SREE VIDYANIKETHAN ENGINEERING COLLEGE

Sree Sainath Nagar, A. Rangampet – 517 102

Department of Information Technology

Lab Manual, II B. Tech – I Semester (SVEC19)

(19BT31232) **SOFTWARE ENGINEERING LAB**

PRE-REQUISITES: ACourse on Software Engineering.

COURSE DESCRIPTION: Software Development Life Cycle activities-requirements specification, SRS preparation, Modeling case studies–Online Ticket Reservation system; Point of sales.

COURSE OUTCOMES: After successful completion of the course, students will be able to:

CO1. Analyse user requirements and prepare software requirements specifications.

CO2. Apply design principles of UML for software design.

CO3. Apply tools for developing UML diagrams.

CO4. Use cost estimation models for project evaluation.

CO5. Work effectively as an individual to design UML models.

CO6. Write and present a substantial technical report/document effectively.

LIST OF EXPERIMENTS:

1. Identify Functional and Non-Functional Requirements for:

   i) Online Ticket Reservation for Railways

Functional

- Every online booking needs to be associated with an account
- One account cannot be associated with multiple users
- Search results should enable users to find the most recent and relevant booking options
- System should enable users to book / pay for their tickets only in a timeboxed manner after tickets being added to the cart
- System should only allow users to move to payment only when mandatory fields such as date, time, location has been mentioned
- System should consider timezone synchronisation when accepting bookings from different timezones
- Booking confirmation should be sent to user to the specified contact details

Non Functional

- Use of captcha and encryption to avoid bots from booking tickets
- Search results should populate within acceptable time limits
- User should be helped appropriately to fill in the mandatory fields, incase of invalid input
- System should accept payments via different payment methods, like PayPal, wallets, cards, vouchers, etc
- System should visually confirm as well as send booking confirmation to the user's contact

ii) Online Auction Sales

Functional Requirements

- Only users with administrative login can access the control admin page and have full access right to the system
- Basic account users can only view sales and bid
- Basic account users are recognised by the strength of their bought items and sellers feedbacks
- Seller account users can only sell items and have to pay a subscription fee
- Seller account users are recognised by how many positive feedbacks and stars a seller has
- Advanced account users can only sell, buy or trade items, advanced account users are recognised by how well is their feedback from both sellers and buyers on the auction system, i.e. advanced account member that has 120 sales to his/her account with all positive feedbacks and stars then this seller/trader is very good
- Allow visitors to sign up for account online, Login and Logout to the system options
- New members have the option to upgrade to seller or advanced account
- Admin user can create accounts and delete accounts
- Basic account users can contact a seller privately
- Sellers can create their own discussion forum for the item on sale
- Enabling a reminder on an item for when its soon to finish
- Users can pay online using a secure payment system
- Sellers can have flash images or videos to add to their advertisement
- Guests can view and search for items but not be able to bid unless they sign up for an account
- Admin members can change sale status, Basic, Seller and Advanced members can delete or edit their account/profile

Non Functional Requirements

- Ease of navigation around the site and use of simple colours and fonts on the system
- All web pages that are generated through out the website should be downloadable in no more than 10 seconds over 40KBps modem connection.
- Interface consistency through out the system.
- The response to a query shouldn't take a very long time and not more than 10 seconds to load on screen.
- When a user sells or bids on the system they should be getting a conformation of what the member did, i.e. adding a new item on sale at the end of the adding to the system they should get a conformation message to say its been added.

2. a) Construct a flow graph for Insertion sort algorithm.

The following steps should be followed for computing Cyclomatic complexity and test cases design.

Step 1 - Construction of graph with nodes and edges from the code

Step 2 - Identification of independent paths

Step 3 - Cyclomatic Complexity Calculation

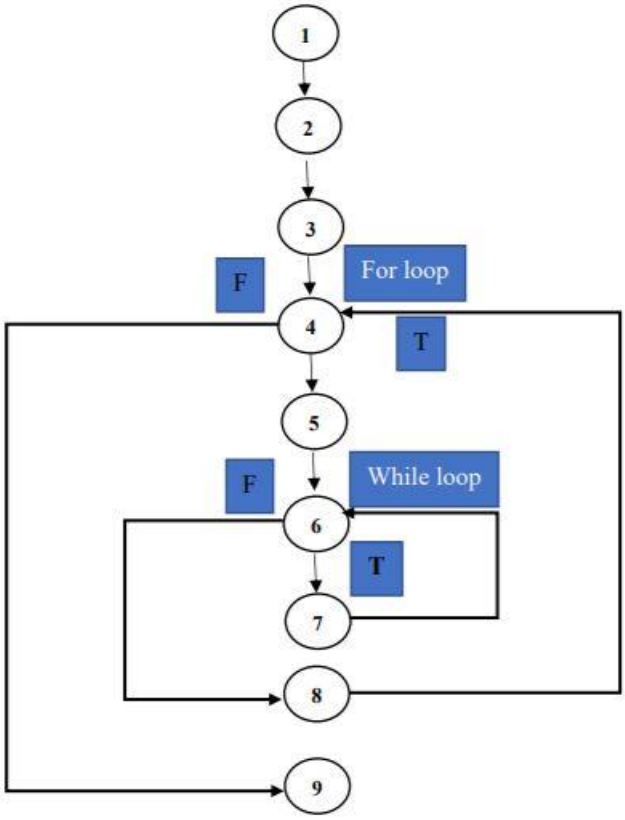
Step 4 - Design of Test Cases

Once the basic set is formed, TEST CASES should be written to execute all the paths.

3. Insertion Sort Algorithm

```
3.  def InsertionSort(A,n)
4.  for i=0 to n {
5.  k = A[i];
    j = i-1;
6.  while (j>=0) and (A[j]>k) {
7.  A[j+1] = A[j]
    j = j-1
    }
8.  A[j+1] = k
    }
9.  print(A)
1.  A=[11,12,13,1,4]
2.  InsertionSort(A,len(A))
```

Flow Graph for Insertion Sort Algorithm



b) Write a program to find Cyclomatic complexity for the above flow graph (**Hint**: McCabe's cyclomatic matrices V(G) of a graph G with n vertices and edges is V(G)=e-n+2)

Number of edges:10

Number of Nodes:9

Cyclomatic complexity : 3

3.3. Manual Calculation of Cyclomatic Complexity for Insertion Sort

Mathematically, it is set of independent paths through the graph diagram. The Code complexity of the program can be defined using the formula –

V(G) = E-N+2

Where,

E - Number of edges

N - Number of Nodes

V(G) = P +1

Where P = Number of predicate nodes (node that contains condition)

- V(G) = E – N + 2 = 10 - 9 + 2 = 3
- V(G) = P + 1 = 2 + 1 = 3 (Condition nodes are 4 and 6 nodes)

CASE STUDIES:

Case studies given below should be Modeled using Visual Modeling tools in different views i.e. Use case view, logical view, component view, Deployment view.

CASE STUDY 1: ONLINE TICKET RESERVATION FOR RAILWAYS

Problem Statement: Computer play an integral part of the day in today's life. It makes the entire job easier and faster, every job is computerized so as the ticket reservation we can book over the online ticket reservation system. During the booking of the ticket reservation passenger has to select origin, date of journey, destination, class of train etc. The reservation counter keeps track of passenger's information. Thus the system will have all the details about the trains and facilities provided by them. There are various trains with the different level of convenience for the passengers. The whole database will be maintained by database administrator. There are varieties of trains where the passengers can select the train according to the convenience for their destination journey. The journey could be within the state or across the India. Each train has the three types of classes i.e. Sleeper class, First class and the AC compartment. Design the application for the above problem description.

