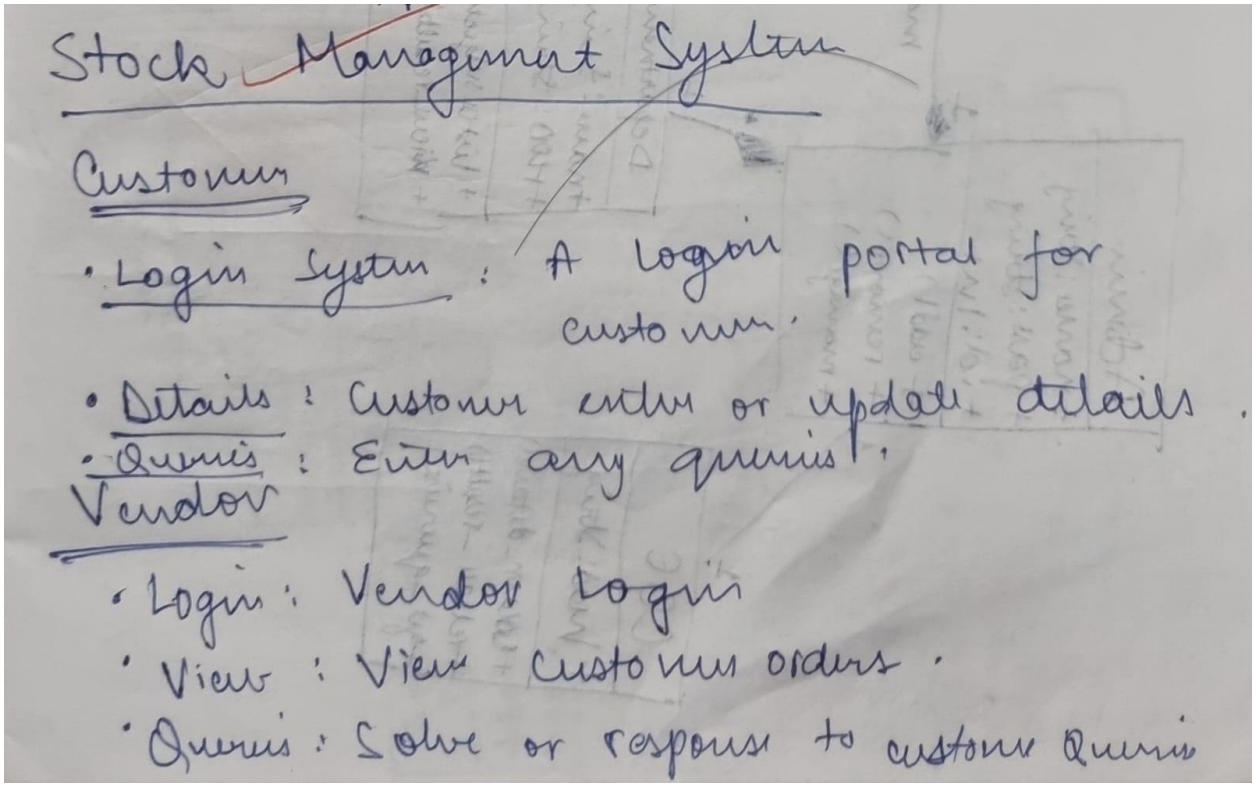
6. advanced activity diagram


Admin
Allotment
Database
Login
[fail]
[success]
Book Room
Pick Room
Confirm Details
Check Details
[fail]
[success]
Room Confirm


Admin
Allotment
Database
Login
failed
successful
BookRoom
Pick Choice
Confirm Details
check Details
failed
successful
Confirm Amount and Payment
Logout

Exercise 3: Stock Management System

1. SRS


Stock Management System

Customer

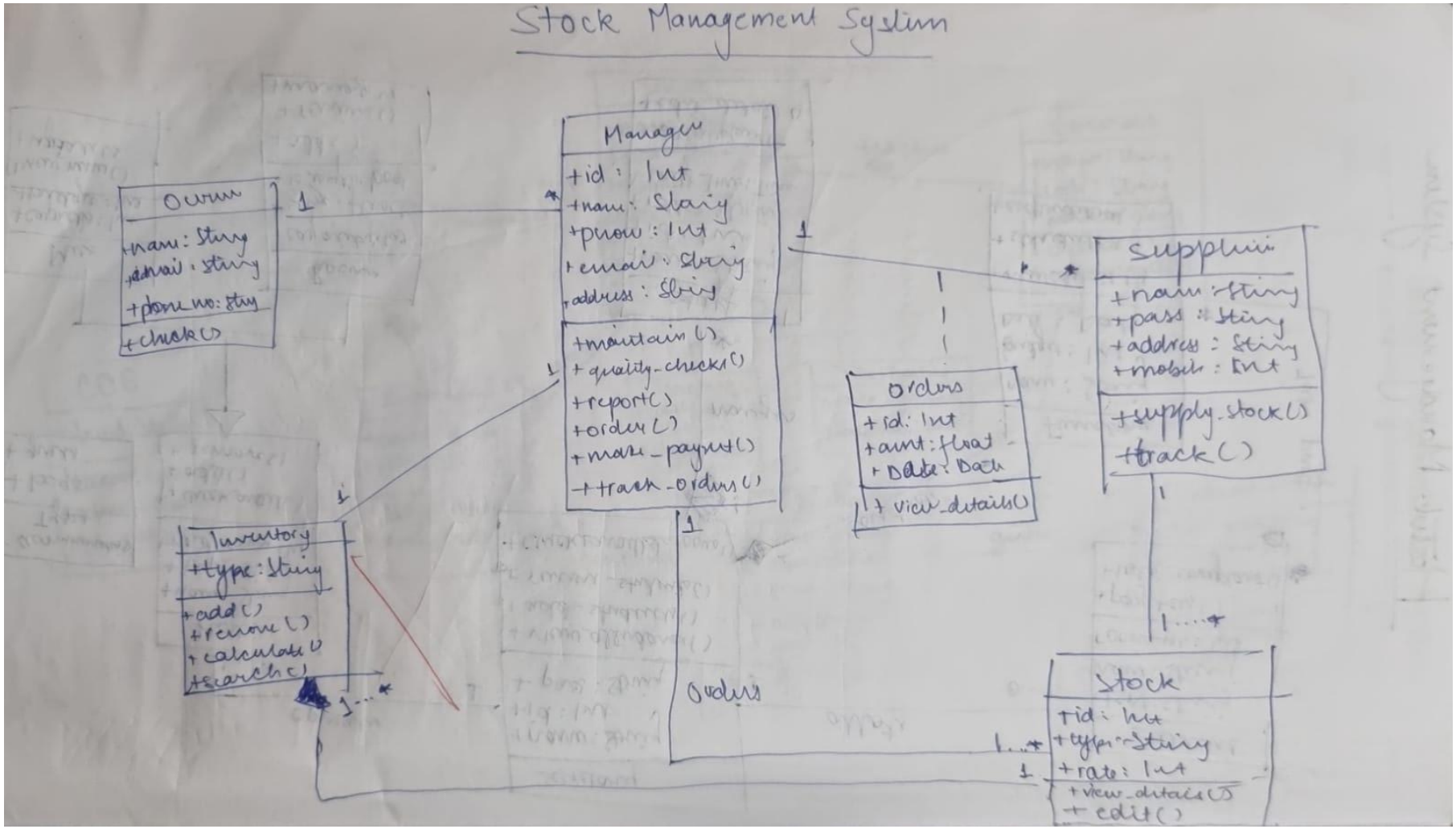
- Login System : A login portal for customer.
- Details : Customer enter or update details.
- Queris : Enter any queris.

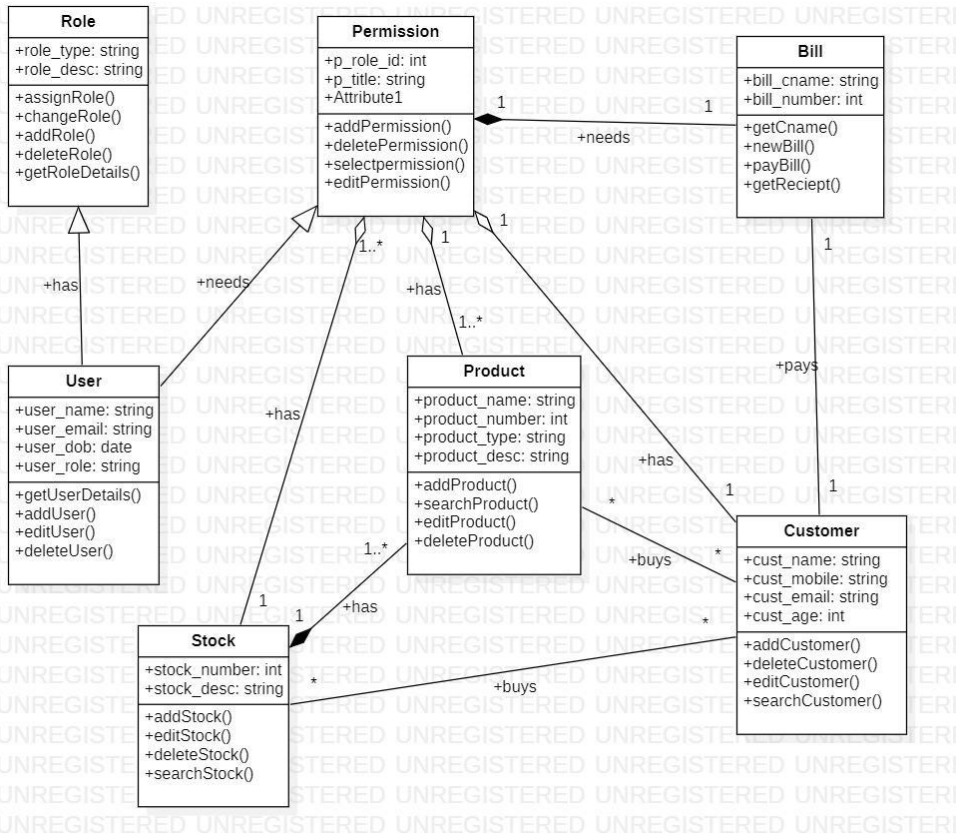
Vendor

- Login : Vendor Login
- View : View Customer orders.
- Queris : Solve or response to customer Queris

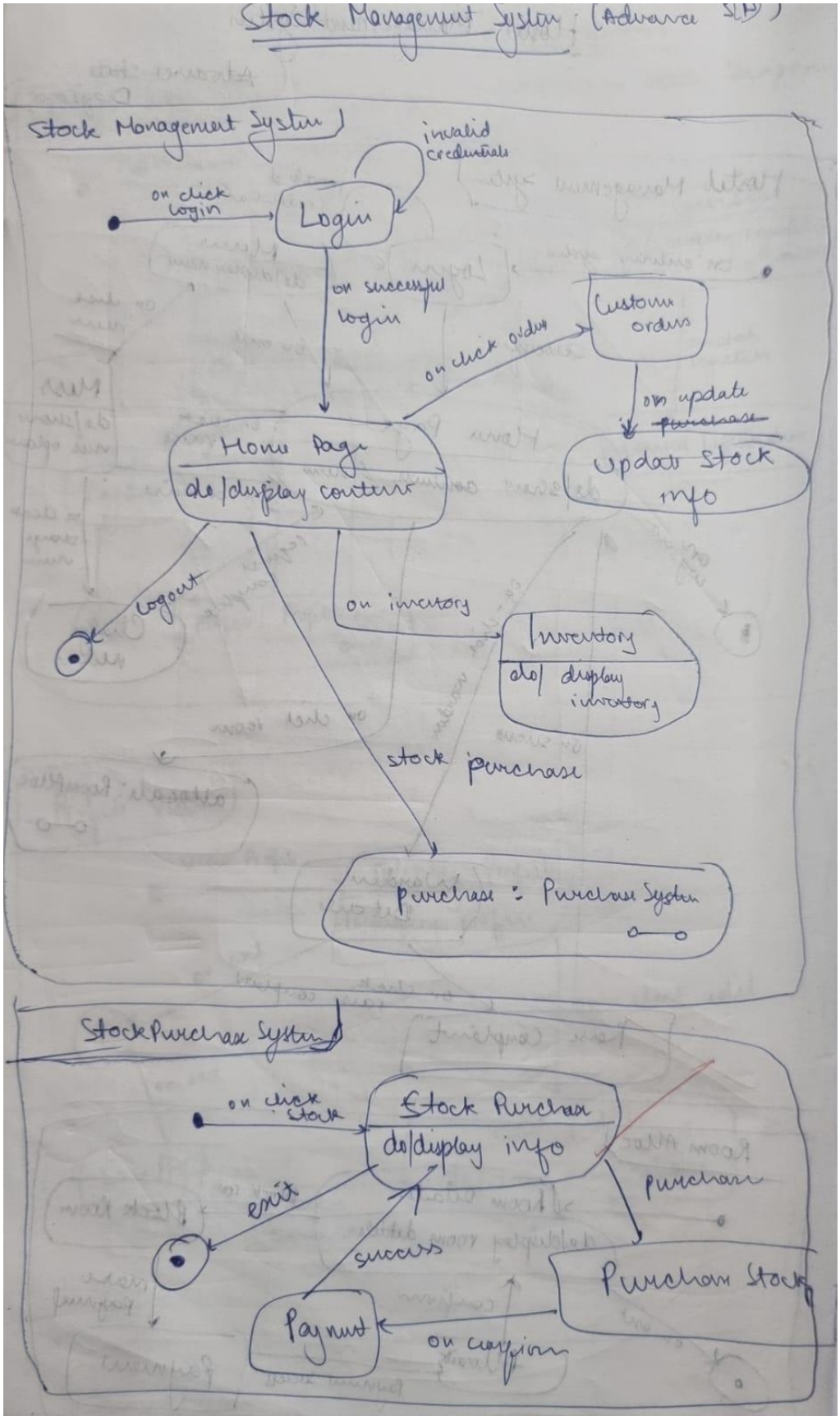
2. advanced class diagram

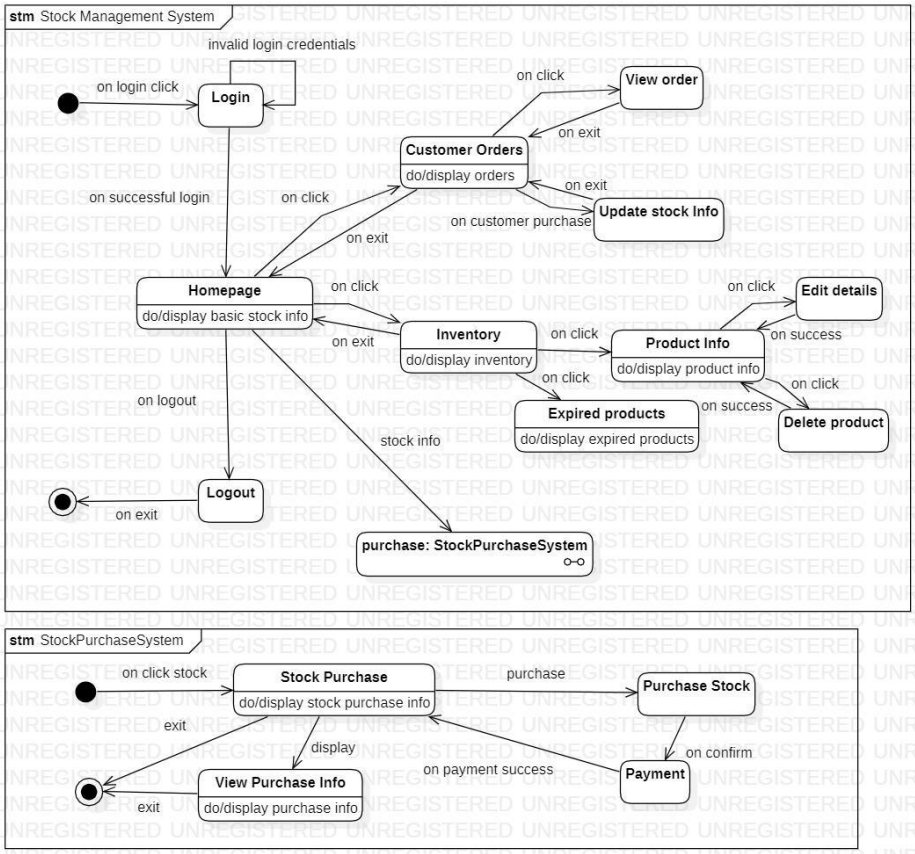
Stock Management System



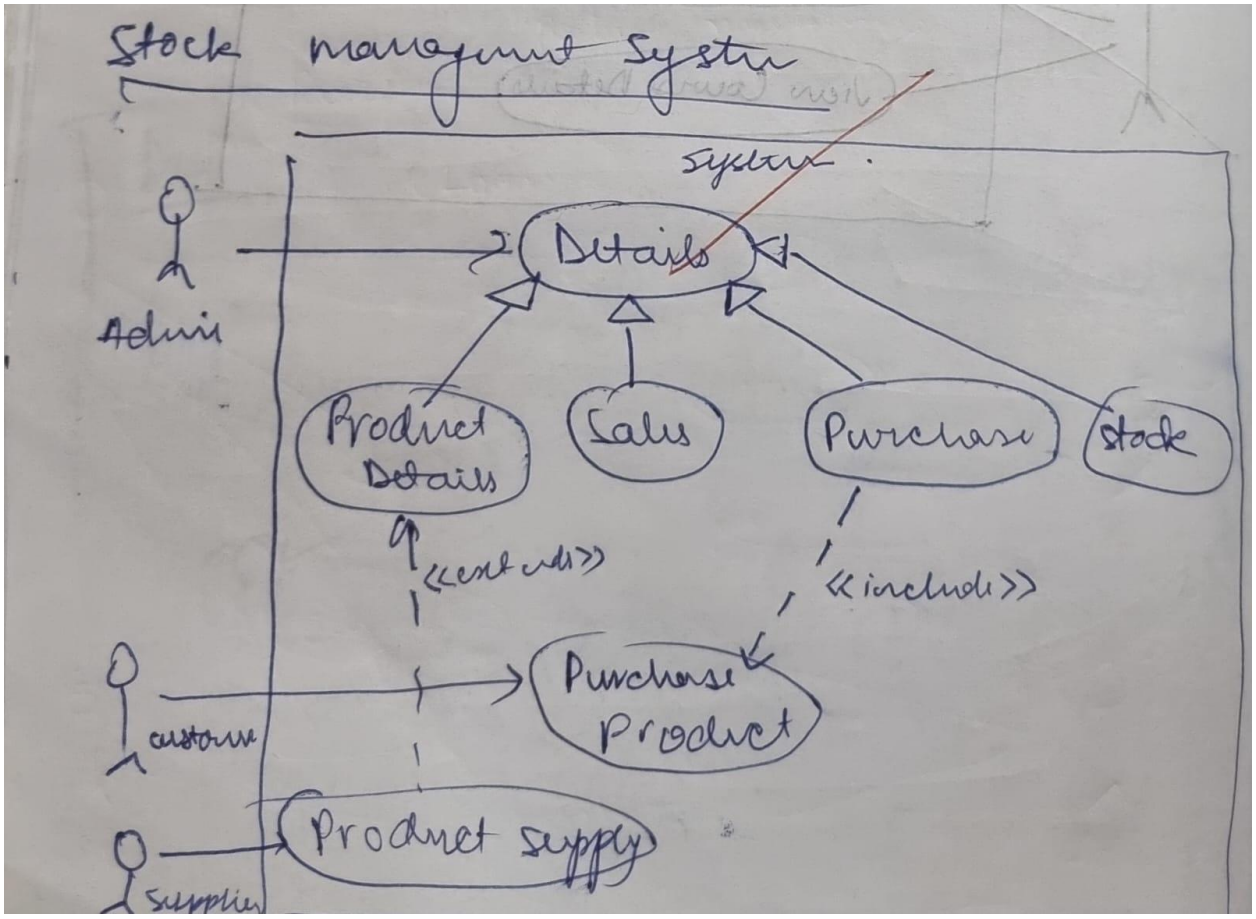


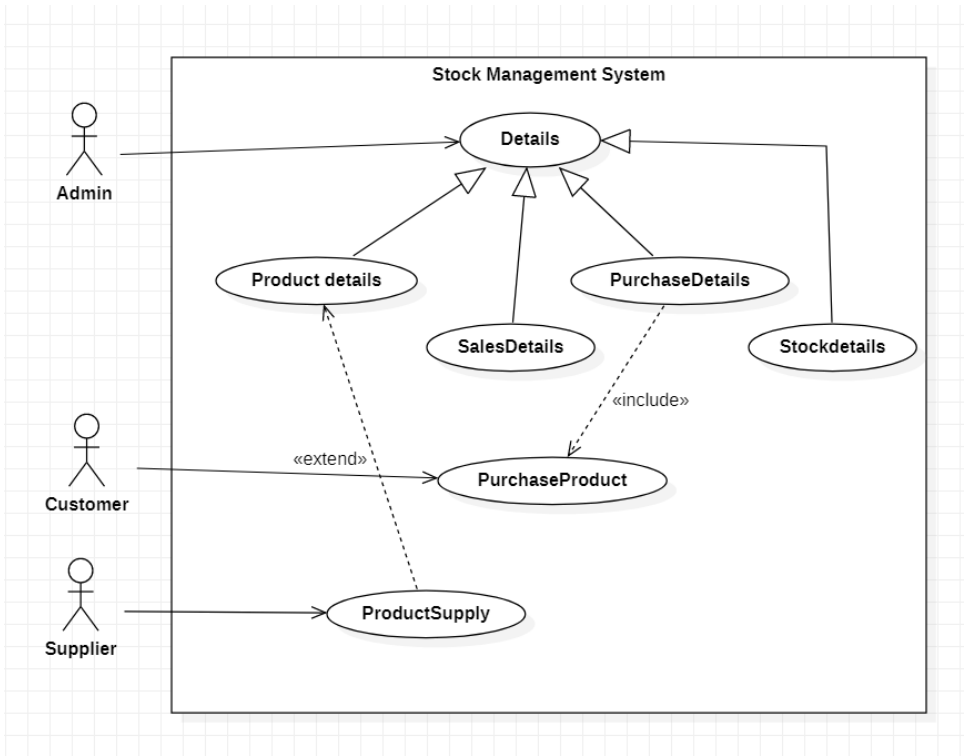
3. advanced state diagram


Stock Management System (Advance
Stock Management System
invalid credentials
on click login
Login
on successful login
Custom order
on click order
on update purchase
Home Page
do/display content
Update Stock info
logout
on inventory
Inventory
do/ display inventory
stock purchase
purchase : Purchase System
StockPurchase System
on click stock
Stock Purchase
do/display info
exit
success
purchase
Purchase Stock
Payment
on confirm

