

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT on

## Object Oriented Analysis and Design (20CS6PCOMD)

*Submitted by*

**PranavKumar**  
**1BM19CS114**

*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 Analysis and Design ” carried out by **Pranav Kumar(1BM19CS114)**, 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 Analysis and Design (20CS6PCOMD)** work prescribed for the said degree.

Name of the Lab-Incharge  
Designation  
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	4
2	Hostel Management System	10
3	Stock Maintenance System	16
4	Coffee Vending Machine	22
5	Online Shopping System	28
6	Railway reservation system	34
7	Graphics Editor	40

## Course Outcome

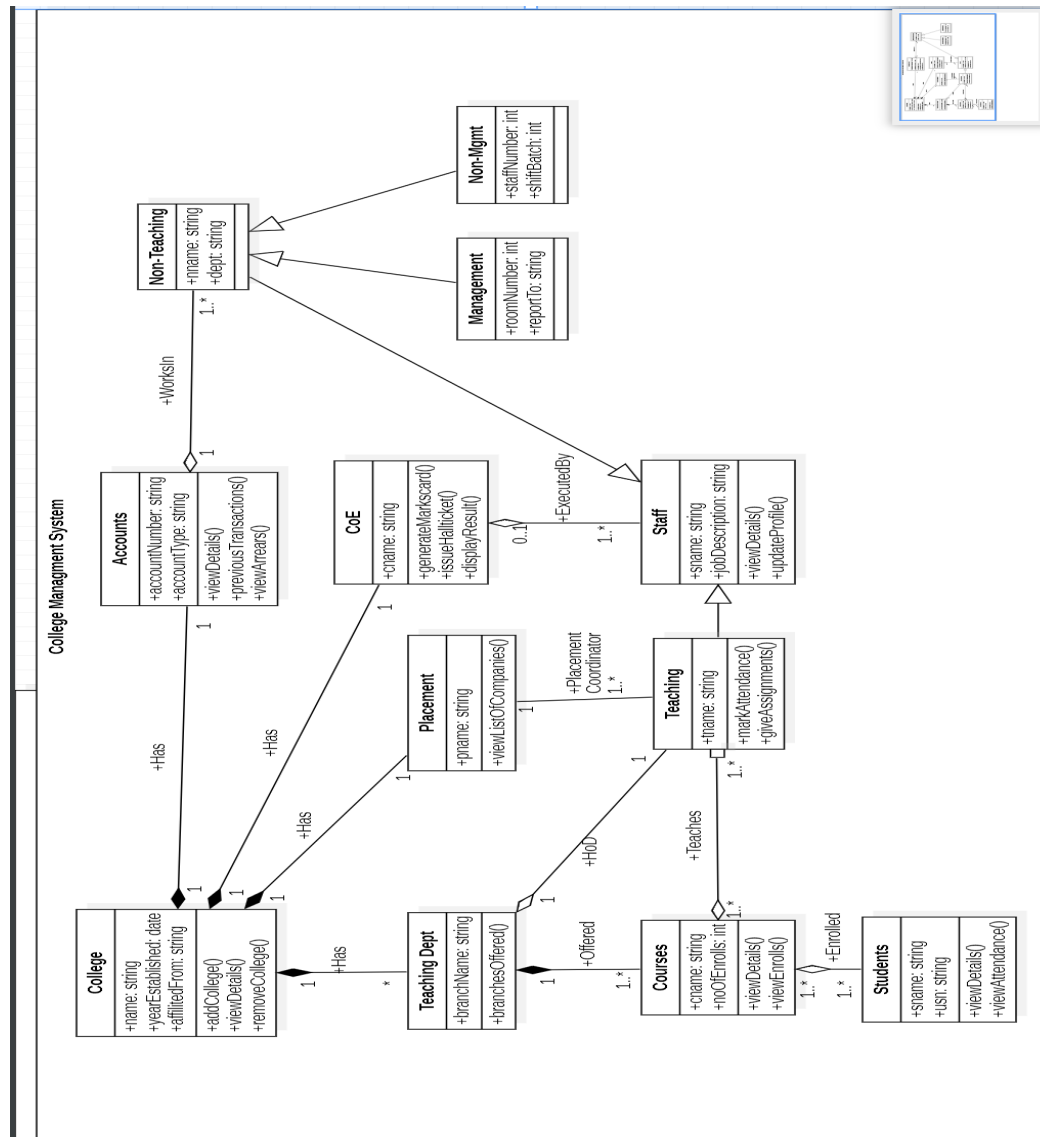
CO4	Ability to conduct practical experiment to solve a given problem using Unified Modeling language.
-----	---

# **1.COLLEGE INFORMATION SYSTEM**

## **Software Requirement Specification**

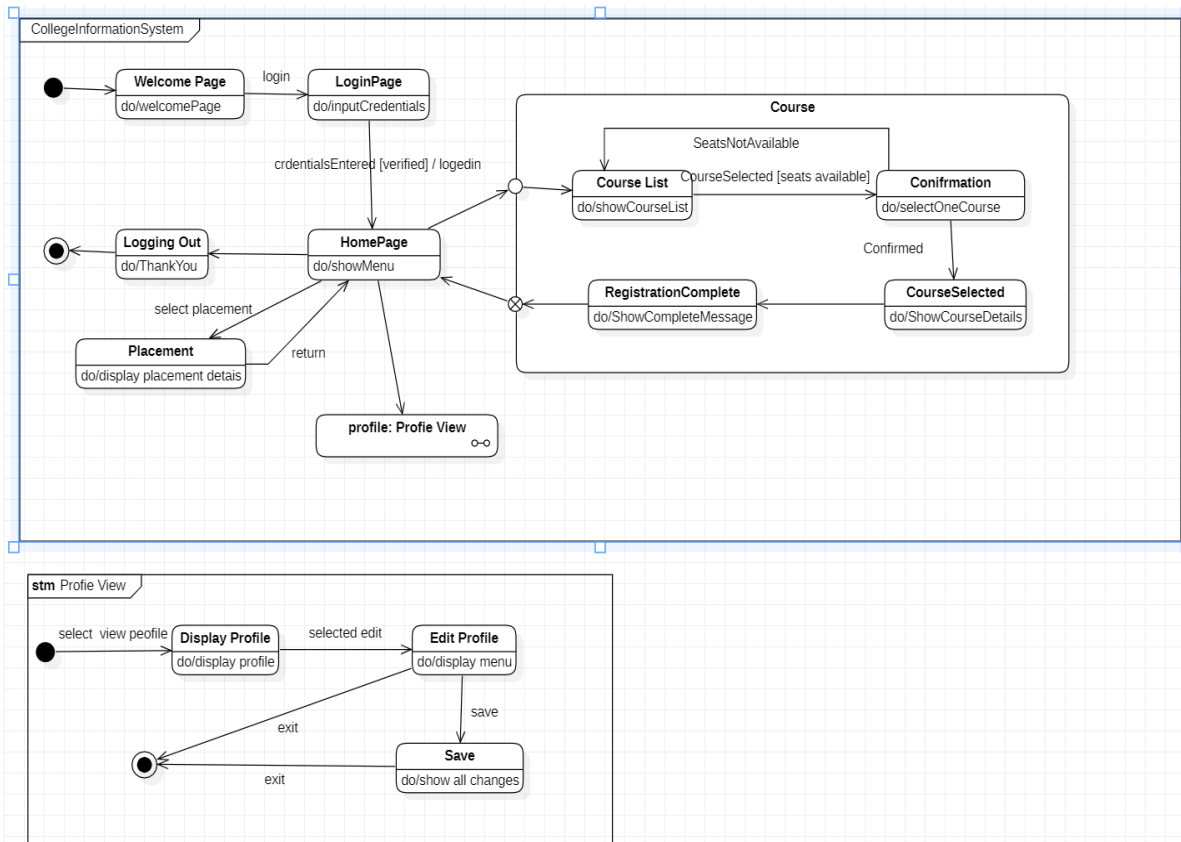
- College information system has admin who manages the staff ,student and department.
- Admin can view and modify the student's records like student's profile, attendance, fee, results, and details of teachers and other employees in college, their personal information and their attendance for their salaries.
- In this system, user authentication will be done by login by user name and password and classified by user type.
- Staff in college teach more than one course to many students and the staff who are teachers conduct examinations for students of the college
- The students of the college register themselves in the department and for the courses they are interested in and join the college by taking admission and following all the admission procedures.
- There are different types of examination conducted by the college for the students. Internals and semester end examination are two of them.
- Every course has a name and its unique name. Every course has different subjects and every subject has its own unique name.
- Department will provide the details about departments within a college with their name and every department have its Department name.

## Advance Class Diagram :



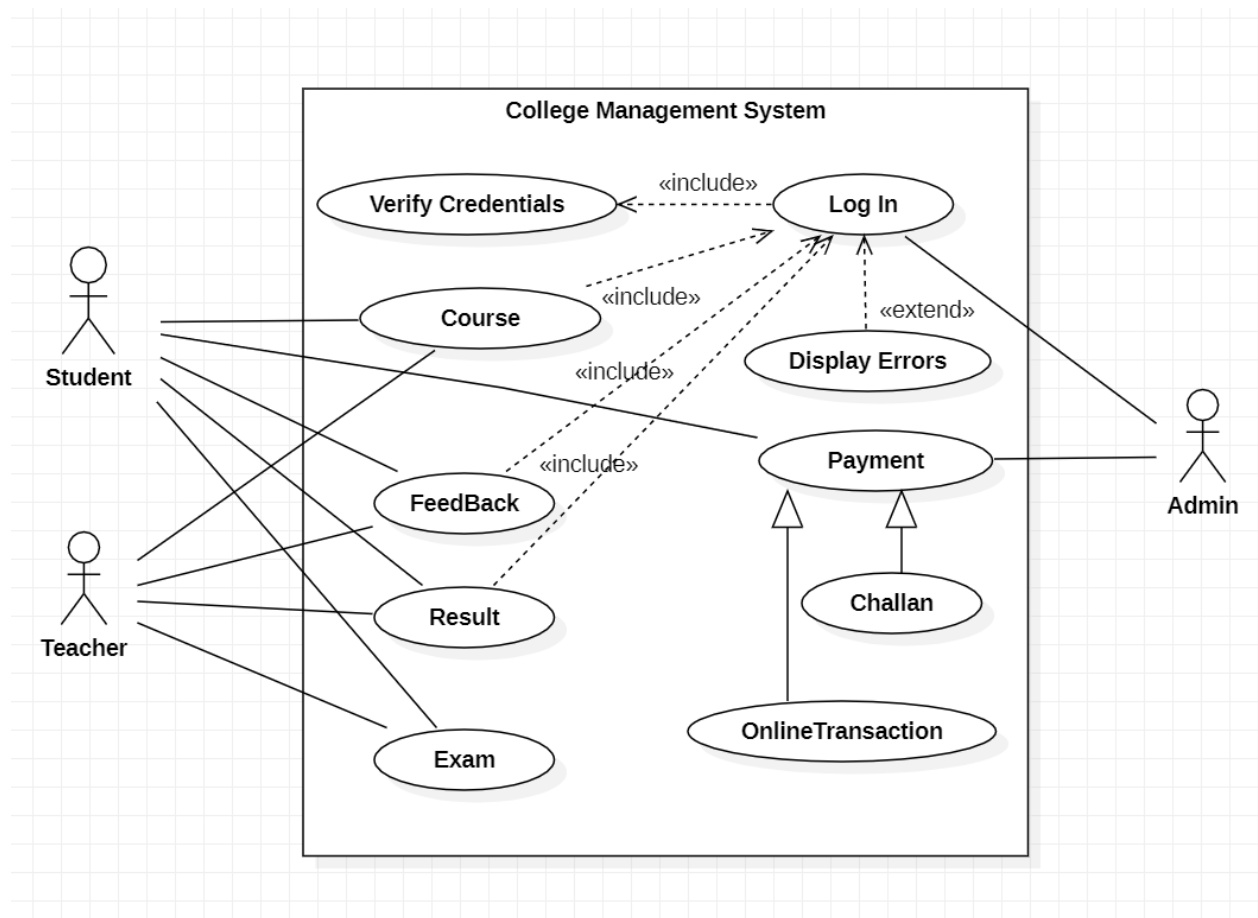
Admin can view and modify the student's records, teachers and department details. The students of the college register themselves in the department and examination and for the courses they are interested in and join the college by taking admission and following all the admission procedures. College conducts Internals and semester end examination for students.

## Advance State Diagram :



The above state diagram describes the states the admin goes through in uploading information of student, staff and department. The admin first needs to login which then leads to the validate state, where the login id and password are validated. If invalid it then goes back to the login state or goes to the get information state. Upon receiving the correct information it goes to the upload state and then to commit state to save all changes. The admin first needs to login and be cleared of their permissions. The admin can then manage information related to the student, teacher, or department. After necessary changes the admin can update the information and logout from the system.

## Advanced Use Case Diagram :



Actors:

Admin: the person who manages everything Student: A person who uses the system

COE: A person who is responsible for examinations Staff: A person who works in the college

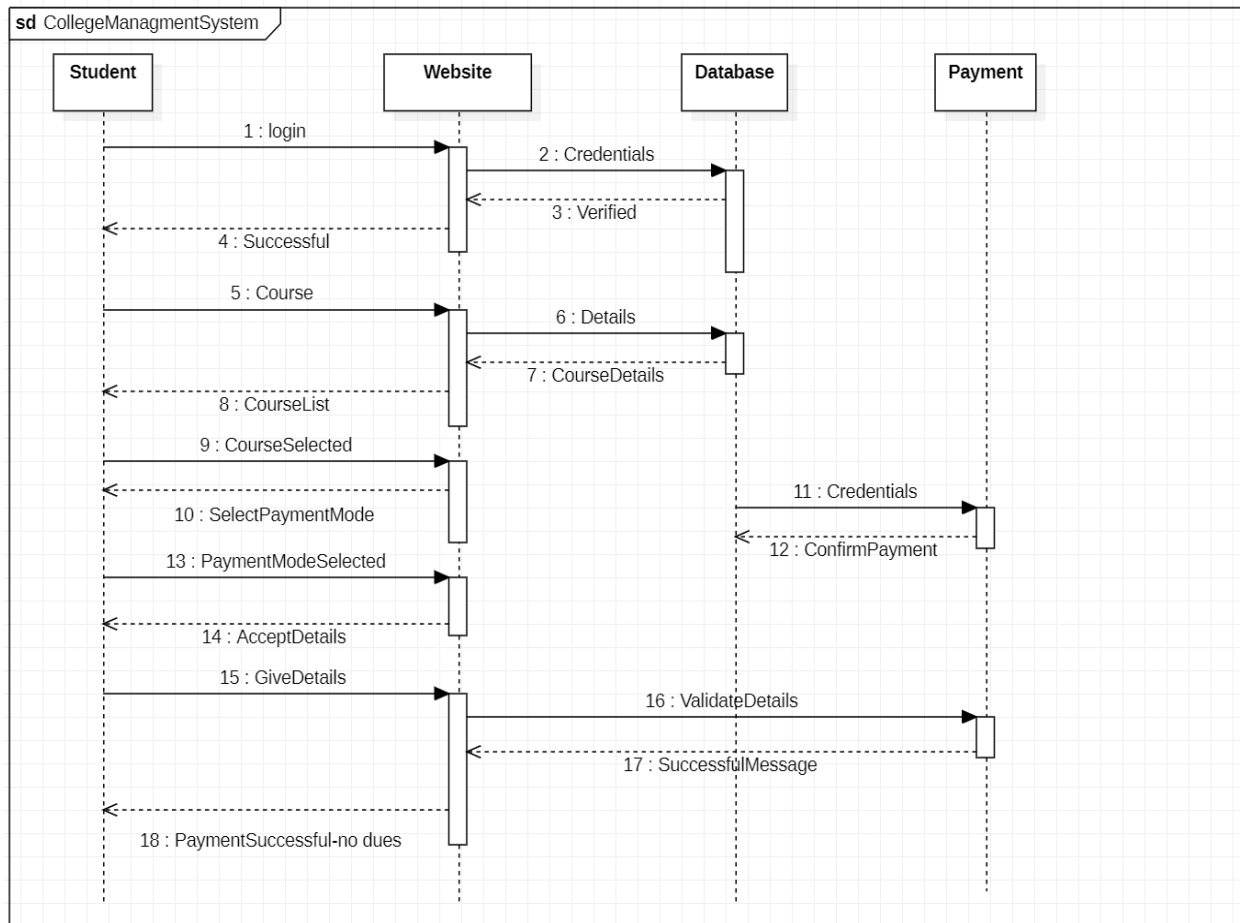
Use Cases:

Manage details: the admin can update, insert or delete the data. View results: displays the result of students.

Subject details: various details related to subject is displayed. View student details: the details of student is displayed

Declare results: the results of exams written by student is displayed.