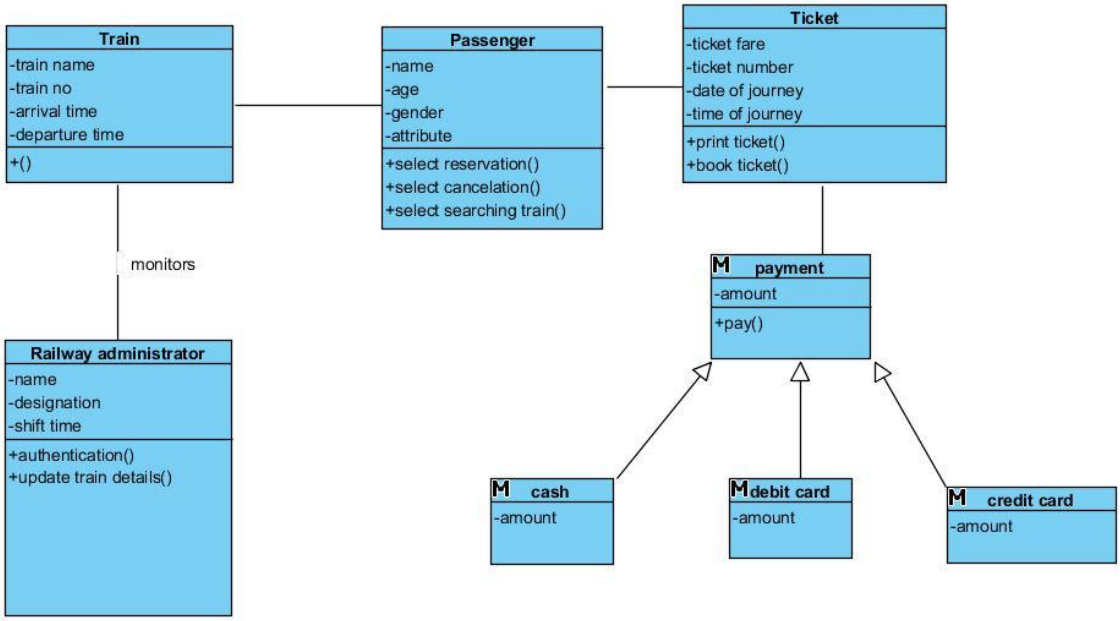


Fig: Class Diagram for **Online Ticket Reservation for Railways**

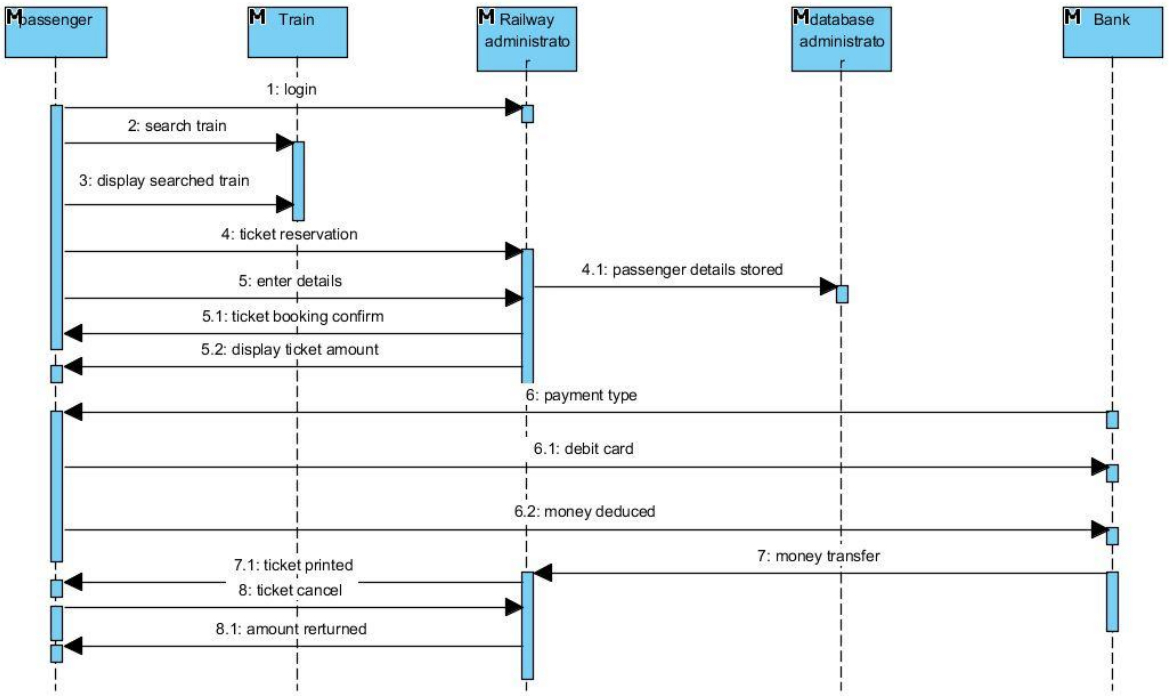


Fig: Sequence Diagram for **Online Ticket Reservation for Railways**

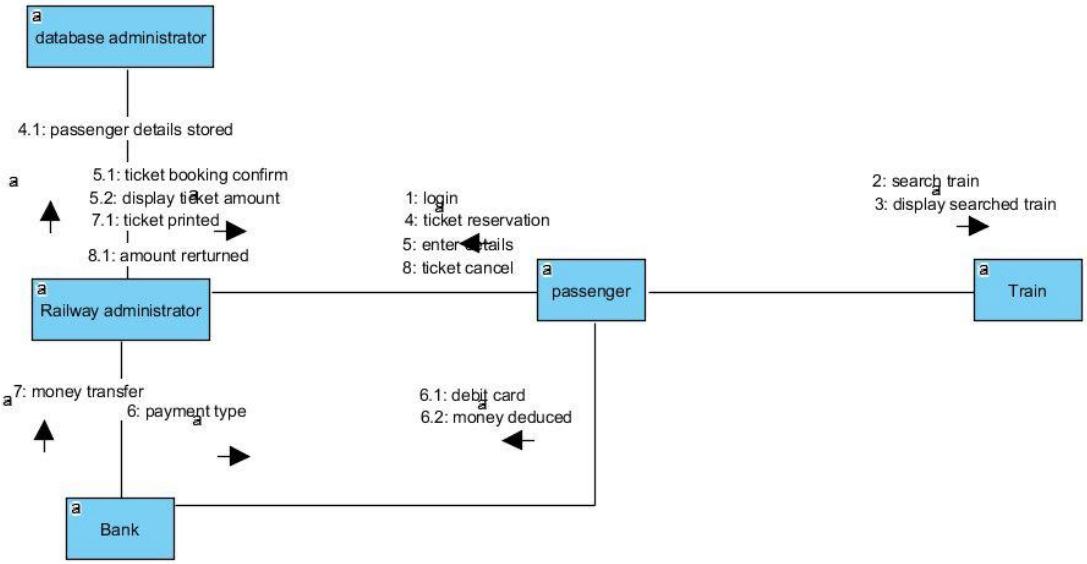


Fig: Collaboration Diagram for **Online Ticket Reservation for Railways**

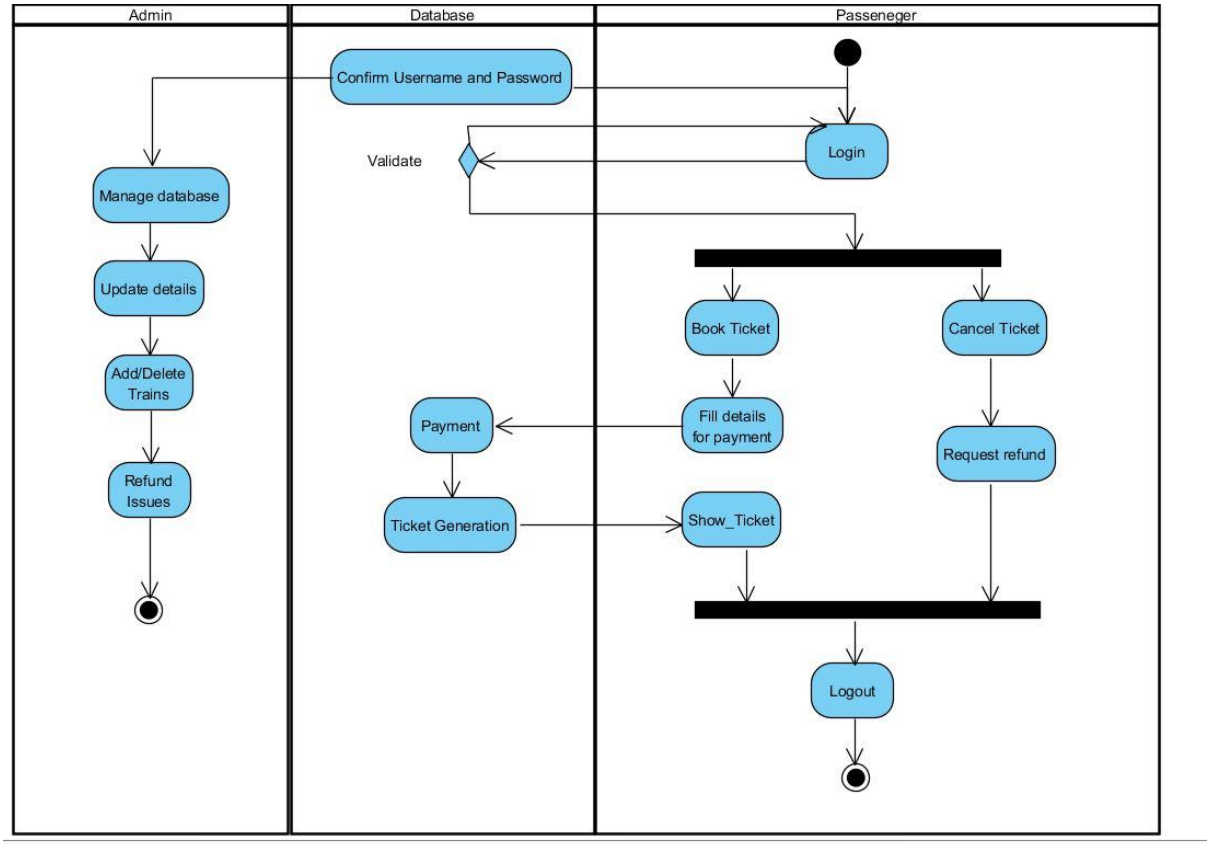


Fig: Activity Diagram for **Online Ticket Reservation for Railways**

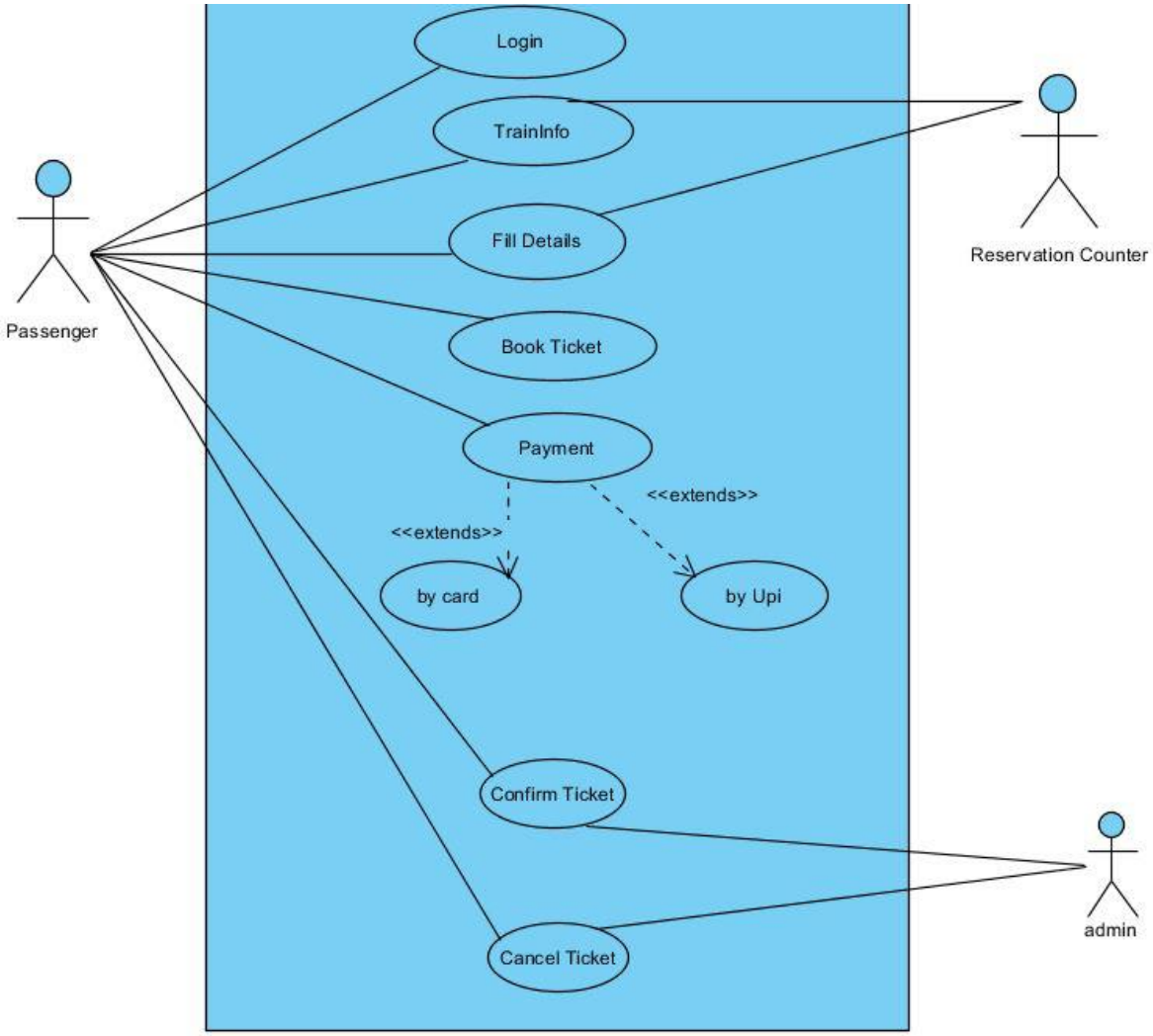


Fig: Usecase Diagram for **Online Ticket Reservation for Railways**

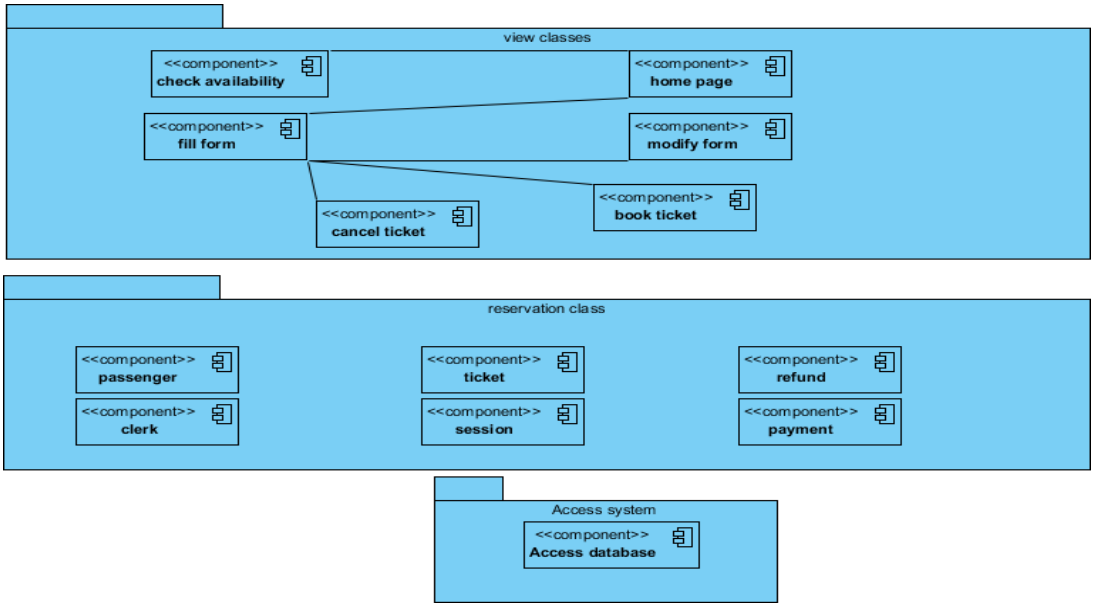


Fig: Component Diagram for **Online Ticket Reservation for Railways**

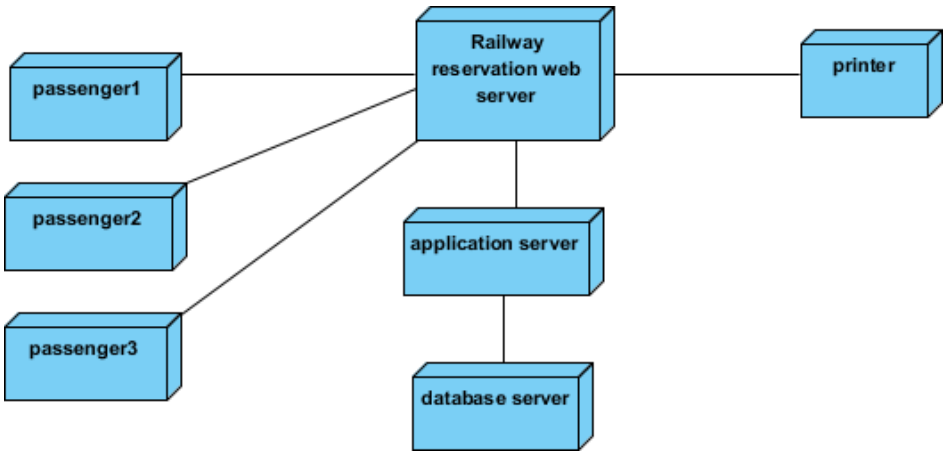


Fig: Deployment Diagram for **Online Ticket Reservation for Railways**

CASE STUDY 2: A POINT OF SALE (POS) SYSTEM

Problem Statement: A POS System is a computerized application used to record sales and handle payments; it is typically used in a retail store. It includes hardware components such as a computer and bar code scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant; that is, even if remote services and temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client – side terminals and interfaces such as browser, PDA's, touch – screens.

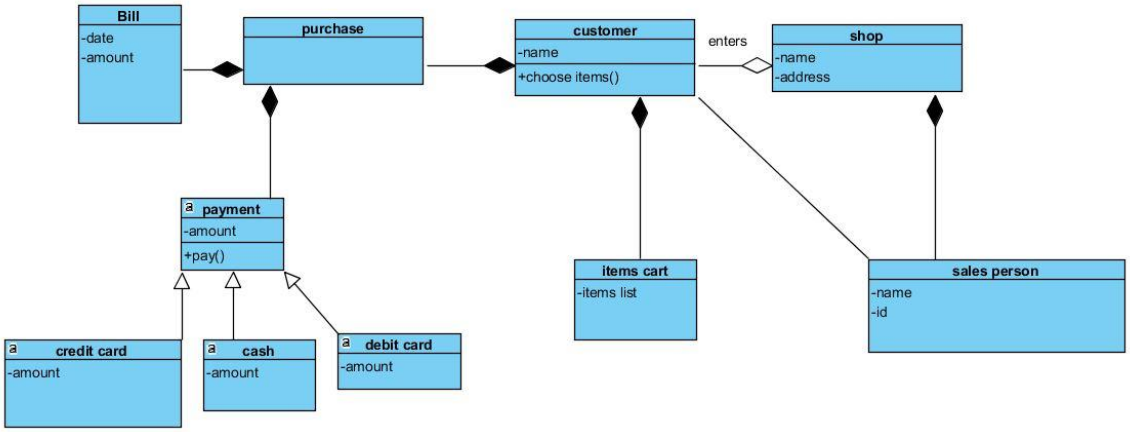


Fig: Class Diagram for **Point of Sale (POS) System**

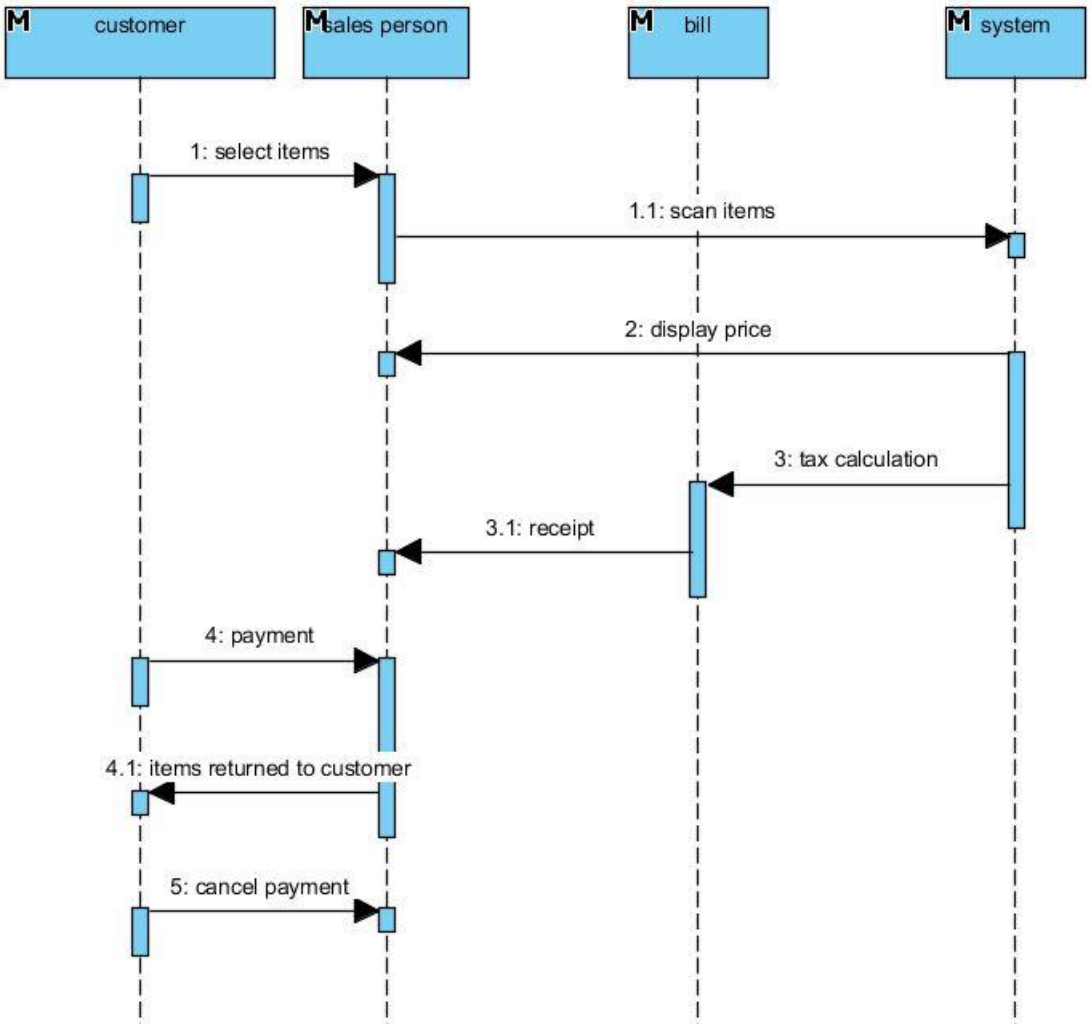


Fig: Sequence Diagram for **Point of Sale (POS) System**

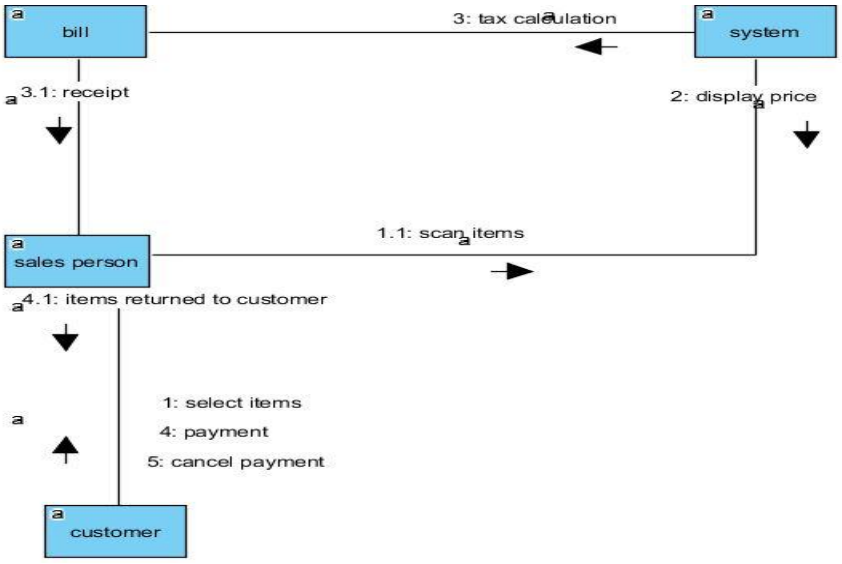


Fig: Collaboration Diagram for **Point of Sale (POS) System**

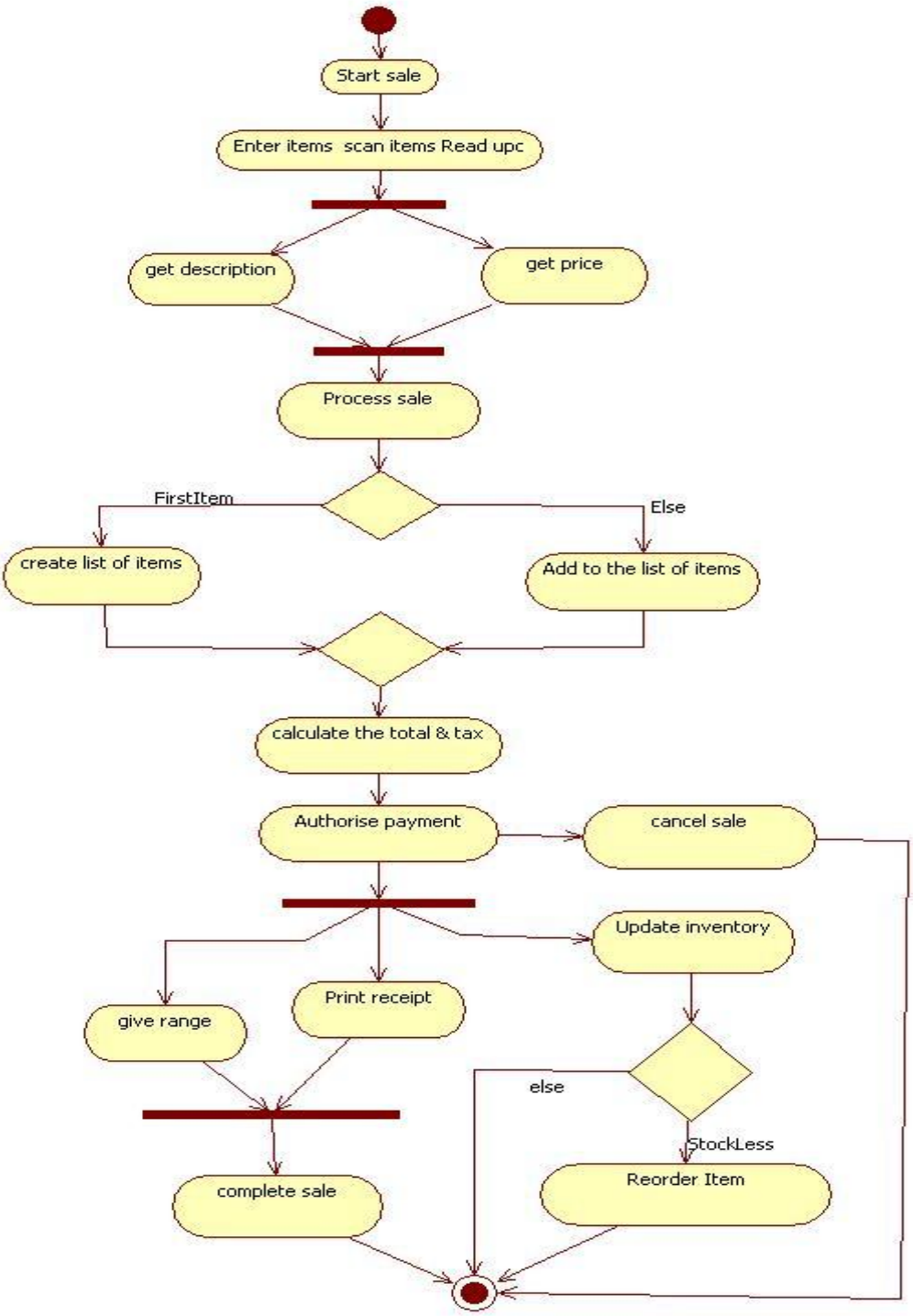


Fig: Activity Diagram for **Point of Sale (POS) System**

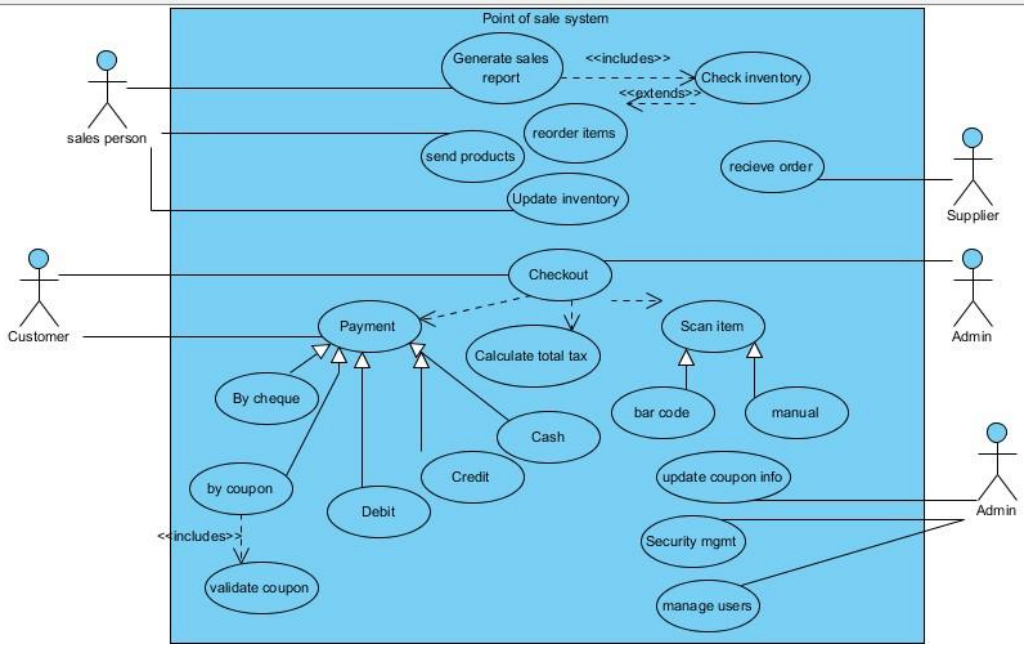


Fig: Usecase Diagram for **Point of Sale (POS) System**

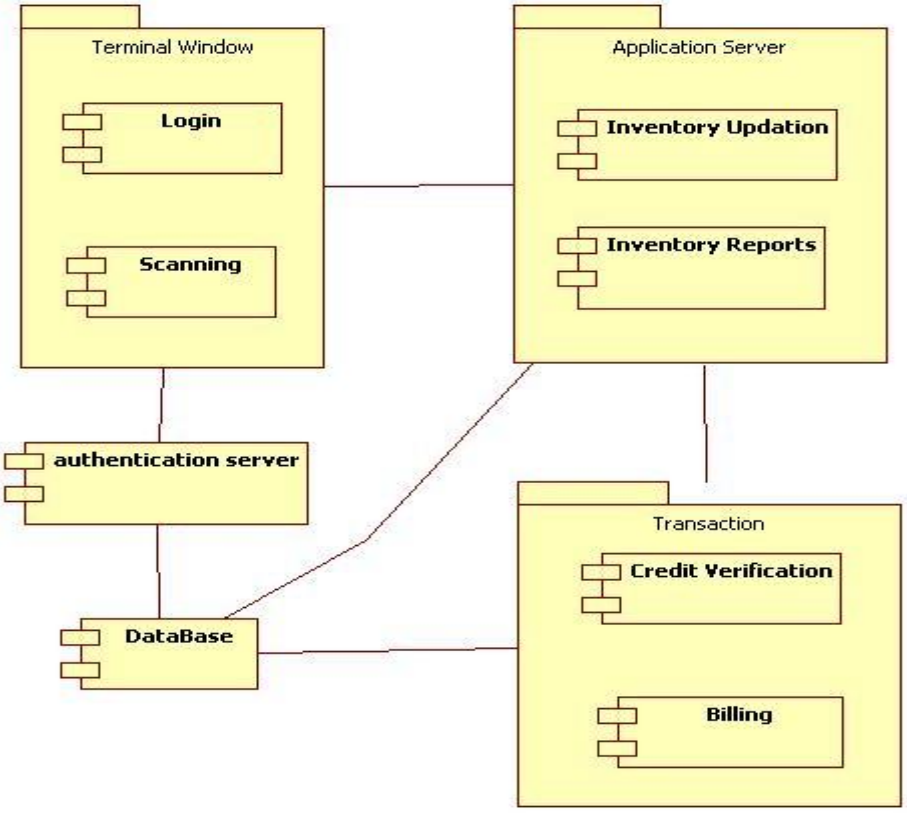


Fig: Component Diagram for **Point of Sale (POS) System**

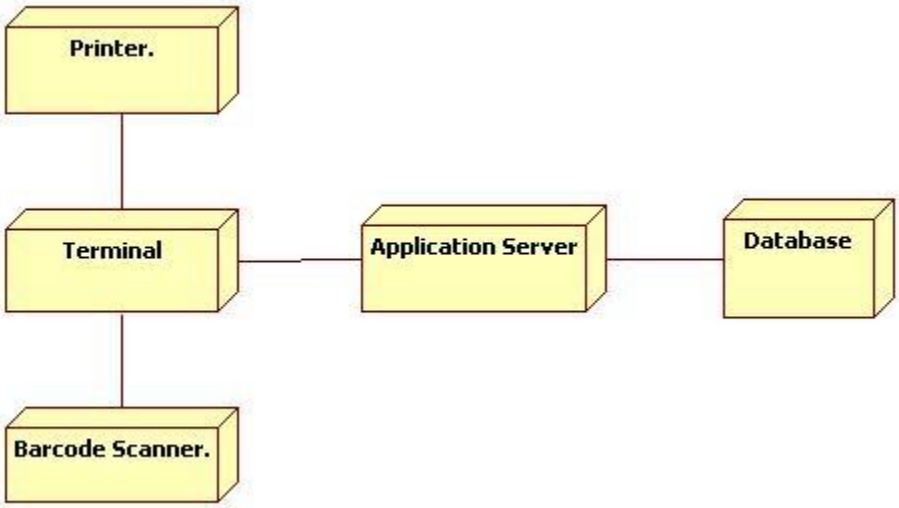


Fig: Deployment Diagram for **Point of Sale (POS) System**

CASE STUDY 3: RECRUITMENT PROCEDURE FOR SOFTWARE INDUSTRY

Problem Statement: In the software industry the recruitment procedure is the basic thing that goes in the hand with the requirement as specified by the technical management team. HR first gives an advertisement in leading Newspapers, Journals, Weeklies and Websites. The job seekers can apply for it through by Post or by e-mail to the company. The technical skill and the experience of the candidates are reviewed and the short listed candidates are called for the interview. There may be different rounds for interview like the written test, technical interview, and HR interview. After the successful completion of all rounds of interview, the selected candidates' names are displayed. Meanwhile HR gives all the details about the salary, working hours, terms and conditions and the retirement benefit to the candidate.

