

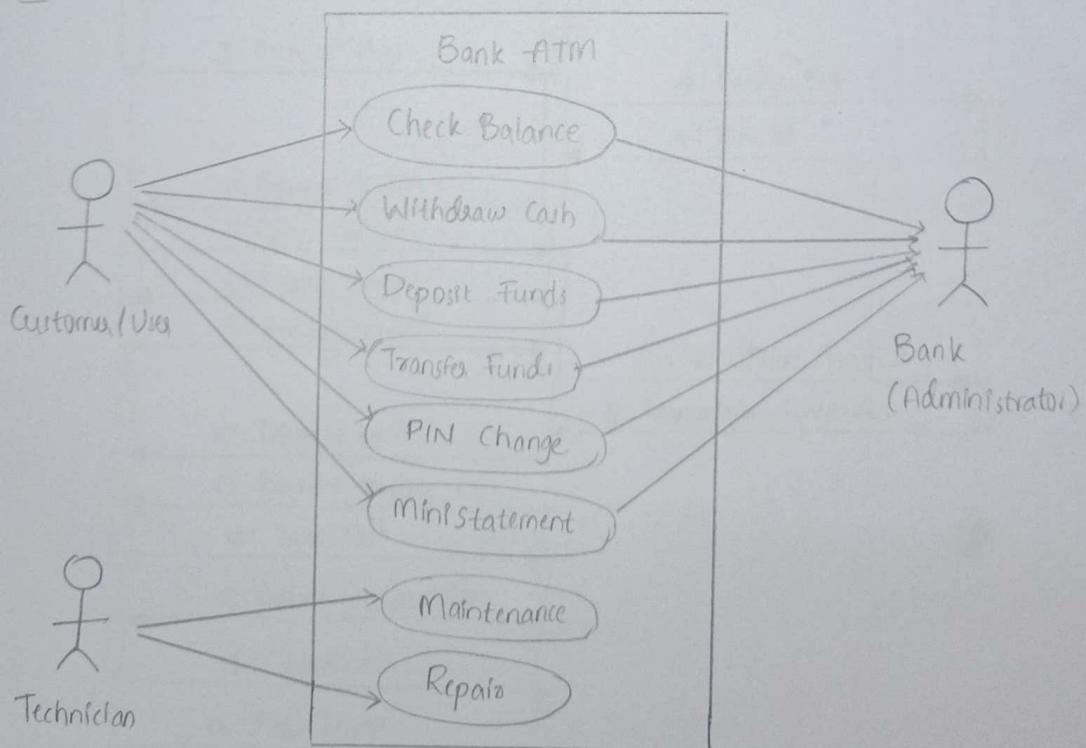
I. ATM Case Study.

1. Use case Diagram for ATM System, with use case specification.

Procedure:

- (i) First an actor is created and named as User/Customer.
- (ii) Secondly a system is created for ATM.
- (iii) A use case Enter PIN, Withdraw money, Check Balance, Deposit funds are created and connected with user as association relationship.
- (iv) Similarly various use cases like Deposit money, Maintenance, Repair, Transfer funds etc are created and appropriate relationships are associated with each of them.

Use Case:

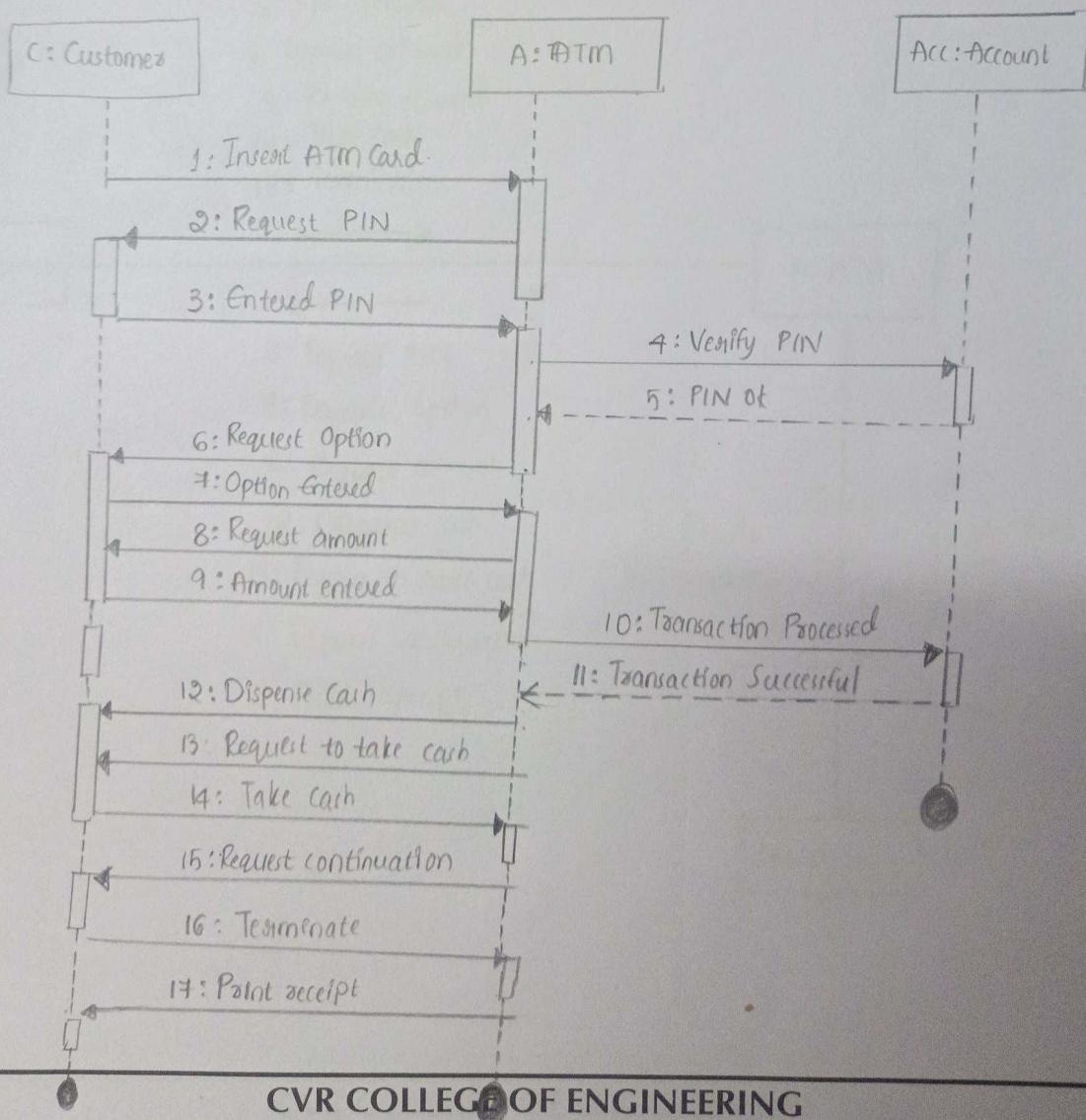


Sequence Diagram for each of the use case identified for ATM system.

Procedure:

- 1) First an actor is created and named as User.
- 2) Secondly an object is created for ATM.
- 3) Timelines and Lifelines are created automatically for them.
- 4) In sequence diagram interaction is done through time ordering of messages. So appropriate messages are passed between user and ATM is as shown below.