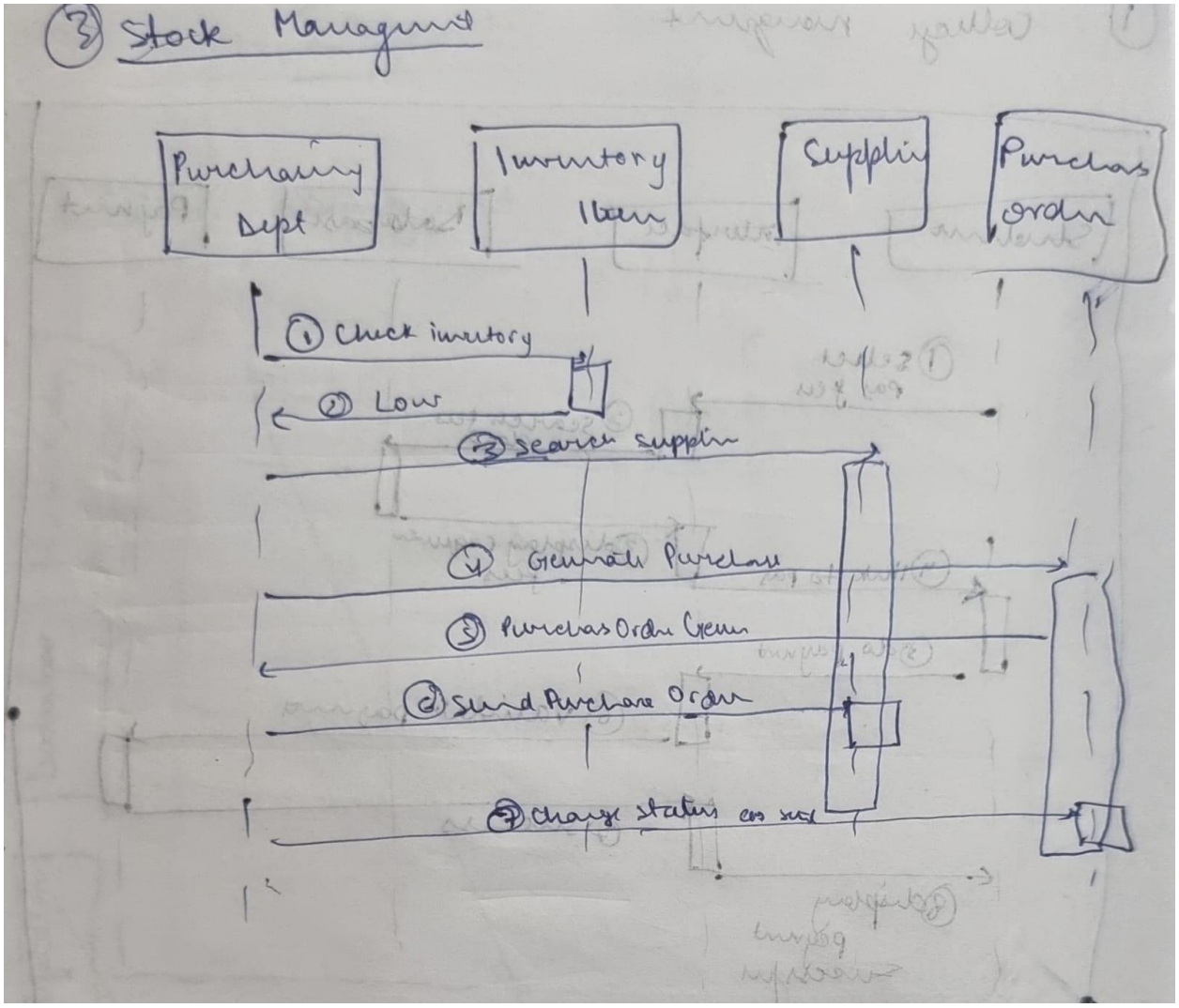


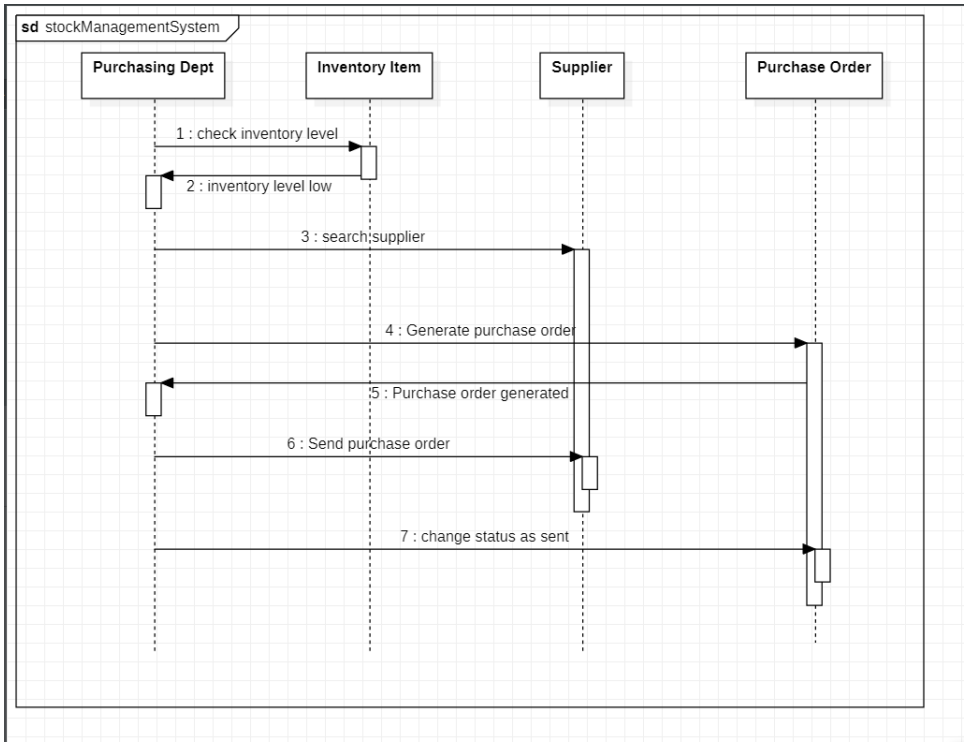
4. wadvanced use case diagram


Stock management Syste
Details
Admin
Product Details
Sales
Purchase
Stock
«extend»
«include»
Purchase Product
Product supply

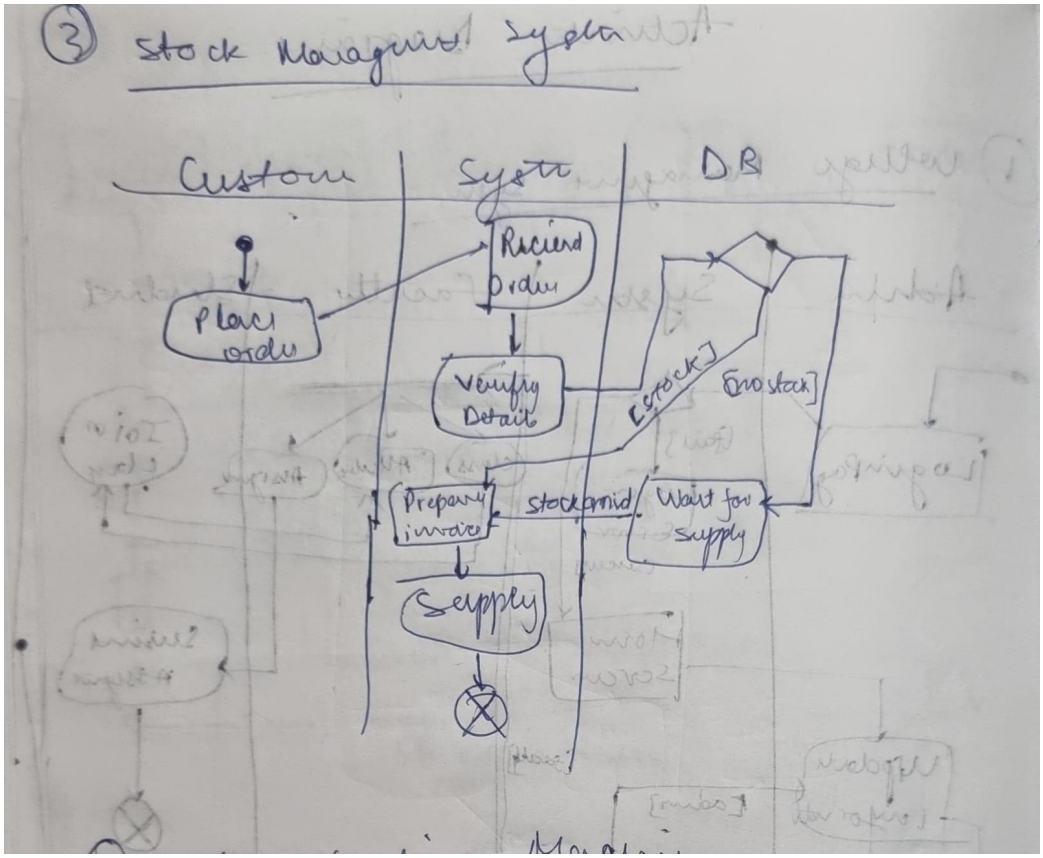

Stock Management System
Admin
Details
Product details
PurchaseDetails
SalesDetails
Stockdetails
«include»
Customer
«extend»
PurchaseProduct
ProductSupply
Supplier

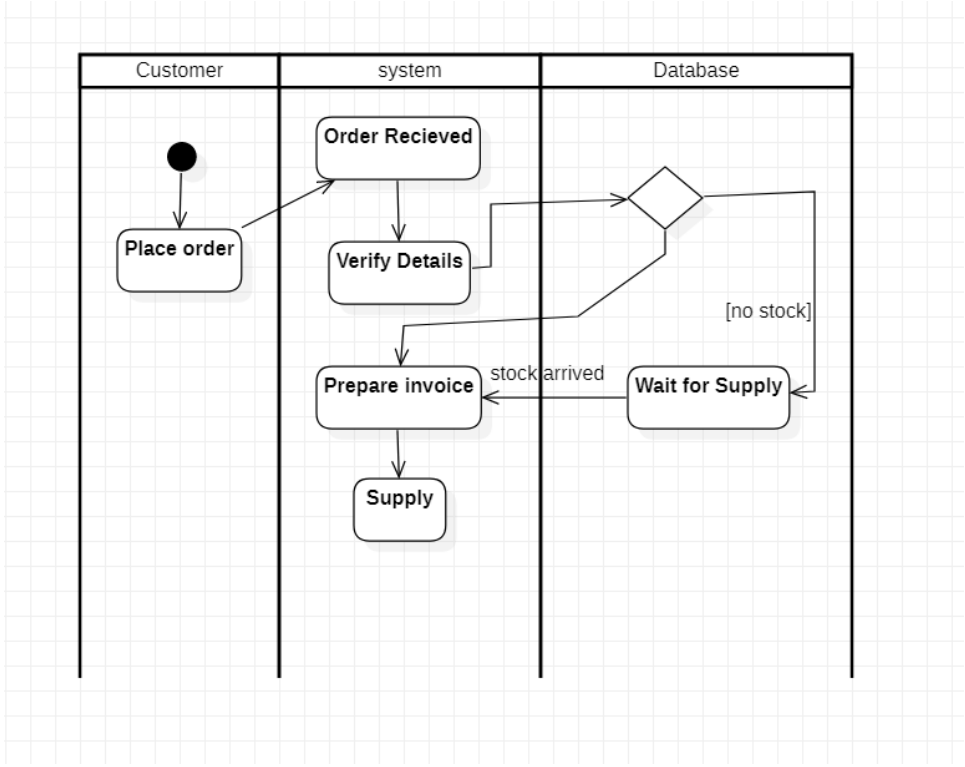
5. advanced sequence diagram





6. advanced activity diagram


Custom
Syste
DB
Recieve Order
Place order
Verify Details
[stock]
[no stock]
Prepare invoice
stockarrived
Wait for supply
Supply


Customer
system
Database
Order Recieved
Place order
Verify Details
[no stock]
Prepare invoice
stock arrived
Wait for Supply
Supply

Exercise 4: Coffee Vending Machine

1. SRS

Coffee Vending Machine (SRS)

① Controller:

Controls the system, takes care of tasks.

- no. of products dispensed
- show menu()
- collect amount()
- check availability()

② Amount Collector:

Takes care of amount collection:

- amount
- status
- collect money()
- update Status()

③ Product Dispensing:

Responsible for dispensing product

④ Products:

Lists different products

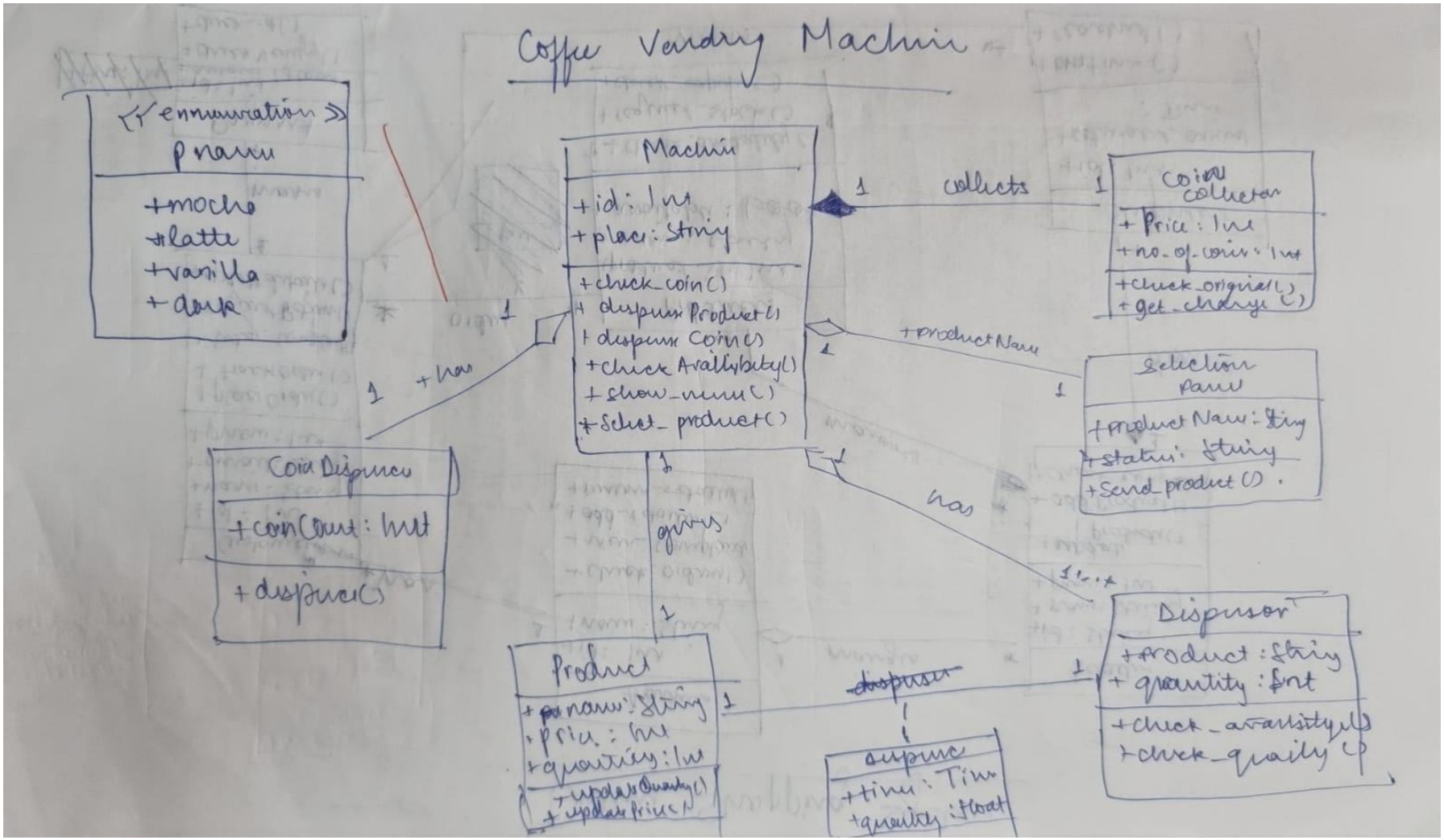
⑤ Product Selection:

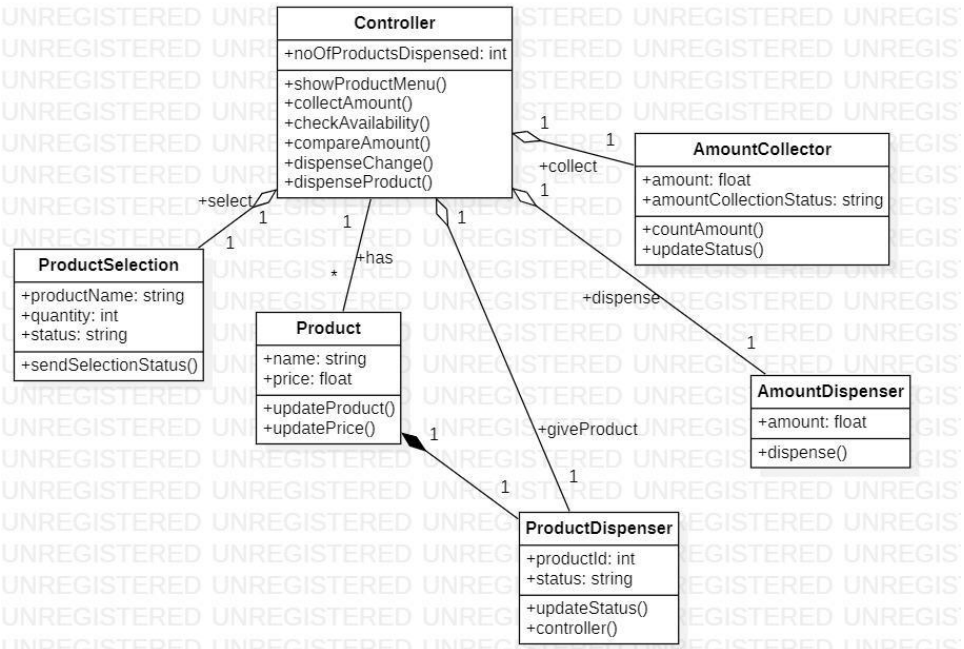
Handles the product selection

products by the user.