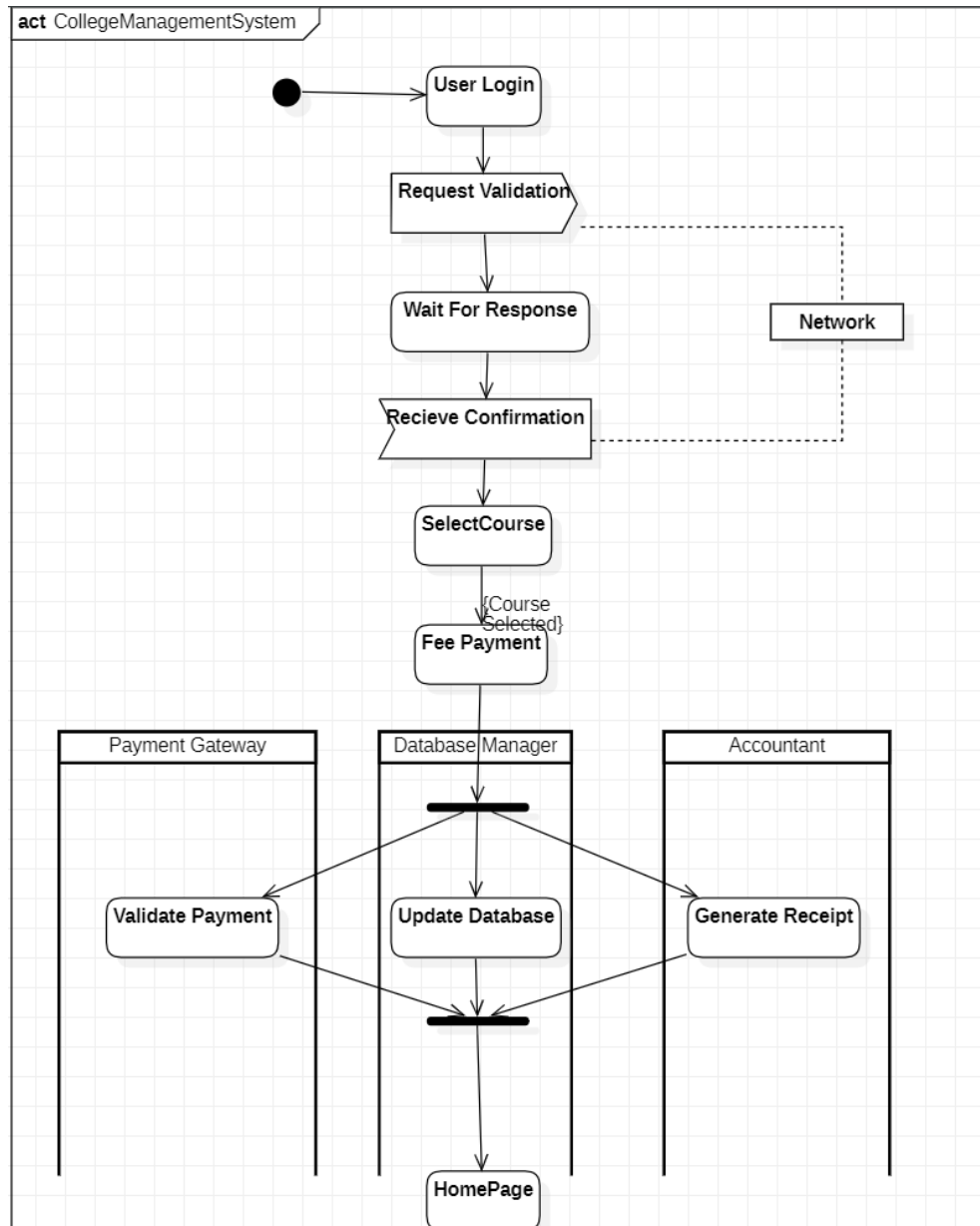
## Advanced Sequence Diagram :



The above sequence diagram gives the interaction between objects while a user is logging into a system. The user enters login information in the website which sends to the server, where the information is validated and the appropriate reply message is displayed to the user.



## Advanced Activity Diagram :



The above activity diagram has three swimlanes mainly admin, database and accountant. The admin can login and manage information. The database verifies the login information and on success has two options. The student can login and select a course .

## **2.HOSTEL MANAGEMENT SYSTEM**

### **Software Requirement Specification**

- Hostel management system has admin who manages the hostel,allot-es and payment methods.The admin will allocate a room to student according to the section or class. The admin will also keep track of the payment made by the student/allot-es .
- As the student's course is over they will vacate their rooms. So it is required for the administrator to remove their records from the database tables.
- The allot-es makes payment according to the bill generated which have the attributes bill number,type and date.
- The details of the students staying in the hostels like name, place, address, contact details is maintained in the database.
- The hostel is categorized into two types I.e boys and girls hostel.Each hostel type has different costs ,warden and name.
- A hostel is made up of mess and rooms.A mess account will also generate. This account having the mess status of the whole month. On the base of this account monthly charges of mess of a student will be defined.
- The hostel management system will allow renewing the student's registration every year.the rooms of hostel are composed of table and beds,where a count of the same is maintained and the allot-es can use them as they wish.

UML Class Diagram for Hostel Management System

```

classDiagram
    class Accounts {
        +accountNumber: int
        +feeStatus: bool
        +paymentMode: PaymentMode
        +giveFeeDetails()
        +arrear()
    }
    class HostelManagement {
        +college: string
        +deleteHostel()
        +viewHostel()
    }
    class Secretary {
        +name: string
        +workHours: int
        +gender: string
        +login()
        +addHostel()
        +addWarden()
        +addStudent()
    }
    class Students {
        +name: string
        +phone: int
        +roomNo: int
        +giveFeedback()
        +leaveHostel()
    }
    class Mess {
        +messType: string
        +workingDays: list
        +getMessMenu()
    }
    class Maintenance {
        +noOfStaffs: int
        +types: int
        +noOfComplaints: int
        +acceptComplaints()
        +assignStaffs()
    }
    class Accommodation {
        +hname: string
        +forGender: string
        +totalRooms: int
        +address: string
    }
    class Rooms {
        +roomNo: int
        +roomType: string
        +roomDesc: string
        +addRoom()
        +deleteRoom()
        +searchRoom()
        +viewVacancy()
    }
    class Warden {
        +name: string
        +gender: string
        +age: int
        +phone: int
        +yearOfExperience: int
        +manageHostel()
        +getStudentList()
    }
    class MultiplesSharing {
        +noOfStudents: int
    }
    class SingleSharing {
        +acornNon: string
    }
    class Cupboard {
        +compartments: int
        +height: int
        +width: int
        +length: int
    }
    class Bed {
        +rawMaterial: string
        +alotedTo: int
        +size: string
    }
    class Table {
        +rawMaterial: string
        +supplier: string
        +alotedTo: int
        +length: int
        +width: int
        +height: int
        +details()
    }
    class PaymentInfo {
        +amount: int
    }
    class PaymentMode {
        creditCard
        debitCard
        netBanking
        cash
    }

    Accounts "1" -- "1" HostelManagement : +Has
    HostelManagement "1" -- "1" Secretary : +Responsible
    HostelManagement "1" -- "1" Students : +Effects
    HostelManagement "1" -- "1" Mess : +Has
    HostelManagement "1" -- "1" Maintenance : +Maintains
    HostelManagement "1" -- "1" Accommodation : +Maintains
    HostelManagement "1" -- "1" Warden : +TakeCareOf
    Students "1" -- "1" Mess : +EatsIn
    Students "1" -- "1" Rooms : +StaysIn
    Students "1" -- "1" Accommodation : +Has
    Mess "1" -- "1" Maintenance : +Maintains
    Accommodation "1" -- "1" Rooms : +Has
    Accommodation "1" -- "1" Warden : +Has
    Rooms "1" -- "1" Warden : +Has
    Rooms "1" -- "1" MultiplesSharing : +Has
    Rooms "1" -- "1" SingleSharing : +Has
    Rooms "1" -- "1" Cupboard : +Has
    Rooms "1" -- "1" Bed : +Has
    Rooms "1" -- "1" Table : +Has
    PaymentInfo "1" -- "1" Accounts : ..
    PaymentInfo "1" -- "1" Students : ..
    PaymentInfo "1" -- "1" Mess : ..
    PaymentInfo "1" -- "1" Maintenance : ..
    PaymentInfo "1" -- "1" Accommodation : ..
    PaymentInfo "1" -- "1" Rooms : ..
    PaymentInfo "1" -- "1" Warden : ..
    PaymentInfo "1" -- "1" MultiplesSharing : ..
    PaymentInfo "1" -- "1" SingleSharing : ..
    PaymentInfo "1" -- "1" Cupboard : ..
    PaymentInfo "1" -- "1" Bed : ..
    PaymentInfo "1" -- "1" Table : ..
    
```

The diagram illustrates the relationships between various components of a Hostel Management System. Key classes include Accounts, HostelManagement, Secretary, Students, Mess, Maintenance, Accommodation, Rooms, Warden, MultiplesSharing, SingleSharing, Cupboard, Bed, Table, PaymentInfo, and PaymentMode. Relationships are defined by associations with multiplicity and roles, such as 'Has', 'Effects', 'Maintains', 'TakeCareOf', 'EatsIn', 'StaysIn', and 'Has'.

UML Class Diagram for Hostel Management System

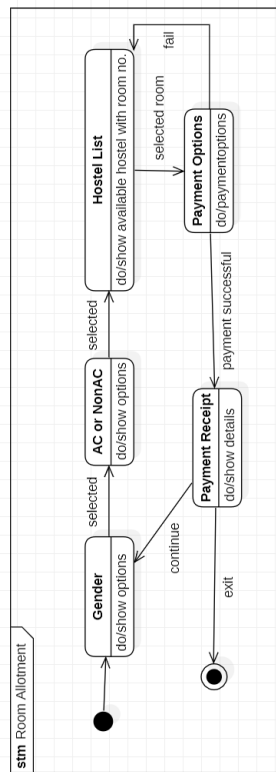
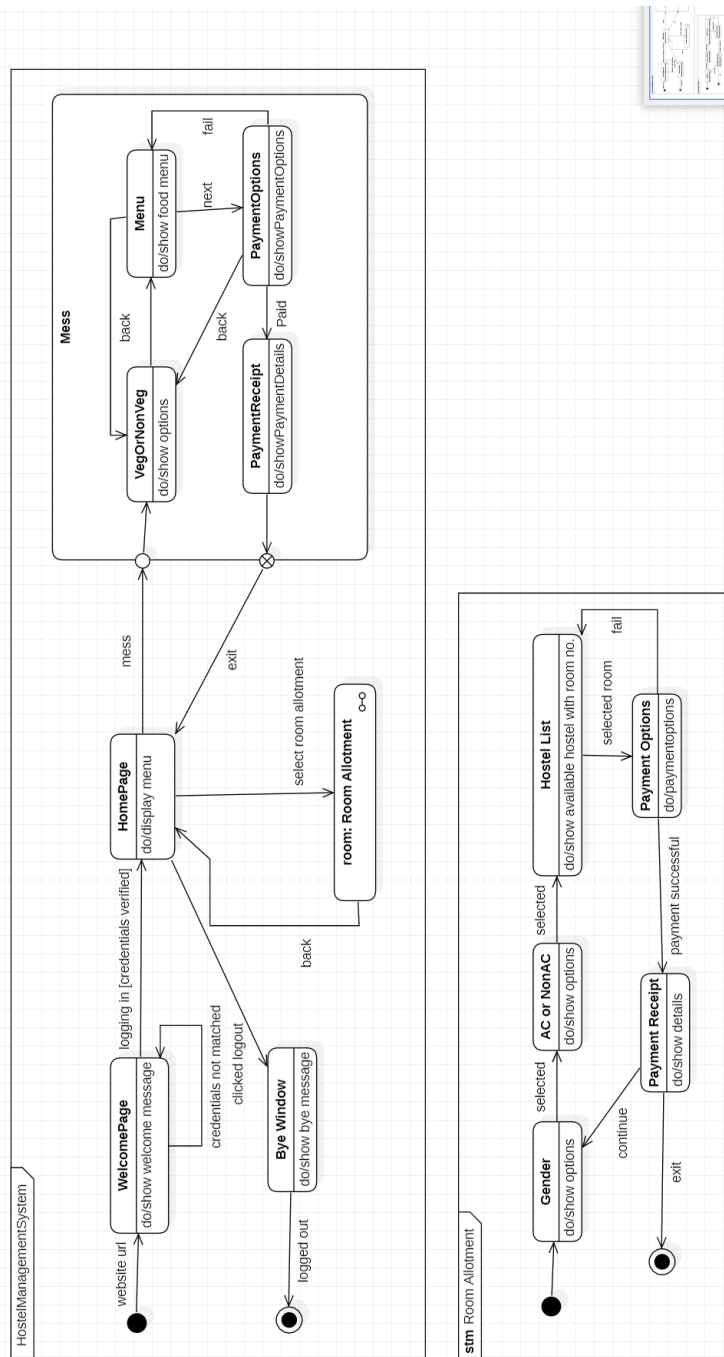
```

classDiagram
    class Accounts {
        +accountNumber: int
        +feeStatus: bool
        +paymentMode: PaymentMode
        +giveFeeDetails()
        +arrear()
    }
    class HostelManagement {
        +college: string
        +deleteHostel()
        +viewHostel()
    }
    class Secretary {
        +name: string
        +workHours: int
        +gender: string
        +login()
        +addHostel()
        +addWarden()
        +addStudent()
    }
    class Students {
        +name: string
        +phone: int
        +roomNo: int
        +giveFeedback()
        +leaveHostel()
    }
    class Mess {
        +messType: string
        +workingDays: list
        +getMessMenu()
    }
    class Accommodation {
        +hname: string
        +forGender: string
        +totalRooms: int
        +address: string
    }
    class Rooms {
        +roomNo: int
        +roomType: string
        +roomDesc: string
        +addRoom()
        +deleteRoom()
        +searchRoom()
        +viewVacancy()
    }
    class Warden {
        +name: string
        +gender: string
        +age: int
        +phone: int
        +yearOfExperience: int
        +manageHostel()
        +getStudentList()
    }
    class MultiplesSharing {
        +noOfStudents: int
    }
    class SingleSharing {
        +acornNo: string
    }
    class Cupboard {
        +compartments: int
        +height: int
        +width: int
        +length: int
    }
    class Bed {
        +rawMaterial: string
        +alotedTo: int
        +size: string
    }
    class Table {
        +rawMaterial: string
        +supplier: string
        +alotedTo: int
        +length: int
        +width: int
        +height: int
        +details()
    }
    class PaymentInfo {
        +amount: int
    }
    class PaymentMode {
        creditCard
        debitCard
        netBanking
        cash
    }
    class Maintenance {
        +noOfStaffs: int
        +types: int
        +noOfComplaints: int
        +acceptComplaints()
        +assignStaffs()
    }

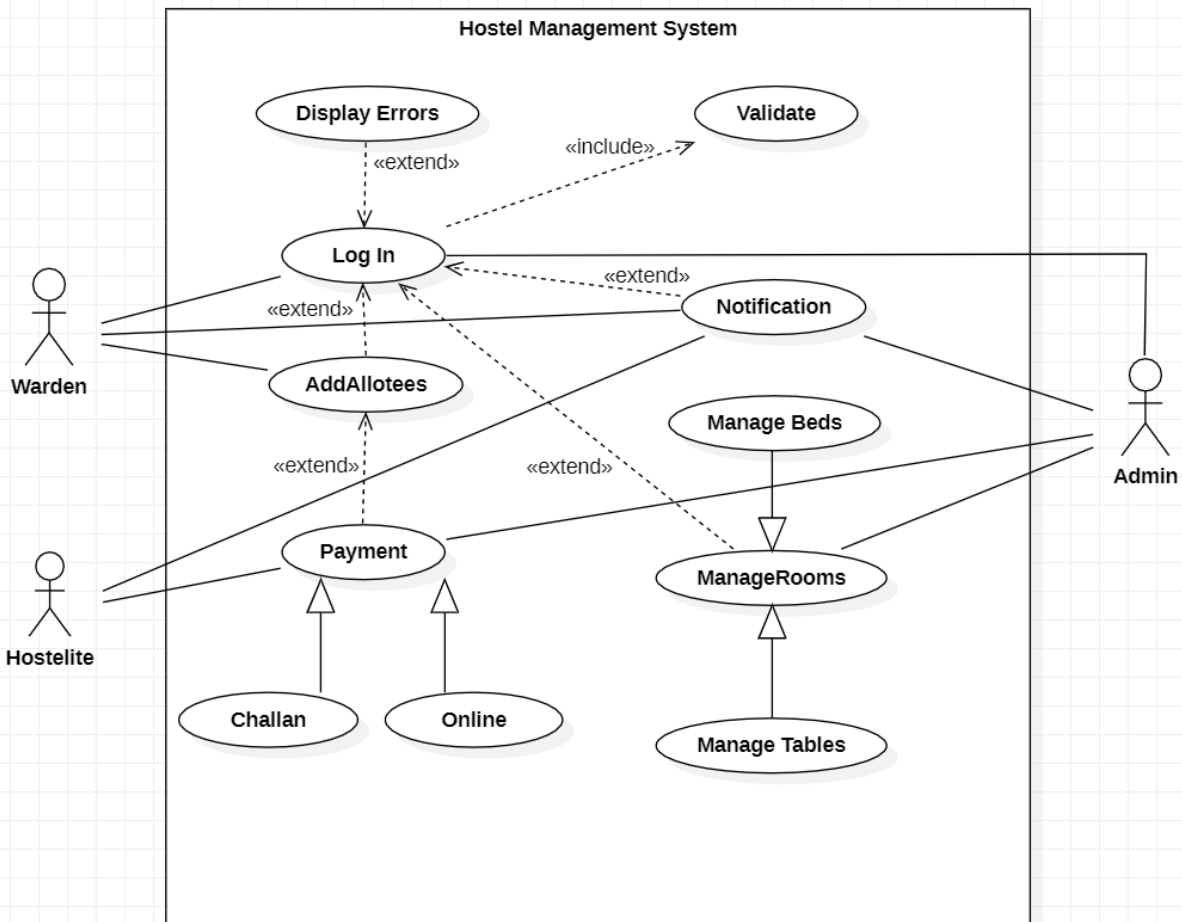
    Accounts "1" -- "1..*" HostelManagement
    Accounts "1" -- "1" PaymentInfo
    HostelManagement "1" -- "1" Secretary
    HostelManagement "1" -- "1" Students
    HostelManagement "1" -- "1" Mess
    HostelManagement "1" -- "1" Accommodation
    HostelManagement "1" -- "1" Warden
    Secretary "1" -- "1" HostelManagement : +Responsible
    Secretary "1" -- "1" Accommodation : +Maintains
    Secretary "1" -- "1" Warden : +TakeCareOf
    Secretary "1" -- "1" Maintenance : +Maintains
    Students "1..*" -- "1" Mess : +EatsIn
    Students "1..*" -- "1" Rooms : +StaysIn
    Students "1..*" -- "1" Accommodation : +Has
    Students "1..*" -- "1" Warden : +Has
    Students "1..*" -- "1" MultiplesSharing : +Has
    Students "1..*" -- "1" SingleSharing : +Has
    Students "1..*" -- "1" Cupboard : +Has
    Students "1..*" -- "1" Bed : +Has
    Students "1..*" -- "1" Table : +Has
    Mess "1" -- "1..*" Accommodation : +Maintains
    Mess "1" -- "1..*" Rooms : +Maintains
    Accommodation "1" -- "1..*" Rooms : +Has
    Accommodation "1" -- "1..*" Warden : +Has
    Rooms "1" -- "1..*" Warden : +Has
    Rooms "1" -- "1..*" MultiplesSharing : +Has
    Rooms "1" -- "1..*" SingleSharing : +Has
    Rooms "1" -- "1..*" Cupboard : +Has
    Rooms "1" -- "1..*" Bed : +Has
    Rooms "1" -- "1..*" Table : +Has
    Warden "1" -- "1..*" MultiplesSharing : +Has
    Warden "1" -- "1..*" SingleSharing : +Has
    Warden "1" -- "1..*" Cupboard : +Has
    Warden "1" -- "1..*" Bed : +Has
    Warden "1" -- "1..*" Table : +Has
    MultiplesSharing "1" -- "1..*" Cupboard : +Has
    MultiplesSharing "1" -- "1..*" Bed : +Has
    MultiplesSharing "1" -- "1..*" Table : +Has
    SingleSharing "1" -- "1..*" Cupboard : +Has
    SingleSharing "1" -- "1..*" Bed : +Has
    SingleSharing "1" -- "1..*" Table : +Has
    Cupboard "1" -- "1..*" Bed : +Has
    Cupboard "1" -- "1..*" Table : +Has
    Bed "1" -- "1..*" Table : +Has
    PaymentInfo "1" -- "1..*" Accounts
    PaymentInfo "1" -- "1..*" Students
    PaymentInfo "1" -- "1..*" Mess
    PaymentInfo "1" -- "1..*" Accommodation
    PaymentInfo "1" -- "1..*" Rooms
    PaymentInfo "1" -- "1..*" Warden
    PaymentInfo "1" -- "1..*" MultiplesSharing
    PaymentInfo "1" -- "1..*" SingleSharing
    PaymentInfo "1" -- "1..*" Cupboard
    PaymentInfo "1" -- "1..*" Bed
    PaymentInfo "1" -- "1..*" Table
    Maintenance "1" -- "1..*" Secretary
    Maintenance "1" -- "1..*" Accommodation
    Maintenance "1" -- "1..*" Warden
    Maintenance "1" -- "1..*" MultiplesSharing
    Maintenance "1" -- "1..*" SingleSharing
    Maintenance "1" -- "1..*" Cupboard
    Maintenance "1" -- "1..*" Bed
    Maintenance "1" -- "1..*" Table
  