Sequence Diagram:

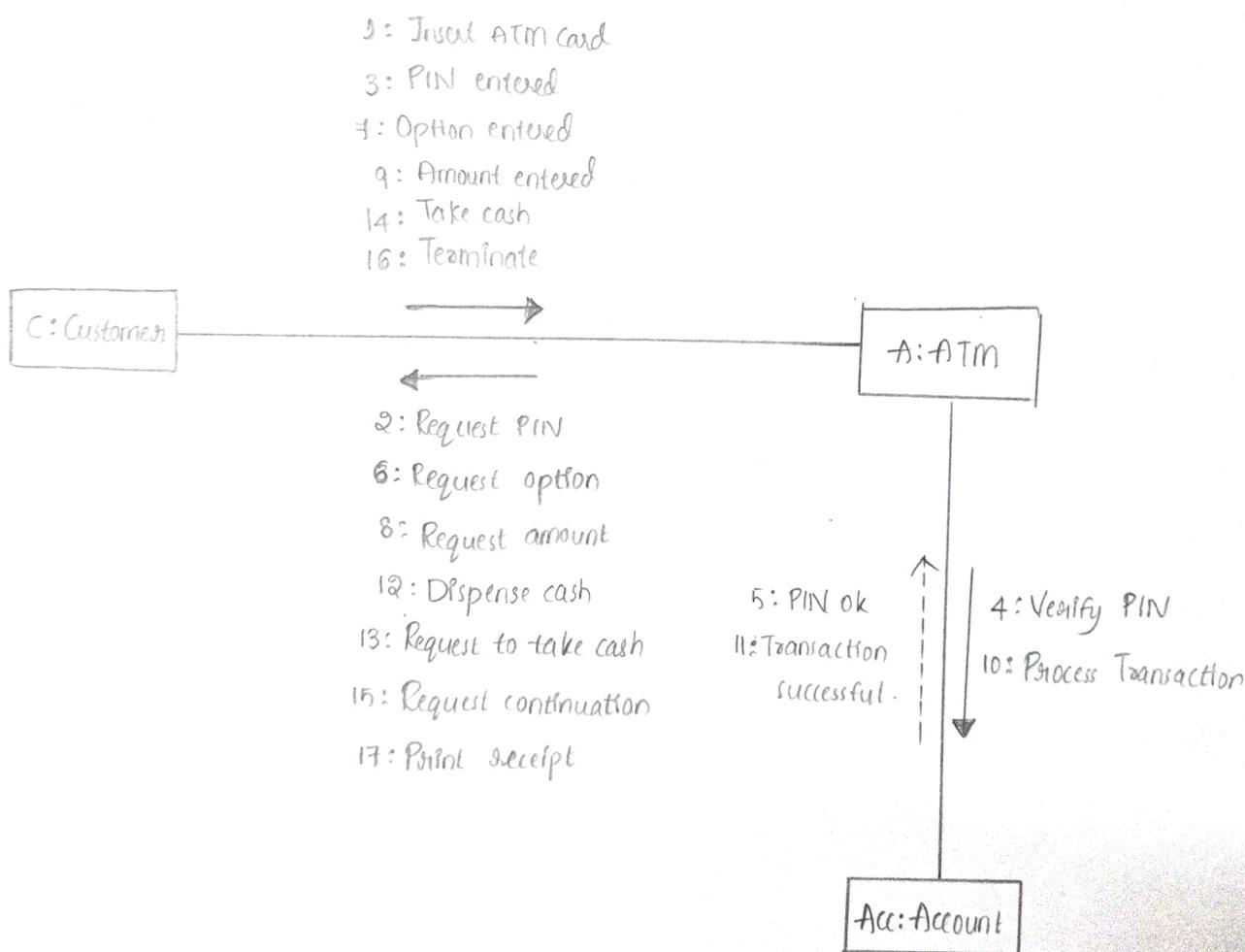


Collaboration Diagram for ATM System. (Communication Diagram)

Procedure:

- 1) An actor is created and named as user.
- 2) An object is created for use ATM.
- 3) In collaboration diagram interaction is done through organization.
- 4) So appropriate messages are passed between user and ATM.

Collaboration Diagram:

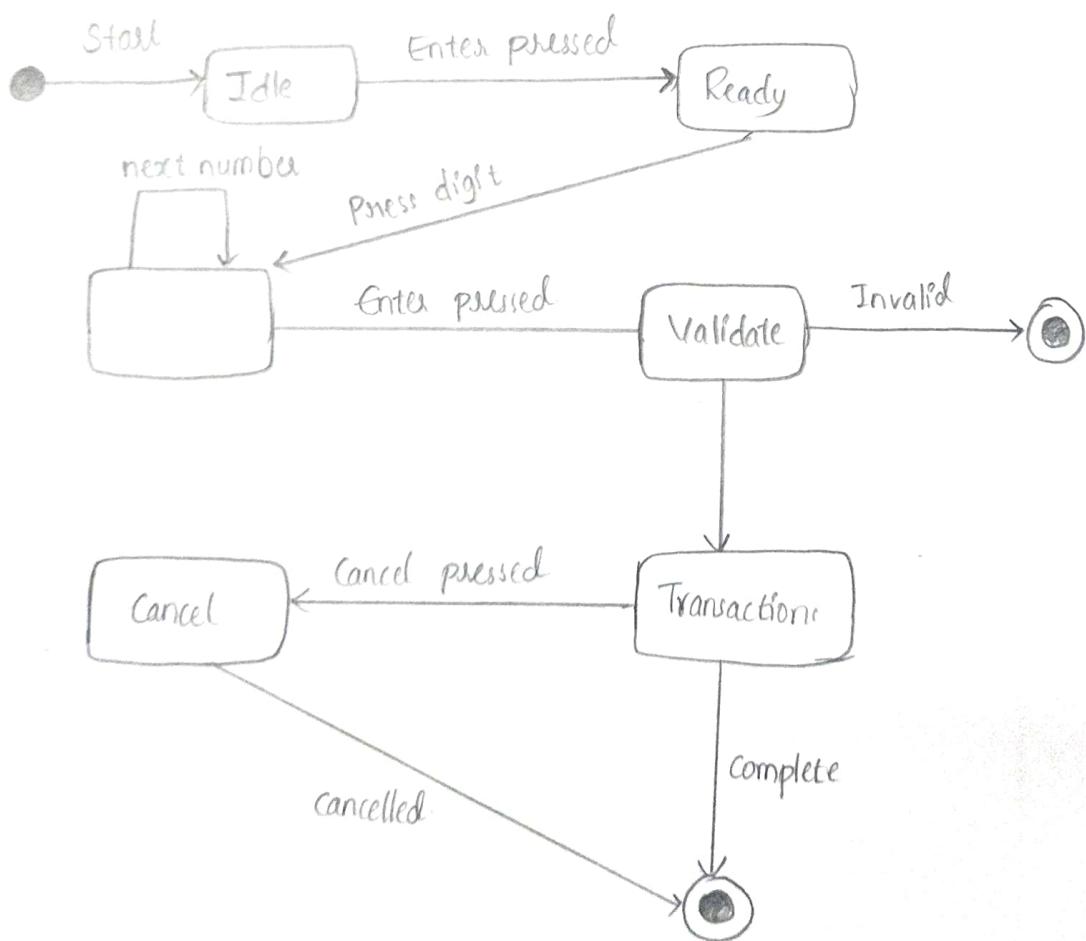


State Chart Diagram for ATM System

Procedure:

- 1) Create states such as Idle, Ready, next Number, Invalid etc. using "State" tool.
- 2) Use the "Transition" tool to connect states with transition based on the triggers or conditions.
- 3) Add initial state and final state. Connect them accordingly.