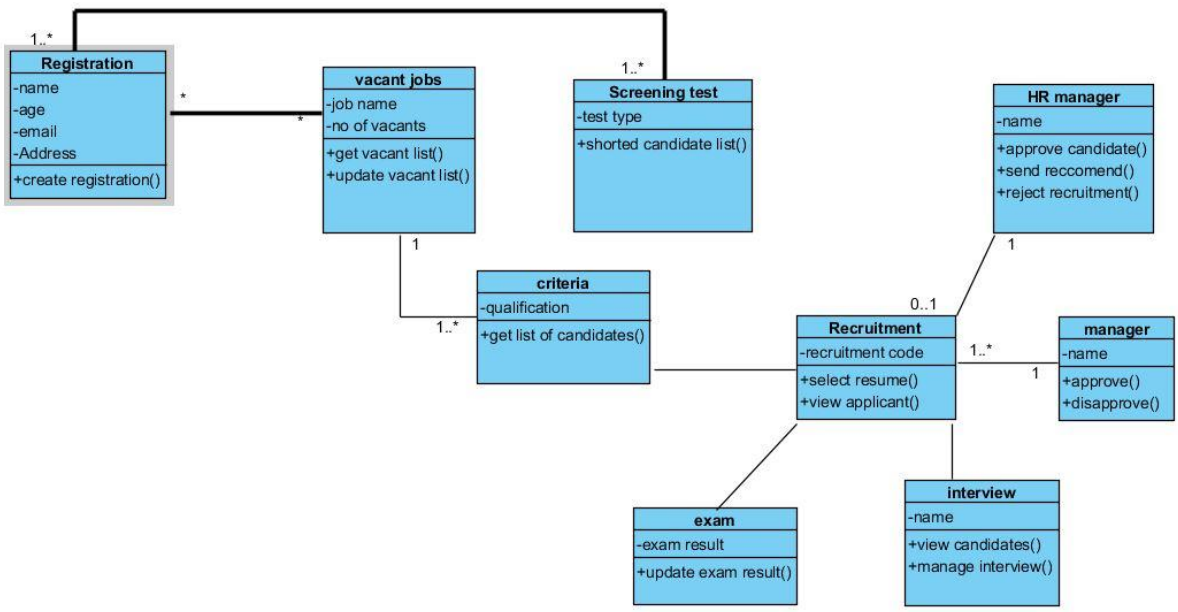


Fig: Class Diagram for **Recruitment Procedure for Software Industry**

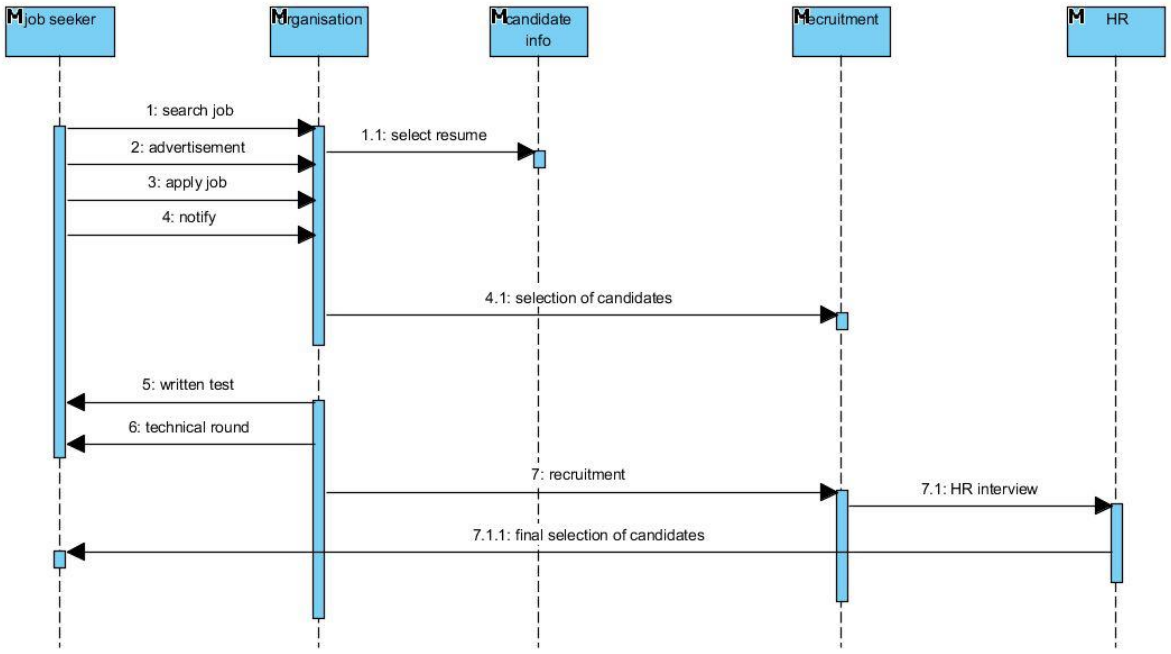


Fig: Sequence Diagram for **Recruitment Procedure for Software Industry**

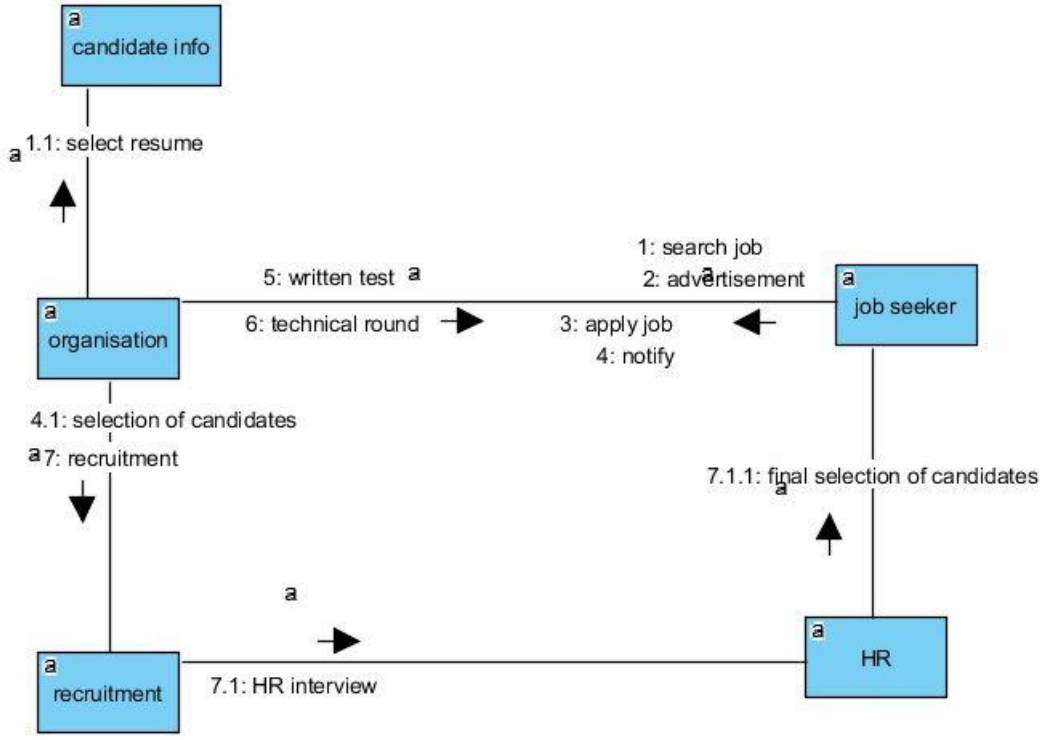


Fig: Collaboration Diagram for **Recruitment Procedure for Software Industry**

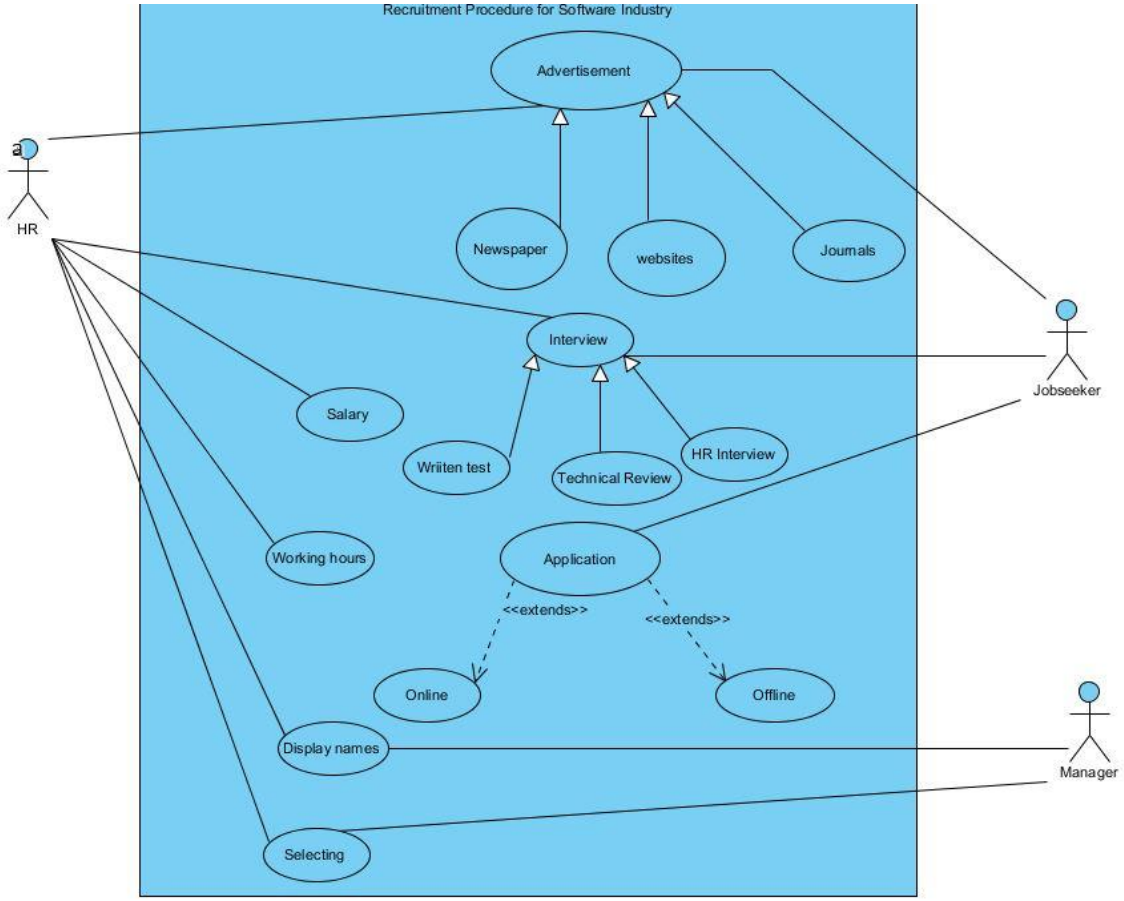


Fig: Usecae Diagram for **Recruitment Procedure for Software Industry**

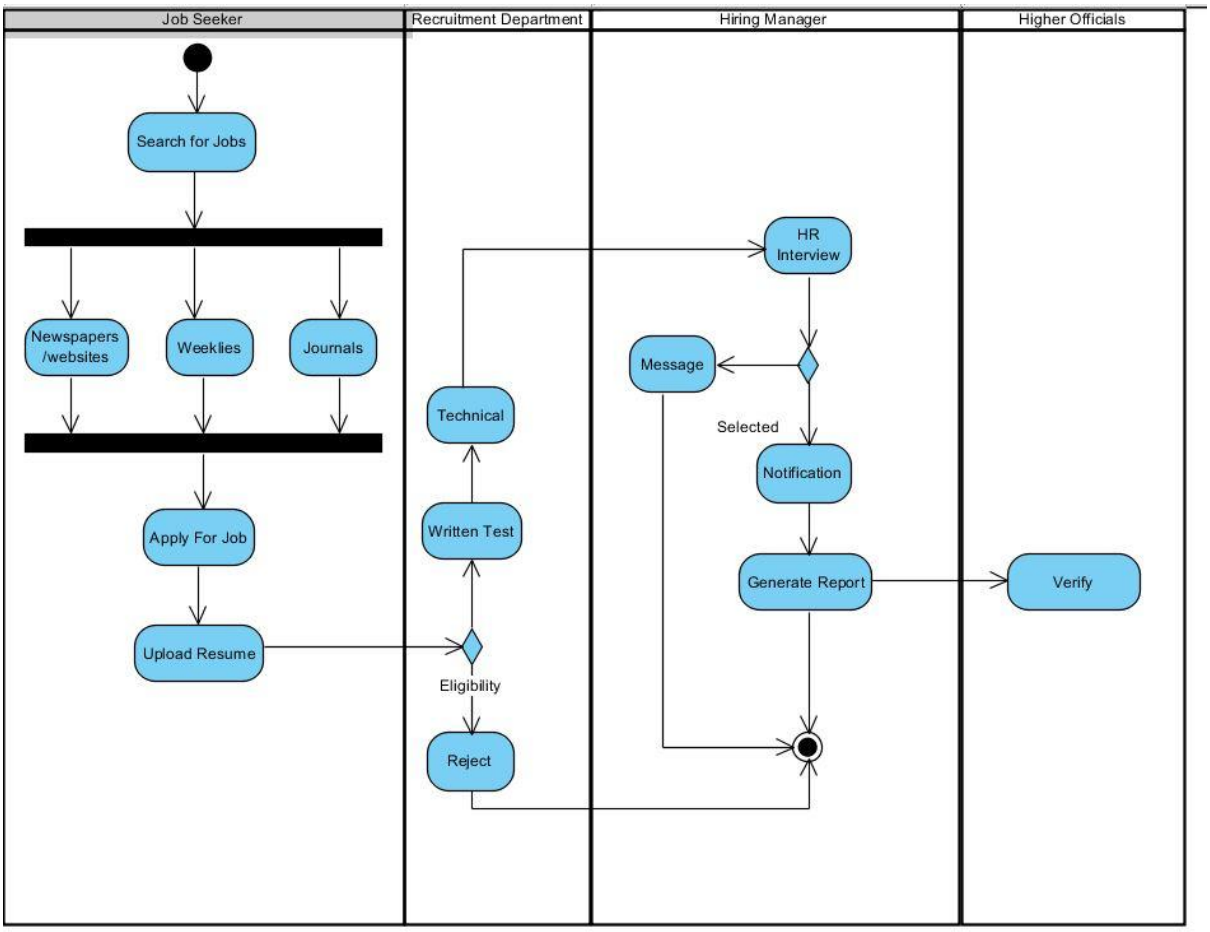


Fig: Activity Diagram for **Recruitment Procedure for Software Industry**

CASE STUDY 4: ONLINE AUCTION SALES

Problem Statement: The online auction system is a design about a website where sellers collect and prepare a list of items they want to sell and place it on the website for visualizing. To accomplish this purpose the user has to access the site. Incase it's a new user he has to register. Purchaser's login and select items they want to buy and keep bidding for it. Interacting with the purchasers and sellers through messages does this. There is no need for customer to interact with the sellers because every time the purchasers bid, the details will be updated in the database. The purchaser making the highest bid for an item before the close of the auction is declared as the owner of the item. If the auctioneer or the purchaser doesn't want to bid for the product then there is fixed cutoff price mentioned for every product. He can pay that amount directly and own the product. The purchaser gets a confirmation of his purchase as an acknowledgement from the website. After the transition by going back to the main menu where he can view other items.

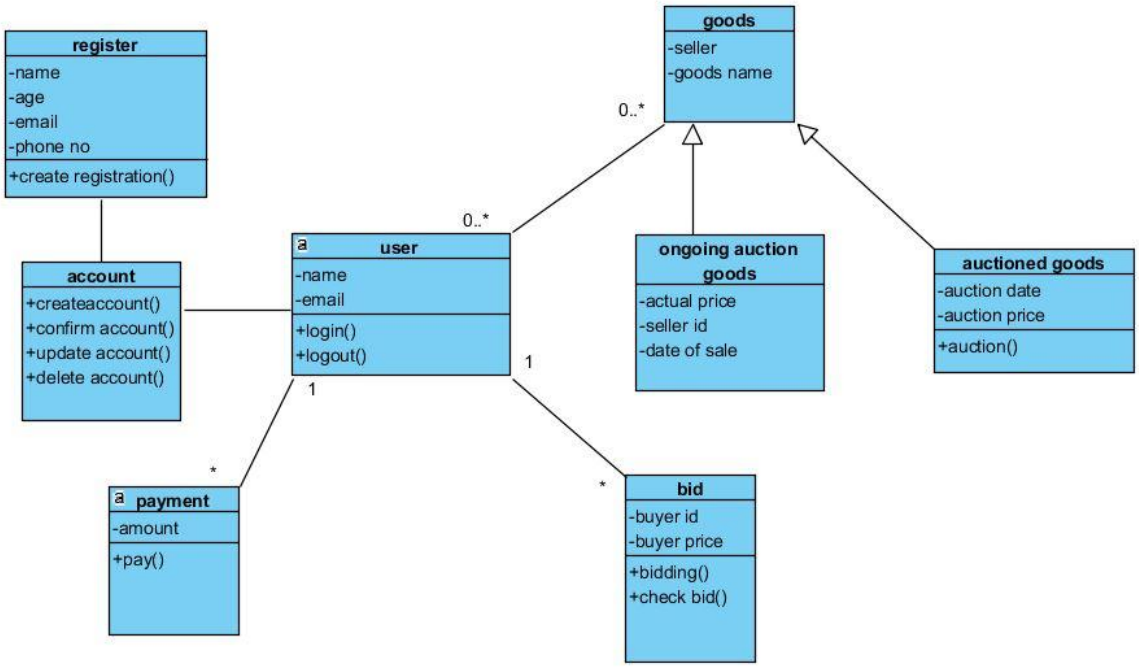


Fig: Class Diagram for **Online Auction Sales**

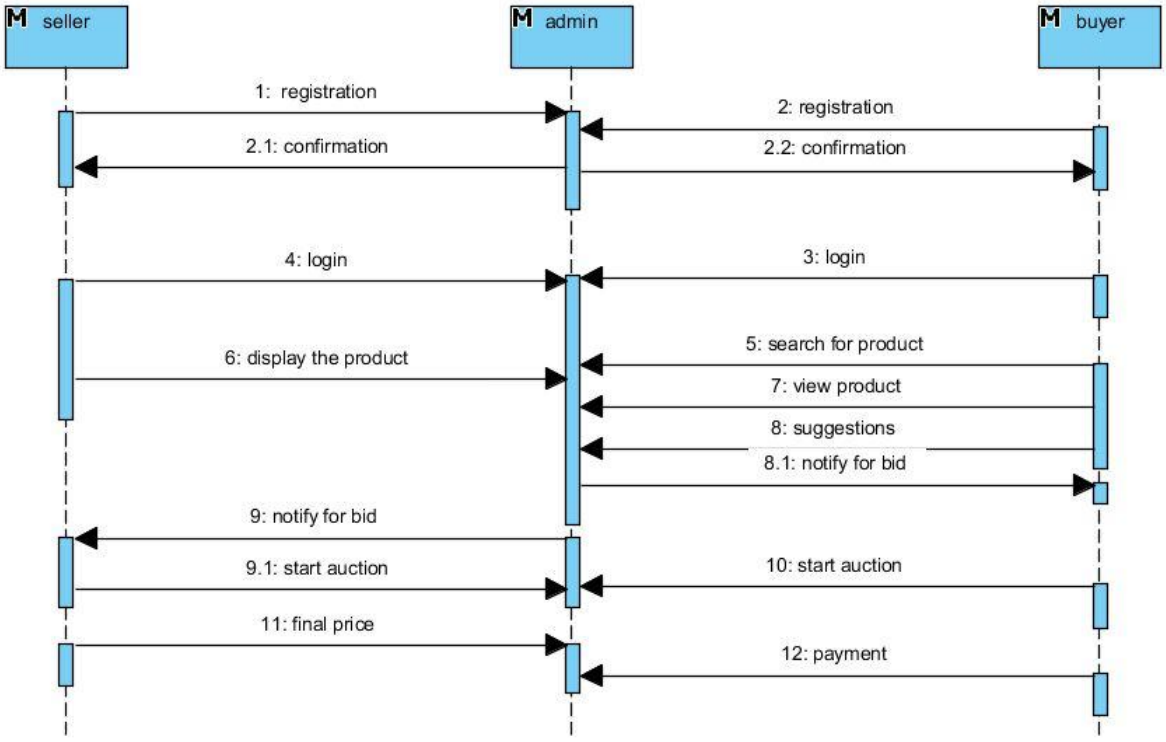


Fig: Sequence Diagram for **Online Auction Sales**

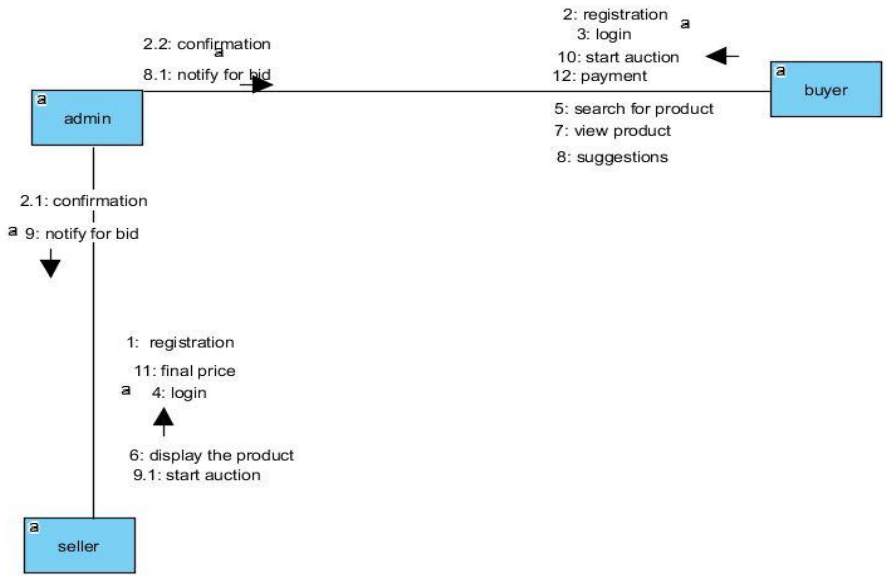


Fig: Collaboration Diagram for **Online Auction Sales**

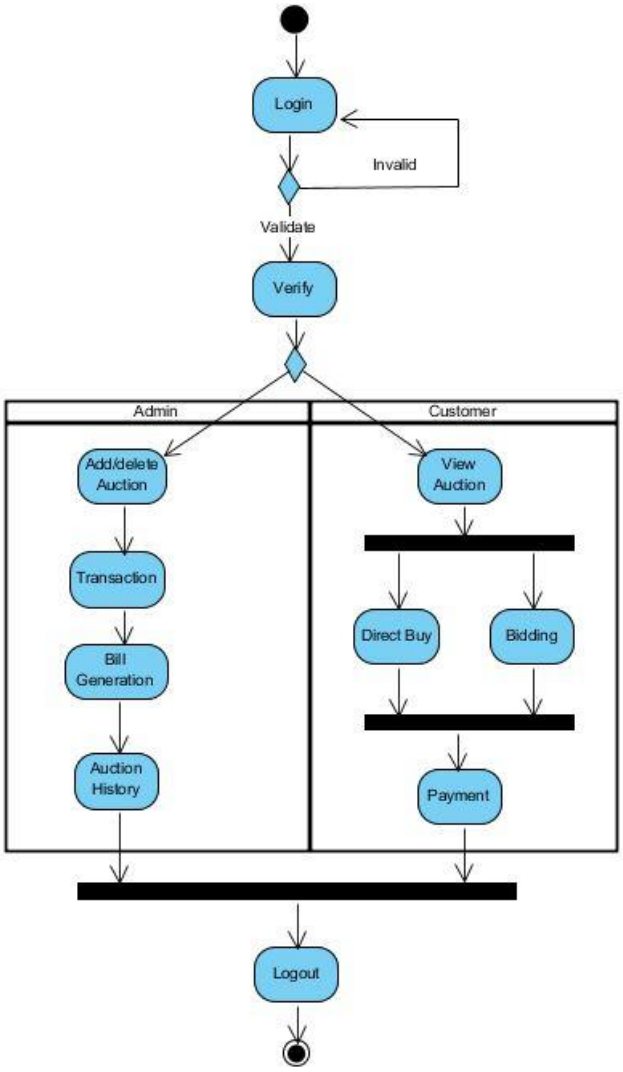


Fig: Activity Diagram for **Online Auction Sales**

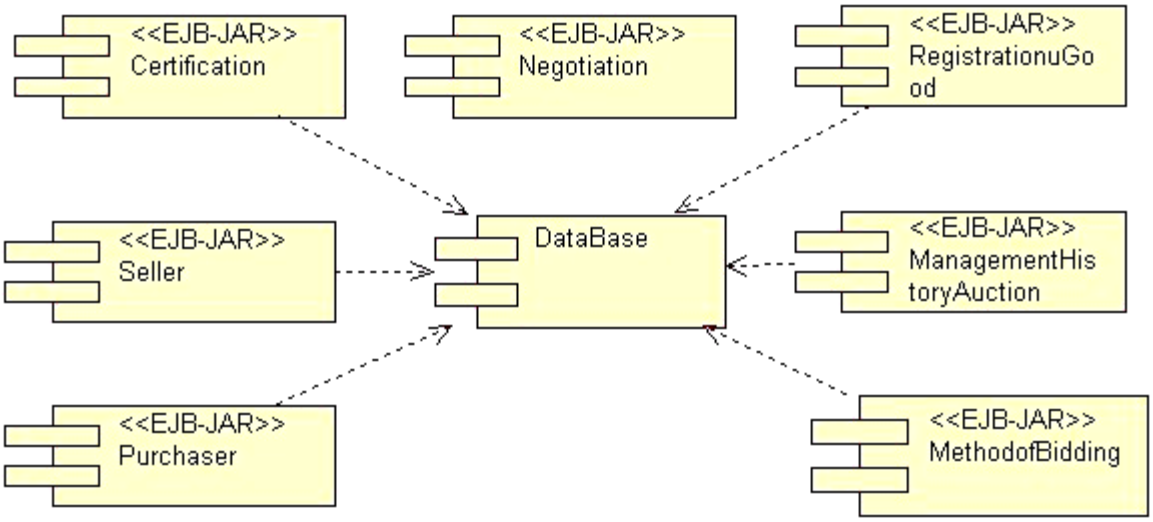


Fig: Component Diagram for **Online Auction Sales**

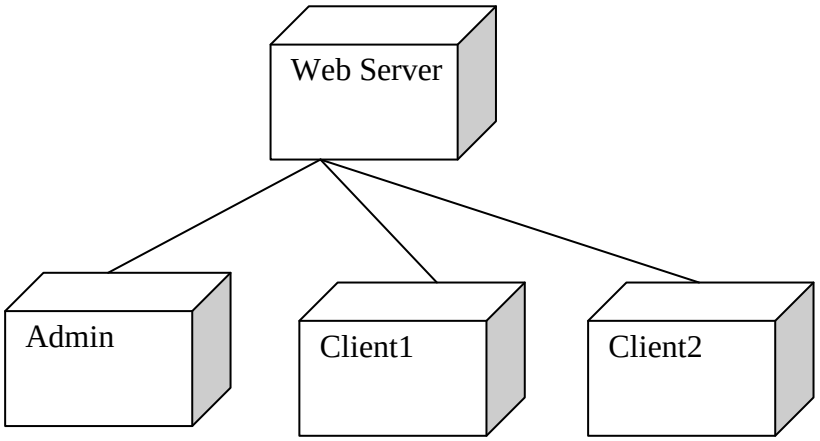


Fig: Deployment Diagram for **Online Auction Sales**

CASE STUDY 5: TWO FLOOR ELEVATOR SIMULATOR

Problem Statement: The elevator has the basic function that all elevator systems have, such as moving up and down, open and close doors, and of course, pick up passengers. The elevator is supposed to be used in a building having floors numbered from 1 to MaxFloor, where the first floor is the lobby. There are car call buttons in the car corresponding to each floor. For every floor except for the top floor and the lobby, there are two hall call buttons for the passengers to call for going up and down. There is only one down hall call button at the top floor and one up hall call button in the lobby. When the car stops at a floor, the doors are opened and the car lantern indicating the current direction the car is going is illuminated so that the passengers can get to know the current moving direction of the car. The car moves fast between floors, but it should

be able to slow down early enough to stop at a desired floor. When an elevator has no requests, it remains at its current floor with its doors closed.

In order to certificate system safety, emergency brake will be triggered and the car will be forced to stop under any unsafe conditions.