```

The diagram illustrates the relationships between various entities in a hostel management system. Key classes include Accounts, HostelManagement, Secretary, Students, Mess, Accommodation, Rooms, Warden, MultiplesSharing, SingleSharing, Cupboard, Bed, Table, PaymentInfo, and PaymentMode. Relationships are defined by associations with multiplicity and roles, such as 'Responsible', 'Maintains', 'TakeCareOf', 'EatsIn', 'StaysIn', 'Has', and 'Effects'.

## Advance State Diagram :



## Advance Use Case Diagram :



Actors:

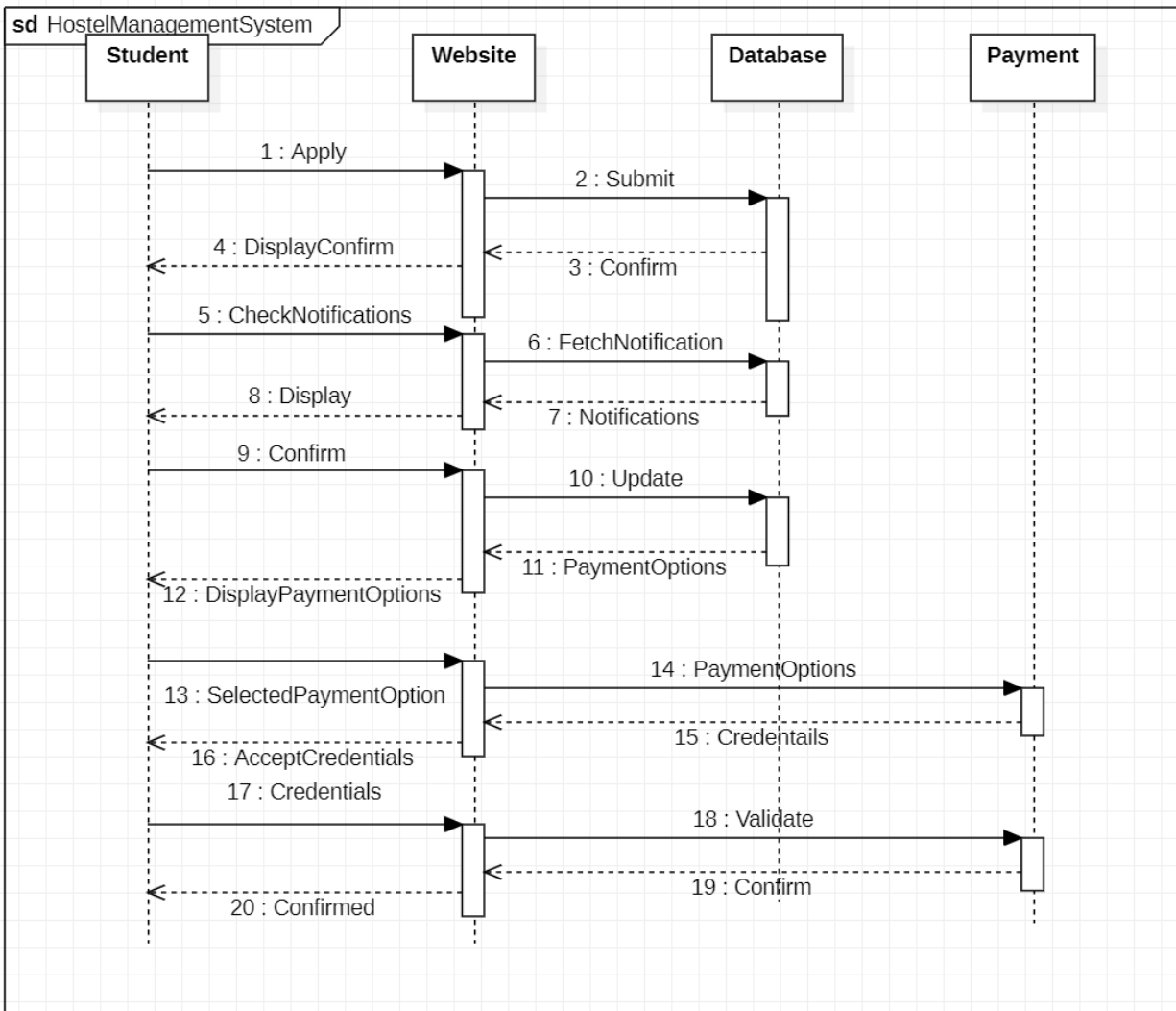
Admin: the person who manages the whole system  
Warden : the person who manages the allottees  
Student : the person who uses the hostel system

Use Cases:  
Manage hostel : allows actor to update delete or add information  
Login : allows actors to login into the system.

Add allottee: the students are allotted hostel rooms

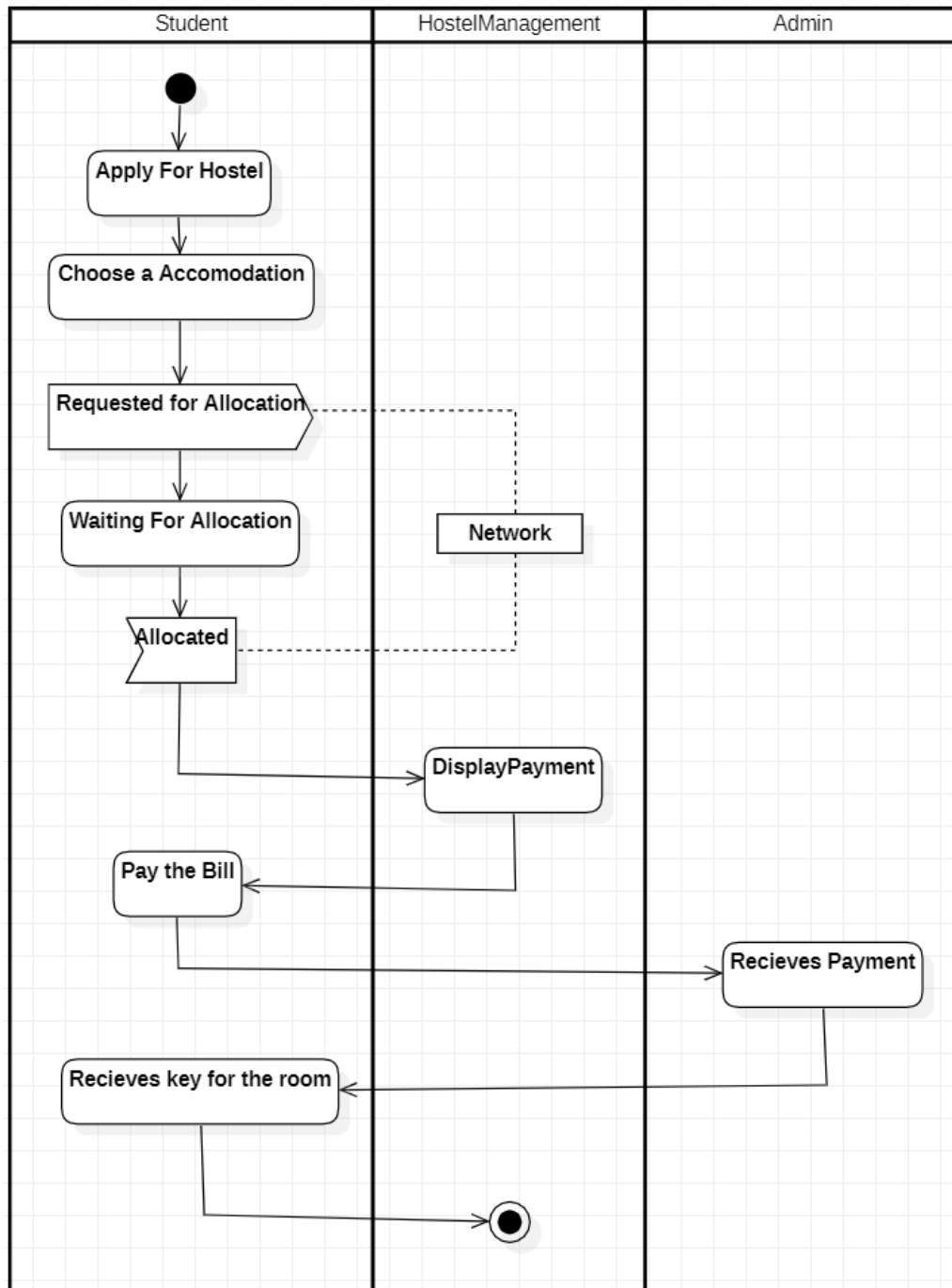
Book hostel: the student can select the hostel they wish to stay in. Pay fees: the fees payment is done by this use case.

## Advance Sequence Diagram :



The above sequence diagram give the steps involved in a student logging in, booking a room, which is verified in the database and the payment for the same is made by the student.

## Advance Activity Diagram:



The activity diagram tells about the activities involved in payment of fees. The student first needs to login and then pay fees for both hostel and mess and finally give feedback.

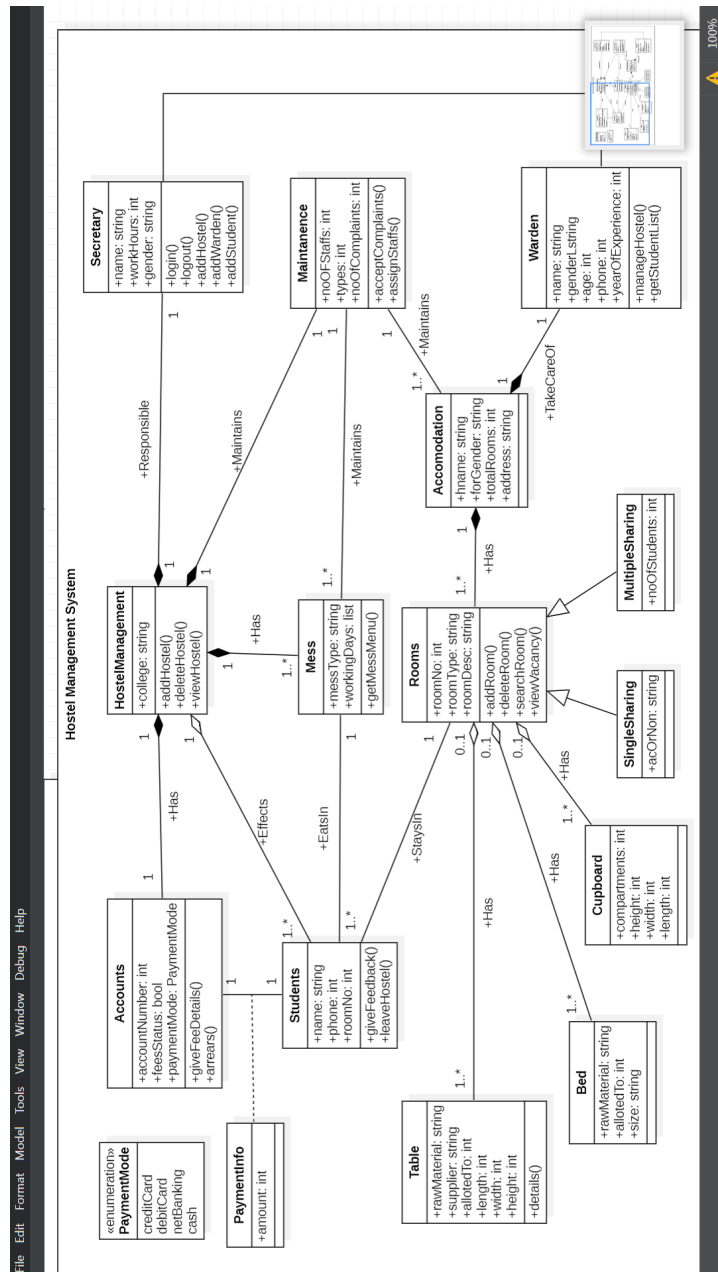
### **3.STOCK MAINTENANCE SYSTEM**

#### **Software Requirement Specification**

- The customer can purchase one or more product on any day,which will have a code price and quantity.
- The customer will need to pay the bill for the products he or she has purchased.the bill number,type description and customer who is paying the bill is maintained.
- The stock of the products is maintained separately,The stock deals with information about the details of the product that the concern handling.
- Stock consist of details such as the name of the product, id generated, quantity, cost, etc. This information is retrieved during the sales and purchase of a product.
- The vendor deals with the information about the details of the suppliers giving product to the organization.
- Vendor consist of details such as vendor name, address, email id, sales tax number etc. This information is retrieved when a Purchase is done
- The products are displayed in stores across the city or world. All the information regarding the store such as store id,name,address and type are used to locate any product.The stores can be of many types.Some of them are departmental stores,super markets and ware houses where the products are kept for display.

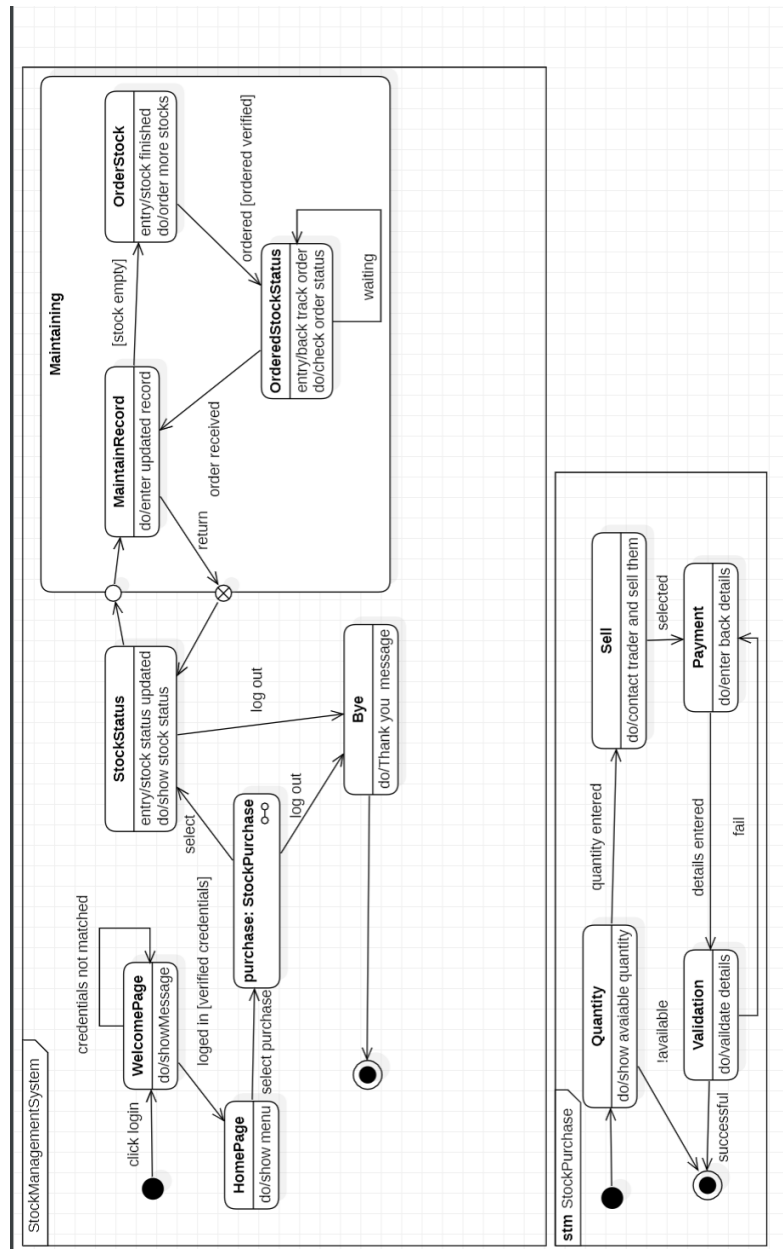


## Advance Class Diagram :



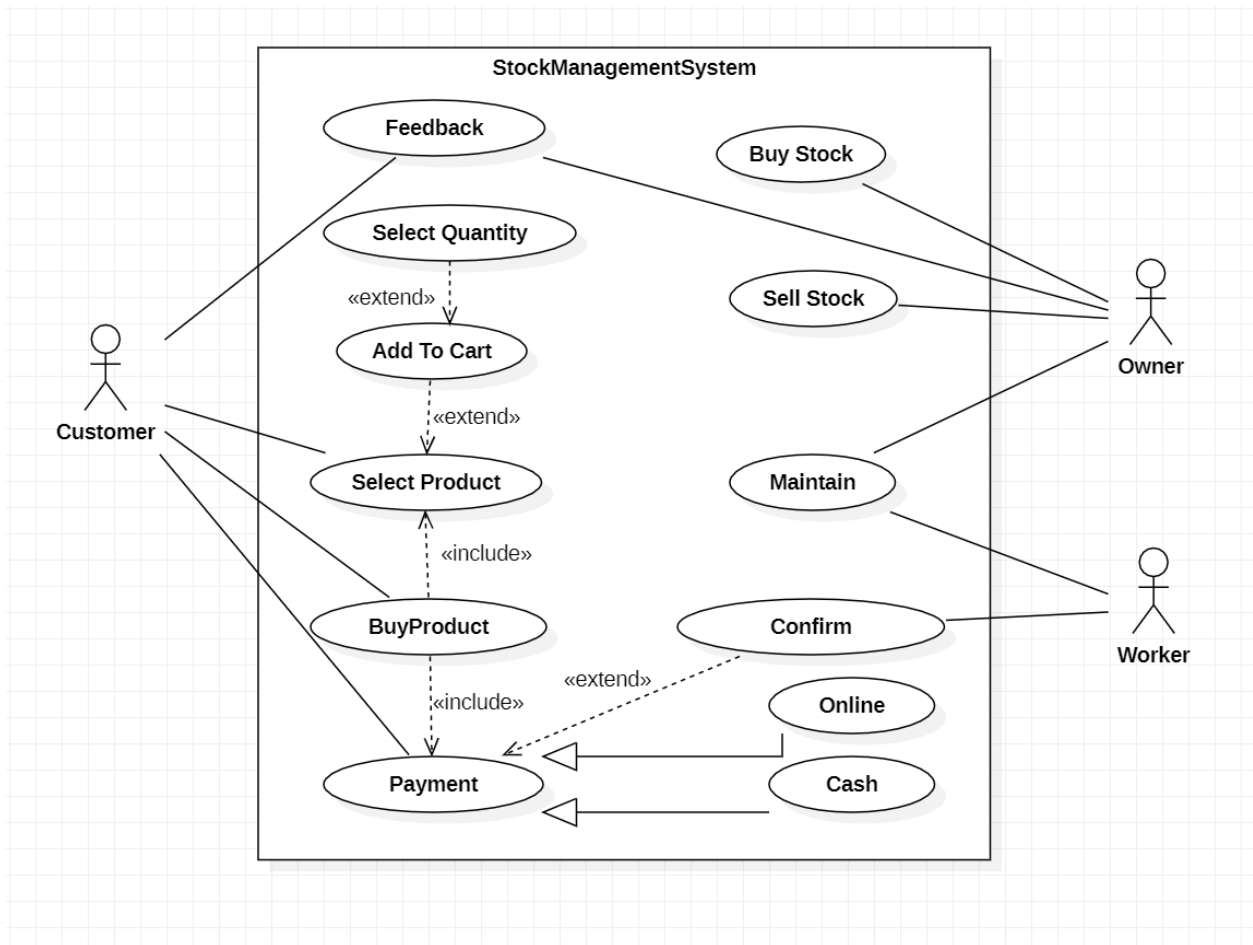
The products are displayed in stores across the city or world. All the information regarding the store are used to locate any product. The stores can be of many types. Some of them are departmental stores, super markets and ware houses where the products are kept for display. The vendor deals with the information about the details of the suppliers giving product to the organization. The stock of the products is maintained separately. The stock deals with information about the details of the product that the concern handling.

## Advance State Diagram :



The state diagram above gives us the states involved in purchasing an product and placing the order for the same. There is first an inventory check ,where is stock of products is noted and if the stock is less than minimum an order is placed by first searching for suitable trader . if a suitable trader is found , the order is placed and verified by the accountant. After the accountant has verified a payment is made for the products purchased.

## Advance Use Case Diagram :



Actors:

Customer: a person who purchases the products

Owner: a person who keeps check of the stock

Worker: a person who dispatches the products

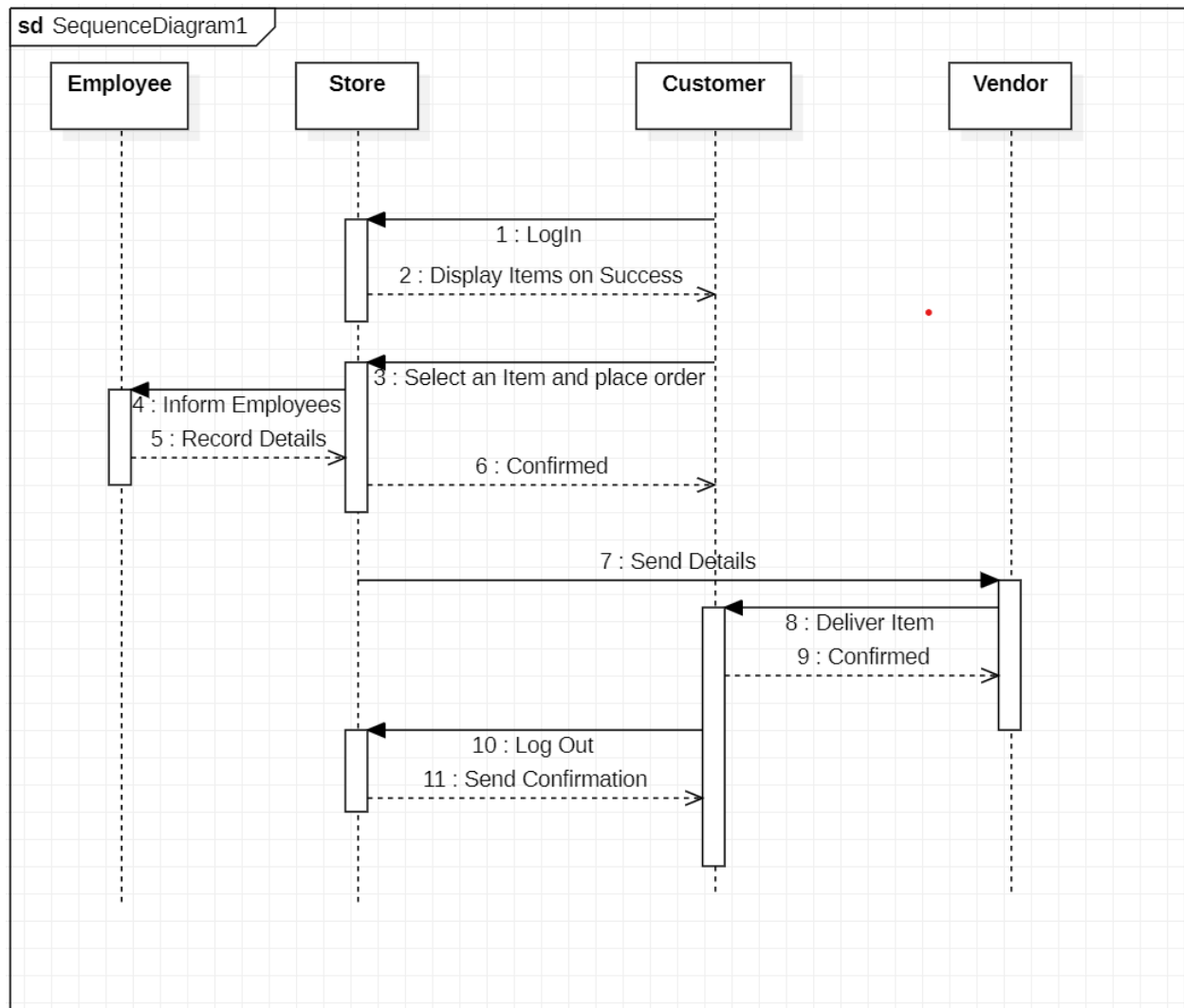
Use Cases:

Purchase item: allows a user to purchase any product Make payment: accepts the payment

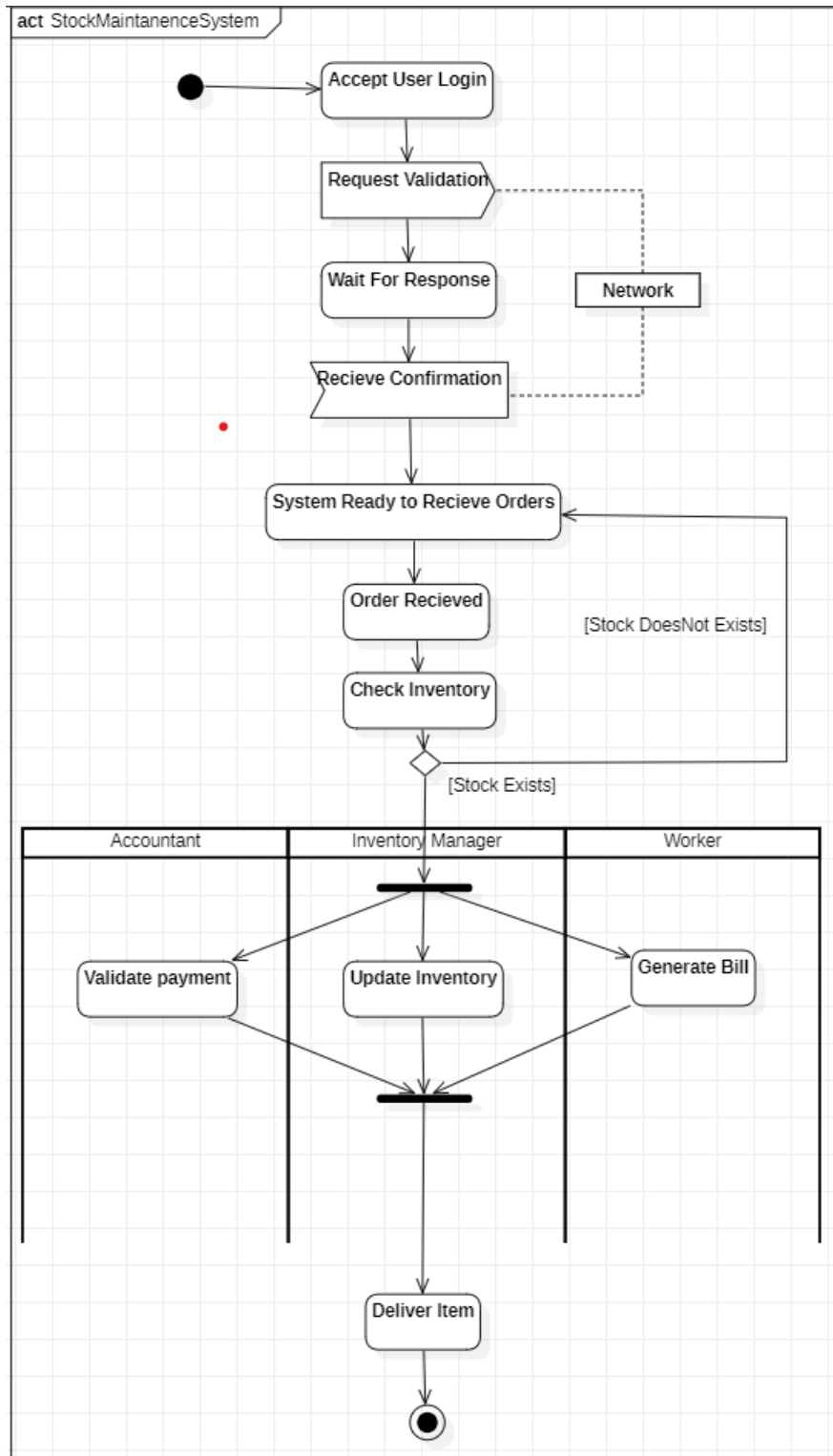
Sell stock: keeps track of the stock supplied Find traders: provides a list of traders

Buy stock: the stock list is updated by the stock person Order goods: the products coming to an end are ordered Prepare bill: a bill for products purchased is made

## Advance Sequence Diagram :



## Advance Activity Diagram:

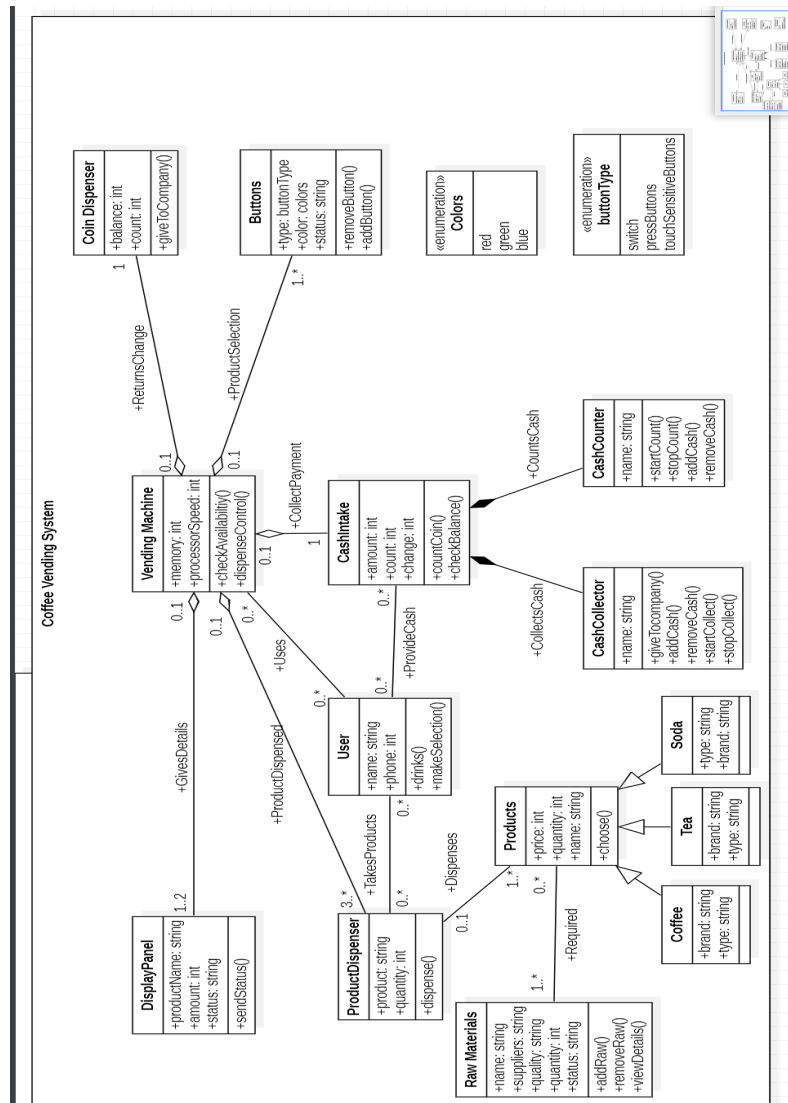


## **4.COFFEE VENDING MACHINE**

### **Software Requirement Specification**

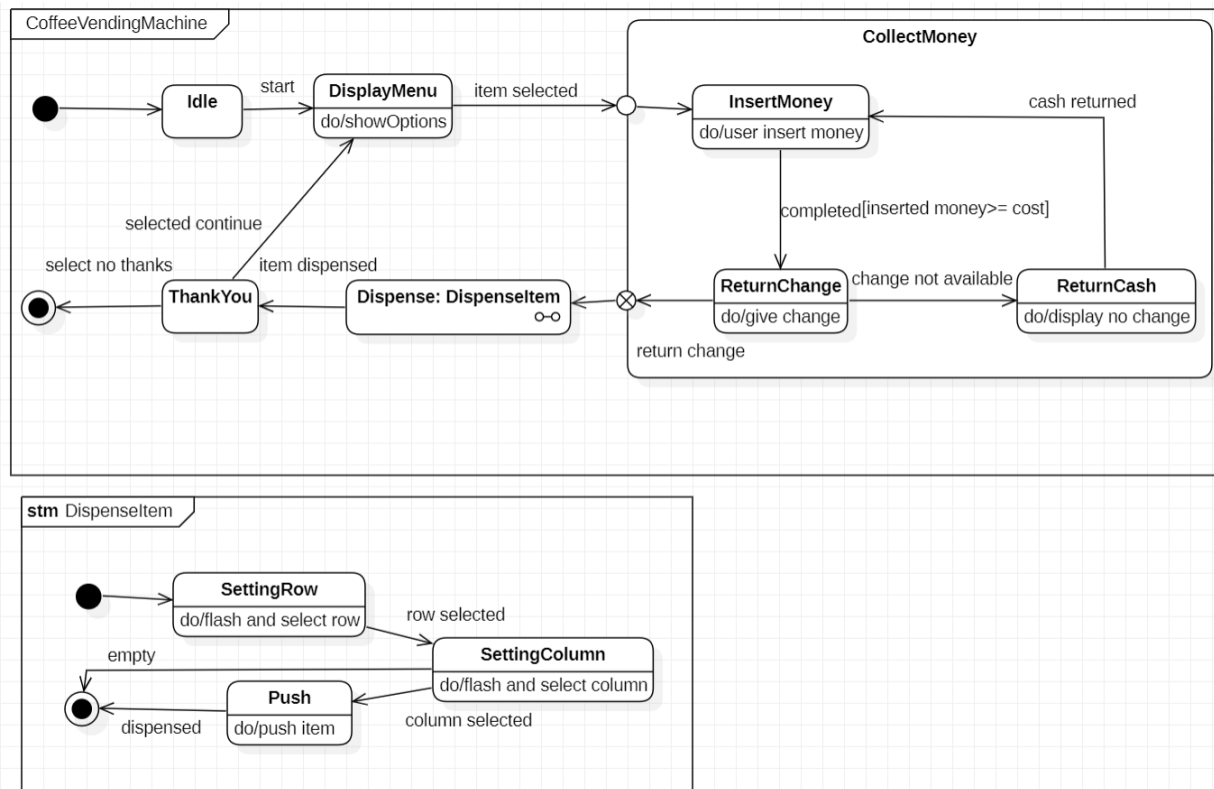
- The vending machine must have money box, coin slot, display screen and products i.e coffee for the machine to be used.
- The user on selecting a coffee ,the coffee machine must be able to dispense the selected coffee to the user.
- The user shall get empty cup placed right below the filter.The user shall be able to choose his preferred beverage from the list of options(buttons).
- There must be buttons(start,pause,stop,coffee,tea,milk) for user to interact with the system.
- The user shall be able to purchase one kind of available drink at a time and get back the exact changes if he has put extra money. The user shall be able to quit the dispense of any beverage at any time during the dispensing.
- The system(machine) shall check for properly inserted coins.
- The system shall be able to dispense coffee(or selected beverage) after a coin has been inserted.
- The system must accept coins of different amount and the system must compare the item cost with entered coin.
- The system must check the validity of coins.
- The system shall be able to detect the low amount of ingredients and low number of cups and indicate with an indicator(small LED).

## Advance Class Diagram :



The vending machine must have money box, coin slot, display screen and products i.e coffee for the machine to be used. The user on selecting a coffee, the coffee machine must be able to dispense the selected coffee to the user. The user shall get empty cup placed right below the filter. The user shall be able to choose his preferred beverage from the list of options. There are different types of coffee such as cappuccino, black coffee, cold coffee and latte. Each type of coffee has a price and a name. A customer can buy their choice of coffee by selecting the button of their coffee and paying for the same through the coin box.

## Advance State Diagram :

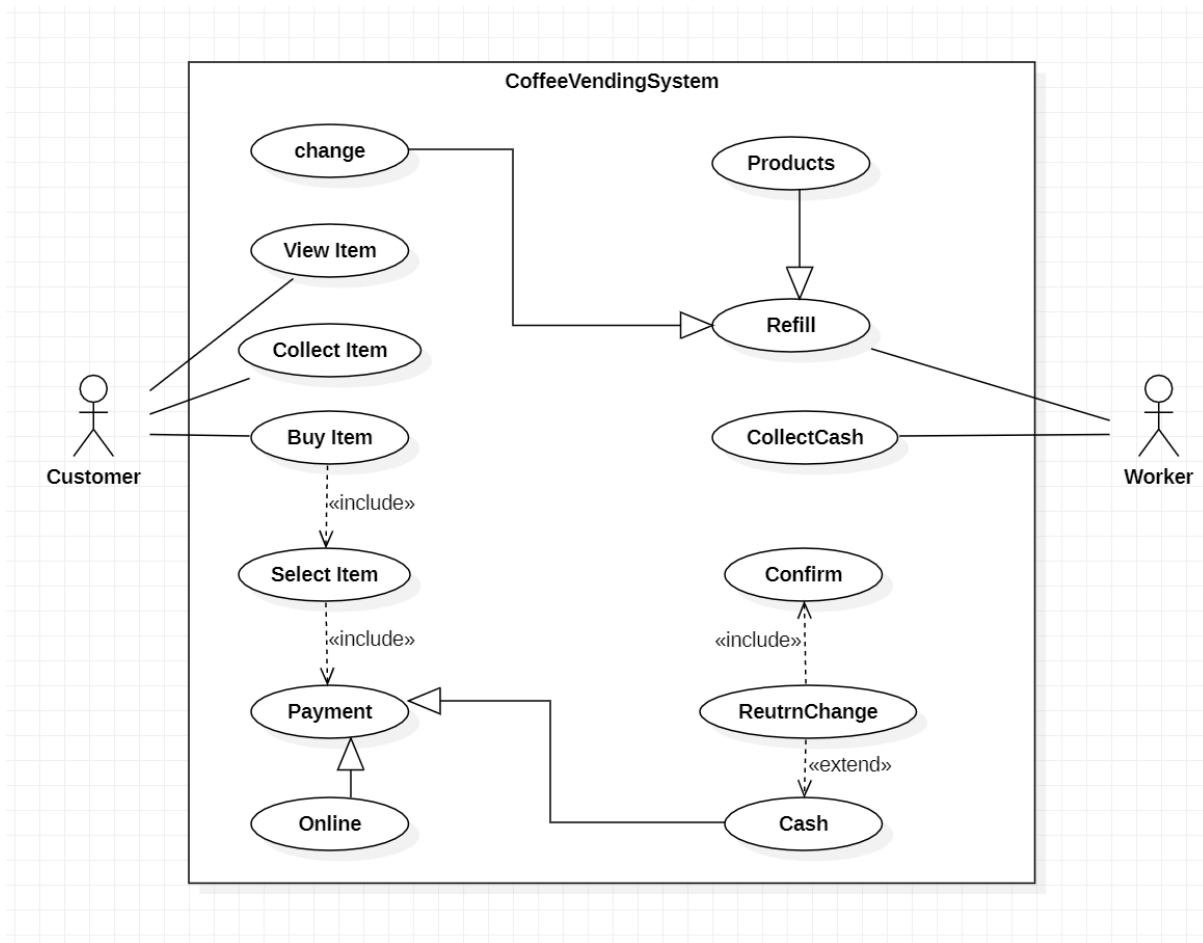


Initially the vending machine is in the waiting state. The machine displays the selected item selected by the user. When the person inserts a coin the machine adds the amount to he cumulative balance. After adding some coins, a person can select nay item. If item is empty or balance is insufficient,the machine waits for another selection. Otherwise the machine dispense the item and returns the appropriate change.

The state diagram for coffee vending machine has a submachine called dispense Item ,which has the states for dispensing an item from the vending machine.the arm of the machine first moves to an appropriate row,when ready,moves to an appropriate column and when the arm is ready it finally dispences the item from the machine.



## Advance Use Case Diagram :



Actors:

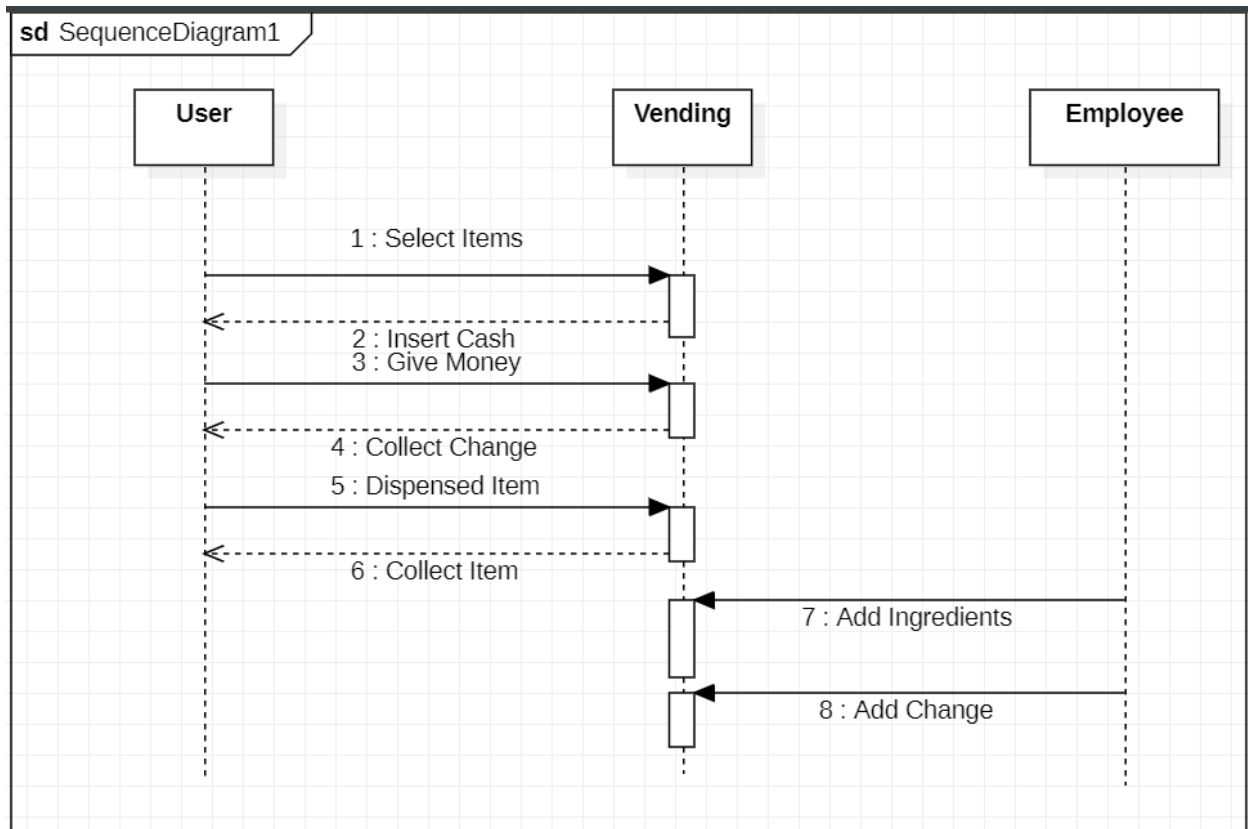
Customer : a person who uses the coffee vending machine  
 Stock clerk : a person who maintains the stock

Coffee vending : a person who maintains the machine

Use Case: Display payment details : displays the payment details  
 Request coffee : allows user to order their coffee  
 Make payment : accepts money for the coffee

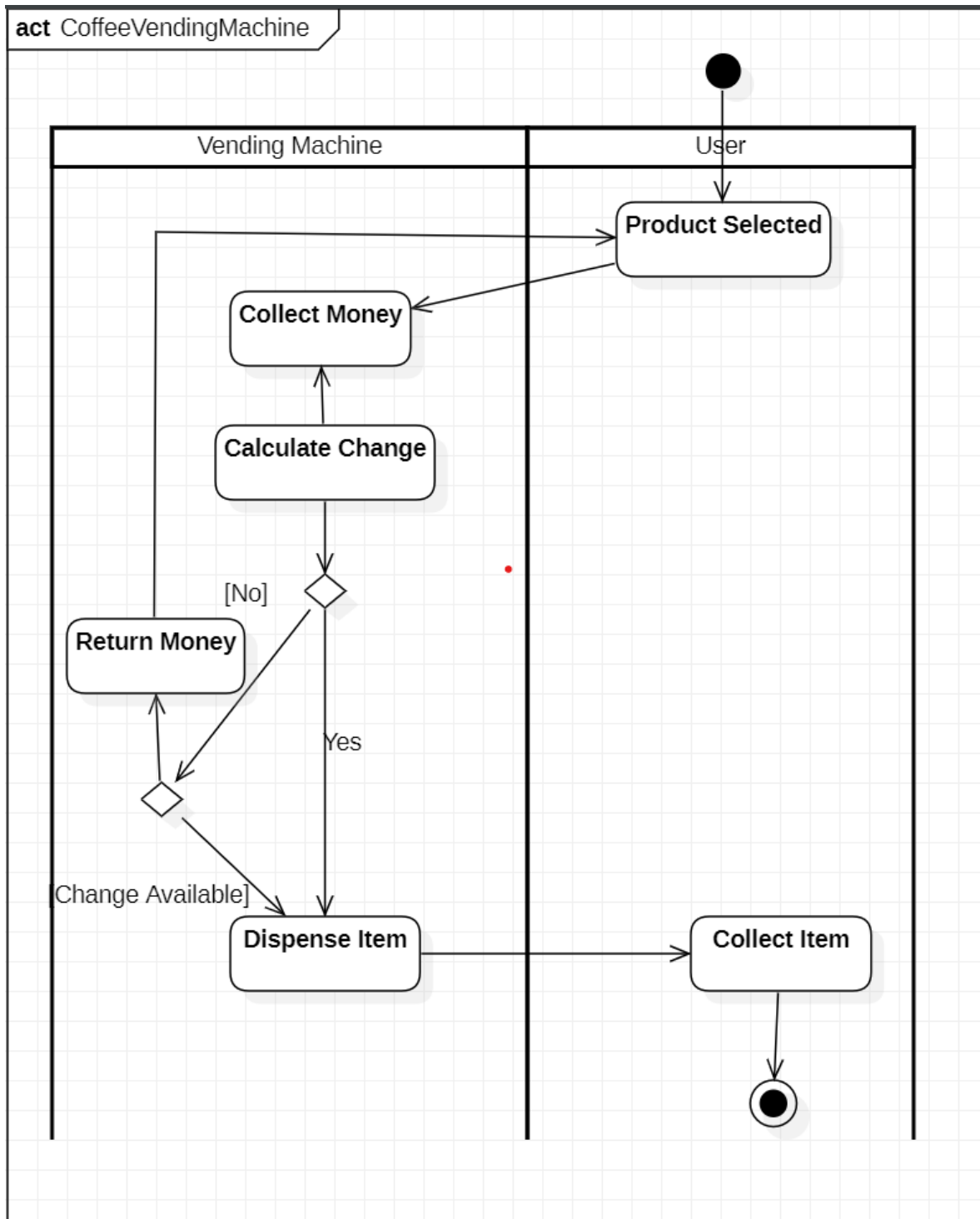
Refill: is the use case where the operator fills the machine with ingredients.

## Advance Sequence Diagram :



The above sequence diagram gives us the steps involved in dispensing a product from the coffee machine. First an enquiry for the product is made and if available the coins are inserted and calculated ,if correct the product is dispensed .

## Advance Activity Diagram:



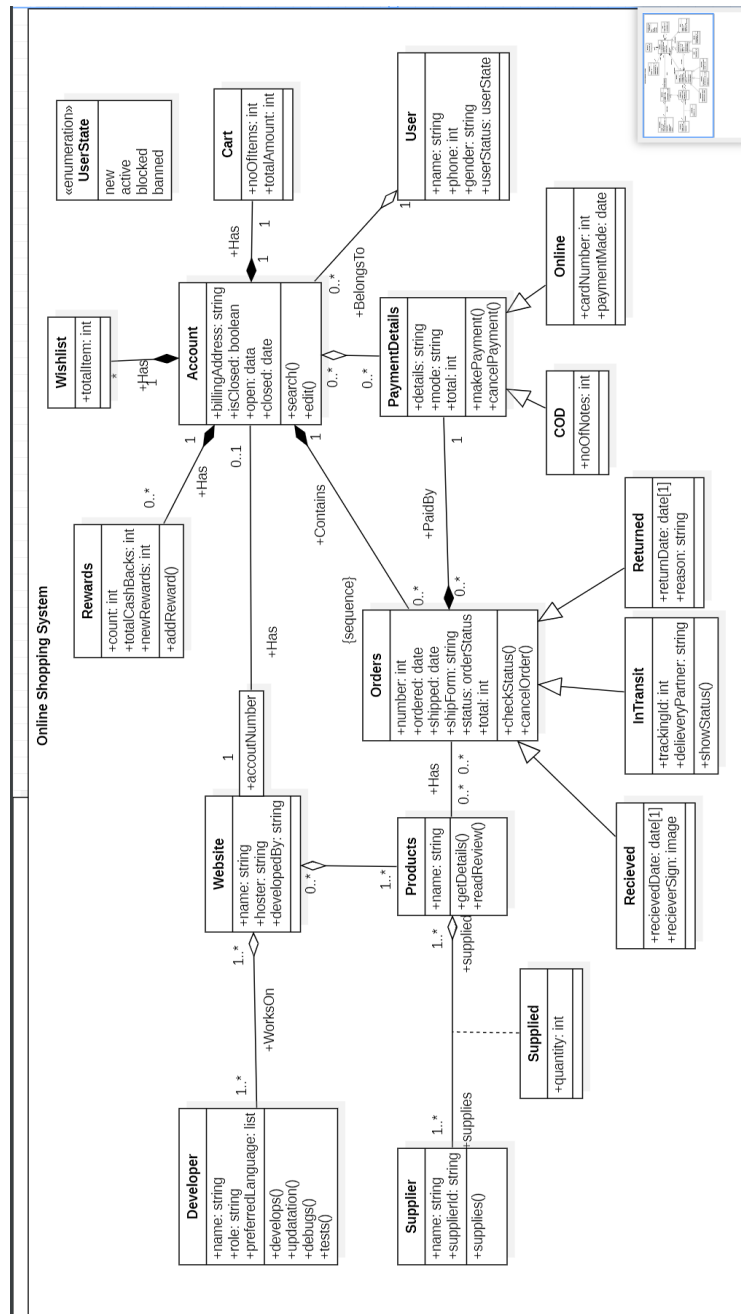
The activity diagram above receives the order and forks into two activities i.e make coffee and display order . on choosing an option if the user wants a receipt it gets printed otherwise no and order ends.

## **5.ONLINE SHOPPING SYSTEM**

### **Software Requirement Specification**

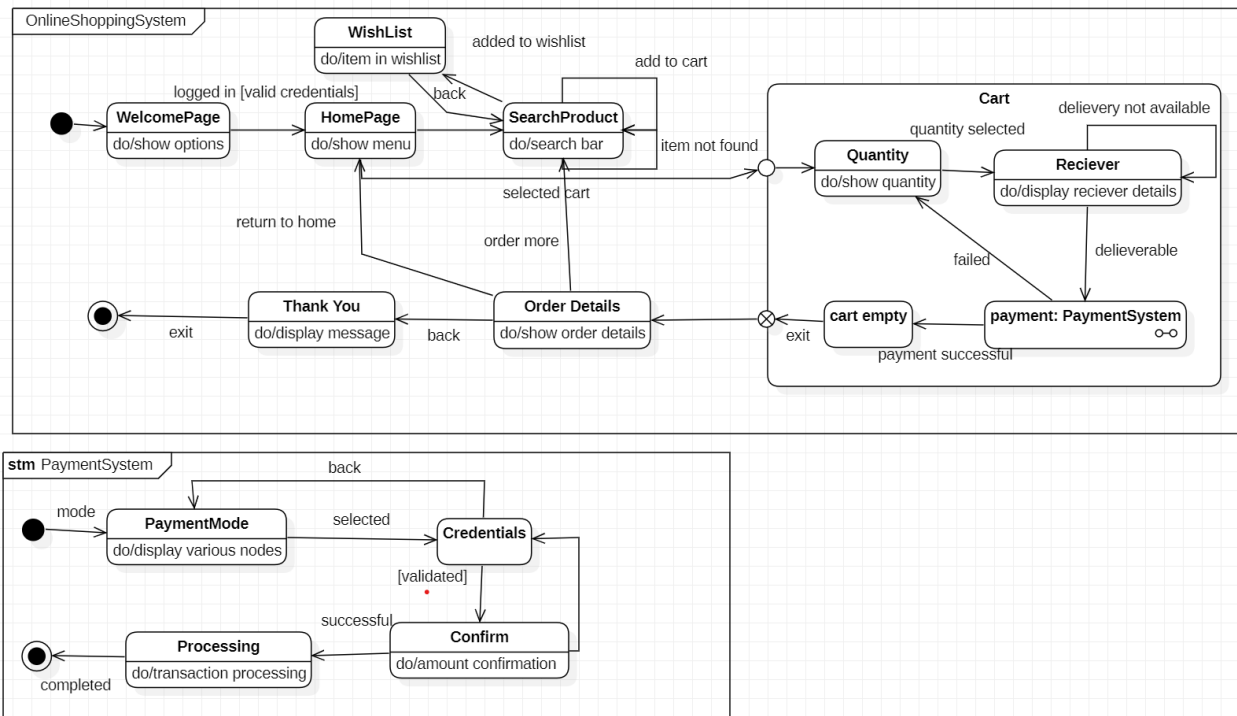
- The customer must have an account in the online website where he/she can purchase products.
- If customer wants to buy the product then he/she must be registered, unregistered user can't go to the shopping cart.
- Customer login to the system by entering valid user id and password for the shopping.
- Changes to cart means the customer after login or registration can make order or cancel order of the product from the shopping cart.
- The products sold for customers are sold for various categories like men,women,kids and home products.
- Customers can view all available products ,compare them and make a choice for purchasing the products.
- For customer there are many type of secure billing will be prepaid as debit or credit card, post paid as after shipping, check or bank draft. The security will provide by the third party like Pay-Pal etc.
- After the payment or surf the product the customer will logged out.

## Advance Class Diagram :



The online shopping system has customers who must have an account in the online website where he/she can purchase products. If customer wants to buy the product then he/she must be registered, unregistered user can't go to the shopping cart. Customer login to the system by entering valid user id and password for the shopping. The products sold for customers are sold for various categories like men, women, kids and home products. After the payment or surf the product the customer will logged out.

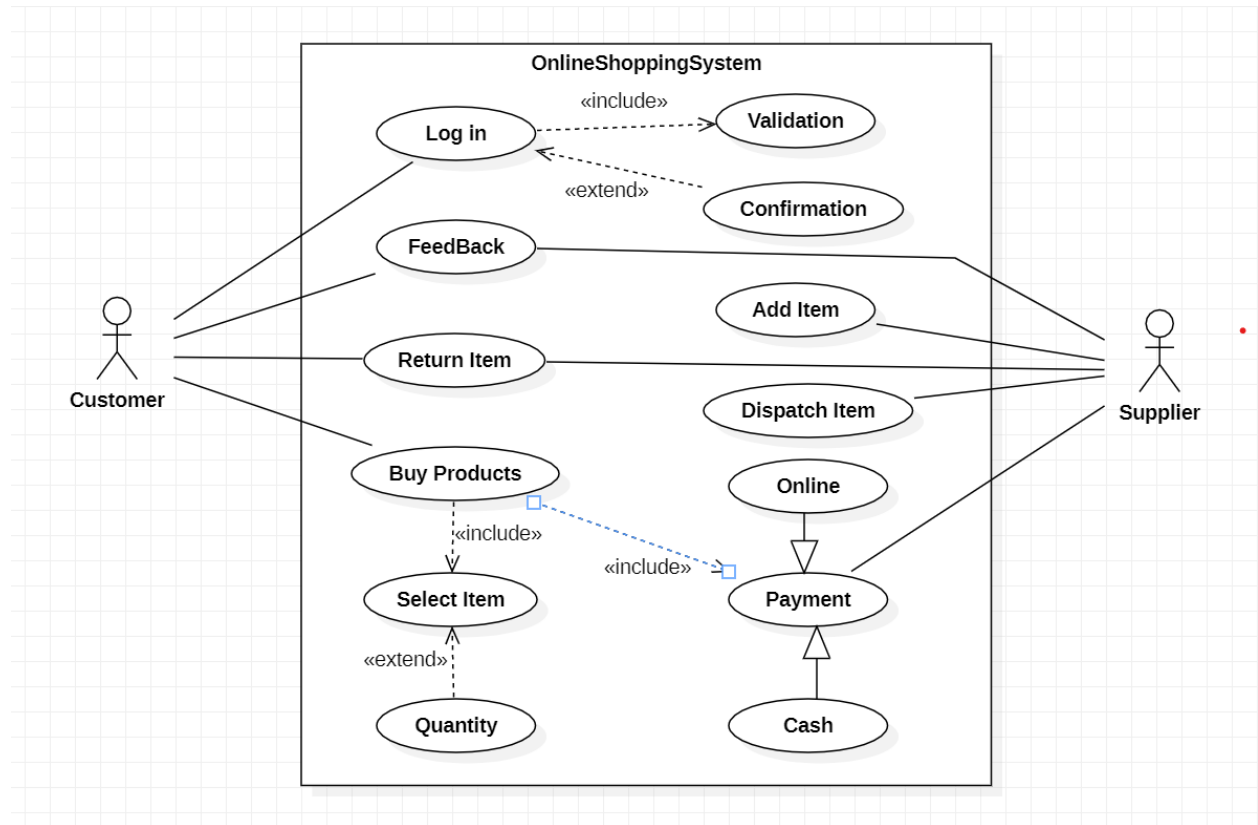
## Advance State Diagram :



The simple state chart diagram gives us states in purchasing a order and paying for the order. The customer is first mad to register and then login into their account. Then the items are displayed,where they can select their choice and add them to cart or reserve or order them. The transaction details are displayed.

The advanced state chart diagram has states explaining the product purchase and payment. It has two sub machines i.e product selection and checkout product. Product selection allows us to select products and add them to cart. Checkout product has states explaining the payment methods and validating the methods.

## Advance Use Case Diagram :



### Actors:

Customer: a person who uses the online shopping system

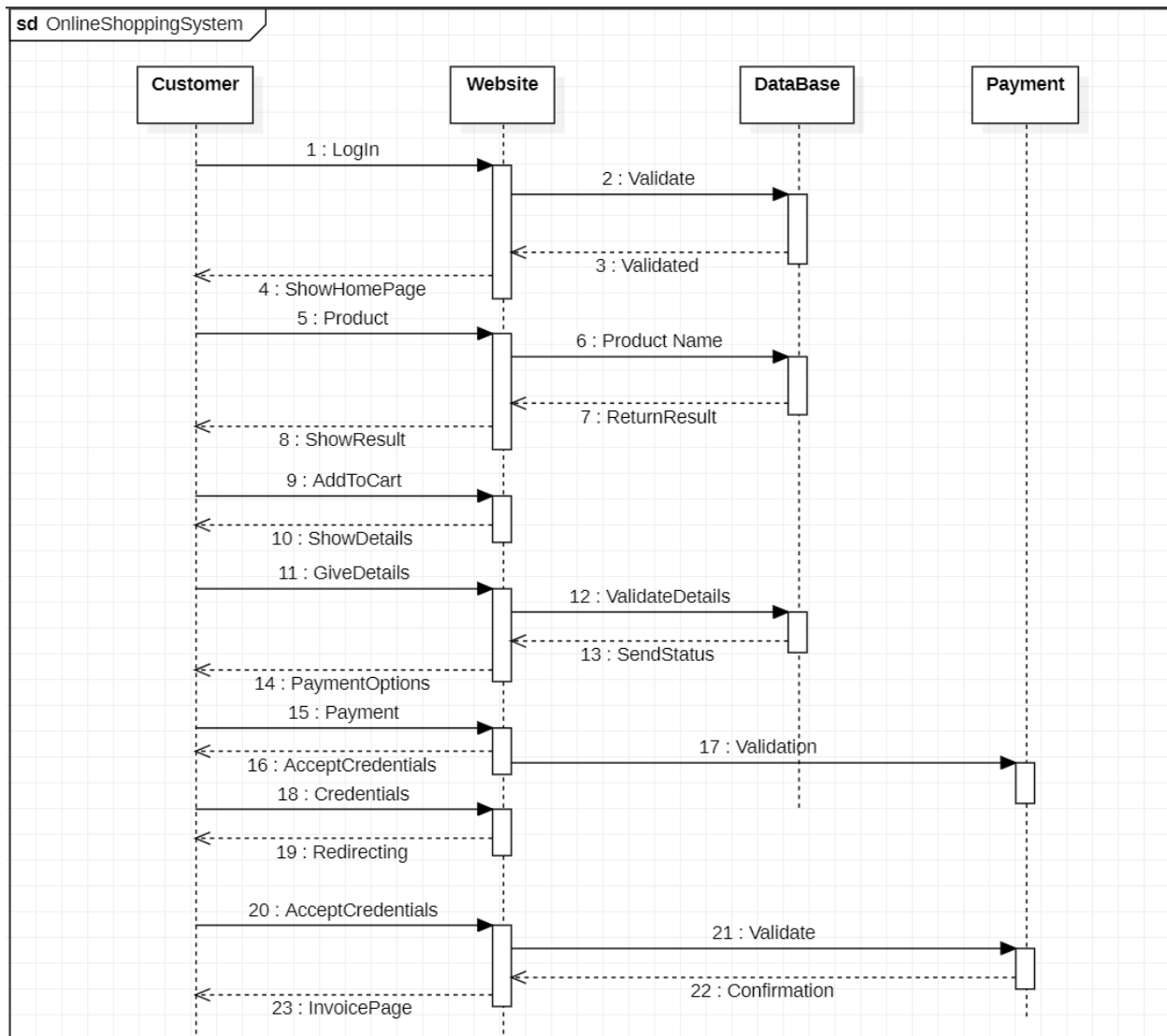
Supplier : person who supplies products

### Use Case:

view product details : displays all product details

Place order : order the items present in the car.

## Advance Sequence Diagram :



The customer logs into the online interface The items are displayed

The customer adds items into cart and reply from interface is sent The customer places the order

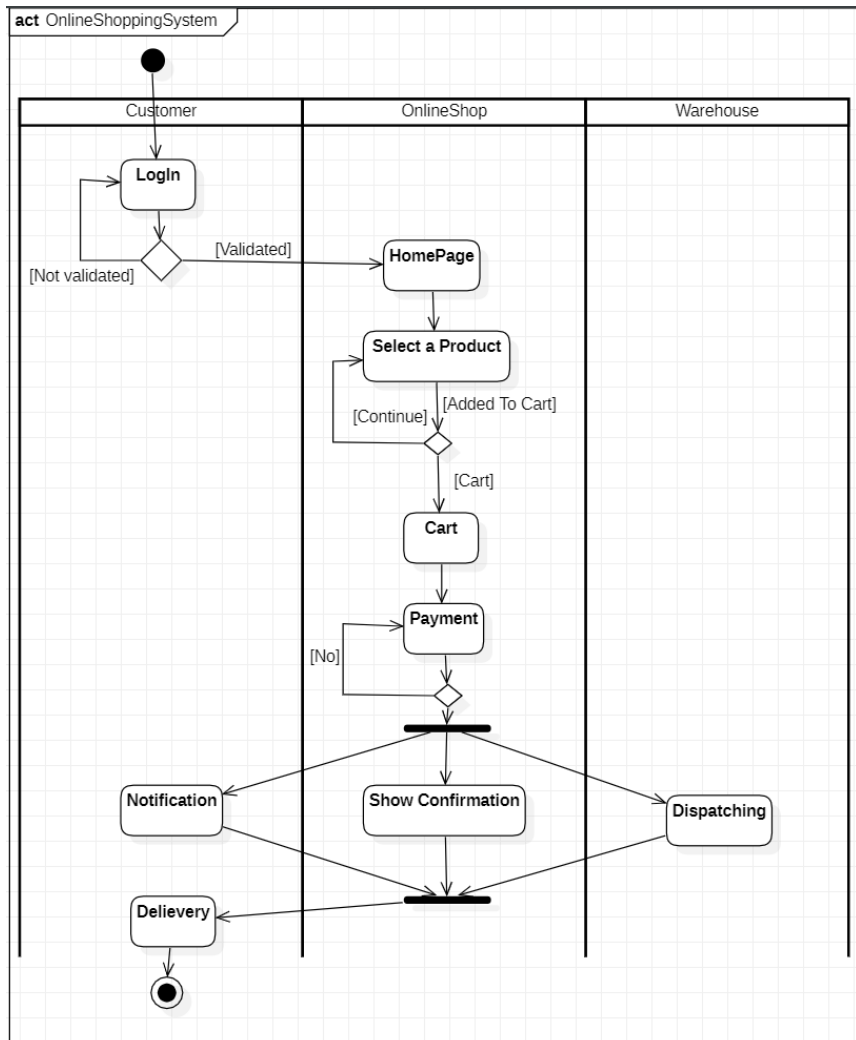
The online interface requests for payment

The customer provides details and confirmation is sent The customer logs out

The logout confirmation is sent to the customer



## Advance Activity Diagram:



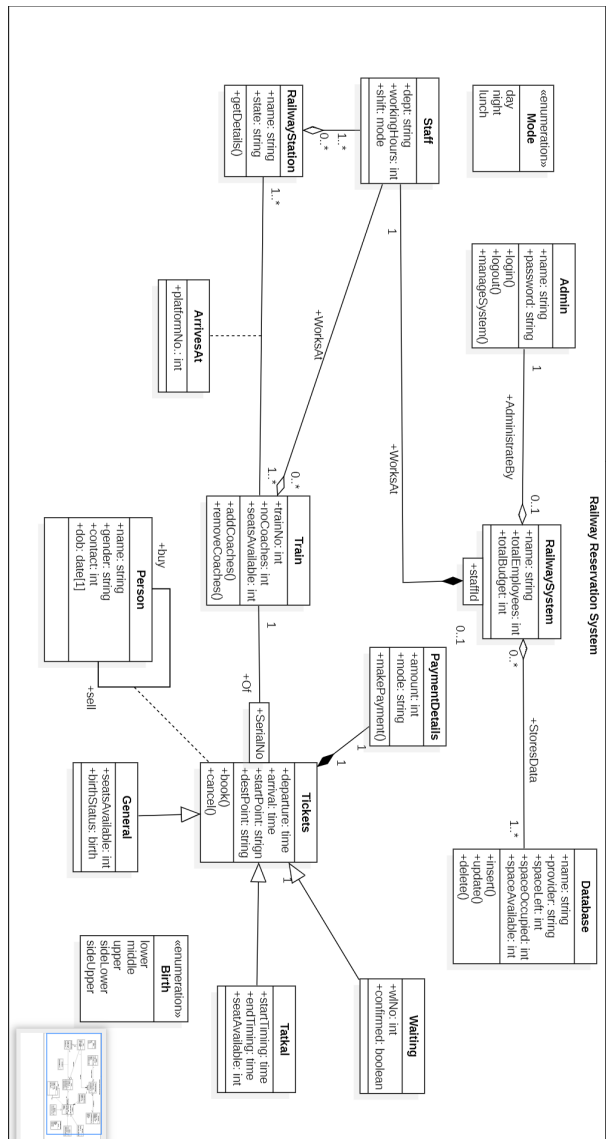
The advanced activity diagram has two swim lanes i.e customer and online shop. The customer can add product to basket and login/register and confirm order. The online shop can checkout the products, deliver, process payment and send confirmation to customer.

## **6.RAILWAY RESERVATION SYSTEM**

### **Software Requirement Specification**

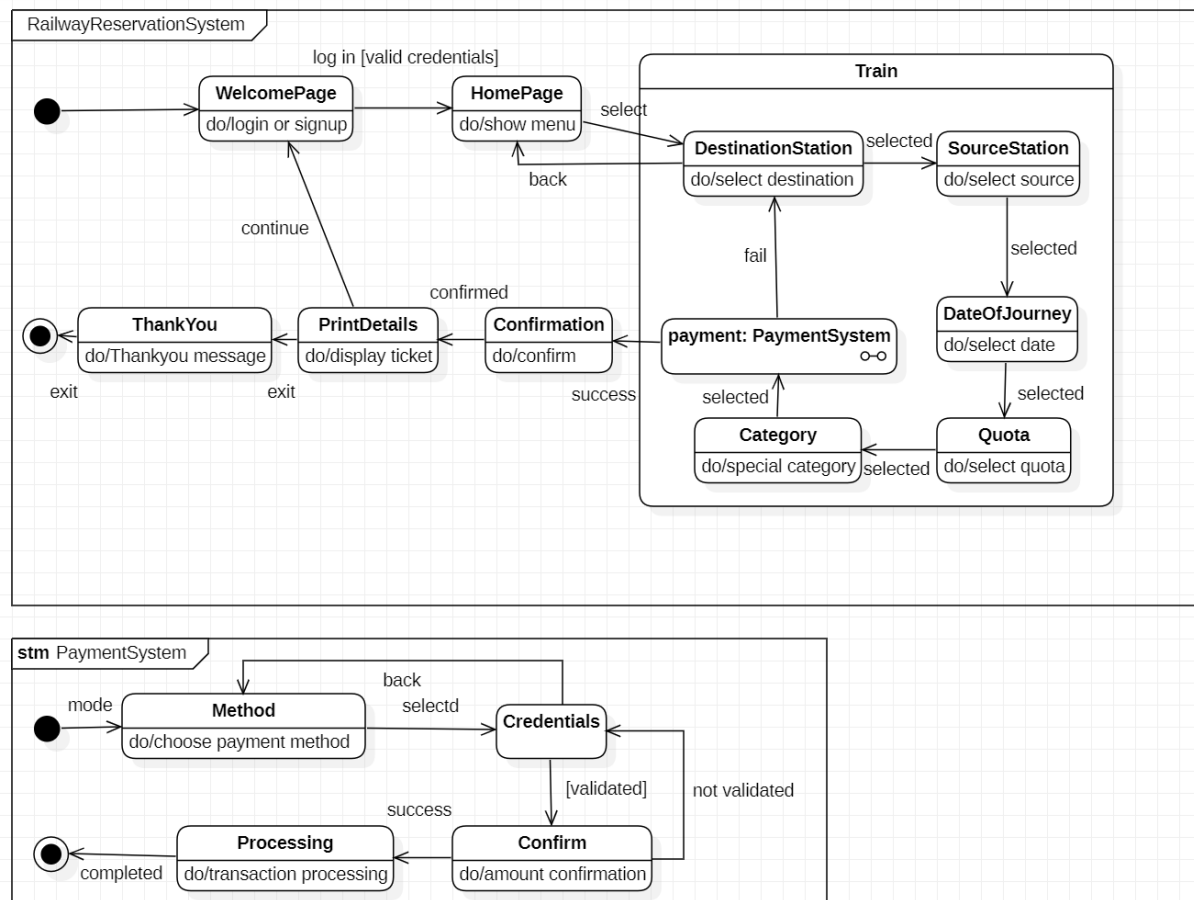
- Each user should have a user id and a password.Record of the users of the system should be kept in the log file. Provision should be made for full backup of the system.
- The customers can view the trains available at any day,the cost and number of tickets available for any train.
- Customer can book a ticket only if the tickets are available.Customer searches for the availability of tickets then if the tickets are available he books the tickets by initially filling details in a form.
- Tickets can be booked in two ways by i-ticket or by e-ticket booking.
- In case of i-ticket booking customer can book the tickets online and the tickets are couriered to Particular customer at their address.But in case of e-ticket booking and canceling tickets are booked and canceled online sitting at the home and customer himself has to take print of the ticket but in both the cases amount for tickets are deducted from customers account.
- For cancellation of ticket the customer has to go at reservation office than fill cancellation form and ask the clerk to cancel the ticket than the refund is transferred to customer account.
- After booking ticket the customer has to checkout by paying fare amount to clerk.
- The system displays the details of train of which user enter the name.The information is saved and the corresponding updating take place in the database.

## Advance Class Diagram :



The admin manages the trains and reservation related to railway reservation system. There are three types of reservation, I.e RAC, waiting and confirmed. The passengers with a reservation goes to one or the other reservation. A train consists of coaches and engine. A passenger pays for the ticket booked. Tickets can be booked in two ways by i-ticket or by e-ticket booking.

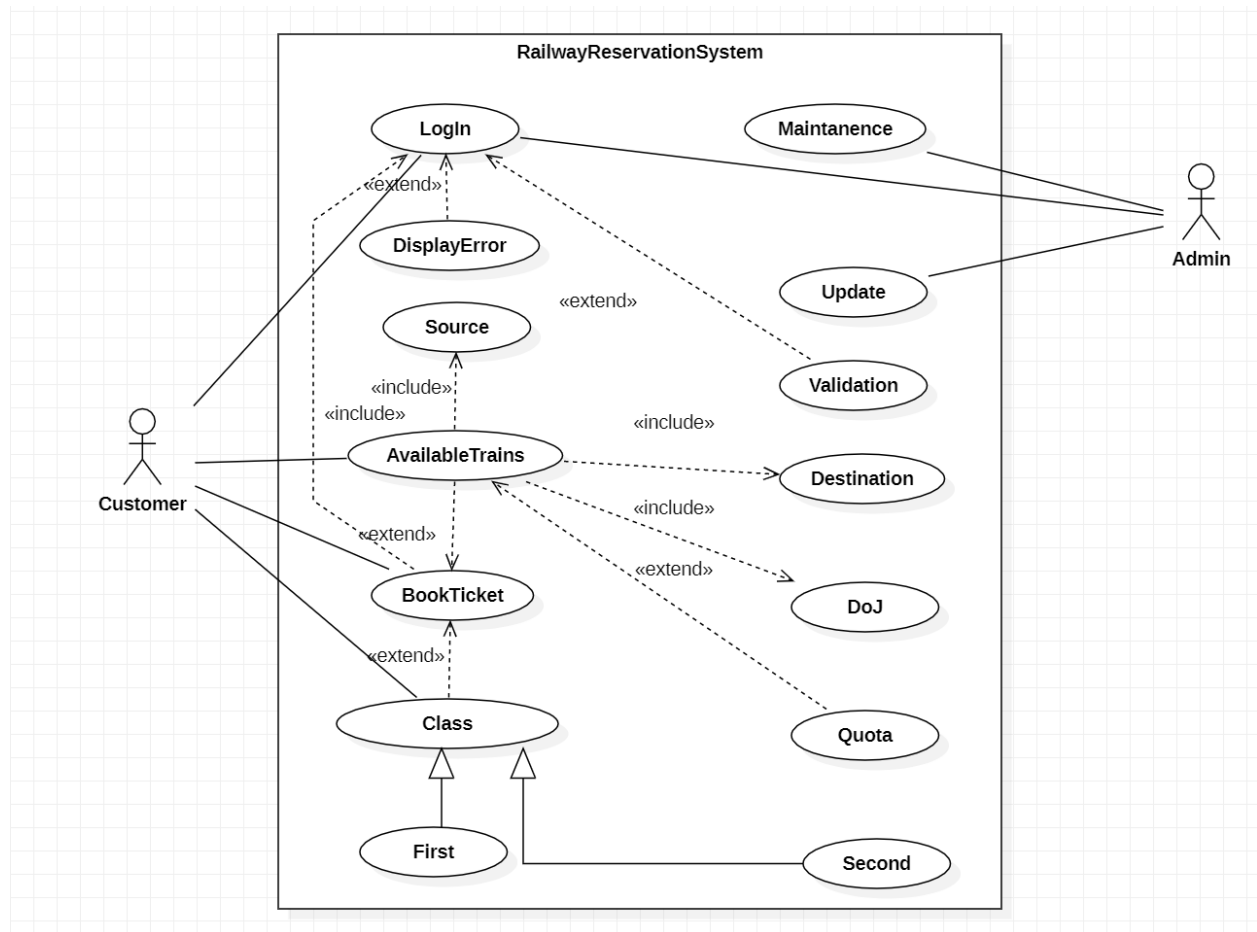
## Advance State Diagram :



The simple state diagram gives the states involved in booking a train ticket and paying for the same. The user can see the train details and book a train for a particular source and destination . on timeout an error message is displayed and redirected to the main page. The user can then select a train and make payment for it

The advanced state diagram has states for paying the ticket. from the ready state the user goes to payment initiation after which the card details are accepted and an OTP is sent to the registered mobile number. On verification the money is deducted and ticket is sent to the customer.

## Advance Use Case Diagram :



Actors:

Customer: uses the railway reservation system. Admin: manages all information

Admin: System that is used for train ticket reservation.

Use Case:

Register: The first time user has to create a account in railway system.

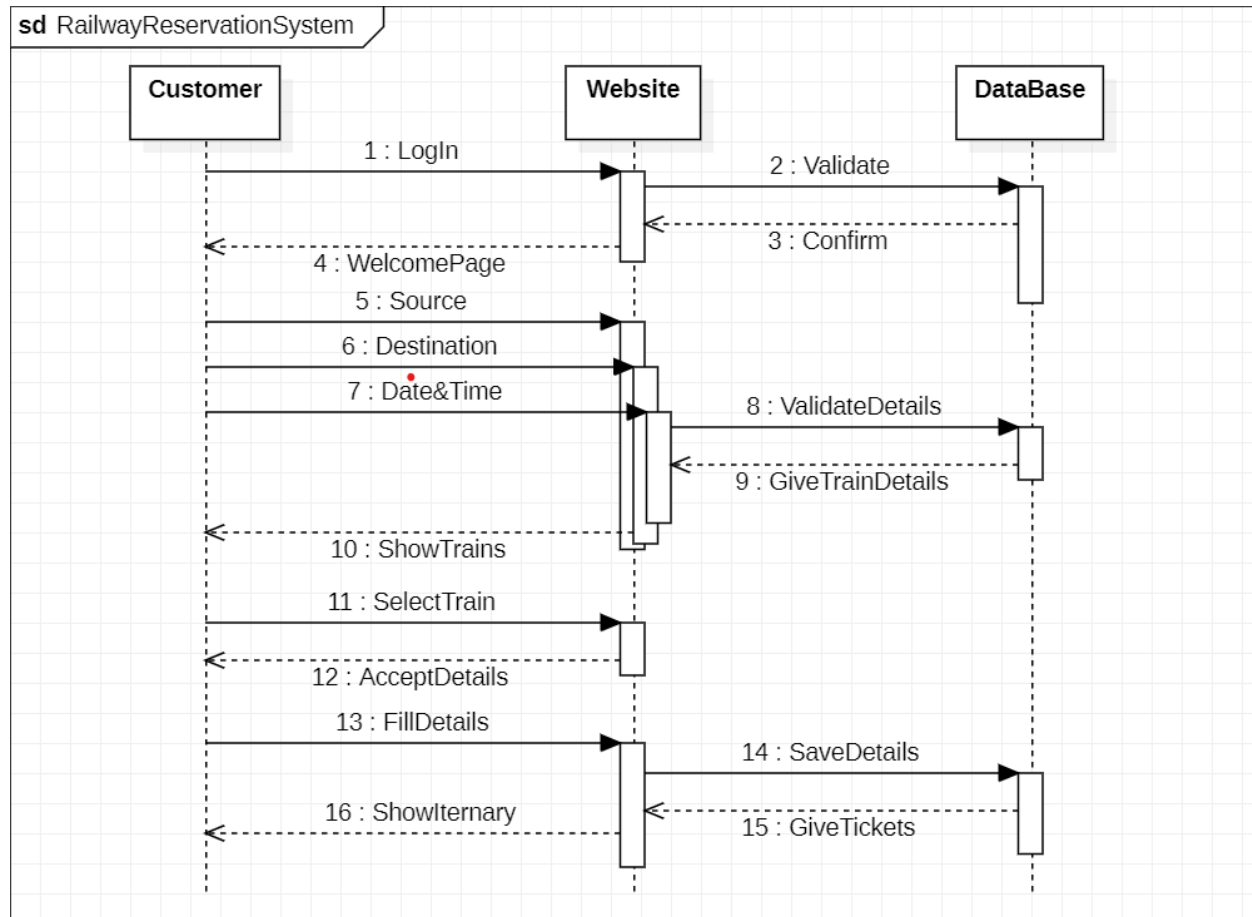
Book Ticket: User can select the type of coach and no of seats and book the ticket.

Make payment: System displays the payment details. User can make his payment.

Cancel Ticket: User can cancel the ticket . The amount will be refunded.

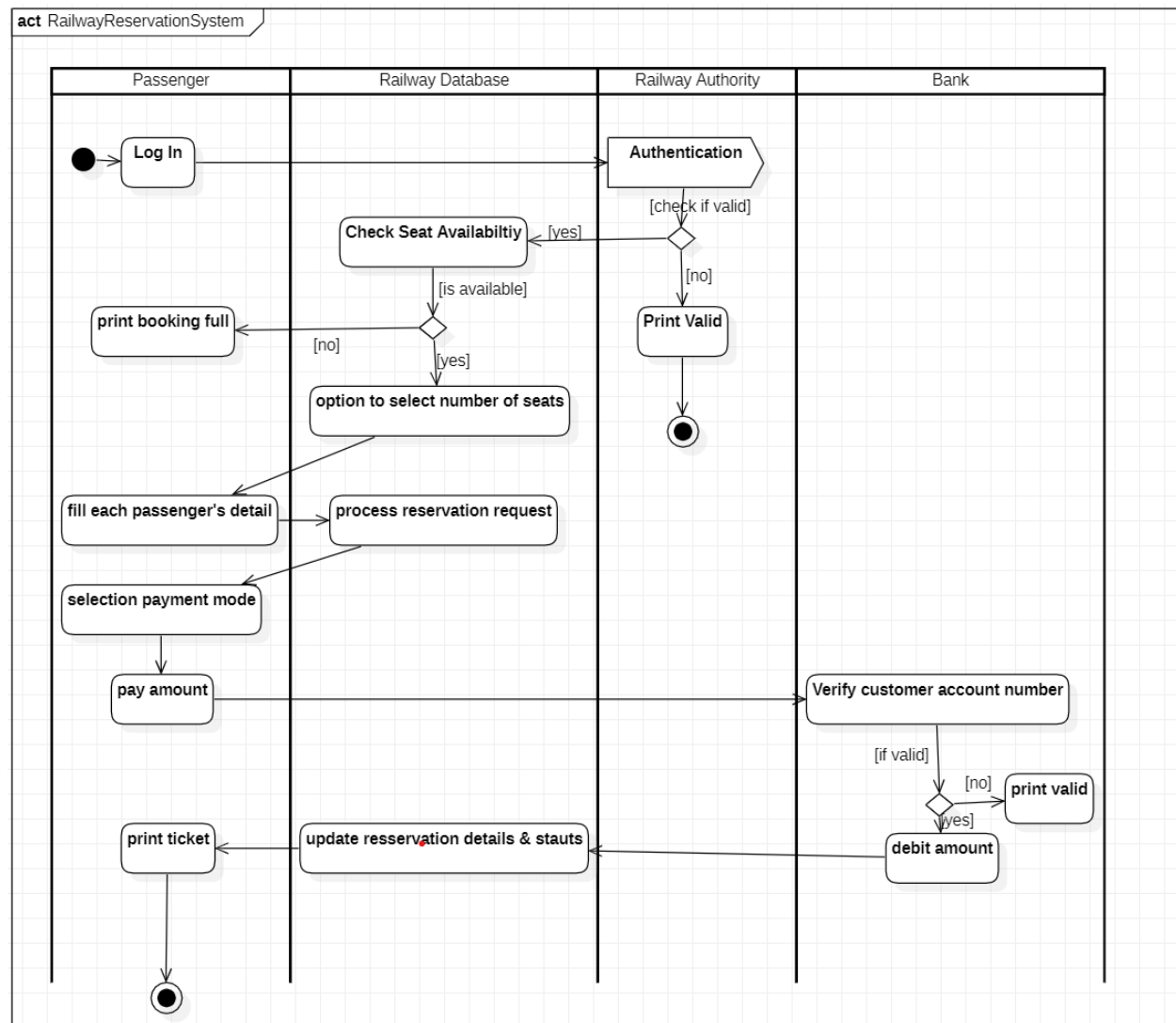
Verify login credentials: The admin verifies the user details, if it matches with the details in database then he allows access to the system.

## Advance Sequence Diagram :



User logs into the railway reservation system. Admin verifies the login details. System establishes secure communication. User checks for availability of trains. Admin updates the train details. System displays the train details. User books tickets. System displays payment details. User makes the payment. System issues the e-ticket. User logs out.

## Advance Activity Diagram:



The activity diagram tells about the steps happening while canceling a ticket which is booked. The user first needs to login and select his ticket, confirm cancellation, request refund and print the canceled ticket and logout.

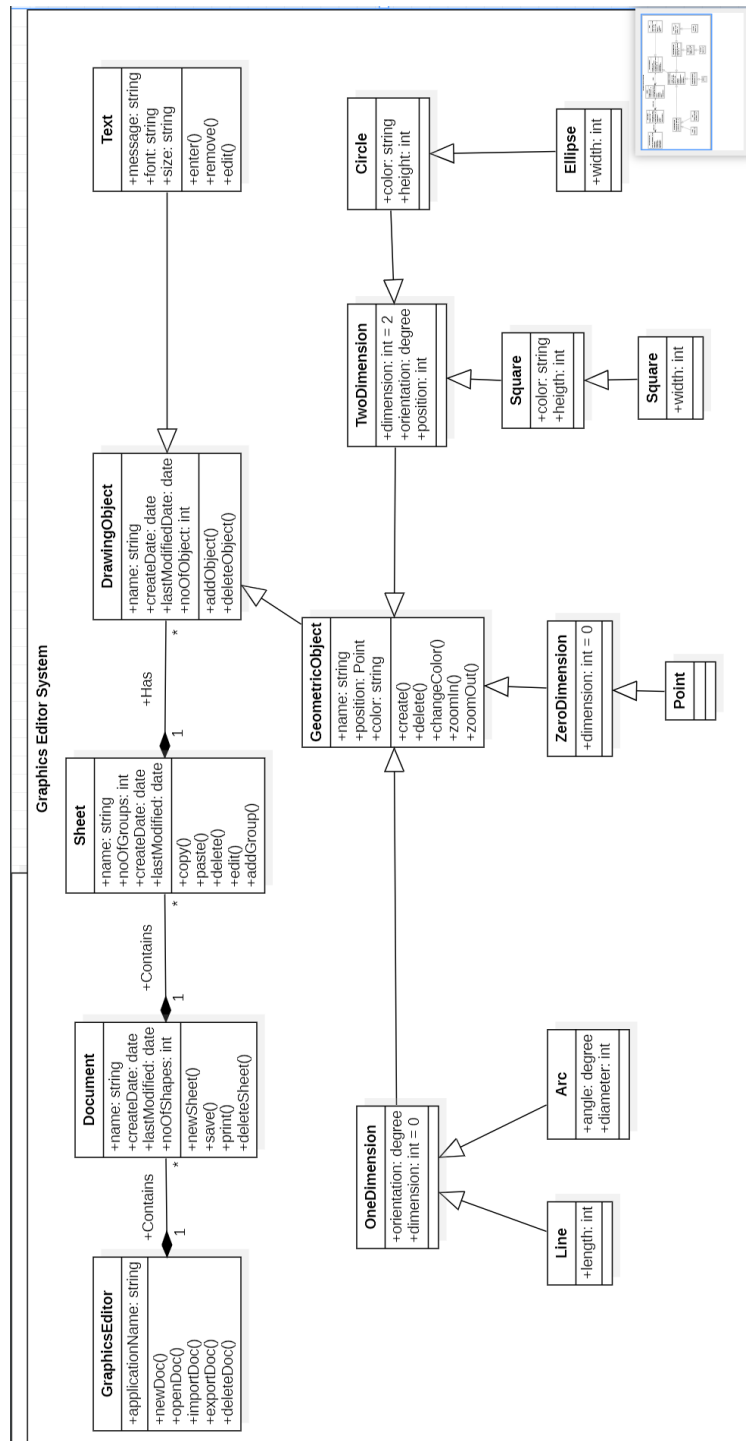
## **7.GRAPHICS EDITOR**

### **Software Requirement Specification**

- The graphical editor consists of a graphical document editor which can be used to create new document,delete document,update or view the document.
- The graphical document editor consists of many documents,where each document can be saved,opened,printed or create a new one
- A document is made up of many sheets which can have graphics included in them.
- Sheets have multiple number of drawing objects,which can be created, grouped or formatted.
- The programmer must provide implementations of functions that draw objects and their connections, as well as functions that add and remove connections. The latter function will be handled by a specific event listener. Any changes made in real-time to the underlying model will also be updated in the diagram through a separate event listener
- The user can also add and remove connections between these objects as needed using the palette supplied, thus modifying the underlying model.
- Each sheet contains drawing objects, including text, geometrical objects and groups.A group is simply a set of drawing objects.
- A geometrical object includes circle, ellipse, rectangles, lines and squares,trapeziums which are identified by their respective constraints.

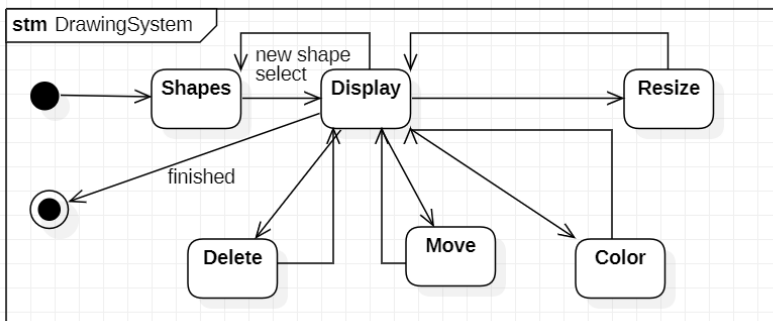
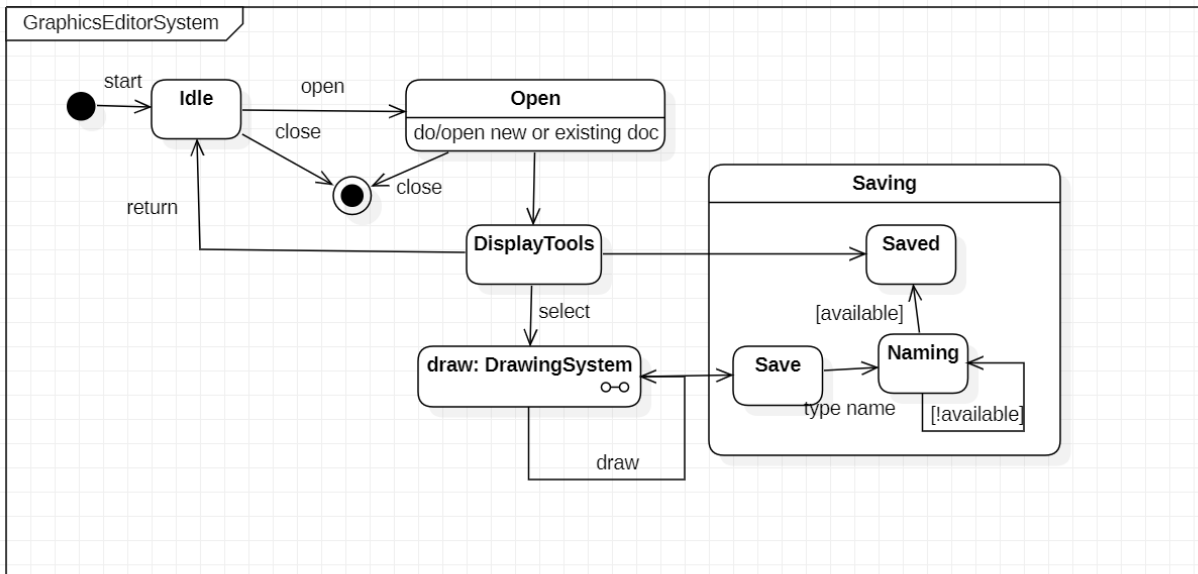


## Advance Class Diagram :



The graphical editor has documents consisting of several sheets. Each sheet contains drawing objects, including text, geometrical objects and groups. A group is simply a set of drawing objects. A geometrical object includes circle, ellipse, rectangles, lines and squares.

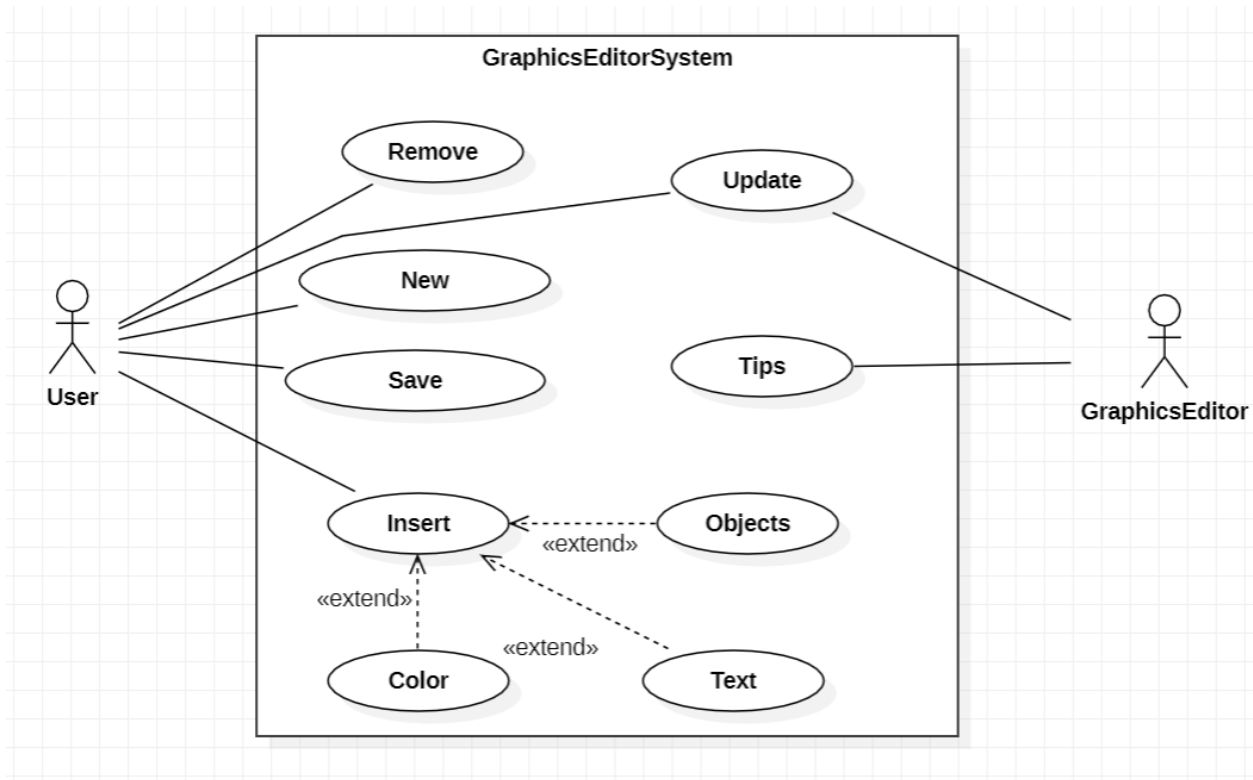
## Advance State Diagram :



The simple state diagram and advanced state diagram gives the states involved in making and saving a graphic file. first the user selects a new document and draws graphics. If there is a mistake he can erase and select a color from the color palet. He can then save the file created.

The advanced state diagram had a composite state called saving where the user can save the file in their desired location.

## Advance Use Case Diagram :



### Actors :

User: the person who uses the graphic system

Graphics system: manages the system

### Use case:

Create document : performs creation of new document

Edit document: performs editing of document

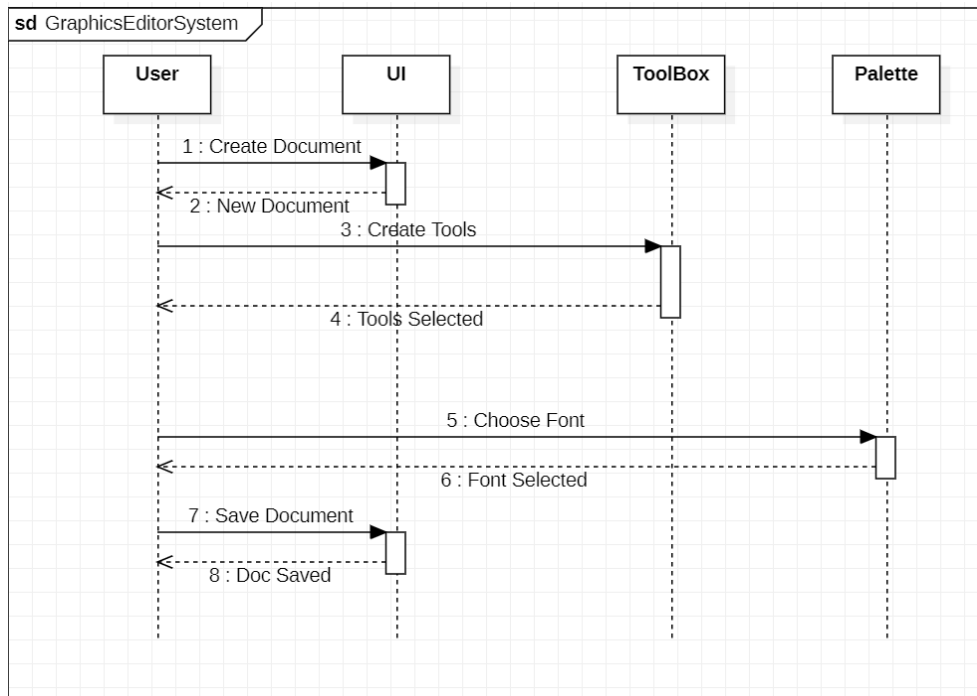
Display toolbox: displays the available tools

Add graphic object : insert a new graphic object

Choose tools from toolbox : allows user to choose tools

Delete document : Permanently deletes the document

## Advance Sequence Diagram :



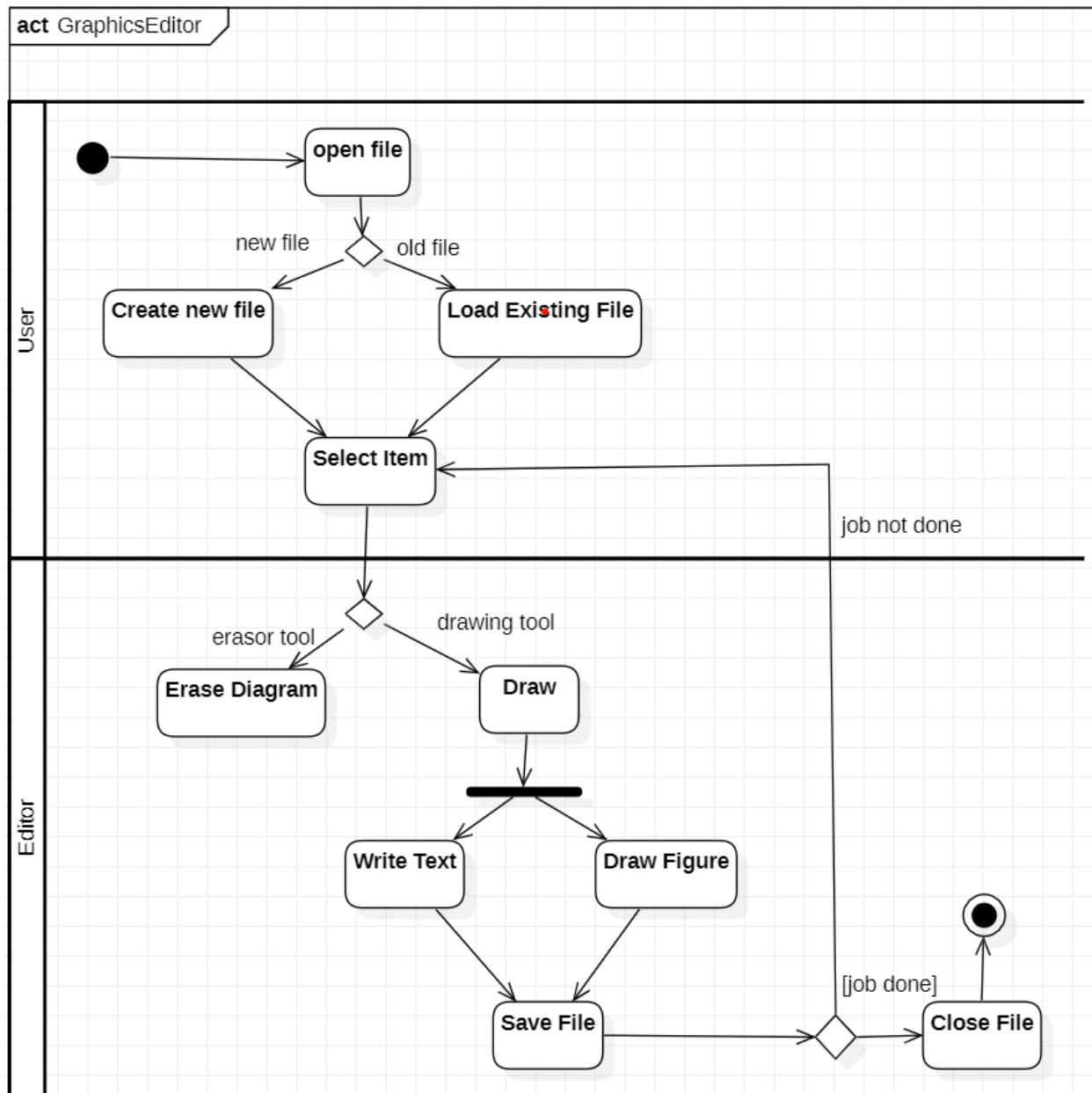
Scenario:

the graphics editor displays options to user The user selects an option

The graphics editor displays shapes The user selects a shape Parameters are asked by the editor

User enters all the required parameter The graphics editor displays the diagram

## Advance Activity Diagram:



The advanced activity diagram gives the states involved in making and saving a graphic file. The user selects a new document and draws graphics, saves the file and closes it.