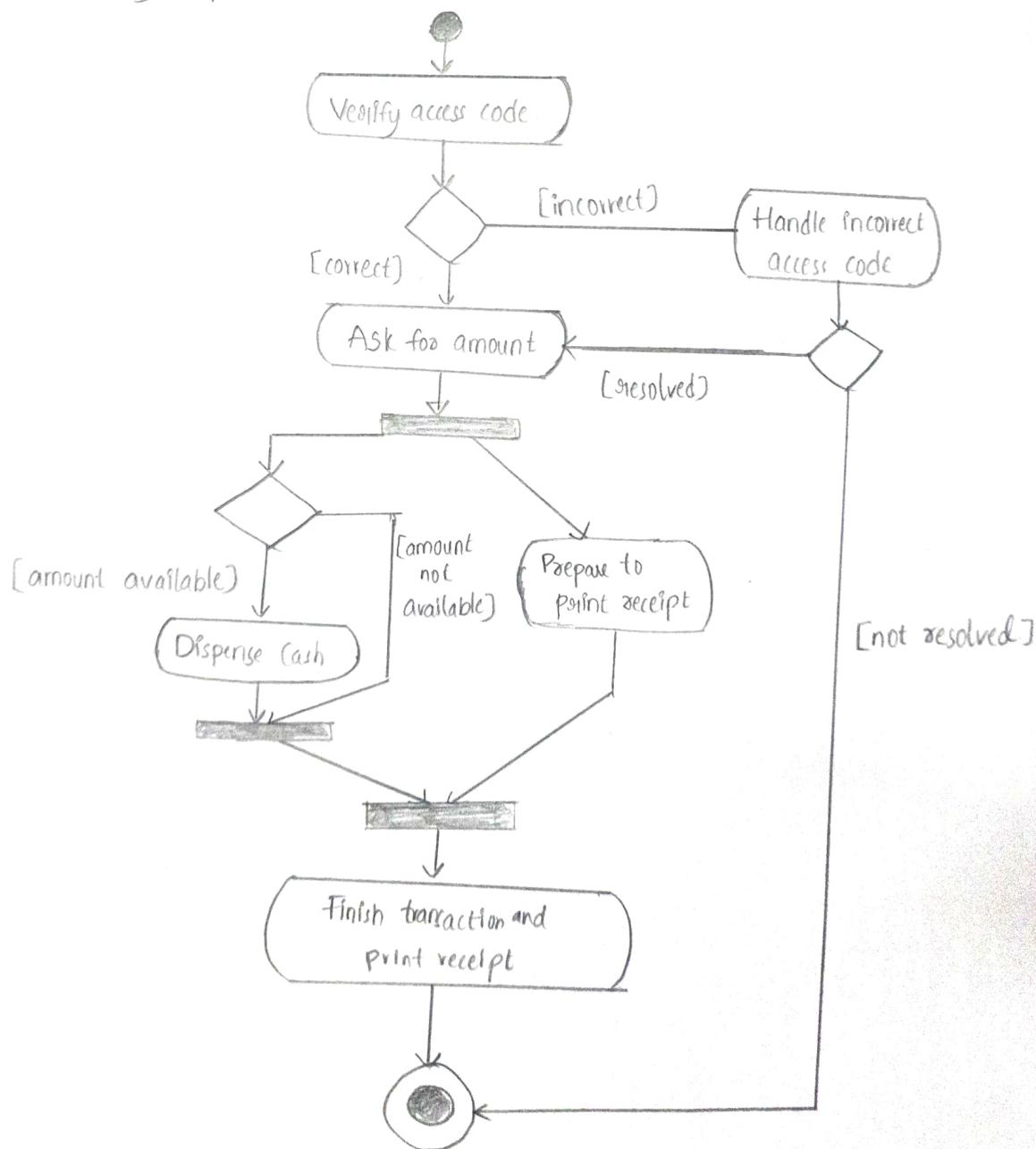
State chart diagram



Activity Diagram

Procedure:

- 1) Initial State is created.
- 2) After that it goes to the action state insert card.
- 3) Next it undergoes transition to the state enter pin.
- 4) In this way it undergoes transitions to the various states.
- 5) Use forking (separation) and joining wherever necessary.

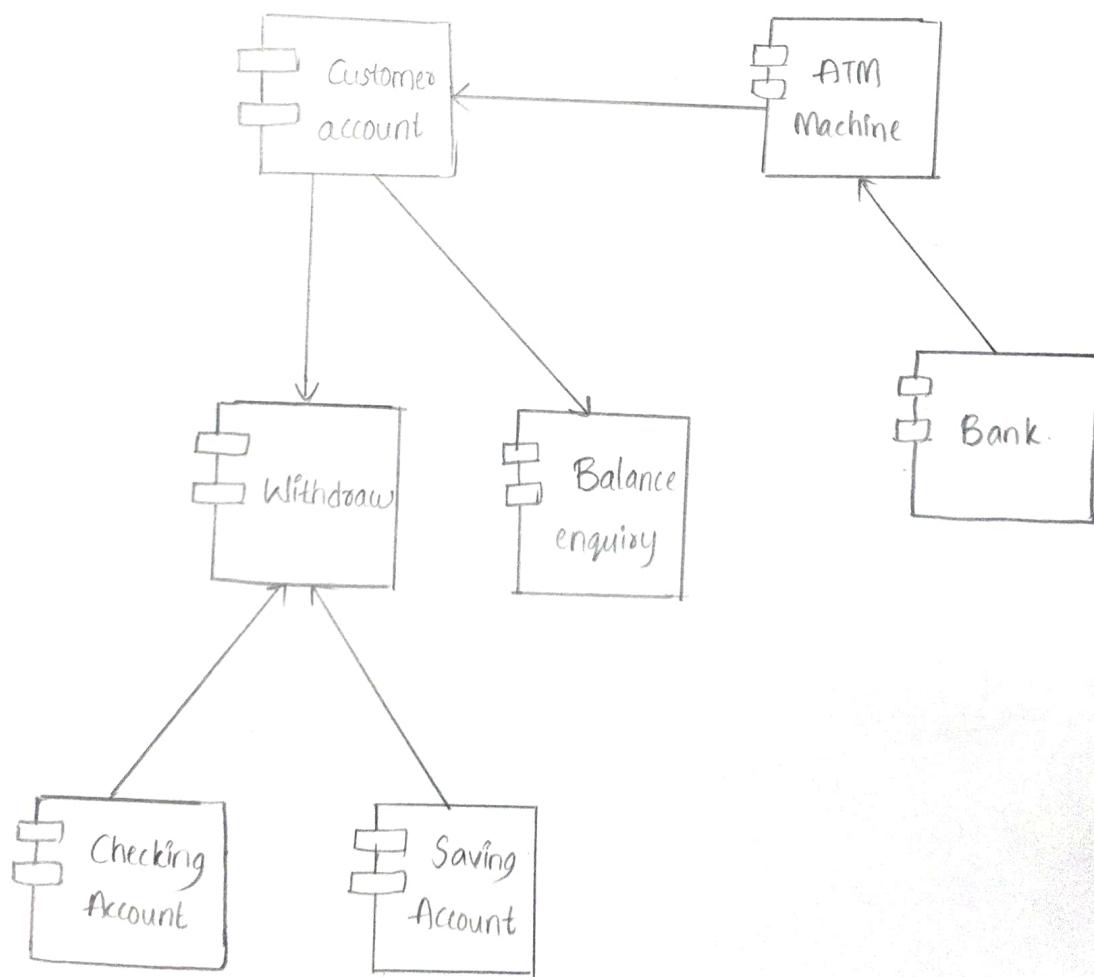


Component Diagram for the ATM System.