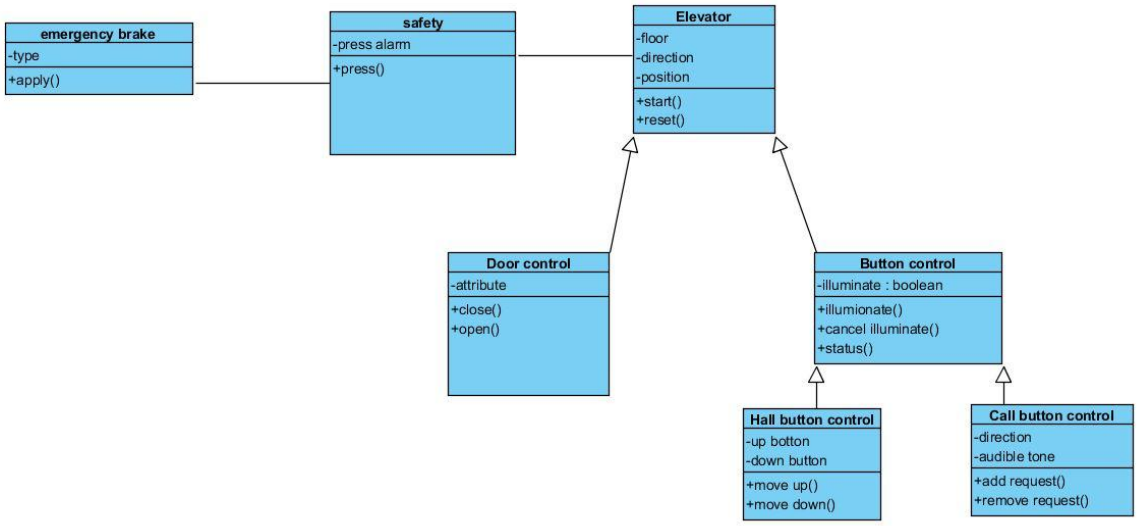


Fig: Class Diagram for **Two Floor Elevator Simulator**

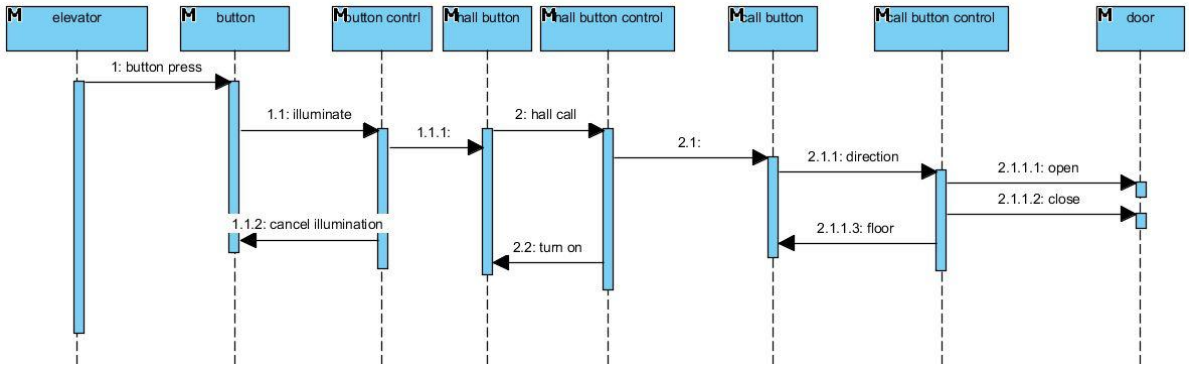


Fig: Sequence Diagram for **Two Floor Elevator Simulator**

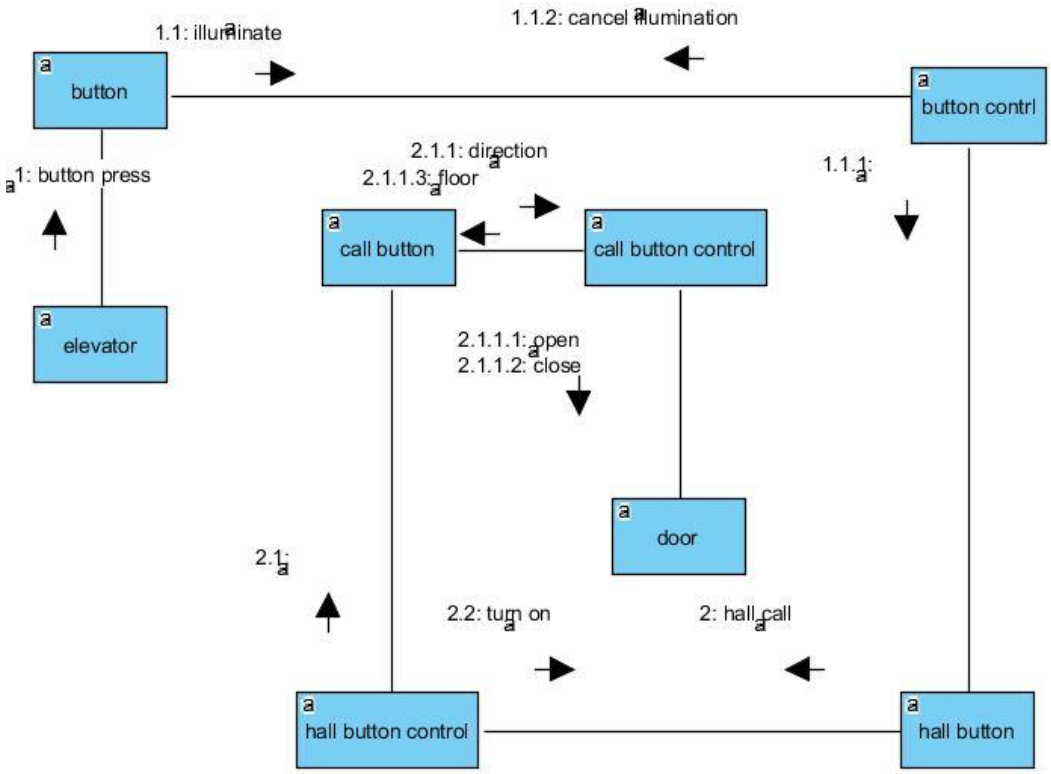


Fig: Collaboration Diagram for **Two Floor Elevator Simulator**

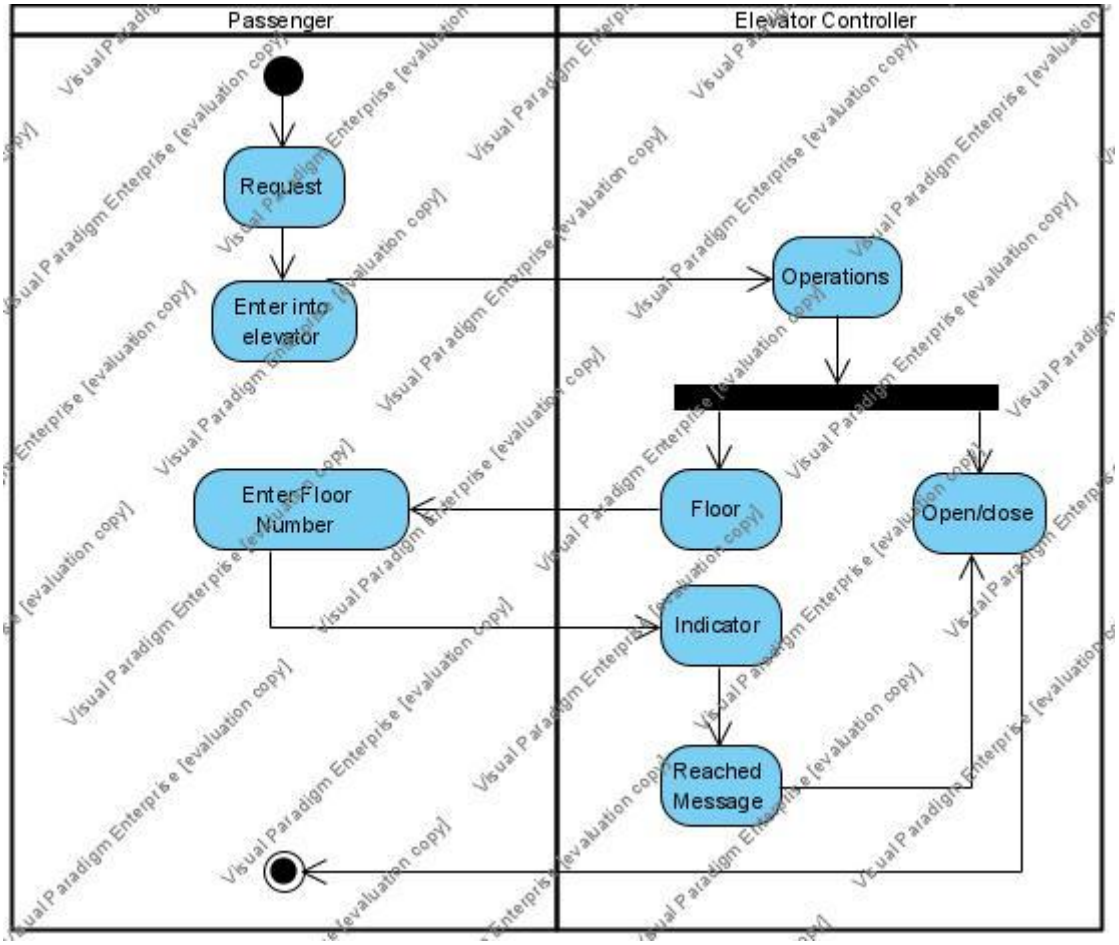


Fig: Activity Diagram for **Two Floor Elevator Simulator**

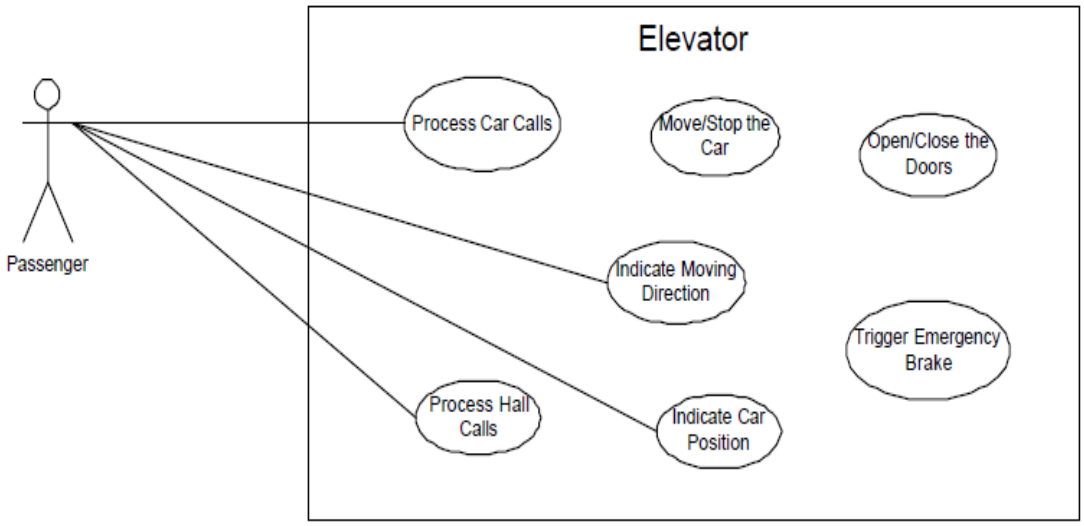


Fig: Usecase Diagram for **Two Floor Elevator Simulator**

CASE STUDY 6: HOME APPLIANCE CONTROL SYSTEM

Problem Statement: A home appliance control system (HACS) is a system which provides various services to remotely operate on home appliances, such as microwave oven, TV, and garage door etc through remote devices such as mobile phone, desktop and palm-top.A home appliance control system (HACS) is a system which is controlled by a remote system such as a mobile phone or a palm-top, and at the same time controls, monitors and coordinates home appliances such as air conditioner, microwave oven, garage doors, TV set, VCR, audio controller, indoor/outdoor lights, water sprinkler, home security system, bath tub controller, etc. In order to activate home appliances and to allow for different ways of cooking, the HACS needs mechanisms for communication between the different devices in the system, and for coordination among the various processes running on such devices. The system administrator of the HACS system has the ability to add a new appliance or delete an existing one. The system administrator has the ability to add a new remote device and configure it with HACS or delete an existing one when it is not used. Also the system administrator can create an account for a new user or delete existing account if it is no longer used.

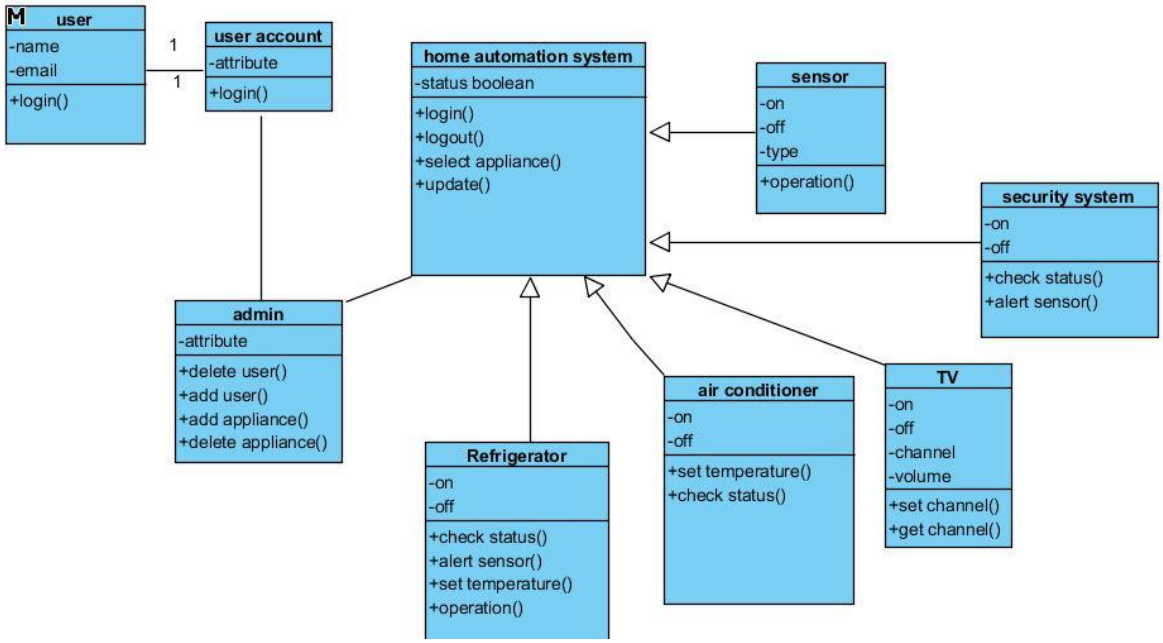


Fig: Class Diagram for **Home Appliance Control System**

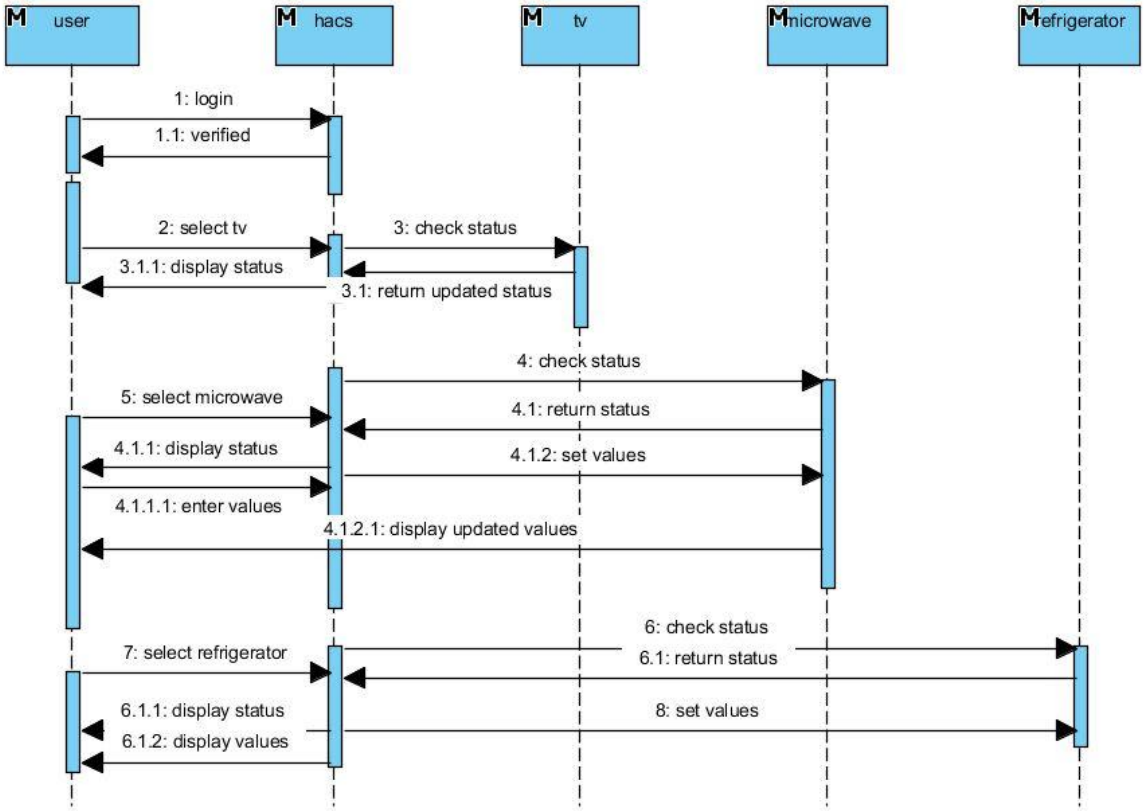


Fig: Sequence Diagram for **Home Appliance Control System**

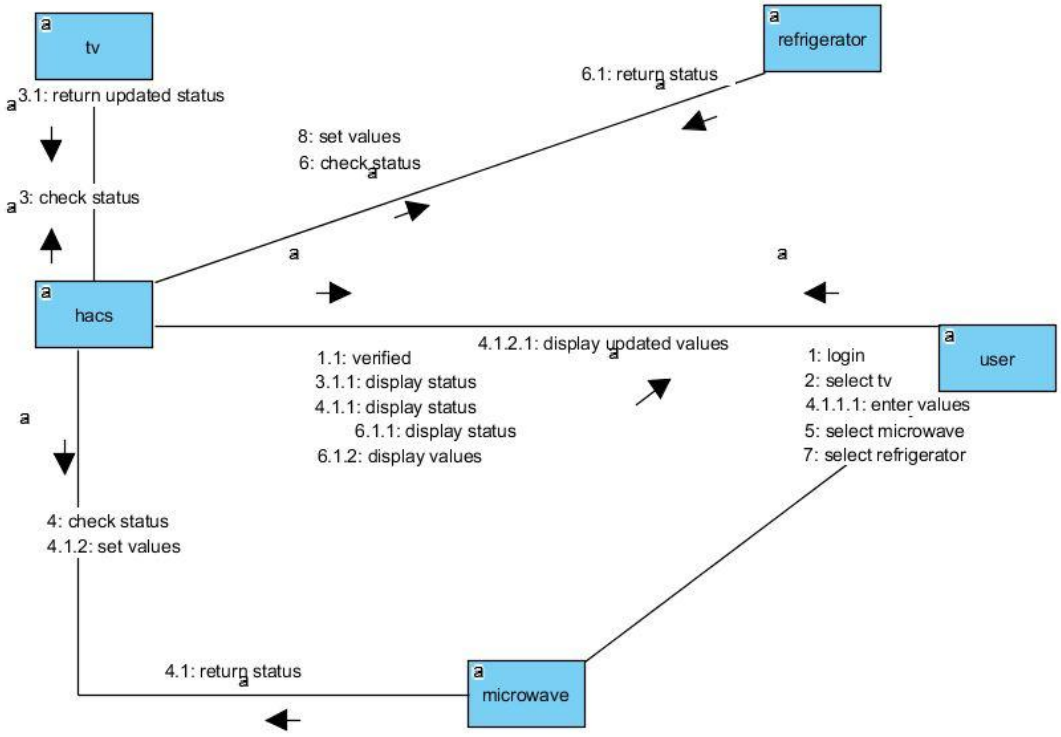


Fig: Collaboration Diagram for **Home Appliance Control System**

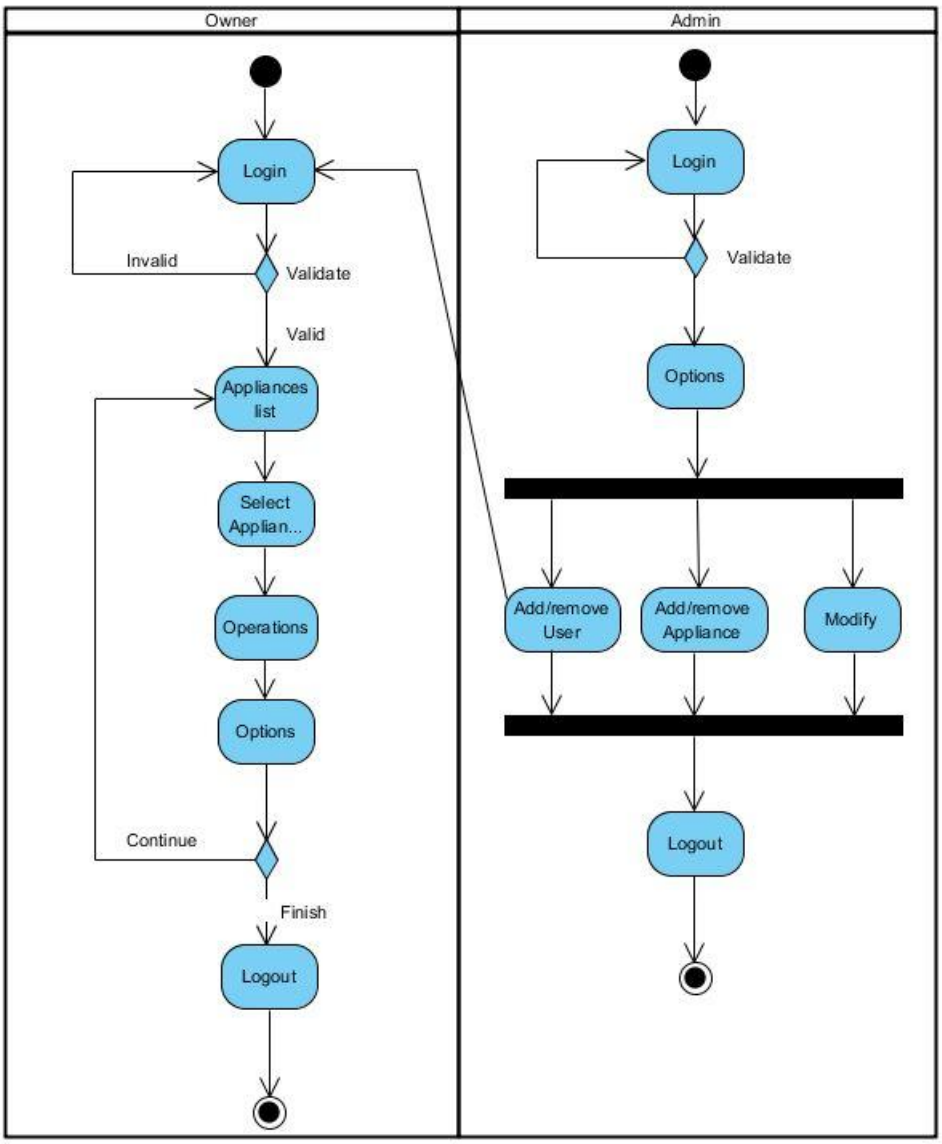


Fig: Activity Diagram for **Home Appliance Control System**

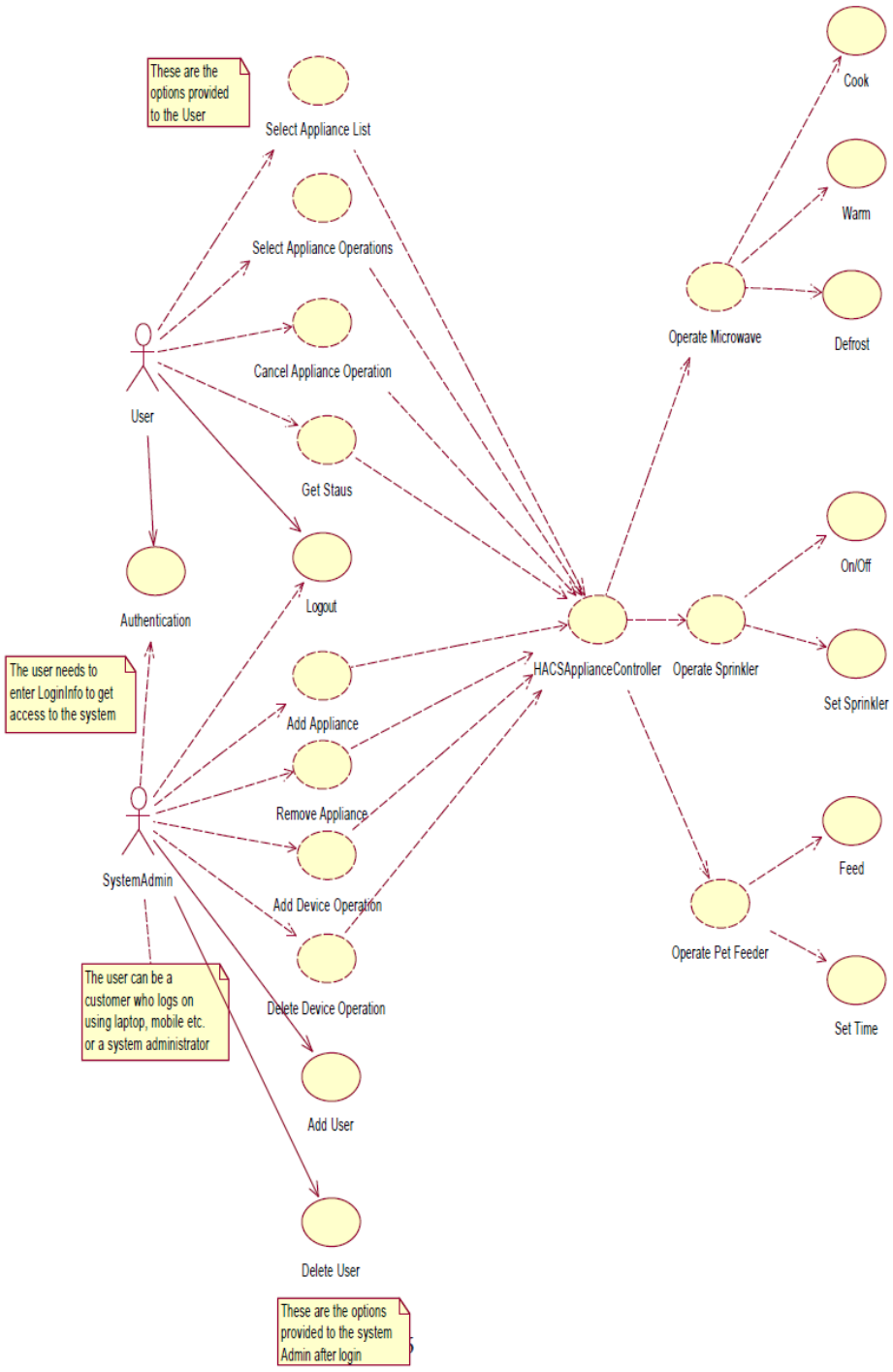


Fig: Usecase Diagram for **Home Appliance Control System**

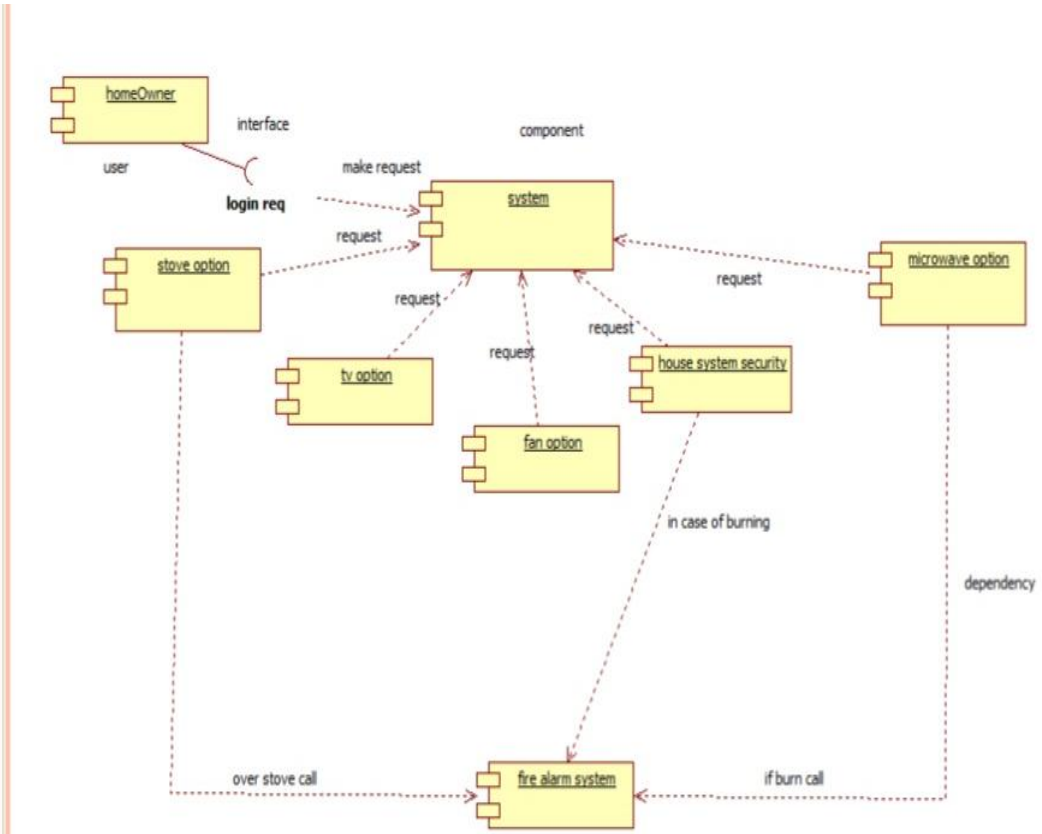


Fig: Component Diagram for **Home Appliance Control System**

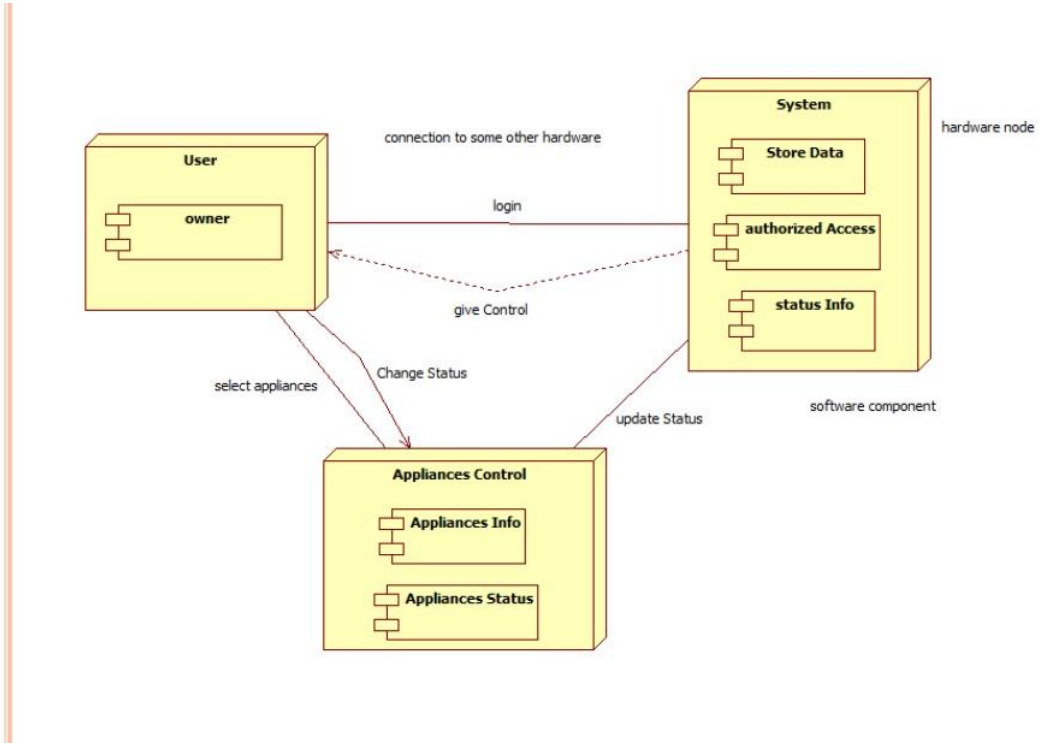


Fig: Deployment Diagram for **Home Appliance Control System**

REFERENCE BOOKS:

1. Grady Booch, James Rum Baugh and Ivar Jacobson, "The Unified Modeling Language User Guide," Second Edition, Pearson Education, 2009.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons and David Fado, "UML 2 Toolkit," WILEY-Dreamtech India Pvt. Ltd., 2003.
3. Rajesh Naik and Swapna Kishore, "Software Requirements and Estimation," Tata McGraw Hill, New Delhi, 2001.

SOFTWARE/TOOLS USED:

Visual Paradigm for modeling diagrams