Problem Statement: An automated machine that handles user order and dispense the correct order.

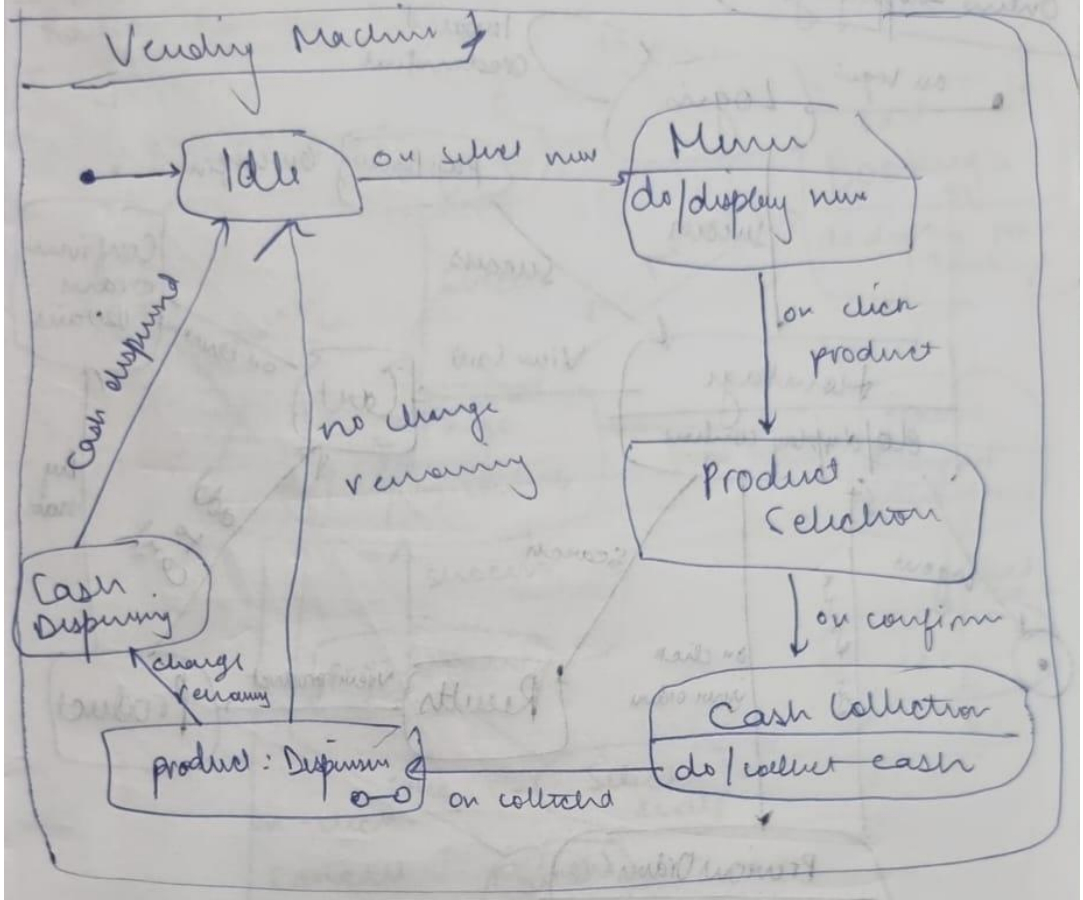
2. advanced class diagram

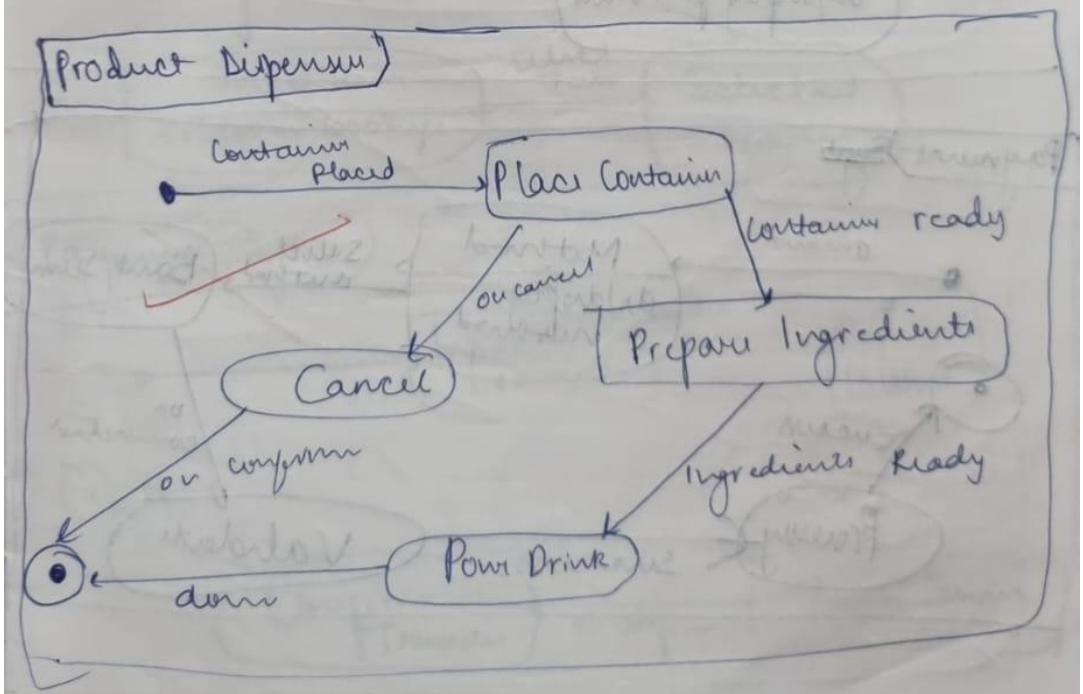

Coffee Vending Machine
<<enumeration>>
Pname
+mocha
+latte
+vanilla
+dark
Machine
+id: Int
+place: String
+check_coin()
+dispense Product()
+dispense Coin()
+check Availibity()
+show_menu()
+select_product()
collects
1
Coin collector
+Price: Int
+no_of_coin: Int
+check_original()
+get_change()
+productName
Selection Pane
+product Name: String
+status: String
+send_product()
has
1..*
Coin Dispenser
+coinCount: Int
+dispense()
gives
Product
+pname: String
+price: Int
+quantity: Int
+updateQuantity()
+updatePrice()
dispense
Dispensor
+product: String
+quantity: Int
+check_availability()
+check_quality()
output
+time: Time
+quality: float

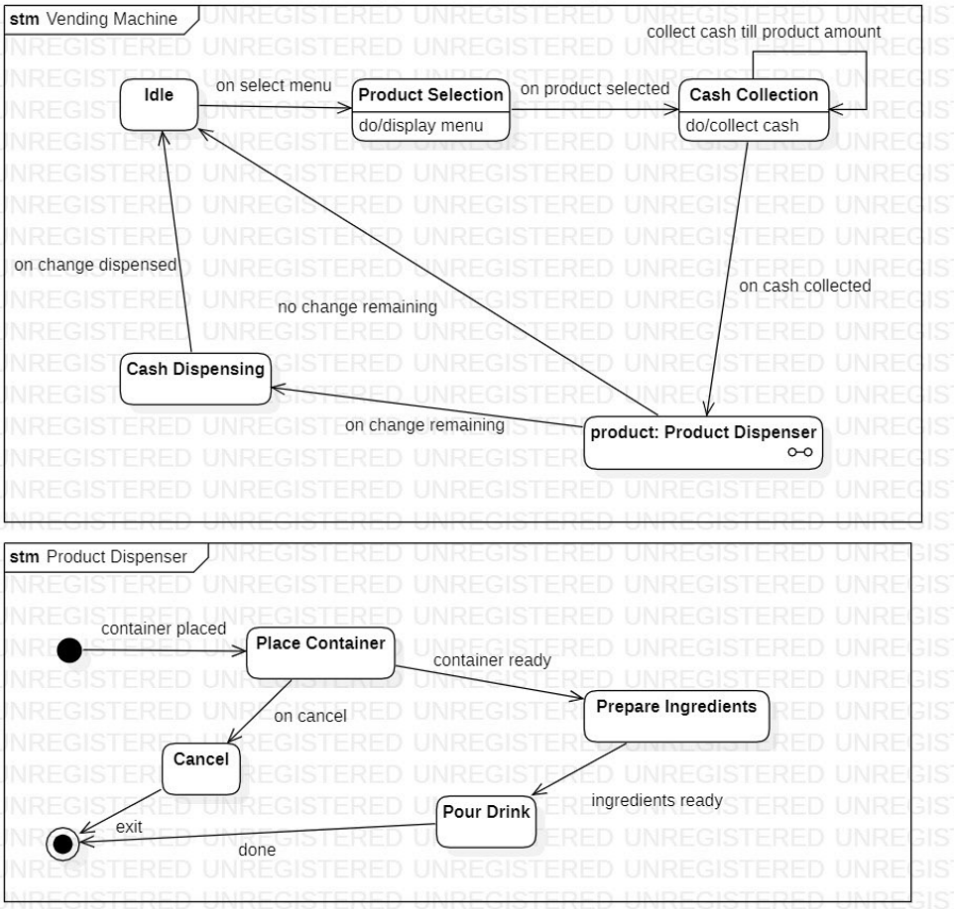


3. advanced state diagram

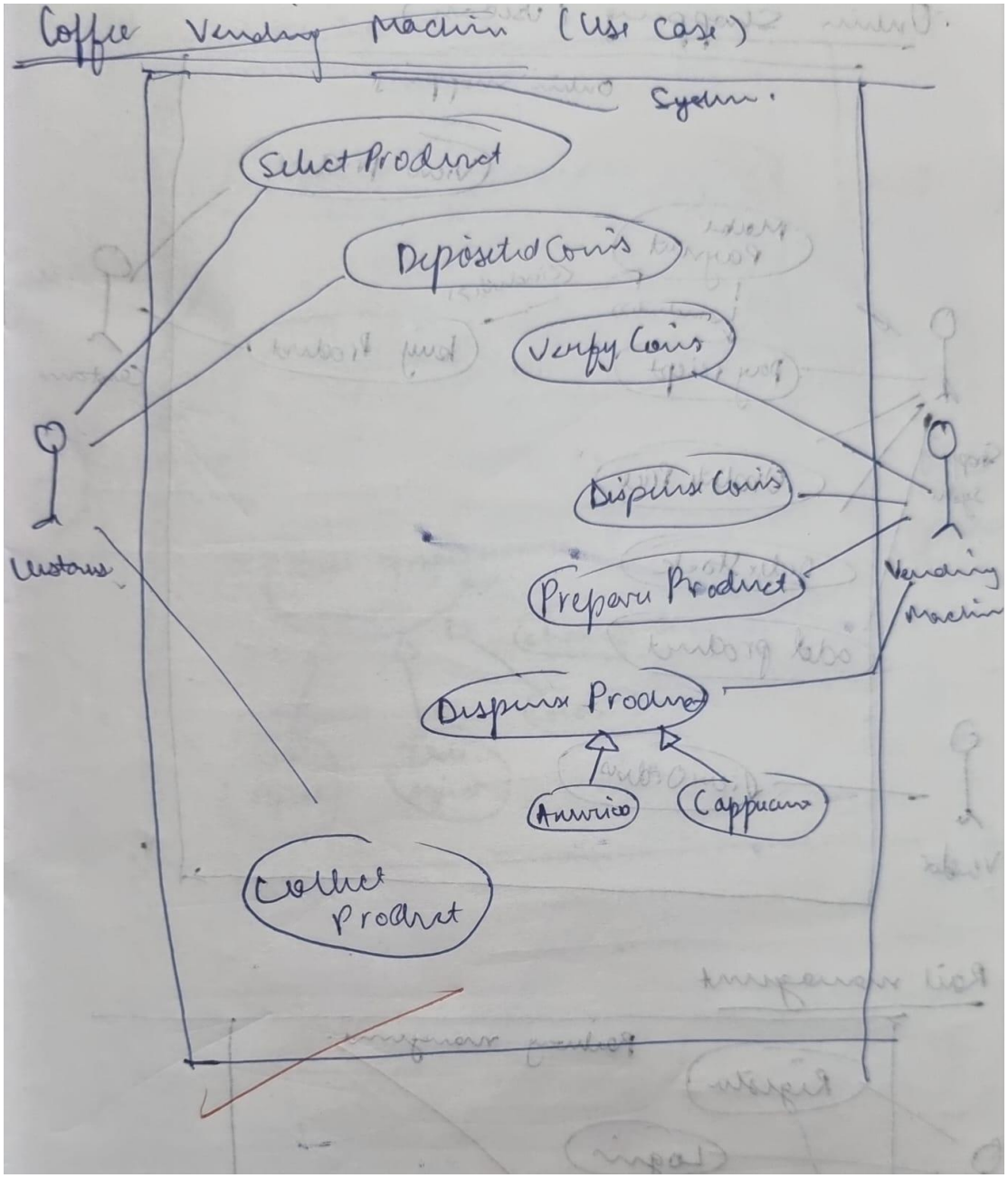
Coffee Vending Machine (4 STM)

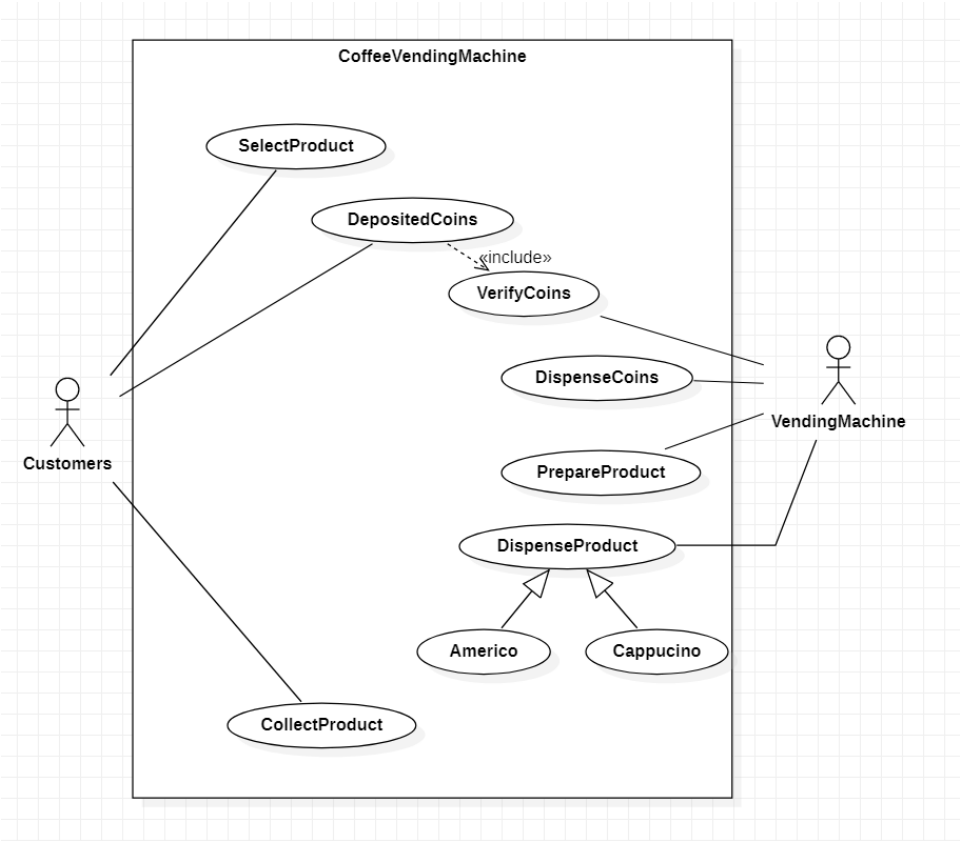




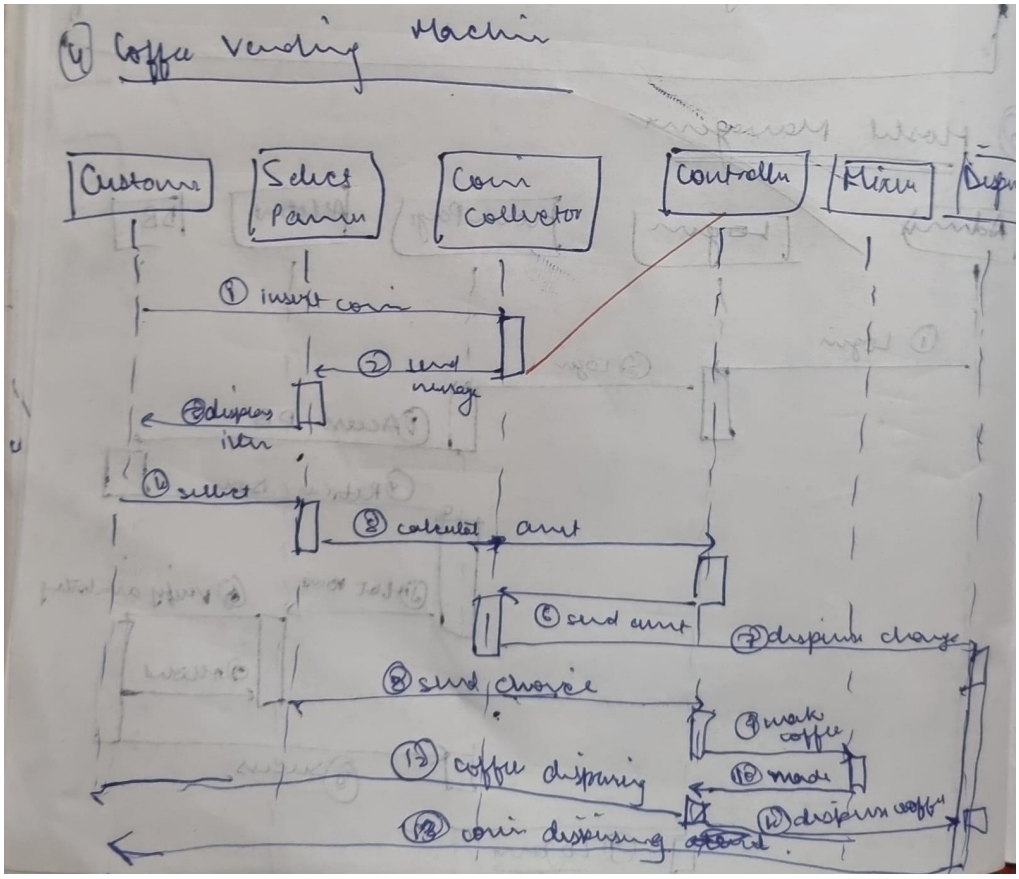


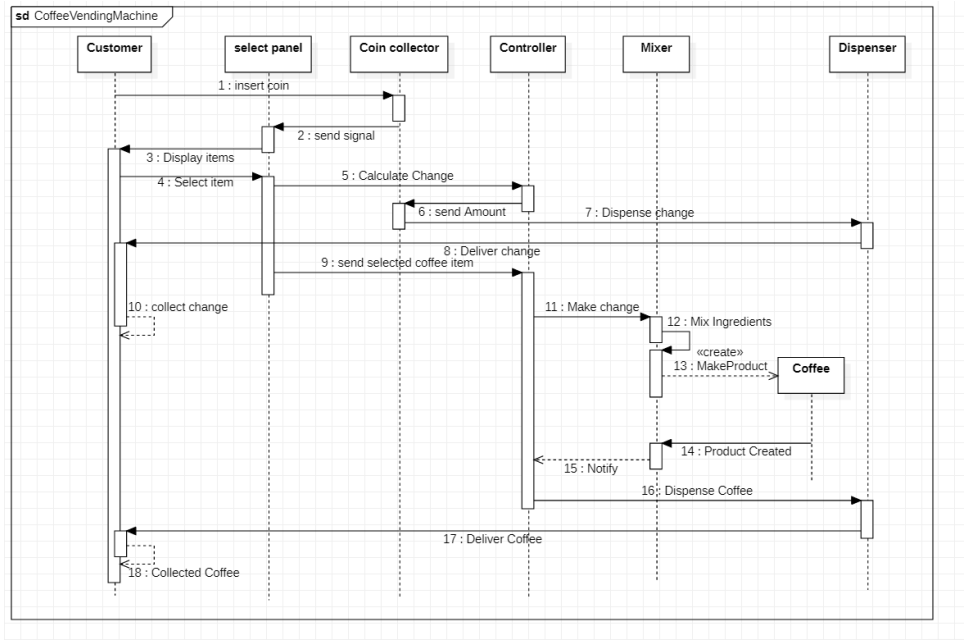
4. advanced use case diagram


Coffee Vending Machine (Use Case)
System
Select Product
Deposited Coins
Verify Coins
Dispense Coins
Prepare Product
Dispense Product
Americano
Cappuccino
Collect Product
Customer
Vending machine

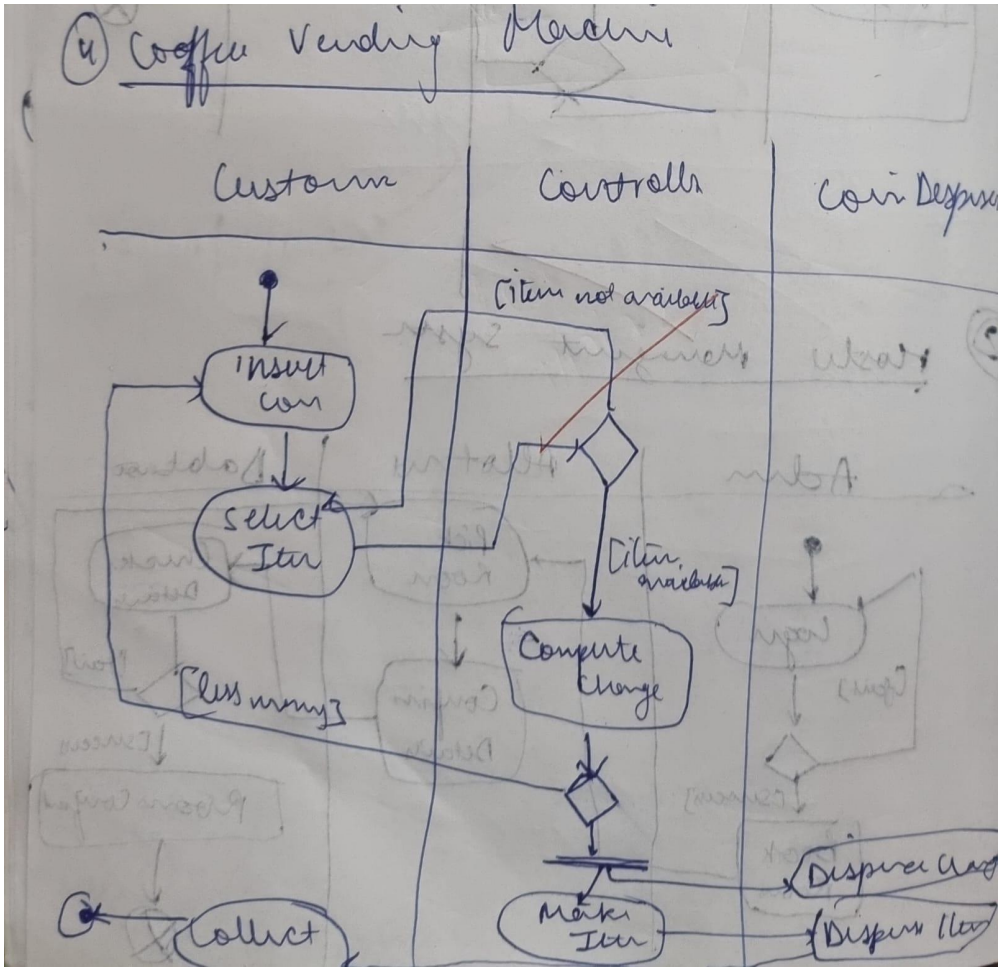


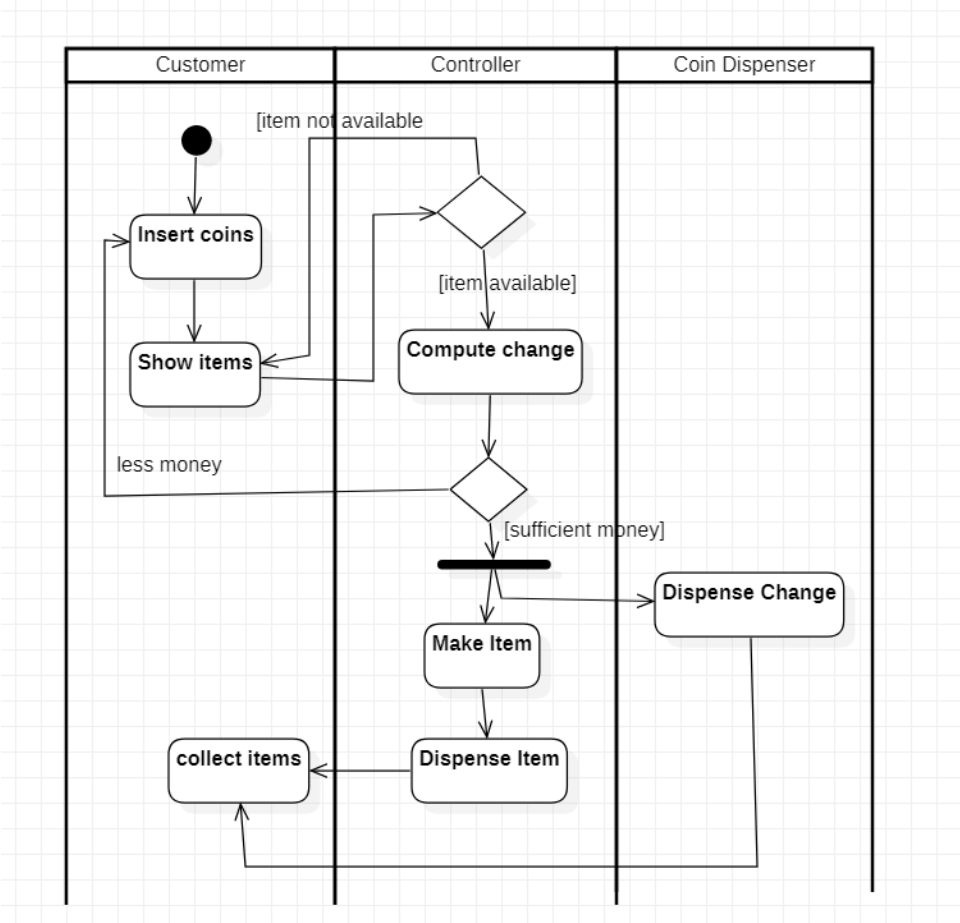
5. advanced sequence diagram





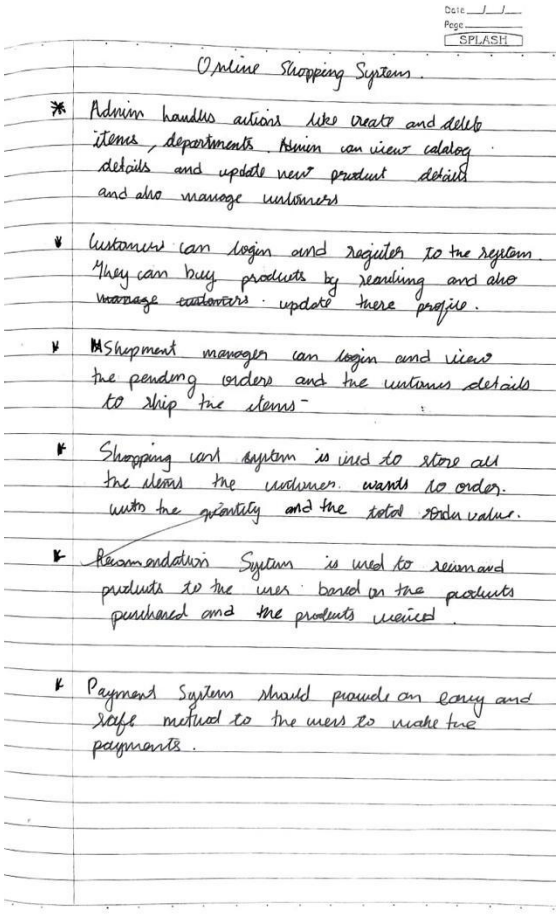
6. advanced activity diagram


4
Coffee Vending
Custom
Controlle
Insert coin
Select Item
[item not available]
[item available]
Compute Change
[less money]
Collect
Make Item



Exercise 5: Online Shopping System

1. SRS



Online Shopping System.

* Admin handles actions like create and delete items, departments. Admin can view catalog details and update new product details and also manage customers

* Customer can login and register to the system. They can buy products by searching and also manage customers. update there profile.

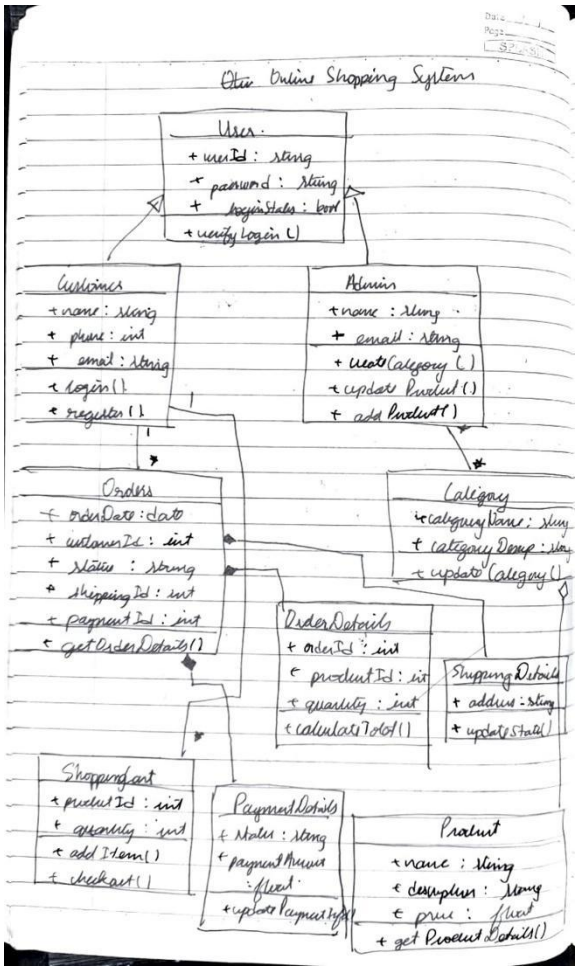
* Shipment manager can login and view the pending orders and the customs details to ship the items.

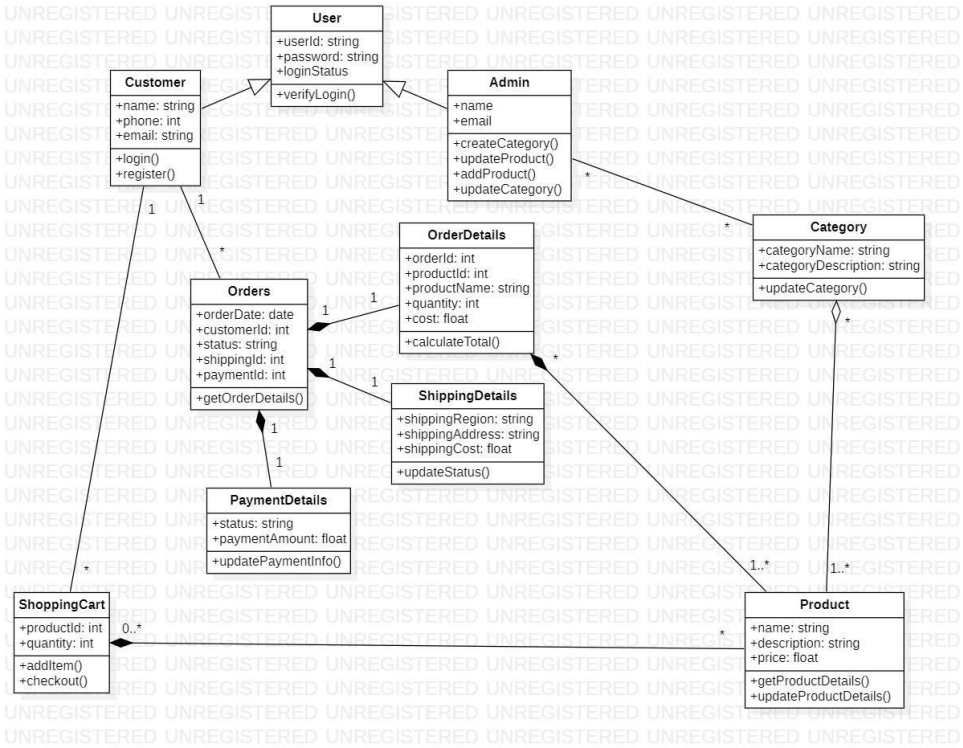
* Shopping cart system is used to store all the items the customer wants to order. with the quantity and the total order value.

* Recommendation System is used to recommend products to the user based on the products purchased and the products viewed.

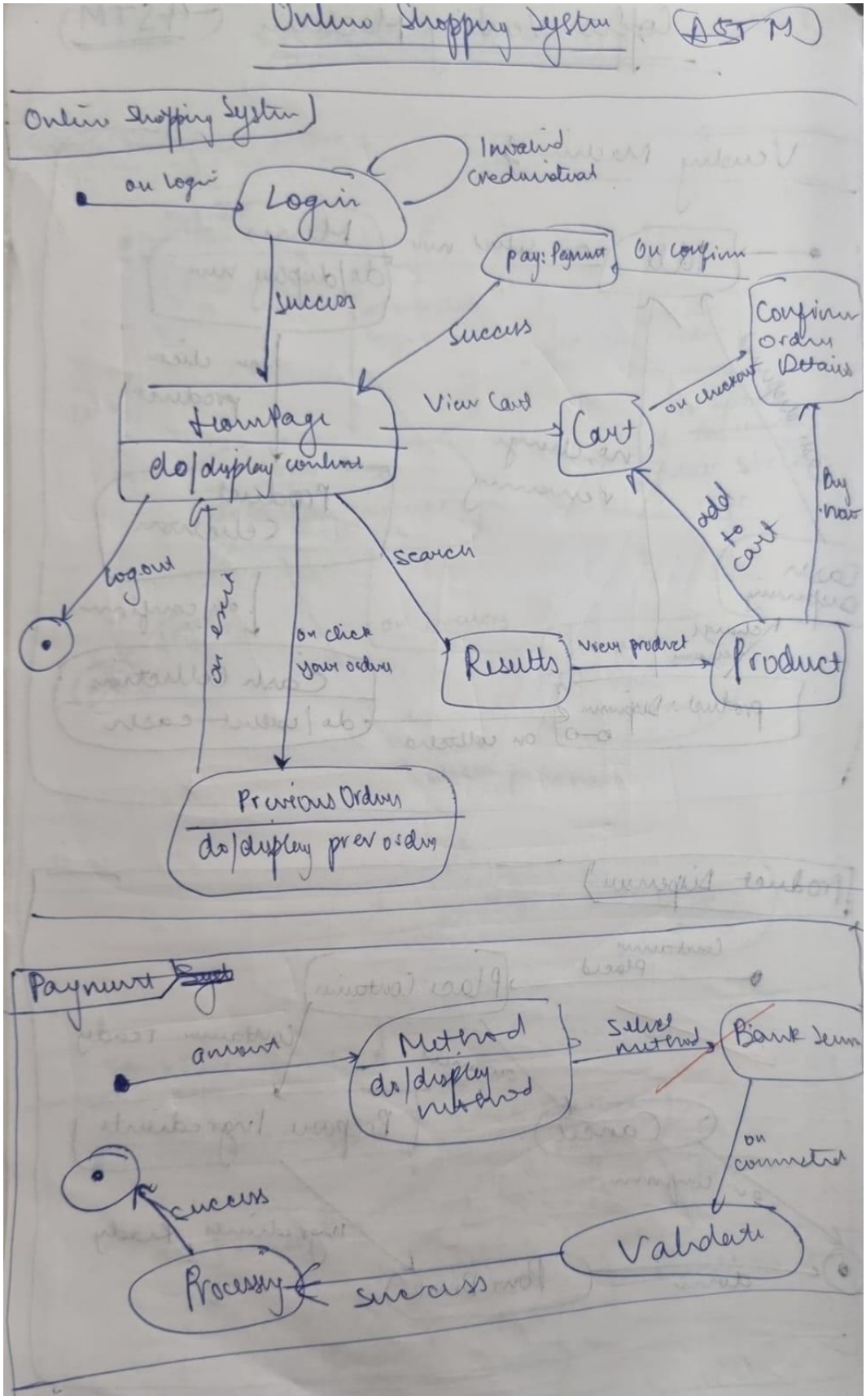
* Payment System should provide an easy and safe method to the users to make the payments.

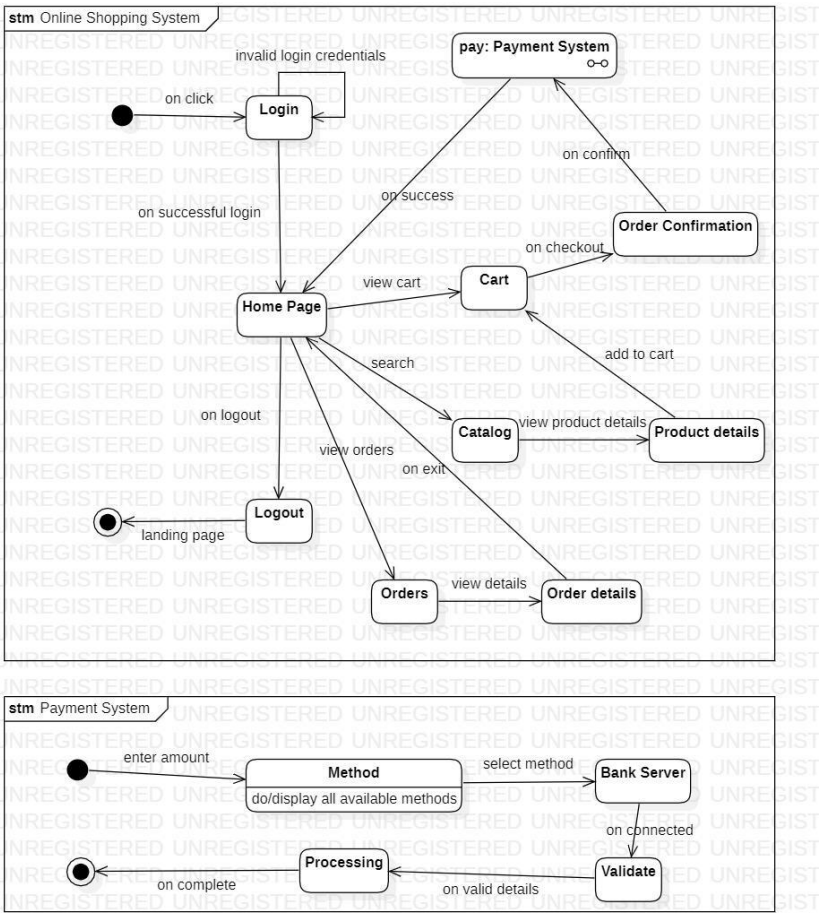
2. advanced class diagram


Date
Page
Online Shopping System
User
+ userId : string
+ password : string
+ loginStatus : bool
+ verifyLogin ()
Customer
+ name : string
+ phone : int
+ email : string
+ login ()
+ register ()
Admin
+ name : string
+ email : string
+ createCategory ()
+ updateProduct ()
+ addProduct ()
Orders
+ orderDate : date
+ customerId : int
+ status : string
+ shippingId : int
+ paymentId : int
+ getOrderDetails ()
Category
+ categoryName : string
+ categoryDesc : string
+ update (category)
OrderDetails
+ orderId : int
+ productId : int
+ quantity : int
+ calculateTotal ()
ShippingDetails
+ address : string
+ updateStatus ()
ShoppingCart
+ productId : int
+ quantity : int
+ addItem ()
+ checkout ()
PaymentDetails
+ status : string
+ paymentAmount : float
+ updatePaymentInfo ()
Product
+ name : string
+ description : string
+ price : float
+ getProductDetails ()

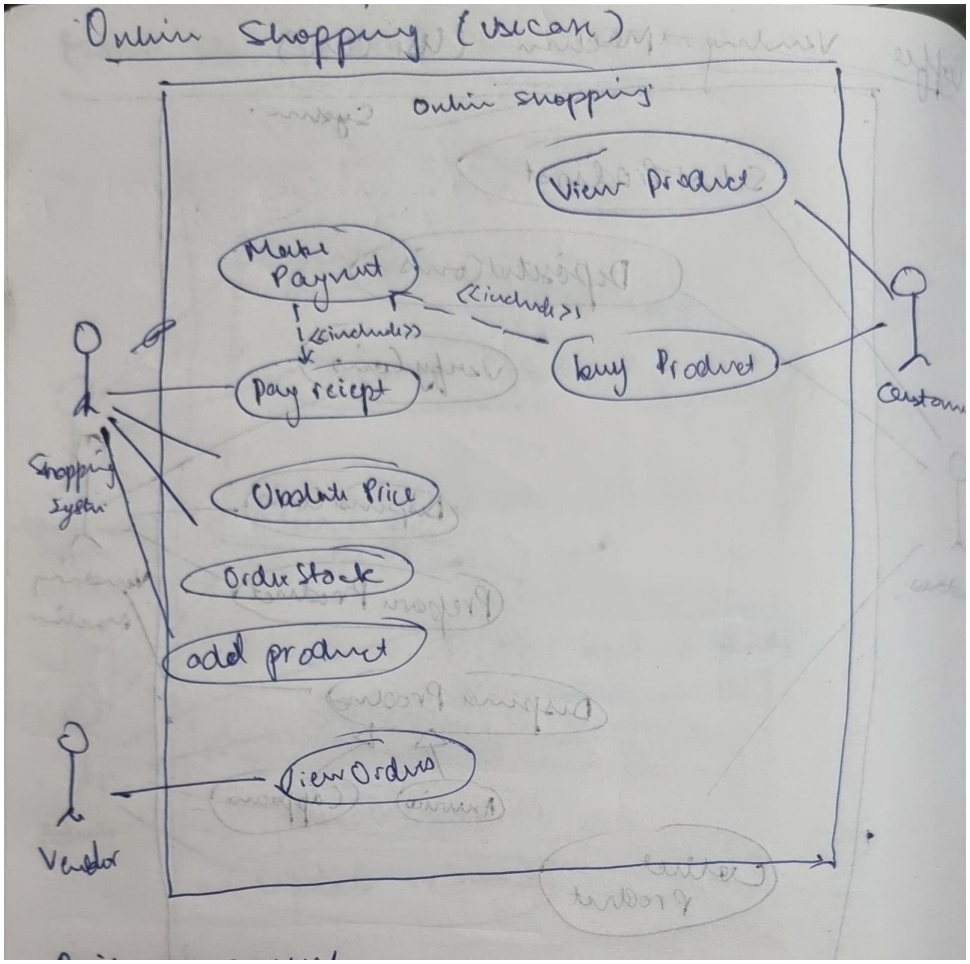


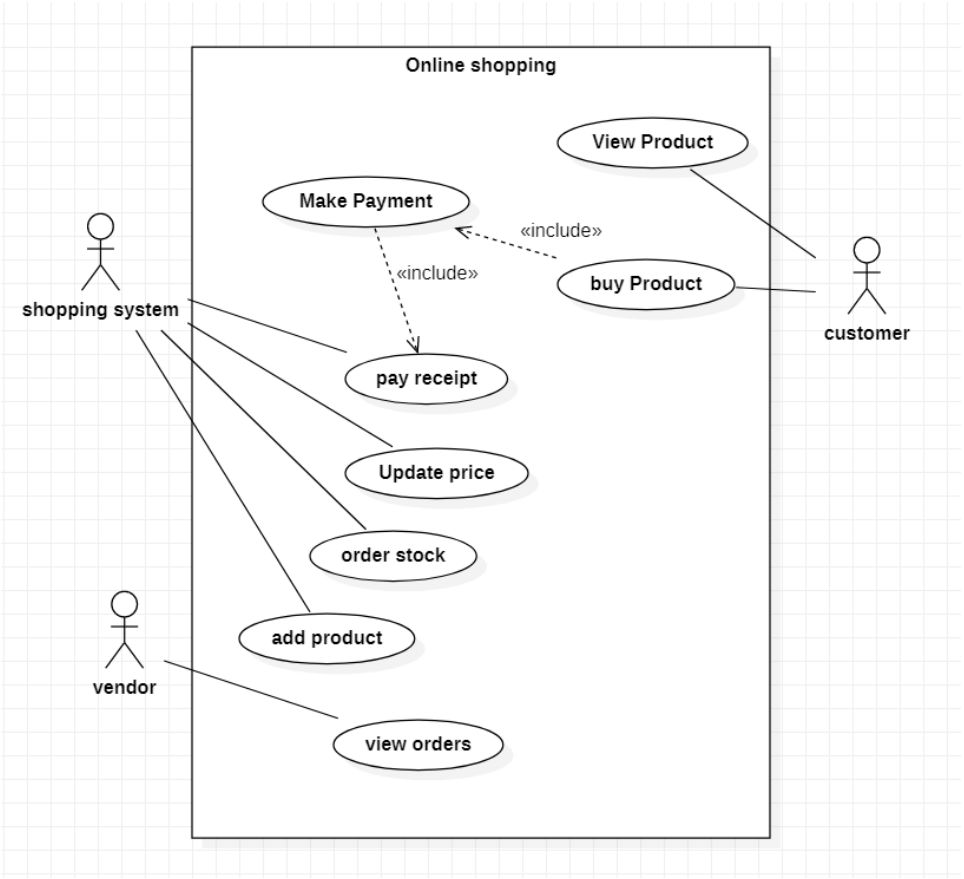
3. advanced state diagram


Online Shopping System (ASTM)
Online Shopping System
on login
Login
Invalid credentials
Success
pay: Payment
On confirm
Success
Confirm order Details
HomePage
do/display content
View Cart
Cart
on checkout
logout
scarch
add to cart
Buy now
on click your orders
Results
View product
Product
Previous Orders
do/display prev orders
Payment
amount
Method
do/display method
select method
Bank sem
on connection
success
Validate
Process
success

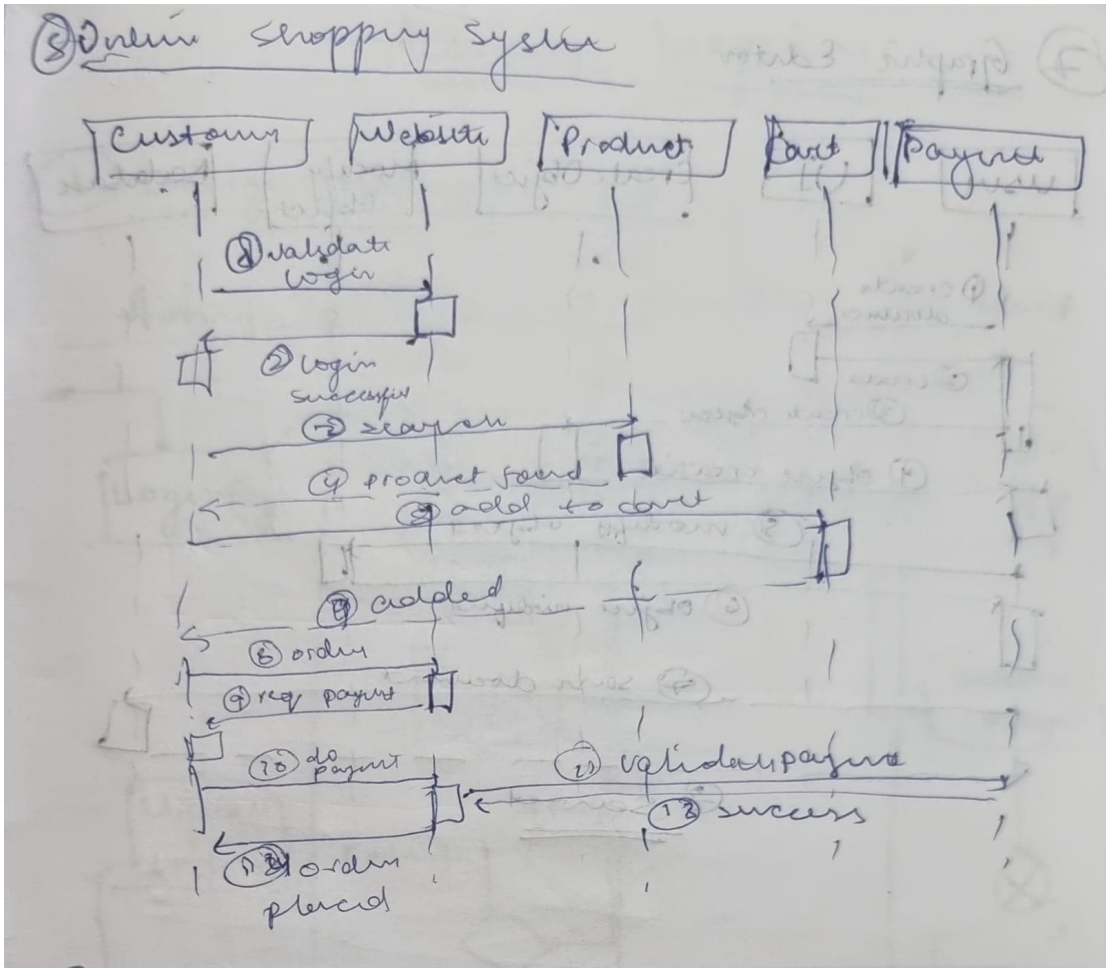


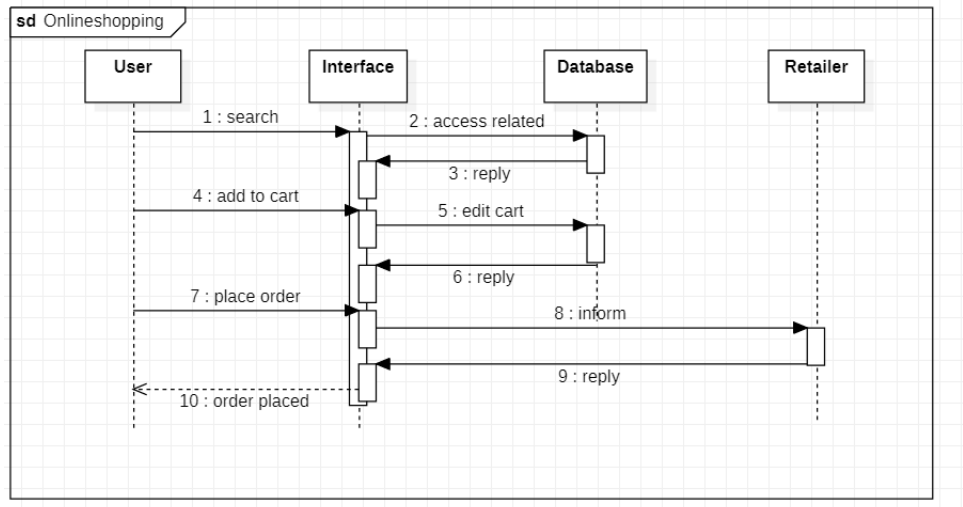
4. advanced use case diagram


Online Shopping (usecase)
online shopping
View product
Make Payment
<<include>>
<<include>>
buy Product
pay reciept
Obtain Price
Order Stock
add product
View Orders
Vendor

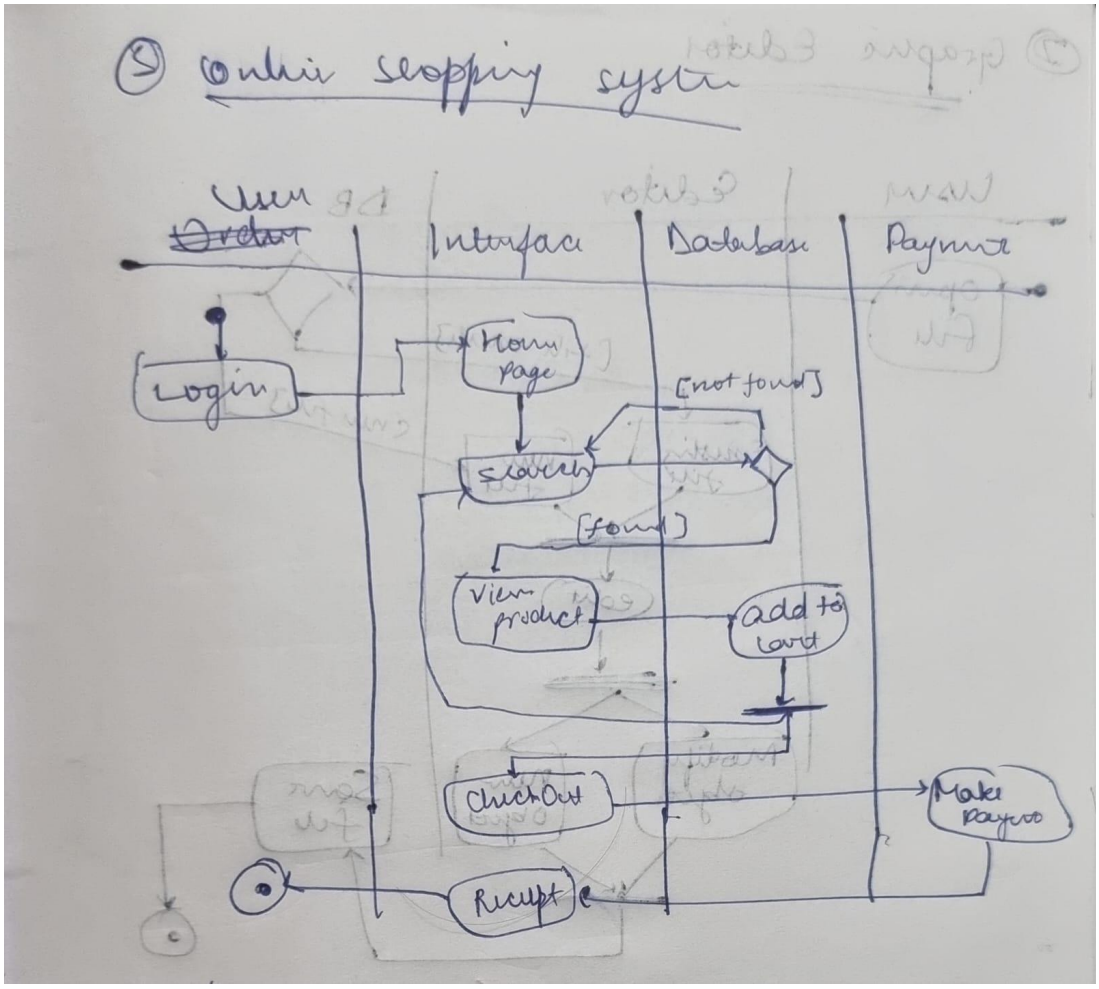


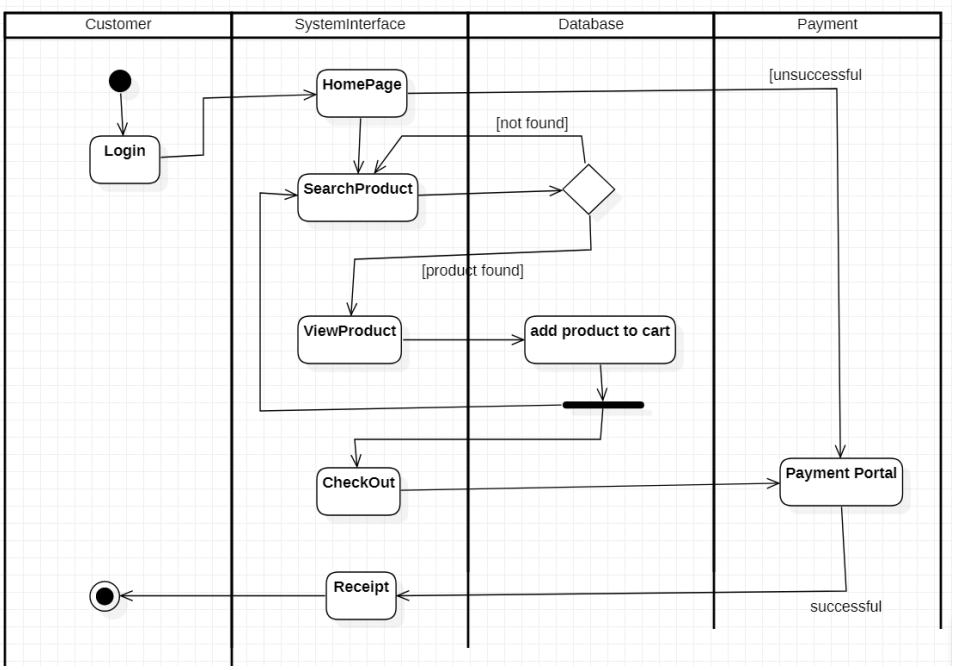
5. advanced sequence diagram





6. advanced activity diagram


Login
[not found]
[found]
[unsuccessful]
Receipt


Customer
SystemInterface
Database
Payment
HomePage
Login
SearchProduct
[not found]
[product found]
ViewProduct
add product to cart
CheckOut
Payment Portal
Receipt
[unsuccessful]
successful

Exercise 6: Railway Reservation System

1. SRS

Railway Reservation System SRS

Problem Statement: An automated ticket booking system that takes care of tasks such as booking ticket, cancelling and checking status etc.

(1) ~~Admin~~ ~~Booking~~: ~~Take~~ Have control over all the system.

→ Id
→ pass
→ manage() → manage the tasks.

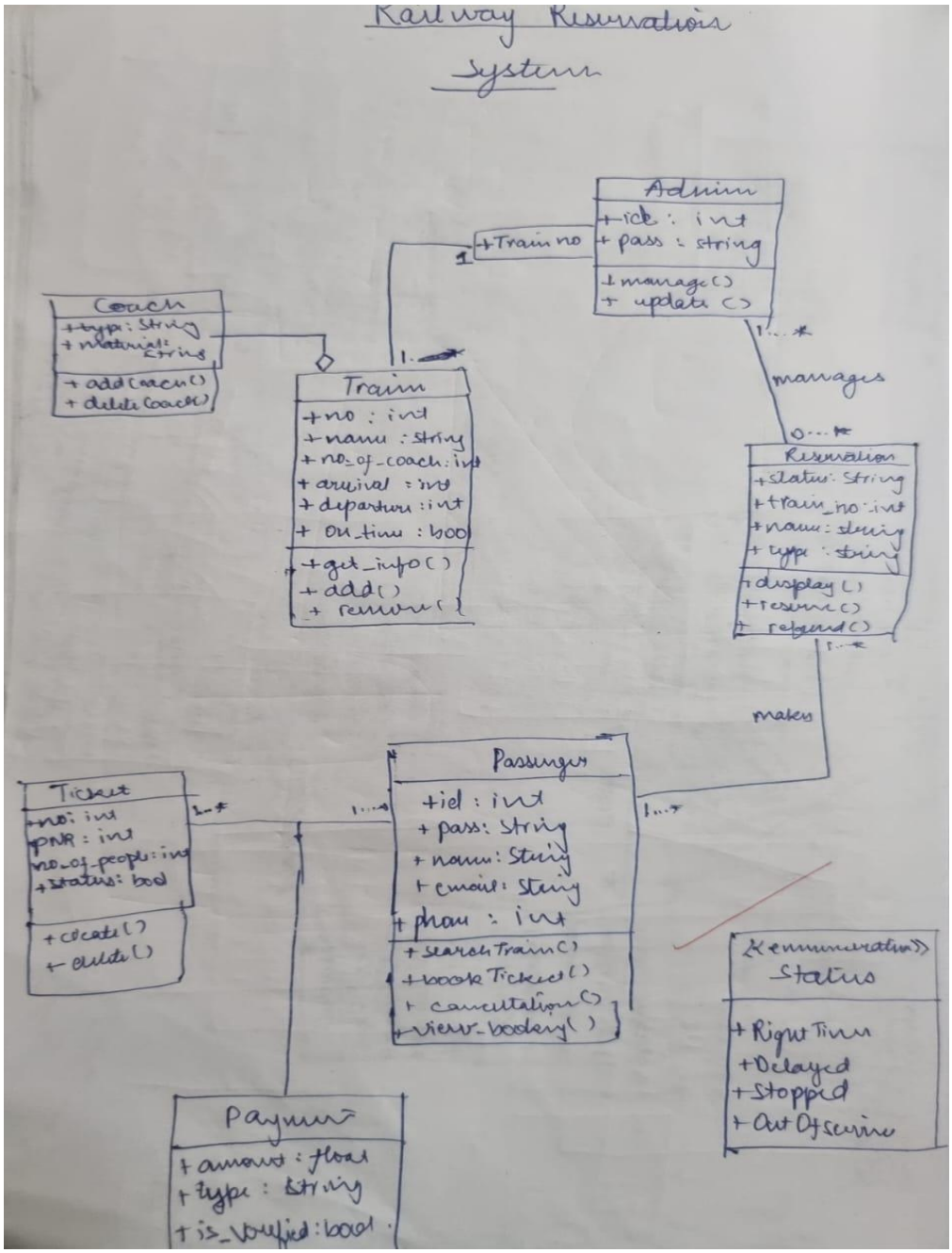
(2) Train: It contains ~~the~~ all the information about trains

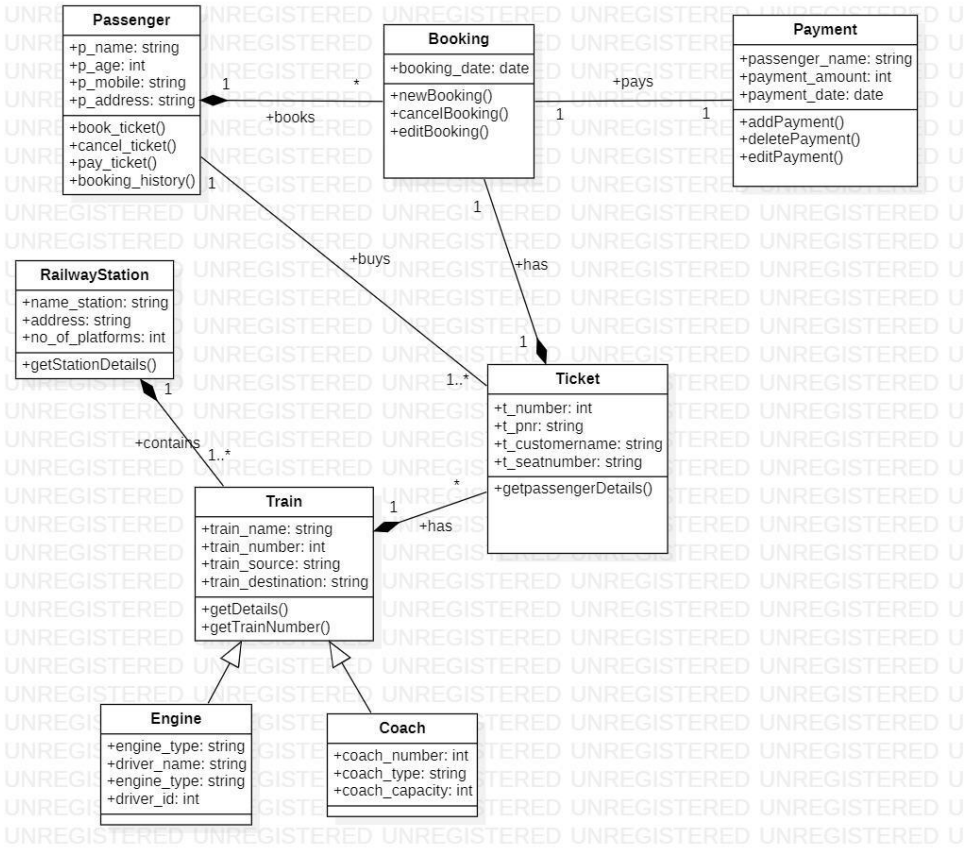
→ Id → no. of coaches
→ name → time
→ getinfo() → give all the detail about the train.
→ add() → to add a new train
→ remove() → to remove some train

(3) Coach: Contains information about coach

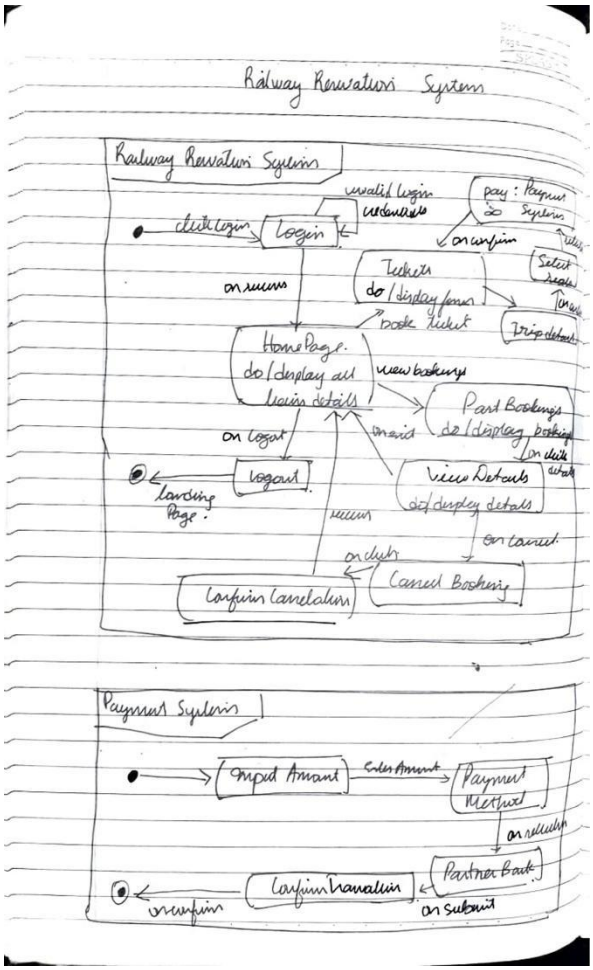
(4) Reservation: Gives users all the information about their reservation.

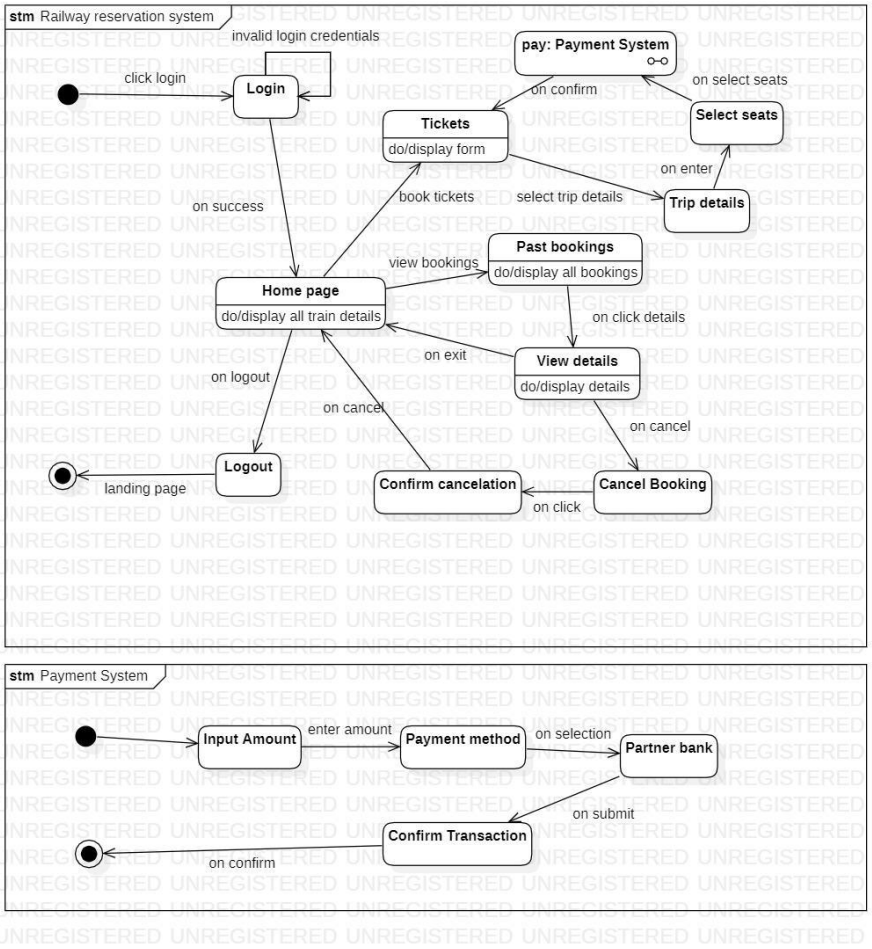
2. Draw the advanced class diagram



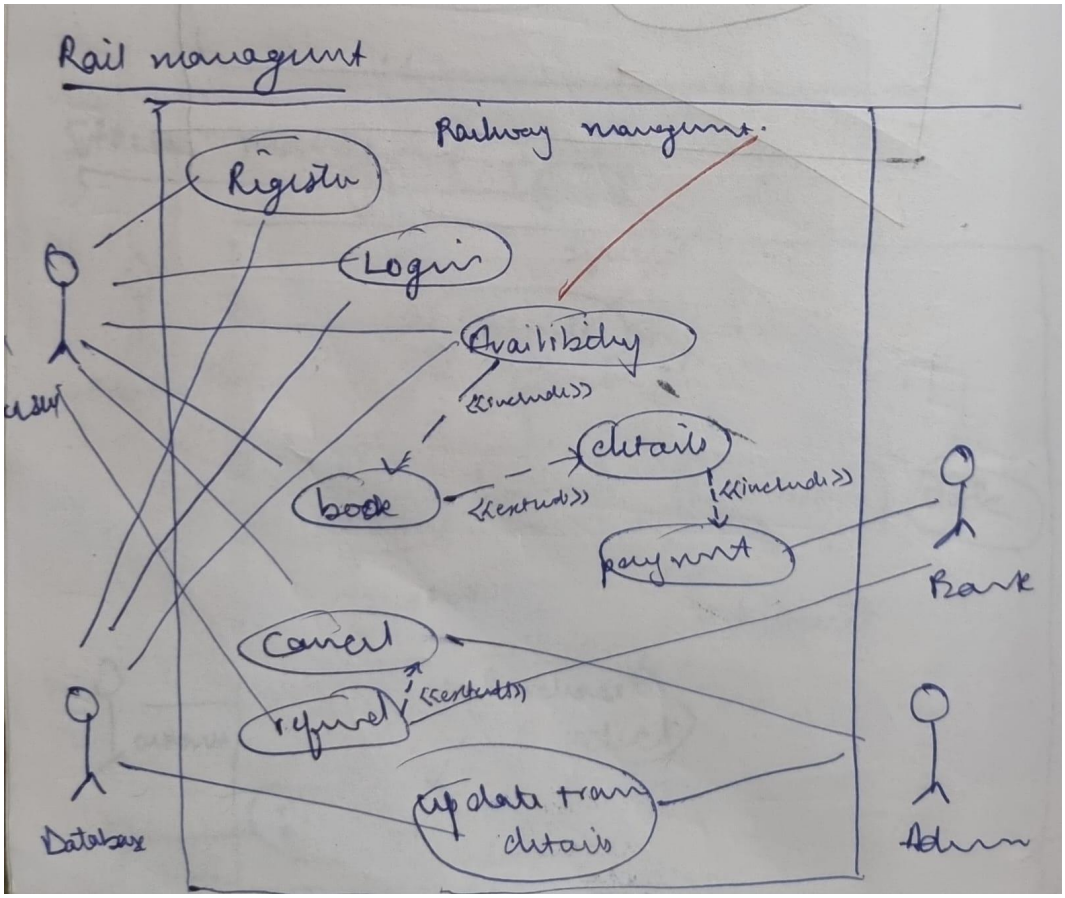


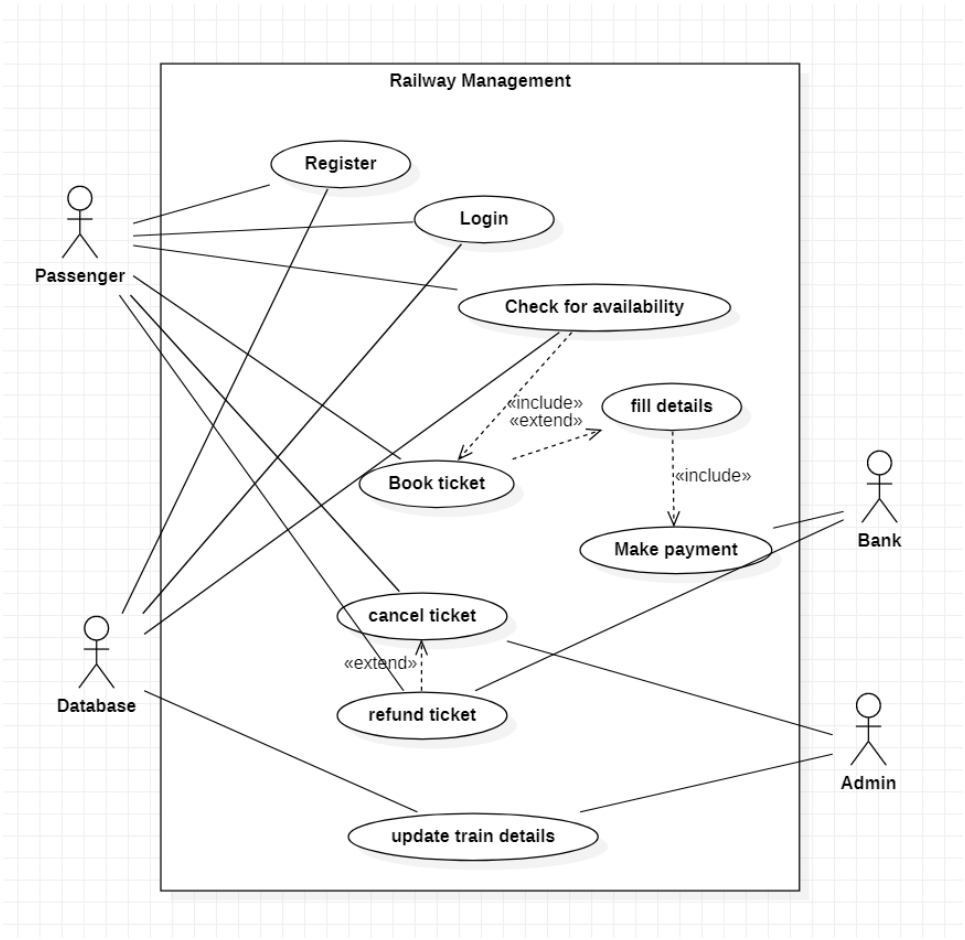
3. advanced state diagram


Railway Reservation System
Railway Reservation System
invalid login credentials
Login
Tickets
do / display forms
book ticket
Trip details
Home Page
do / display all train details
new bookings
Past Bookings
do / display bookings
on cancel
View Details
do / display details
on cancel
Cancel Booking
on click
Confirm Cancelation
on logout
Logout
Landing Page
Payment System
Input Amount
Enter Amount
Payment Method
Partner Bank
Confirm Transaction
on submit

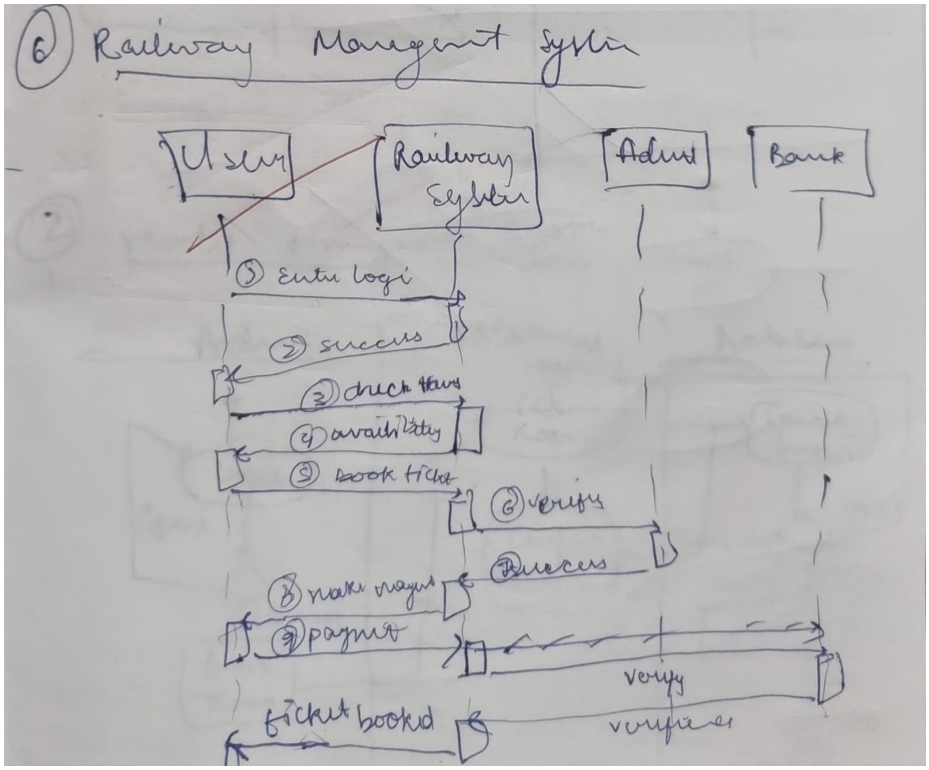


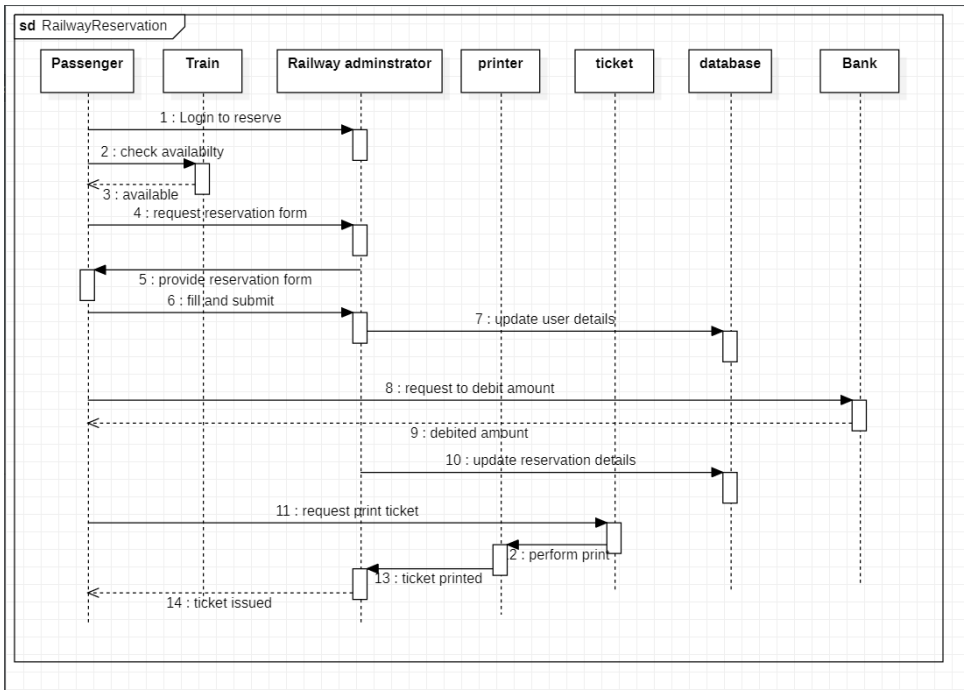
4. advanced use case diagram


Rail managment
Railway managment
Registr
Login
Availibidy
<<extends>>
book
<<extends>>
cbtails
<<includes>>
payment
Bank
Cancel
refund
<<extends>>
update train cbtails
Database
Admin

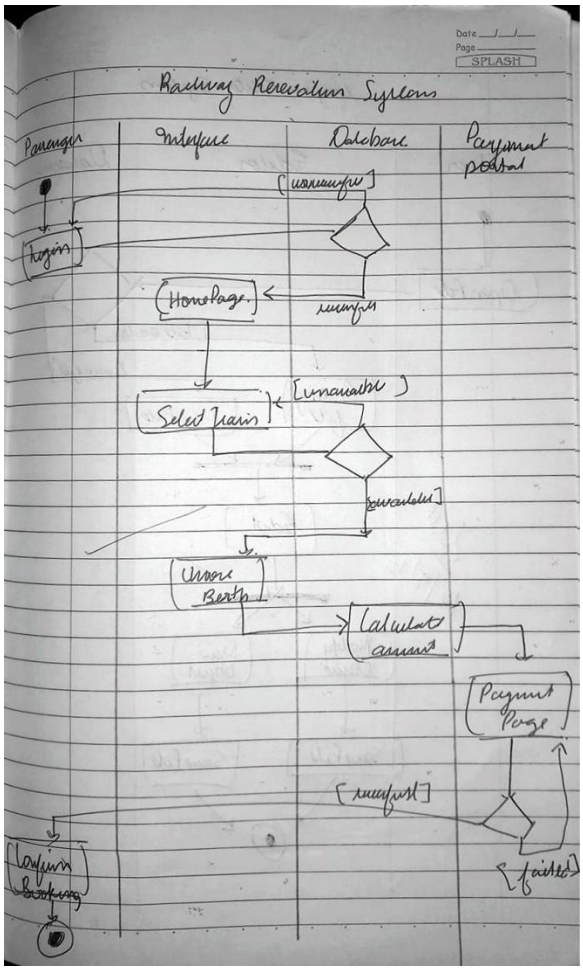


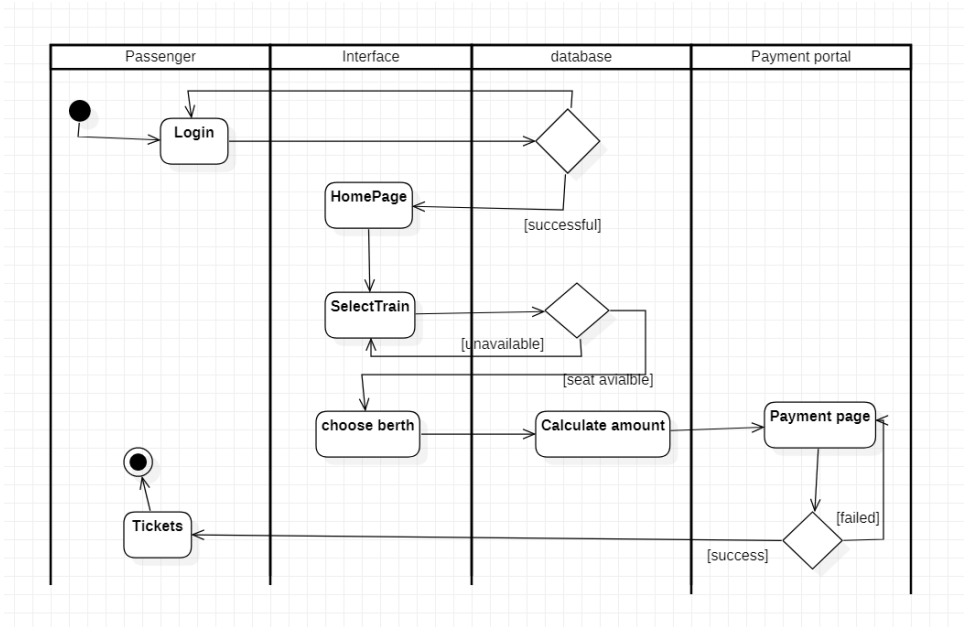
5. advanced sequence diagram





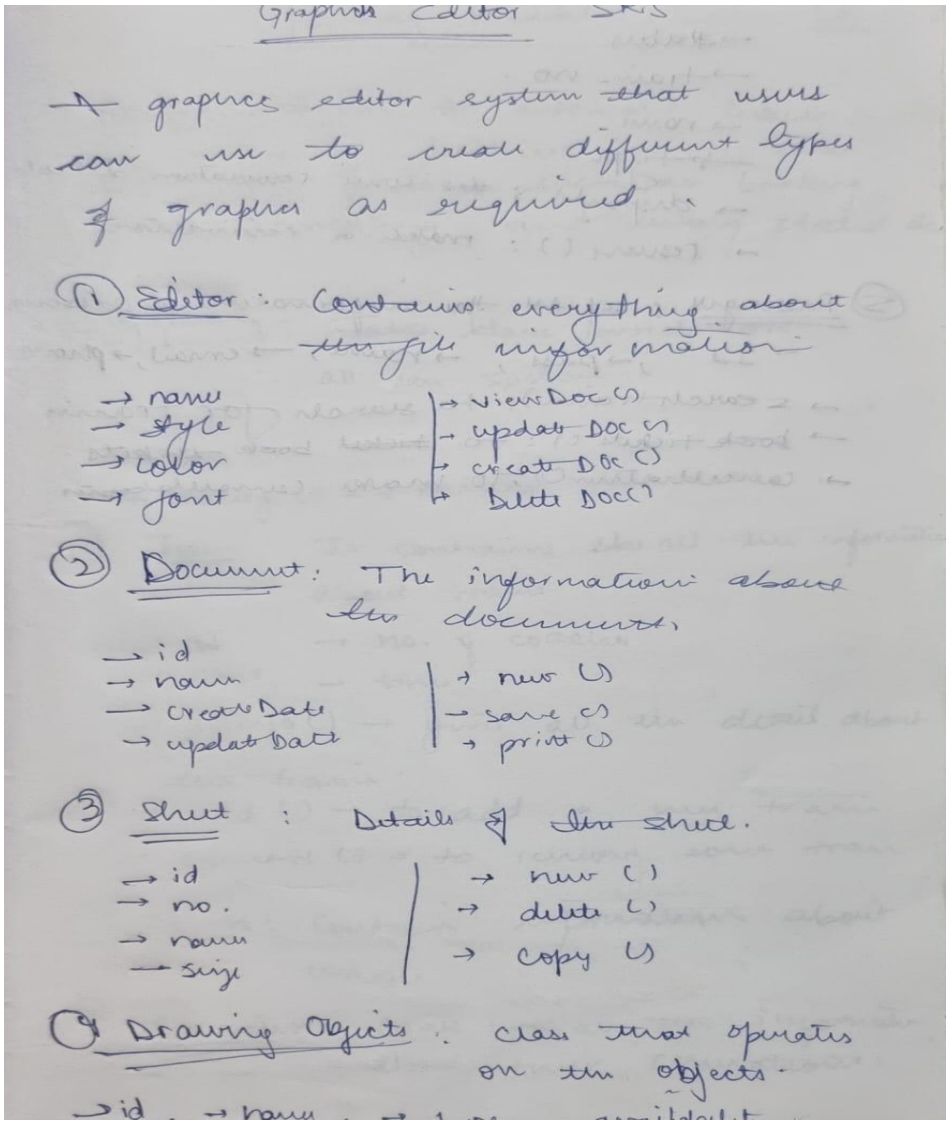
6. advanced activity diagram


Railway Reservation System
Passenger
Interface
Database
Payment portal
Login
HomePage
Select Trains
Choose Berth
Payment Page
Login
Booking

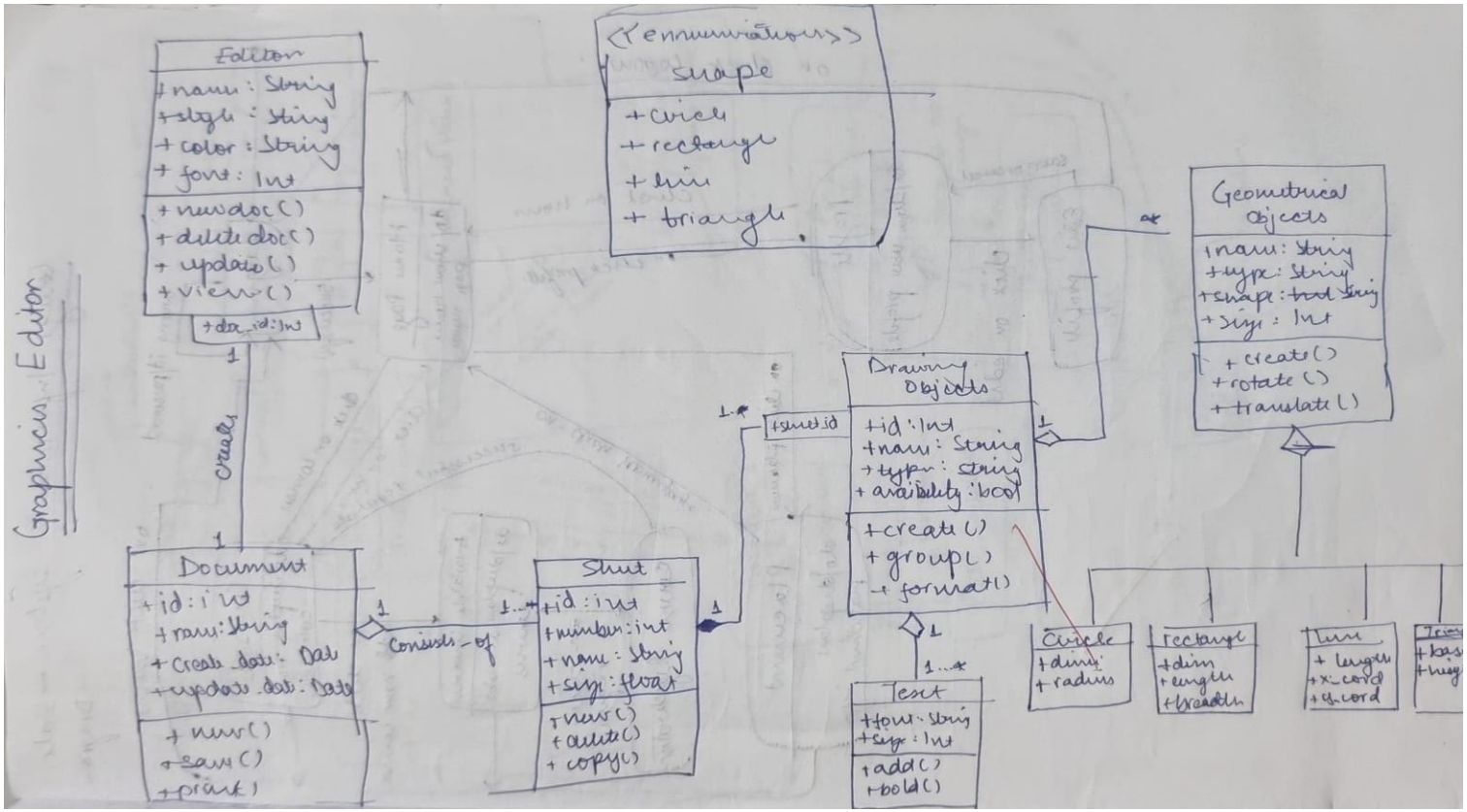

Passenger
Interface
database
Payment portal
Login
HomePage
[successful]
SelectTrain
[unavailable]
[seat avialble]
choose berth
Calculate amount
Payment page
[failed]
[success]
Tickets

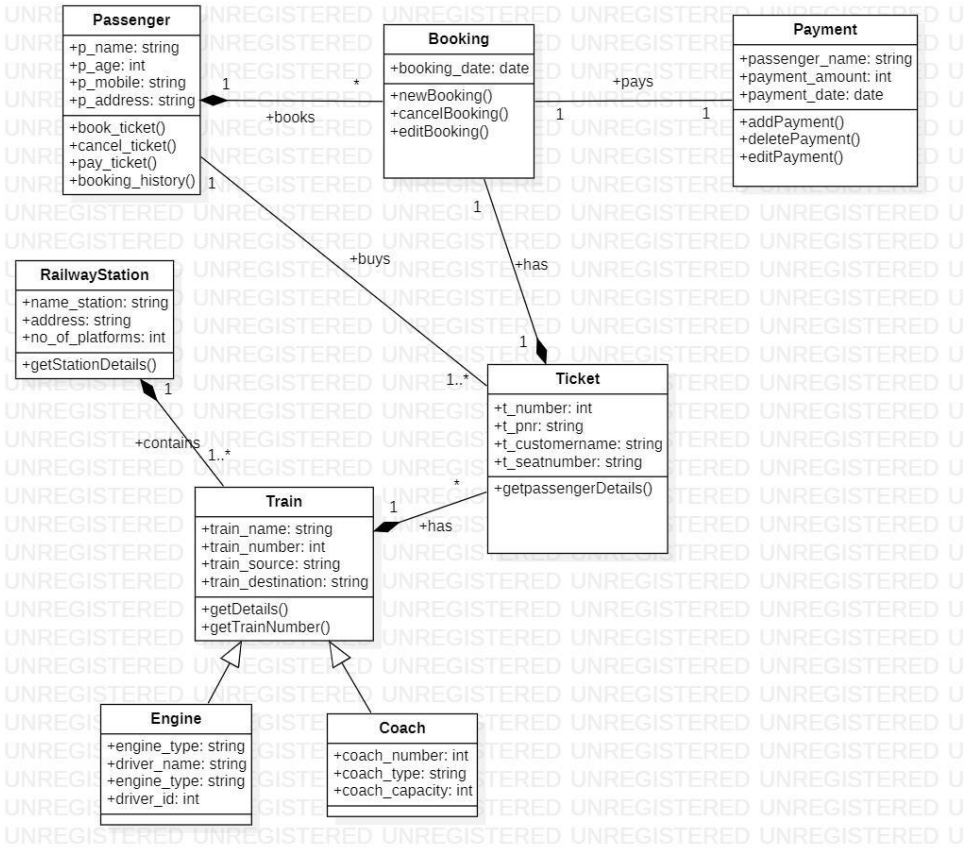
Exercise 7: Graphics Editor System

1. SRS

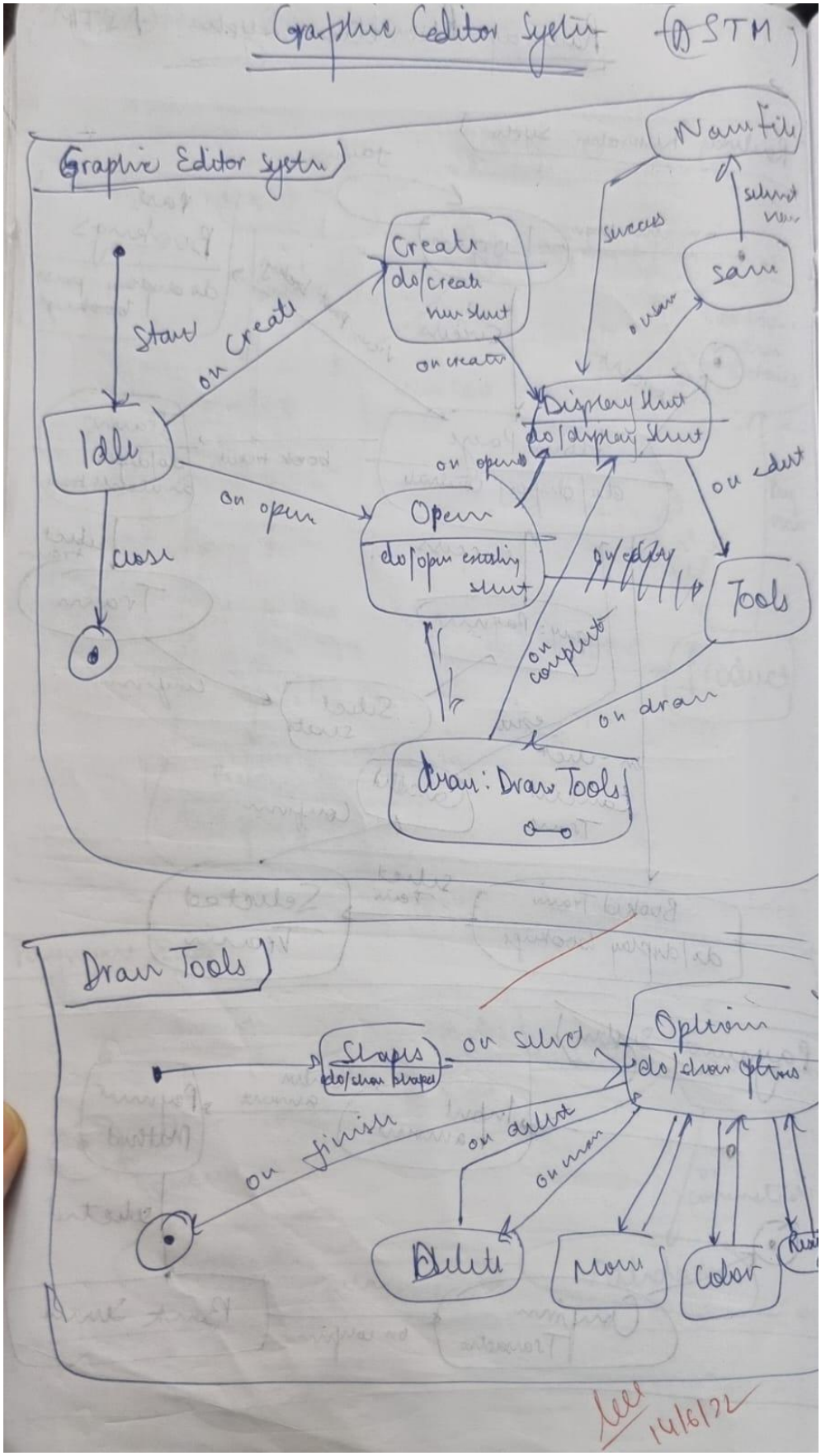


2. advanced class diagram

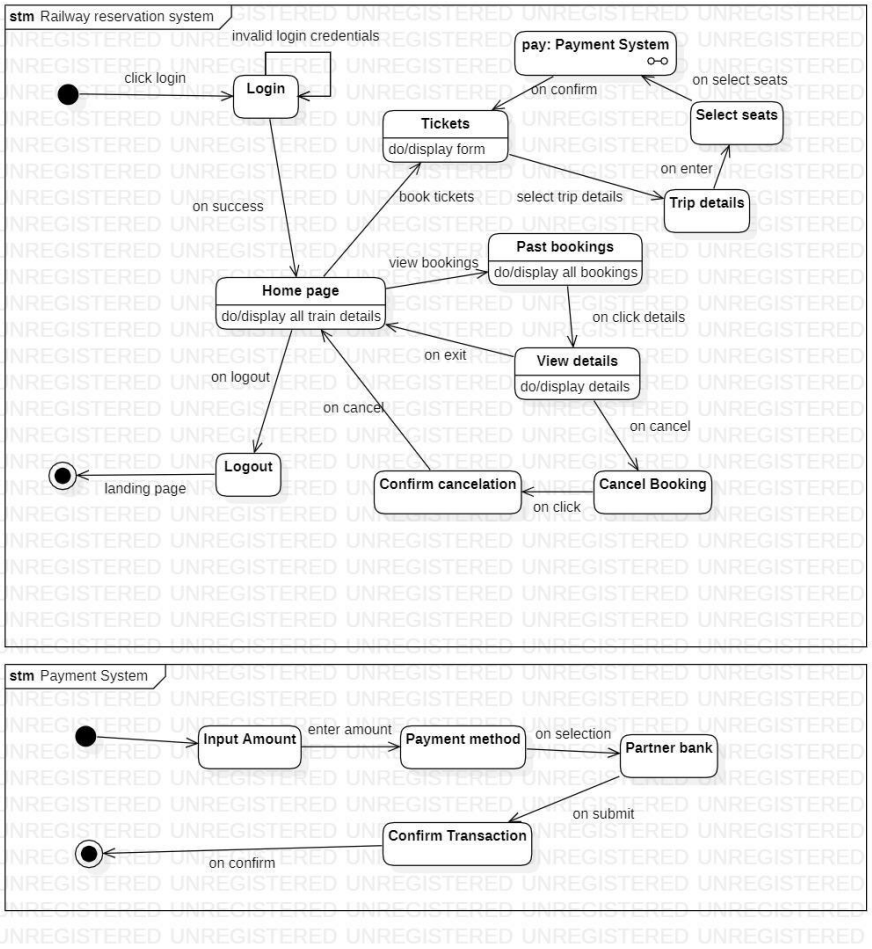

Graphics Editor
Editor
+ name : String
+ style : String
+ color : String
+ font : Int
+ newdoc()
+ deletedoc()
+ update()
+ view()
+ doc_id : Int
creates
1
1
<<enumerations>>
shape
+ circle
+ rectangle
+ line
+ triangle
Geometrical objects
+ name : String
+ type : String
+ shape : bool String
+ size : Int
+ create()
+ rotate()
+ translate()
Drawing Objects
+ id : Int
+ name : String
+ type : String
+ availability : bool
+ create()
+ group()
+ format()
+ struct_id
1..*
1
1
1..*
Document
+ id : Int
+ name : String
+ create_date : Date
+ update_date : Date
+ new()
+ save()
+ print()
1
consists of
1..*
Struct
+ id : Int
+ number : int
+ name : String
+ size : float
+ new()
+ delete()
+ copy()
1
Text
+ font : String
+ size : Int
+ add()
+ bold()
Circle
+ dim
+ radius
Rectangle
+ dim
+ length
+ breadth
Line
+ length
+ x_cord
+ y_cord



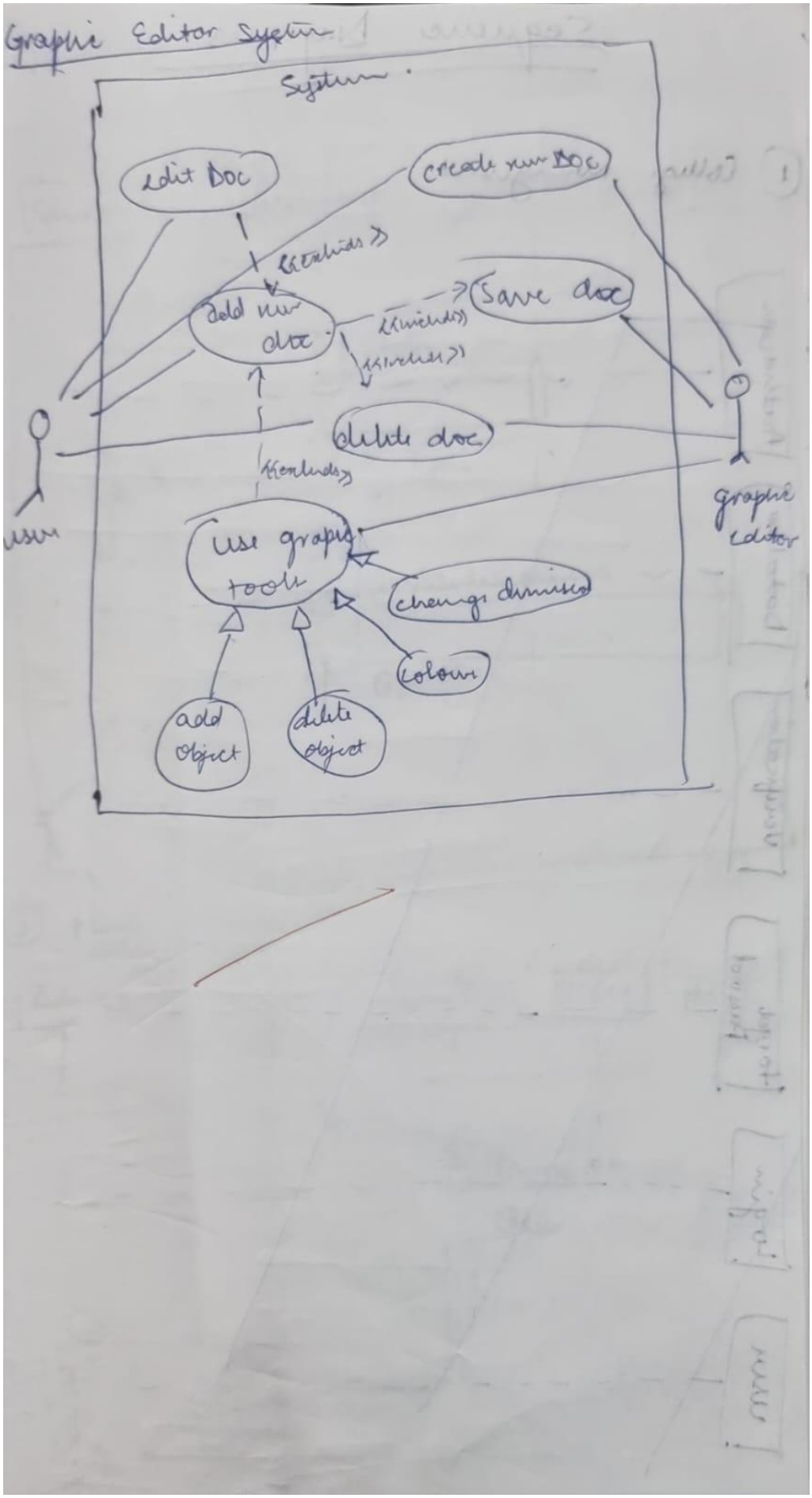
3. advanced state diagram

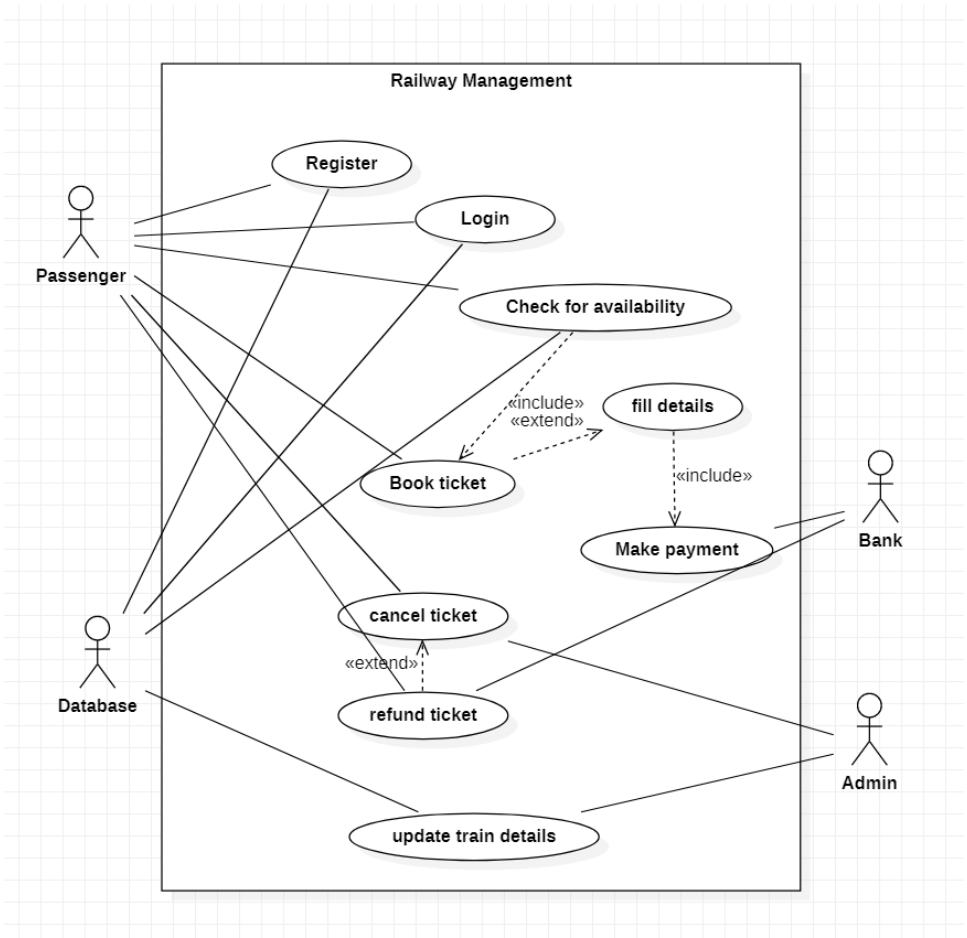

Graphic Editor system (STM)
Graphic Editor system
Name file
success
submit
Create
do/create new shape
Save
start
on create
on create
on save
Idle
Display shape
do/display shape
on open
on close
on open
Open
do/open existing shape
Tools
on complete
on draw
Draw: Draw Tools
Draw Tools
Shapes
do/draw shapes
on select
Options
do/draw options
on finish
on select
on move
Delete
Move
Color

14/6/22

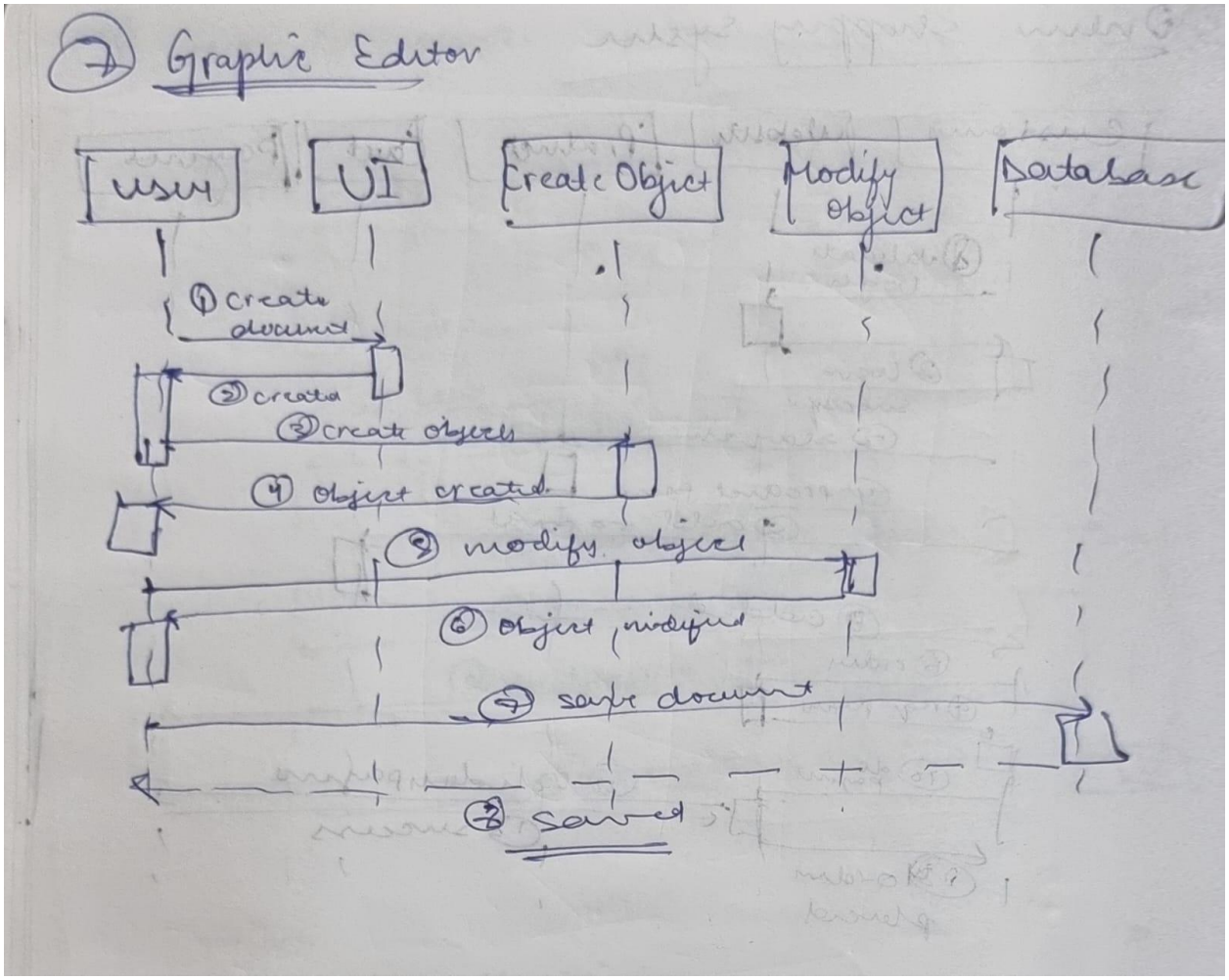


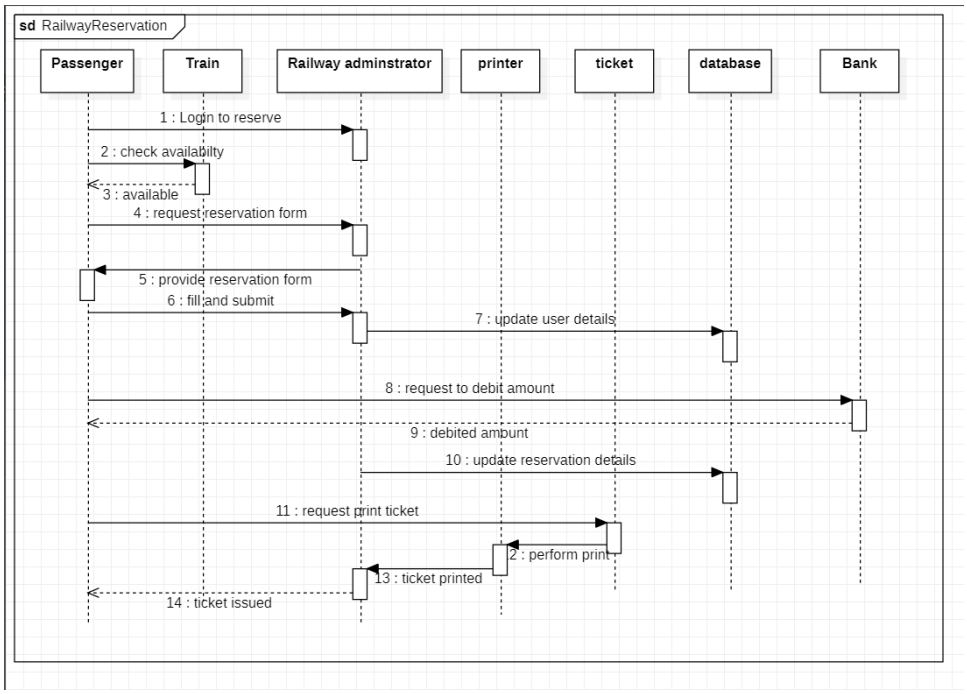
4. advanced use case diagram


Graphic Editor System
System
Edit Doc
Create new Doc
<<extends>>
add new doc
<<include>>
<<include>>
Save doc
delete doc
<<extends>>
user
graphic editor
use graphic tools
change dimension
colour
add object
delete object

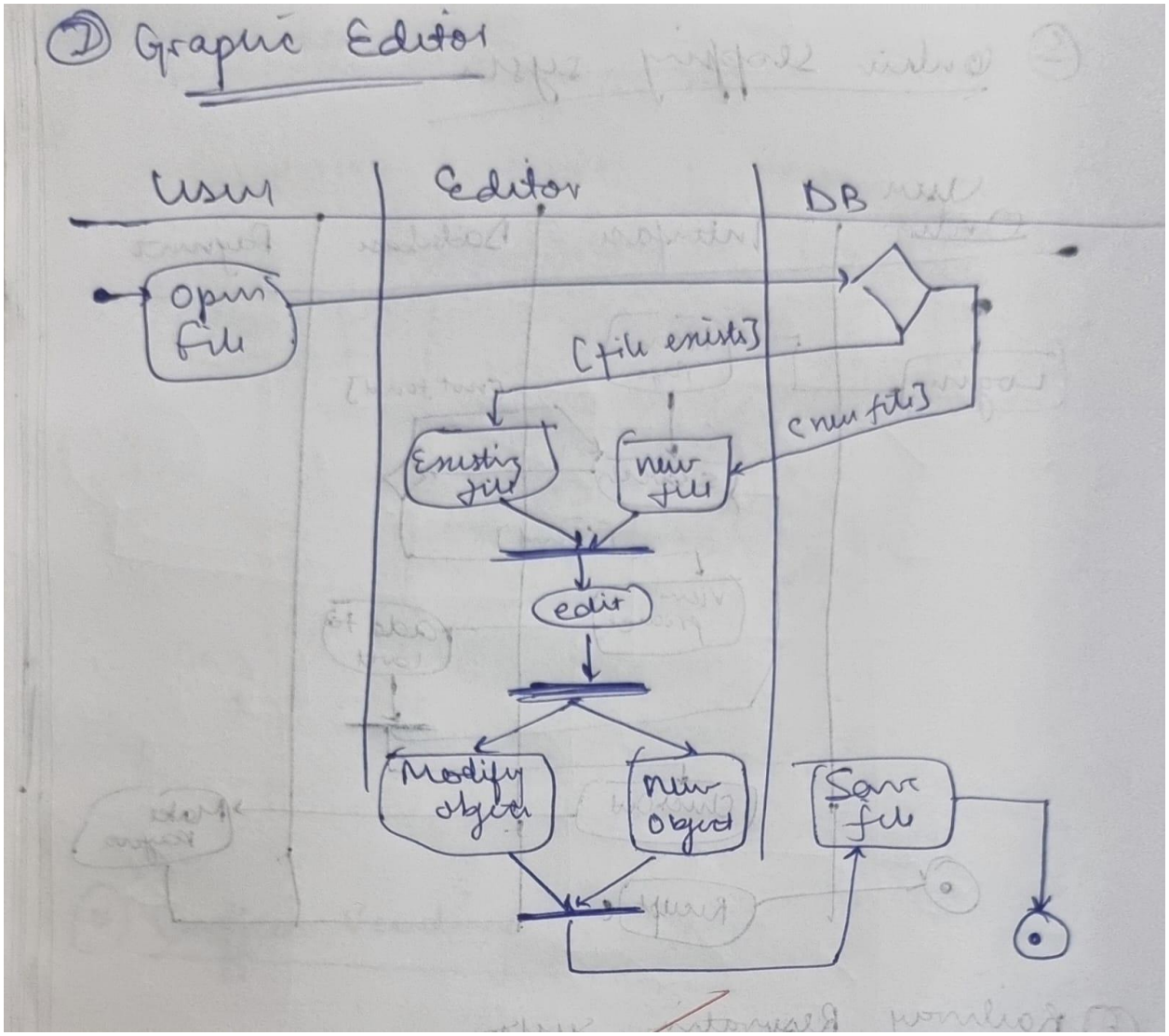


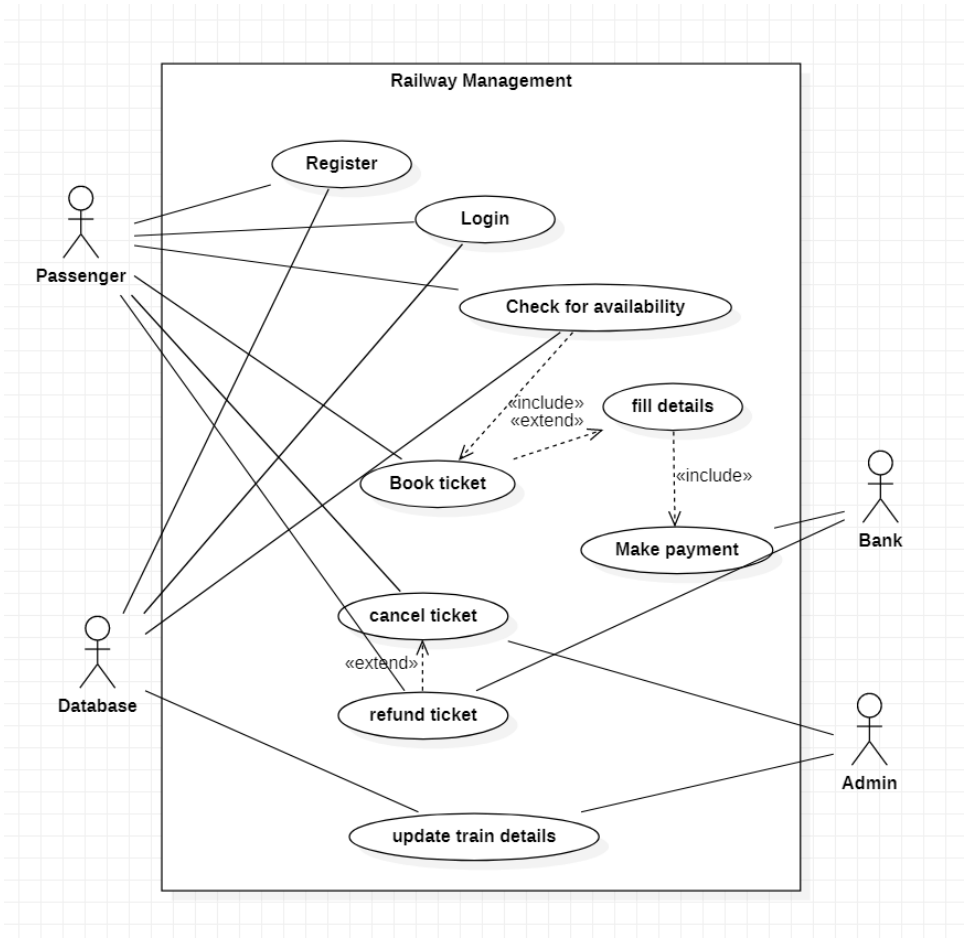
5. advanced sequence diagram


Graphic Editor
User
UI
Create Object
Modify Object
Database
1 create document
2 created
3 create objects
4 object created
5 modify object
6 object modified
7 save document
8 saved


sd RailwayReservation
Passenger
Train
Railway adminstrator
printer
ticket
database
Bank
1 : Login to reserve
2 : check availability
3 : available
4 : request reservation form
5 : provide reservation form
6 : fill and submit
7 : update user details
8 : request to debit amount
9 : debited amount
10 : update reservation details
11 : request print ticket
12 : perform print
13 : ticket printed
14 : ticket issued

6. advanced activity diagram




Railway Management
Register
Login
Passenger
Check for availability
«include»
«extend»
fill details
Book ticket
«include»
Bank
Make payment
cancel ticket
«extend»
Database
refund ticket
Admin
update train details