Procedure:

- 1) User component is created.
- 2) ATM system package is created.
- 3) In it various components such as withdraw money, deposit money, check balance, transfer money etc. are created.
- 4) Association relationship is established between user and other components.

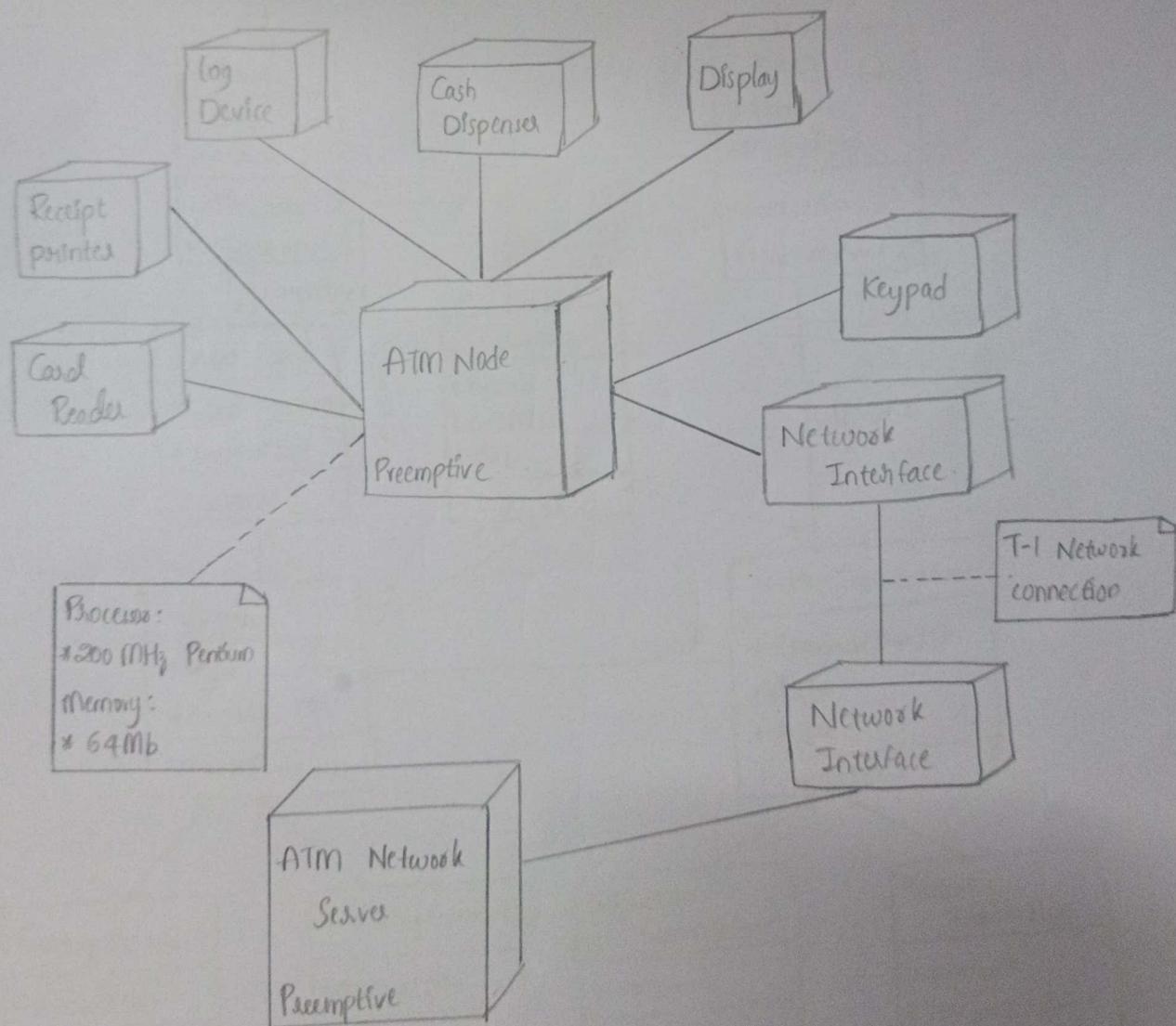
Component Diagram:



Deployment Diagram for ATM system.

Procedure:

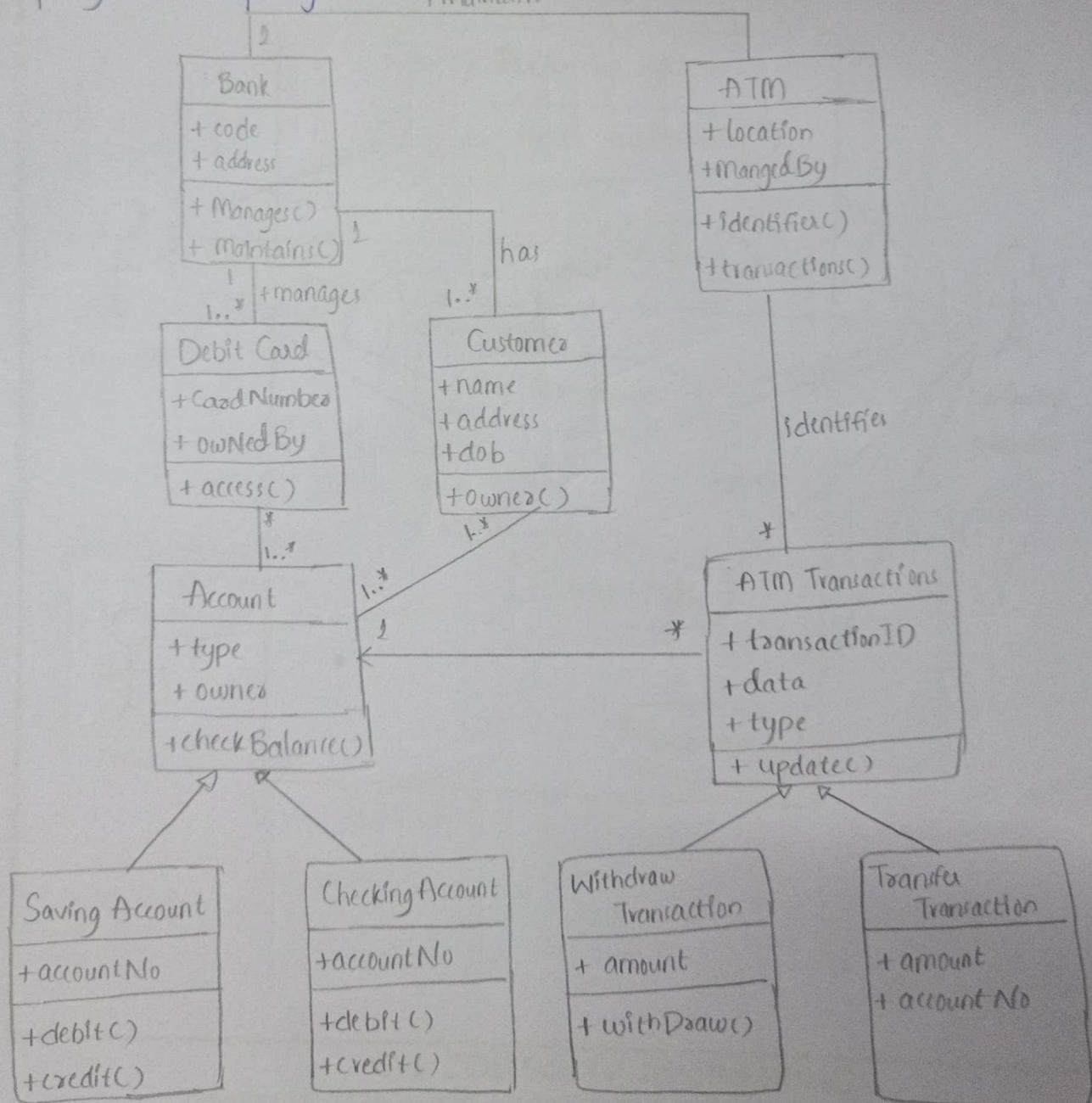
- 1) Add Nodes such as ATM Node, ATM Network Server.
- 2) Add artifacts such as Log device, Display, Keypad, Card Reader etc
- 3) Connect nodes and artifacts according to requirement and finally deploy.
- 4) Add notes (annotations) if necessary.



Class Diagram for ATM System.

Procedure:

- 1) Create classes such as Bank, ATM, Customers, Debit Card etc.
- 2) Add attributes and methods to represent the class's characteristic & behaviour.
- 3) Create appropriate relationships b/w classes as association.
- 4) Specify the multiplicity. + maintains

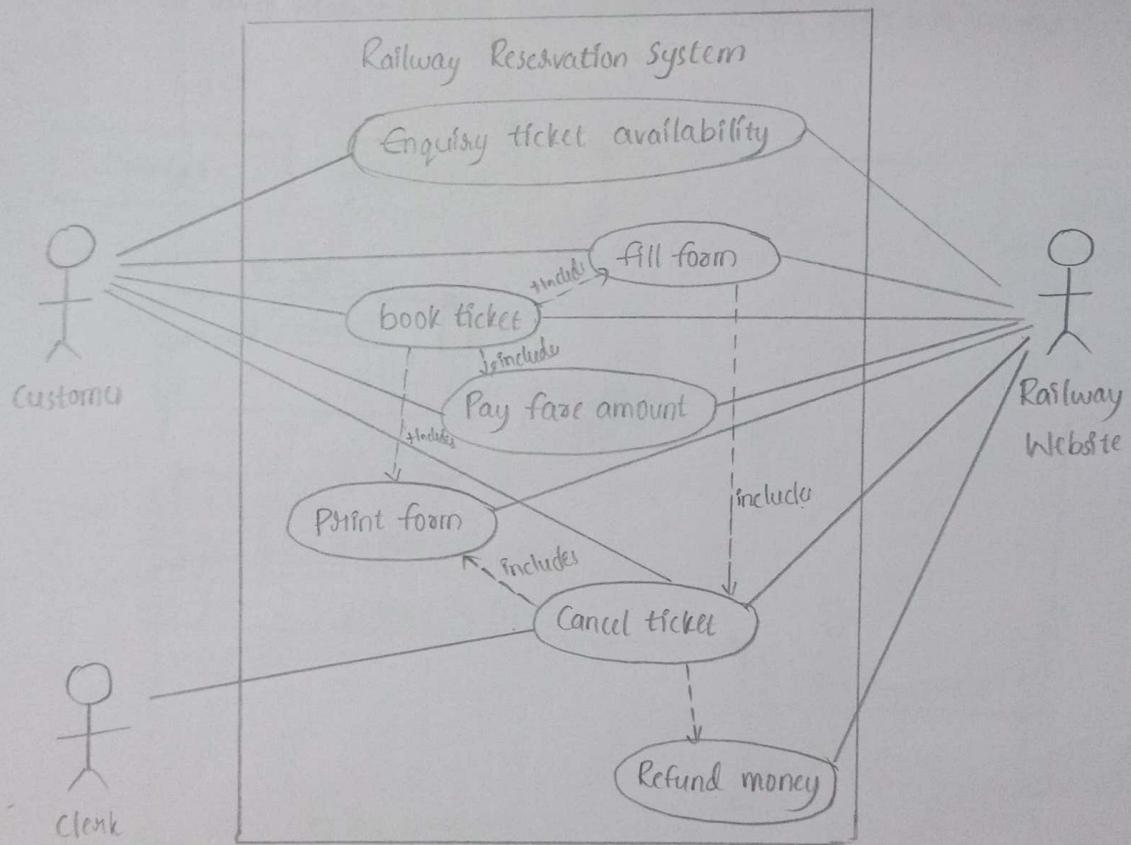


II. Online Railway Reservation System.

1. Use Case Diagram:

Procedure:

- 1) Create actors such as Passenger, Admin and Railway Reservation System -
- 2) Create use cases such as Login, Fill form etc
- 3) Using "Association" tool, connect actors to use cases.

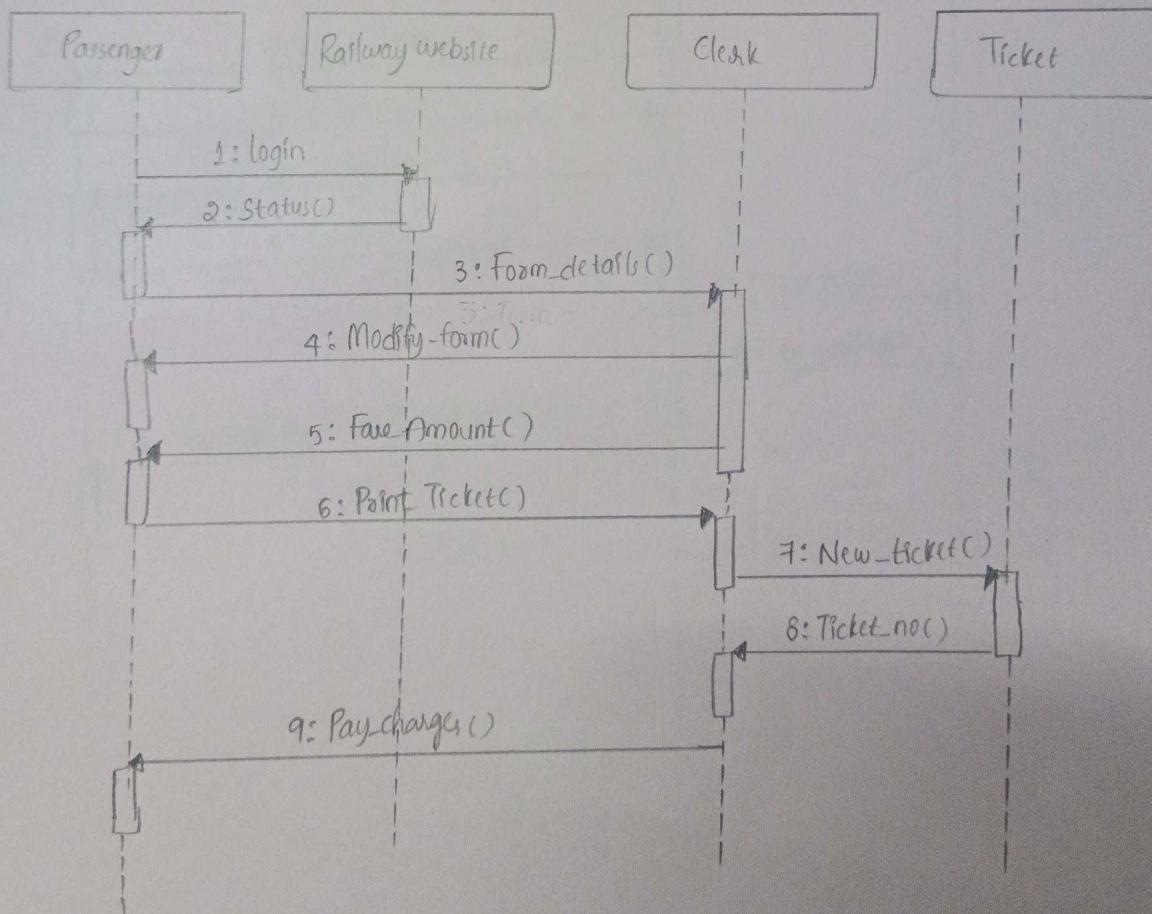


Sequence Diagram:

① For Booking Ticket

Procedures:

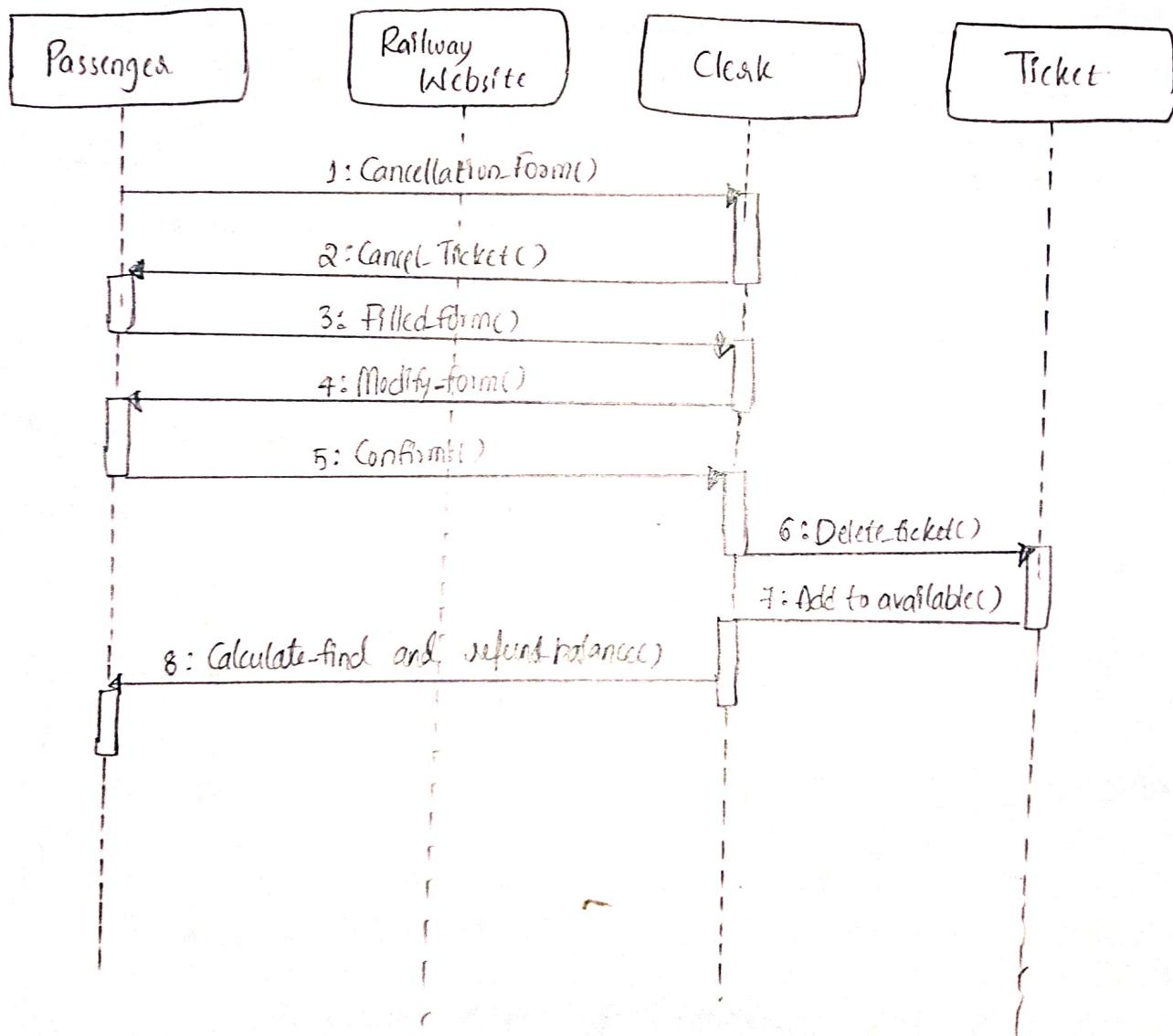
- ② Create Lifelines for the following objects: Passenger, Railway website, Clerk, Ticket.
- ③ Create messages between lifelines in an order.
- ④ Label each message with appropriate operation name and parameters.
- ⑤ If required, using 'Activation Bar' tool, indicate when object is active.



② For Cancelling Tickets [Sequence Diagram]

Procedure:

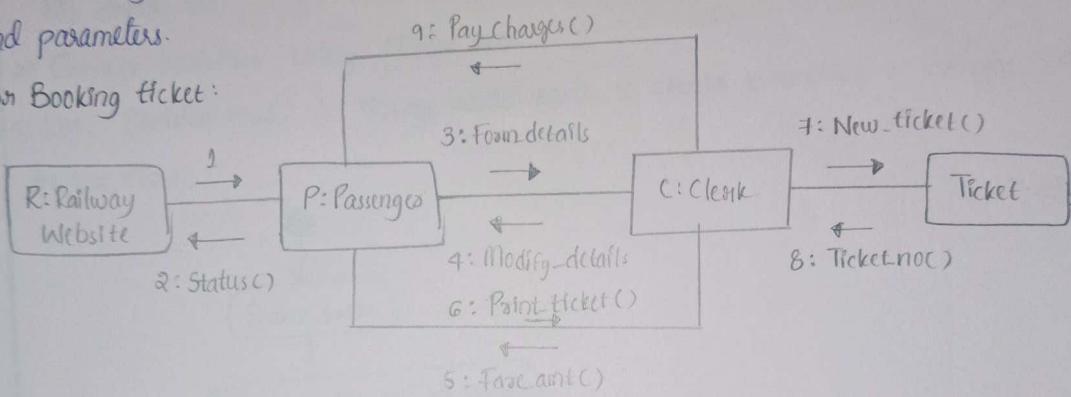
- 1) Create Lifelines for objects: Passenger, Railway website, clerk, Ticket.
- 2) Using 'Message' tool, create messages between lifelines.
- 3) Label each message with appropriate operation name and parameters.



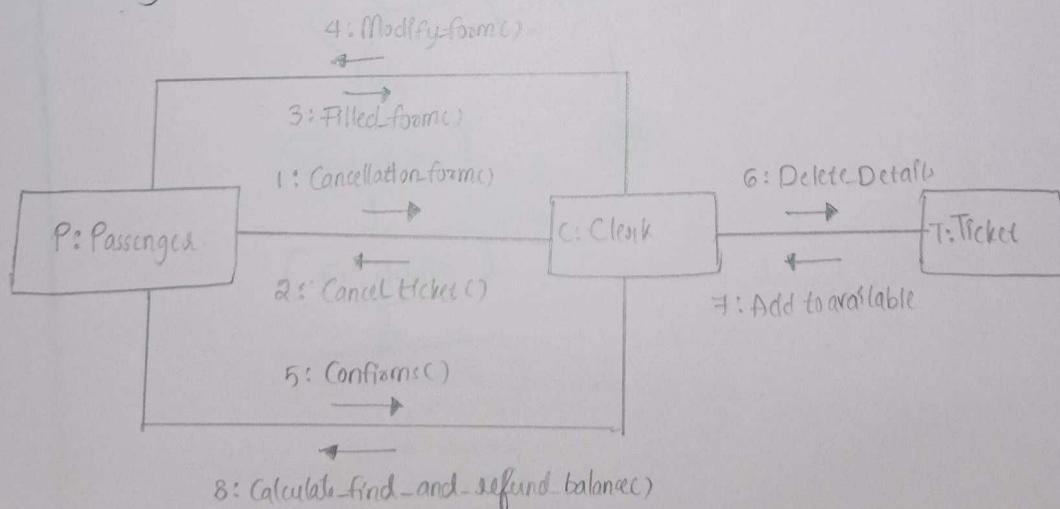
Collaboration Diagram:

Procedure:

- 1) Create required objects (ex: Passenger, Clerk, Ticket).
 - 2) Connect objects with links (associations) to represent their relationships
 - 3) Create messages and label each message with the appropriate operation name and parameters.
- ① For Booking ticket:



② For cancelling ticket:

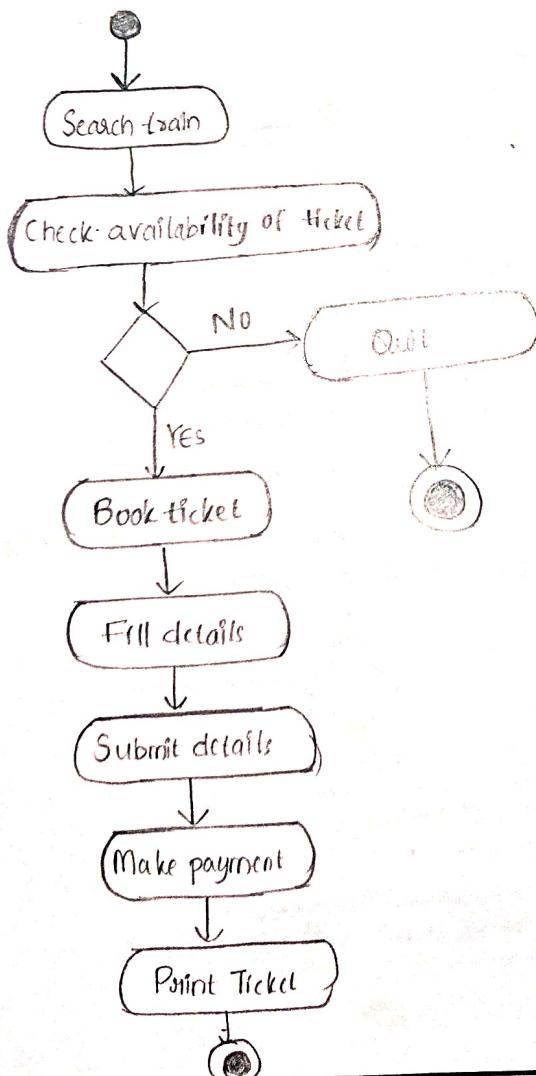


Activity Diagram:

① For Booking System Tickets

Procedure:

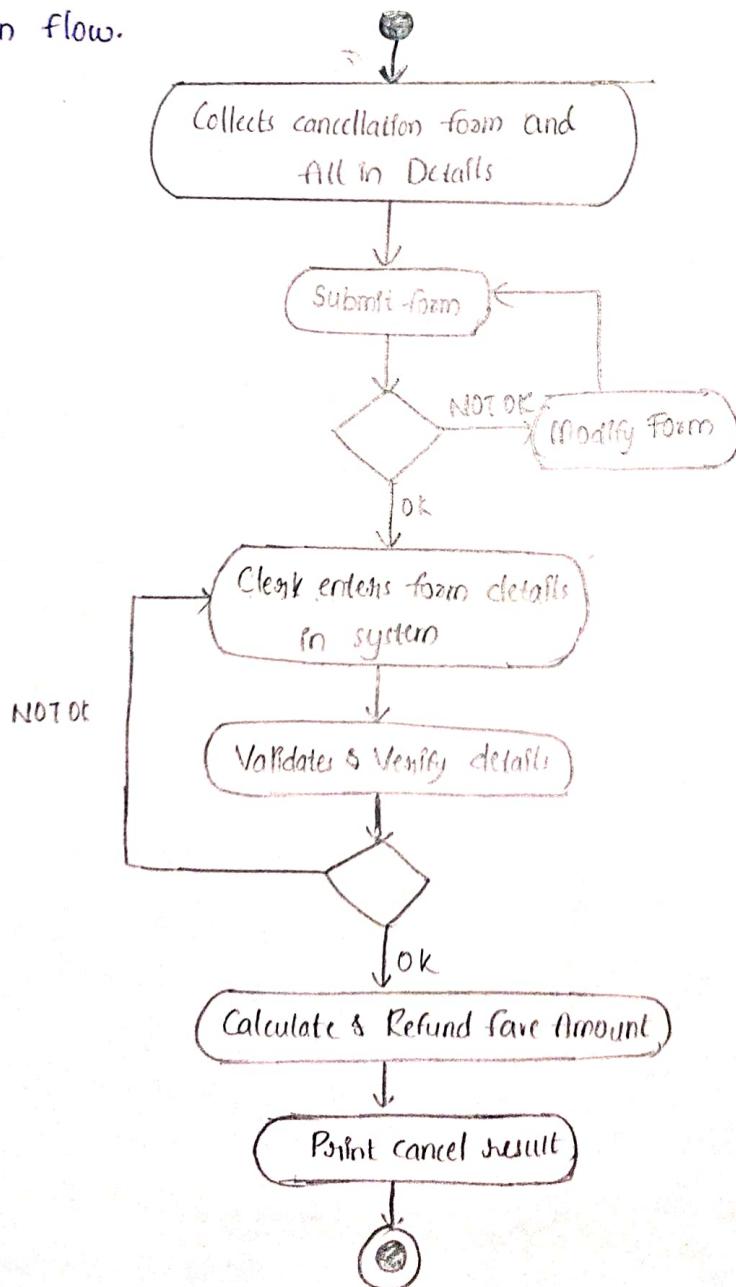
- 1) Create activities such as Search Train, Check Availability of Ticket, Book Tickets, Fill details etc..
- 2) Connect activities using 'Flow' tool.
- 3) Use 'Decision Node' & 'Merge Node' tools to create branching & merging points in the flow.



② For cancelling Tickets

Procedure:

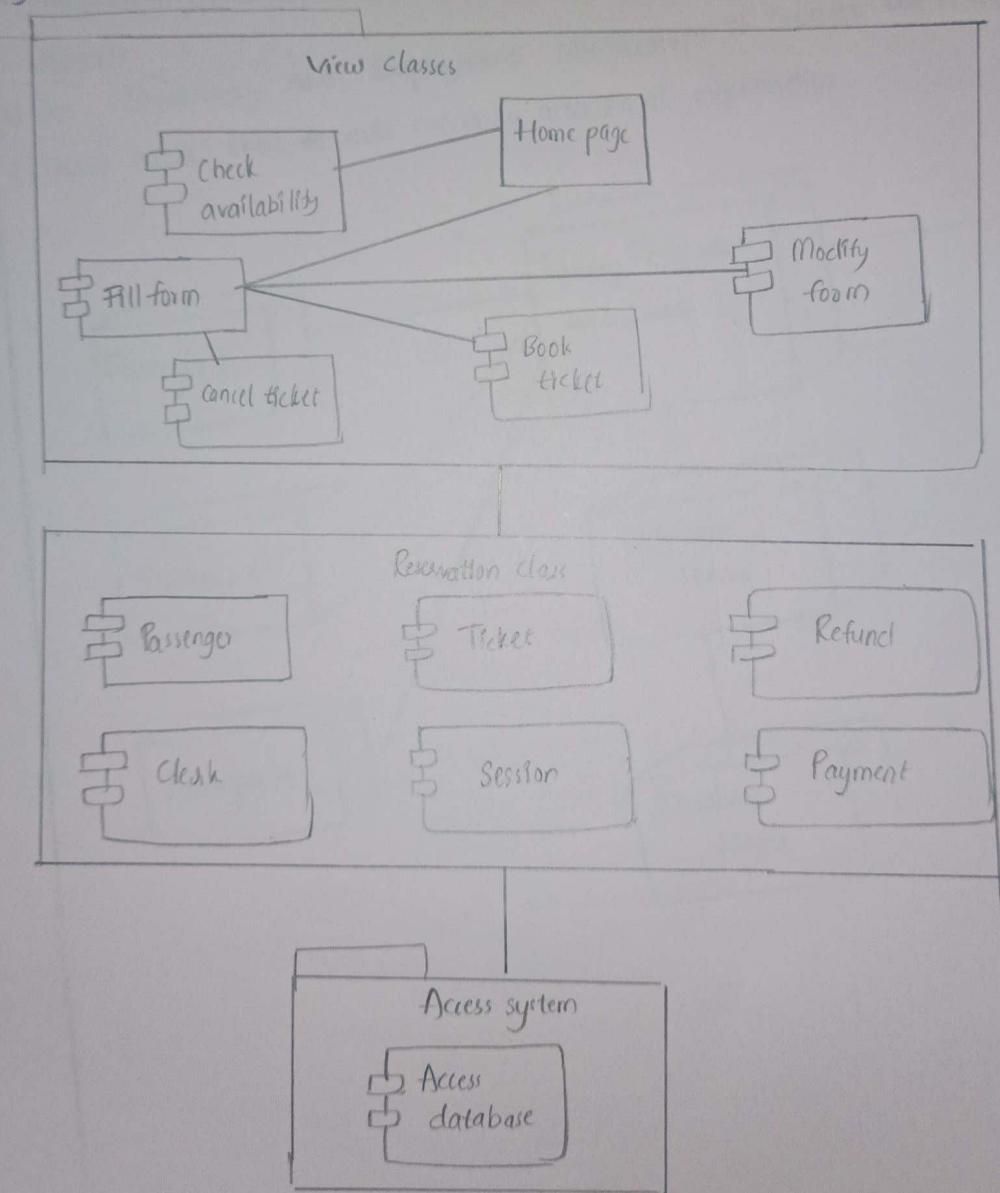
- 1) Create activities such as Collects cancellation form, Submit form, Form modify, Print result etc..
- 2) Connect activities using 'Flow' tool
- 3) Use 'Decision Node' and 'Merge Node' tools to create branching and merging points in flow.



Component Diagram:

Procedure:

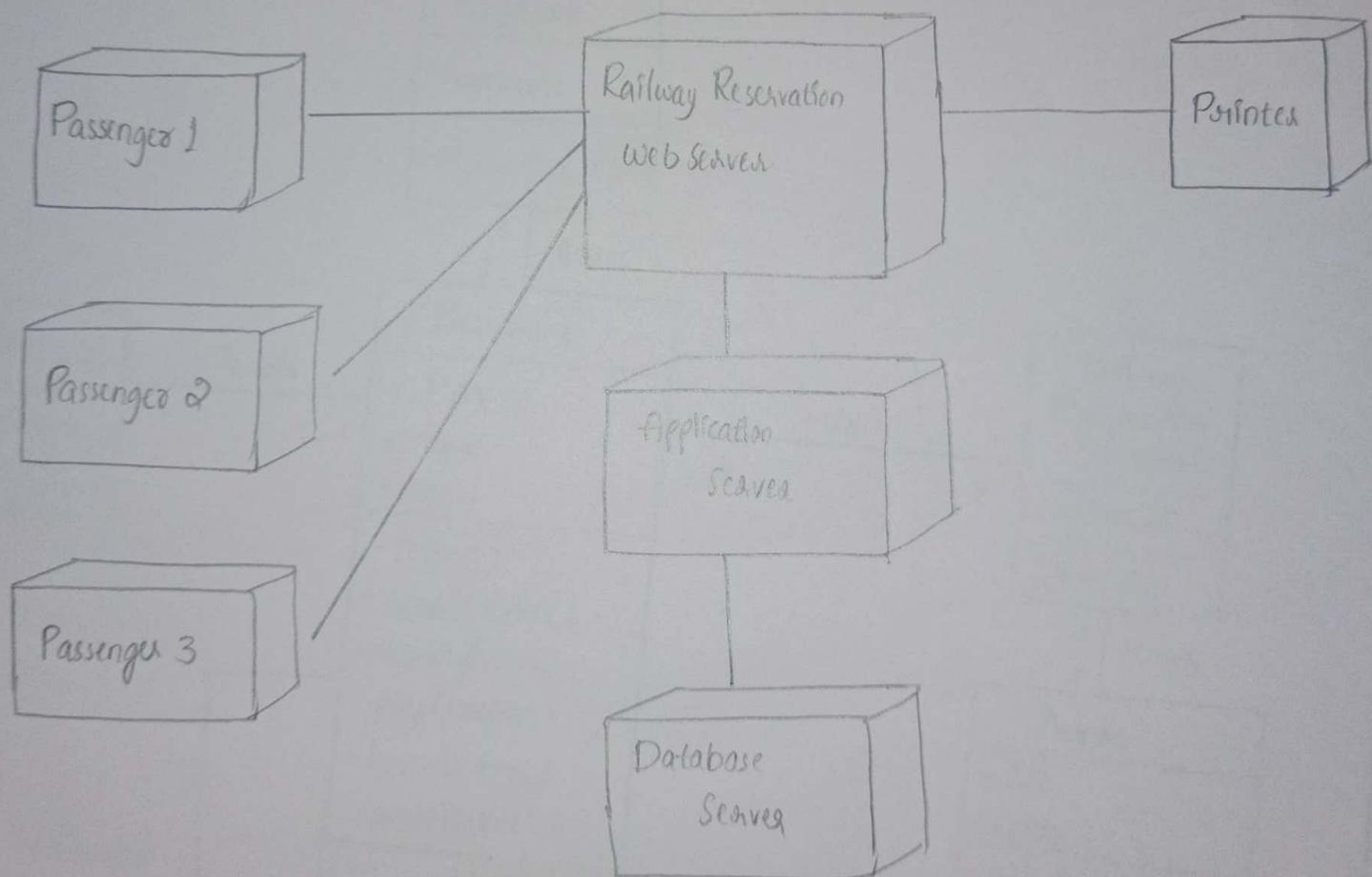
- 1) Create Components : Home page, Fill form, Ticket, Refund, Clerk, Payment etc
- 2) Using 'Dependency' and 'Interface' relationships, connect the components-



Deployment Diagram

Procedure:

- 1) Create the following Nodes: Application Server, Database Server, Printer.
- 2) Create artifacts representing specific components within the nodes such as Libraries.
- 3) Use 'Dependency' and 'Deployment' relationships to connect the nodes.
- 4) Using 'Note' tool, add necessary additional explanation.



Class Diagram for online railway reservation system.

Procedures:

- 1) Create classes such as clerk, railway system, passenger, train etc.
- 2) Add attributes and methods to represent the class's characteristics and behaviour.
- 3) Use 'Association' tool to create relationships b/w classes.
- 4) Specify multiplicity of each relationship.

Class Diagram:

