

UNIFIED MODELING LANGUAGE

Introduction

The unified modeling language (UML) is a standard language for writing software blue prints of the system.

Definition:

The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting the artifacts of a software system.

UML is a language that provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication.

- Vocabulary and rules of a language tell us how to create and real well formed models, but they don't tell you what model you should create and when should create them.

Building Blocks of the UML:

The vocabulary of the UML encompasses three kinds of building blocks:

1. Things
2. Relationships
3. Diagrams

Things are abstractions that are first-class citizens in a model;

Relationships tie these things together;

Diagrams group interesting collections of things.

Things : Things are the most important building blocks of UML. There are four kinds of things in the UML.

1. Structural things
2. Behavioural things
3. Grouping things
4. Annotational things

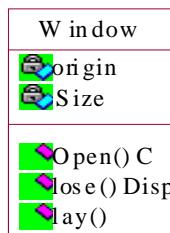
1. STRUCTURAL THINGS: Structural things are the nouns of the UML models.

These are static parts of the model, representing elements that are either conceptual or physical. There are seven kinds of Structural things.

1. Class
2. Interface
3. Collaboration
4. Use case
5. Active class
6. Component
7. Node

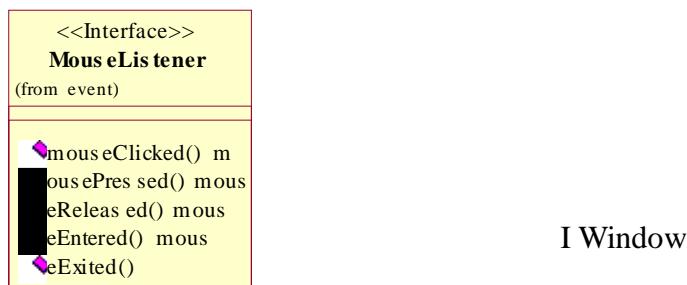
Class:

A class is a description of a set of objects that shares the common attributes, operations, relationships, and semantics. A class implements one or more interfaces. Graphically, a class is represented as a rectangle, usually including its name, attributes and operations, as shown below.

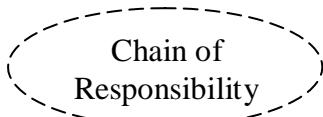


Interface:

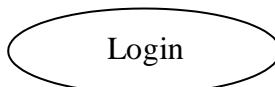
An interface is a collection of operations that specify a service of a class or component. An interface describes the externally visible behaviour of that element. Graphically the interface is rendered as a circle together with its name.



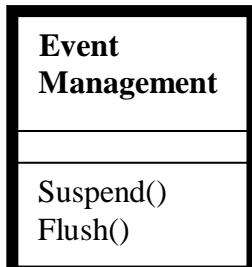
Collaboration: Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behaviour that's bigger than the sum of all the elements. Graphically, collaboration is rendered as an ellipse with dashed lines, usually including only its name.



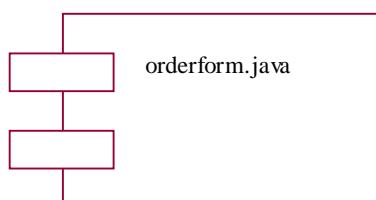
UseCase: Use case is a description of a set of sequence of actions performed by a system for a specific goal for the system. Graphically, Use Case is rendered as an ellipse with dashed lines, usually including only its name as shown below.



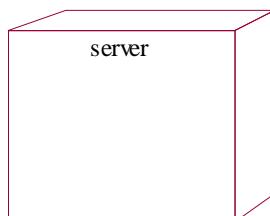
ActiveClass: An active class is a class whose objects own one or more processes or threads and therefore can initiate control activity. Graphically, an active class is rendered just like a class, but with heavy lines usually including its name, attributes and operations as shown below.



Component: Component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs, usually including only its name, as shown below.



Note: A Node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and often, processing capability. Graphically, a node is rendered as a cube, usually including only its name, as shown below.



2. BEHAVIORAL THINGS: Behavioural things are the dynamic parts of UML models. These are the verbs of a model, representing behaviour over time and space.

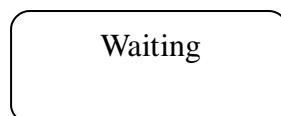
1) *Interaction:*

An interaction is a behaviour that consists of a set of messages exchanged among a set of objects(elements) within a particular context to accomplish a specific task. Graphically, a message is rendered as a direct line, almost always including the name if its operation, as shown below.



2) *State Machine:*

A state machine is a behaviour that specifies the sequence of states of an object in its life cycle. It defines the sequence of states an object goes through in response to events. Graphically, a state is rendered as a rounded rectangle usually including its name and its sub-states, if any, as shown below.

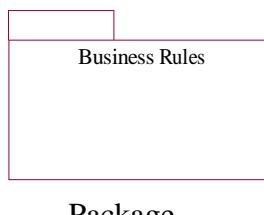


3. GROUPING THINGS:

Grouping things are the organizational parts of the UML models. These are the boxes into which a model can be decomposed. There is one primary kind of grouping thing with “package”.

Package:

A package is a general-purpose mechanism for organizing elements into groups. Package is the only one grouping thing available for gathering structural and behavioural things.



4) ANNOTATIONAL THINGS:

Annotational things are the explanatory parts of the UML models.

Annotational things can be defined as a mechanism to capture remarks, descriptions, and comments of UML model elements.

Note:

A note is simply a symbol for rendering constraints and comments attached to an element or collection of elements. Graphically a note is represented as a rectangle with dog-eared corner together, with a textual or graphical comment, as shown below.



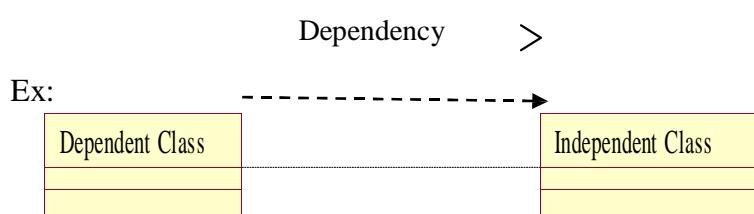
Note

RELATIONSHIPS IN THE UML:

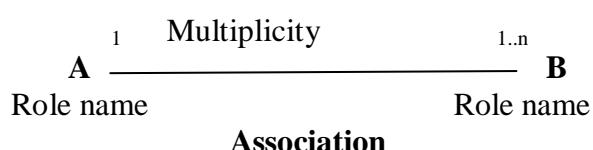
Relationship is another most important building block of UML. It shows how elements are associated with each other and this association describes the functionality of an application. There are four kinds of relationships in the UML:

1. Dependency
2. Association
3. Generalization
4. Realization

Dependency : Dependency is a relationship between two things in which change in one element also affects the other one.

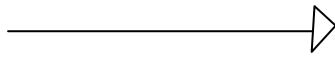


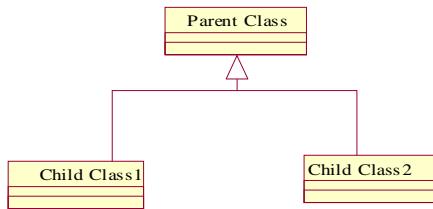
Association: Association is basically a set of links that connects elements of an UML model. It also describes how many objects are taking part in that relationship.



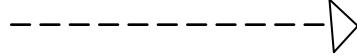
Generalization:

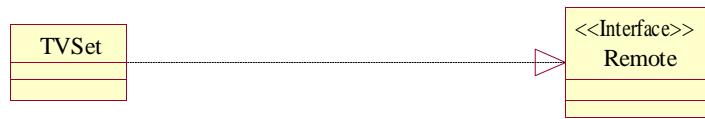
Generalization can be defined as a relationship which connects a specialized element with a generalized element. It basically describes inheritance relationship in the world of objects.

Ex: 



Realization: Realization can be defined as a relationship in which two elements are connected. One element describes some responsibility which is not implemented and the other one implements them. This relationship exists in case of interfaces.

Ex: 



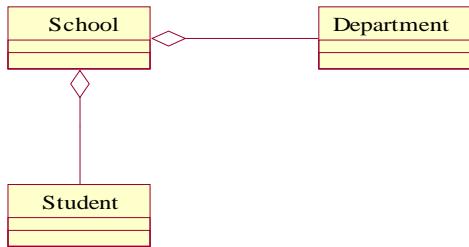
DIAGRAMS IN UML : All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. Each UML diagram is designed to let developers and customers view software system from a different perspective and in varying degrees of abstraction. UML diagrams are the ultimate output of the entire system.

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices(things) arcs (relationships).

UML includes the following nine diagrams:

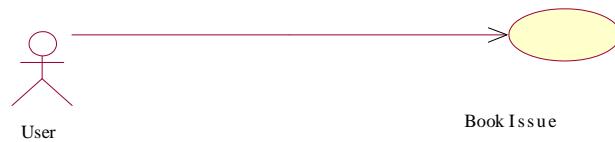
- 1) Class diagram
- 2) Object diagram
- 3) Use case diagram
- 4) Sequence diagram
- 5) Collaboration diagram
- 6) Activity diagram
- 7) State chart diagram
- 8) Deployment diagram
- 9) Component diagram

1. ClassDiagram: Class diagram is a diagram that shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams address the static design view or the static process view of the system. Graphically it is represented as follows:-



2. ObjectDiagram : Object diagram shows a set of objects and their relationships. These diagram the static design view or static process view of a system.

3. UsecaseDiagram : Use Case diagram shows a set of use cases and actors (a special kind of class) and their relationships. These diagrams address the static use case view of a system. Graphically it is represented as follows:-

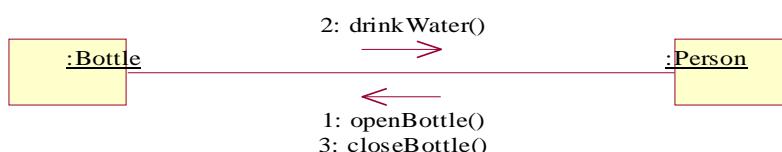


4.SequenceDiagram : Sequence diagram are interaction diagrams. This diagram emphasizes the time-ordering of messages. These diagrams address the dynamic view of a system. Sequence Diagram displays the time sequence of the objects participating in the interaction. This consists of the vertical dimension (time) and horizontal dimension (different objects).

5 CollaborationDiagram

Collaboration diagram are also interaction diagrams. These diagrams emphasize the structural organization of the objects that send and receive messages. These diagrams address the dynamic view of a system. Collaboration Diagram displays an interaction organized around the objects and their links to one another. Numbers are used to show the sequence of messages.

Graphically it is represented as follows:-

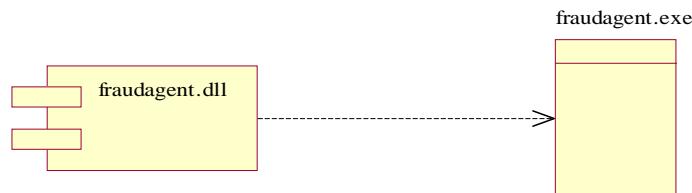


6. StatechartDiagram : State chart diagram shows a state machine, consisting of states, transitions, events and activities. These diagrams address the dynamic view of the system. State Chart diagram displays the sequences of states that an object of an interaction goes through during its life in response to received stimuli, together with its responses and actions.

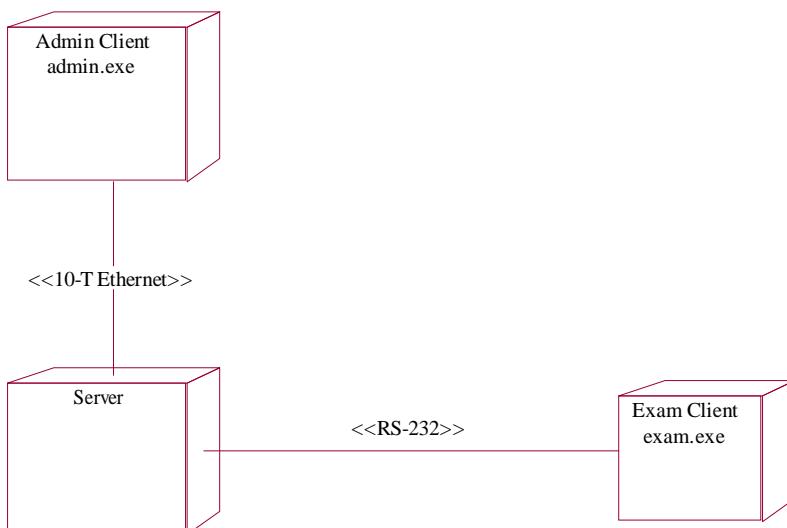
7 .ActivityDiagram : Activity diagram is a special kind of a state chart diagram that shows the flow from activity to activity within a system. These diagrams address dynamic view of a system. Activity Diagram displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states.

8. ComponentDiagram

Component diagram shows the organizations and dependencies among a set of components. These diagrams address the static implementation of view of a system. Component Diagram displays the high level packaged structure of the code itself. Dependencies among components are shown, including source code components, binary code components, and executable components. Some components exist at compile time, at link time, at run times well as at more than one time. Graphically it is represented as follows:-



9. DeploymentDiagram : Deployment diagram shows the configuration of run-time processing nodes and the components that live on them. These diagrams address the static deployment view of architecture. Deployment Diagram displays the configuration of run-time processing elements and the software components, processes, and objects that live on them. Software component instances represent run-time manifestations of code. Graphically it is represented as follows:-



EXPERIMENT : 1

AIM : TO design and implement Electronic Cash Counter system through Class Diagram

HARDWARE / SOFTWARE TOOLS : Star UML / UML Graph / Top cased.

1.0 PROBLEM DEFINITION

To develop an automated banking system, which is required to perform the following functions?

- 1.1 The customer logs into the system using card number and pin number.
The system checks for validation.
- 1.2 The system queries the customer for the type of account either fixed deposit or credit account. After getting the type of account the system shows the balance left.
- 1.3 The system queries the customer for the transaction type either withdrawal or deposit and the required amount. The user enters the amount and the transaction if carries out

2.0 SRS DOCUMENT FOR AUTOMATED BANKING SYSTEM

2.1 INTRODUCTION

2.1.1 Purpose

- 2.1.1.1 The purpose of this SRS is to describe the requirements involved in developing an Automated Banking System (ABS).
- 2.1.1.2 The intended audience is any person who wants
 - 2.1.1.2.1 To create account.
 - 2.1.1.2.2 To withdraw or deposit either in fixed deposit or credit account.

2.1.2 Scope

- 2.1.2.1 The product is titled Automated Banking System (ABS).
- 2.1.2.2 The product will perform the following tasks
 - 2.1.2.2.1 Allow a new user to create an account, either fixed or credit account by entering the details and by depositing an initial amount.
 - 2.1.2.2.2 Allow the existing user to enter his account details like card number, pin number and account type to view his balance.
 - 2.1.2.2.3 Allow the existing user to deposit an amount by entering the amount to be deposited after the balance had been viewed.
 - 2.1.2.2.4 Allow the existing user to withdraw an amount by entering the amount to be withdrawn after the balance had been viewed.
 - 2.1.2.2.5 The primary benefits expected of the system are: user friendly, continuous connectivity without failure, fault tolerant and involves lesser manpower.

2.1.3 Definitions, Acronyms and Abbreviations

2.1.3.1 ABS: Automated Banking System.

2.1.4 References

2.1.4.1 IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description.

2.1.4.2 IEEE Software Requirements Specifications Template
http://www.cas.master.ca/~carette/SE3M04/2003/files/srs_template.doc

2.1.5 Overview

2.1.5.1 The SRS contains an analysis of the requirements necessary to help easy design.

2.1.5.2 The overall description provides interface requirements for the Banking system, product perspective, hardware interfaces, software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.

2.1.5.3 Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect banking system in any fashion.

2.2 THE OVERALL DESCRIPTION

2.2.1 Product perspective

2.2.1.1 Hardware interfaces

2.2.1.1.1 Hard disk: The database connectivity requires a hardware configuration that is on-line. This makes it necessary to have a fast database system (such as any RDBMS) running on high rpm hard-disk permitting complete data redundancy and backup systems to support the primary goal of reliability.

2.2.1.1.2 The system must interface with the standard output device, keyboard and mouse to interact with this software.

2.2.1.2 Software interfaces

2.2.1.2.1 Back End: MS Access 2007

2.2.1.2.2 Front End: Microsoft Visual Basic 6.0

2.2.1.3 Operations

2.2.1.3.1 The user can create a new account.

2.2.1.3.2 The existing user can access his account and view his balance by entering his details.

2.2.1.3.2 The user can deposit and withdraw money from his account.

2.2.2 Product Functions

2.2.2.1 Creating a New Account

The user should provide his personal details to facilitate the bank clerk to create a new account. The user should provide:

- 2.2.2.1.1 Customer Name.
- 2.2.2.1.2 Customer address.
- 2.2.2.1.3 Required account type.
- 2.2.2.1.4 Pin Number.
- 2.2.2.1.5 Initial deposit.

2.2.2.2 Operating with created account

The user should be able to operate with his new account after:

- 2.2.2.2.1 Entering card number.
- 2.2.2.2.2 Entering pin number.
- 2.2.2.2.3 Entering the account type, transaction type and amount involved in the transaction.

2.2.3 User characteristics

- 2.2.3.1 The intended users of this software need not have specific knowledge as to what is the internal operation of the system. Thus the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user
- 2.2.3.2 The Product is absolutely user friendly, so the intended users can be the naïve users.
- 2.2.3.3 The product does not expect the user to possess any technical background. Any person who knows to use the mouse and the keyboard can successfully use this product.

2.2.4 Constraints:

- 2.2.4.1 At the time of creating the new account, each user gives a pin number and is provided with a unique card number that must be used for further transactions. Hence the user is required to remember or store these numbers carefully.
- 2.2.4.2 At the time of creating the new account, the initial deposit should not be less than the specified amount.

2.3 SPECIFIC REQUIREMENTS

2.3.1 Logical Database Requirements

- 2.3.1.1 The system should contain databases that include all the necessary information for the product to function according to the requirements. These include relations such as Customer Details and Account Details.
- 2.3.1.2 Customer details refer to the customer's name and address. Account details of the customer include the card number, account type, transaction type and the pin number given by the user to be used at the time of the transaction at the bank.

2.4 FRONT – END DESCRIPTION

The front end for the Automated Banking System (ABS) is designed using Microsoft Visual Basic 6.0. The front end contains a user-friendly interface. The first form contains a welcome screen that provides an option for the user to either create a new account or to operate through an existing account. The “create

account" module contains a provision to create a new account after collecting the customer name, address and other details. The card number and pin number of the user is obtained every time there is a transaction. The user is requested to select the required type of transaction and the amount involved in the transaction.

2.5 BACK – END DESCRIPTION

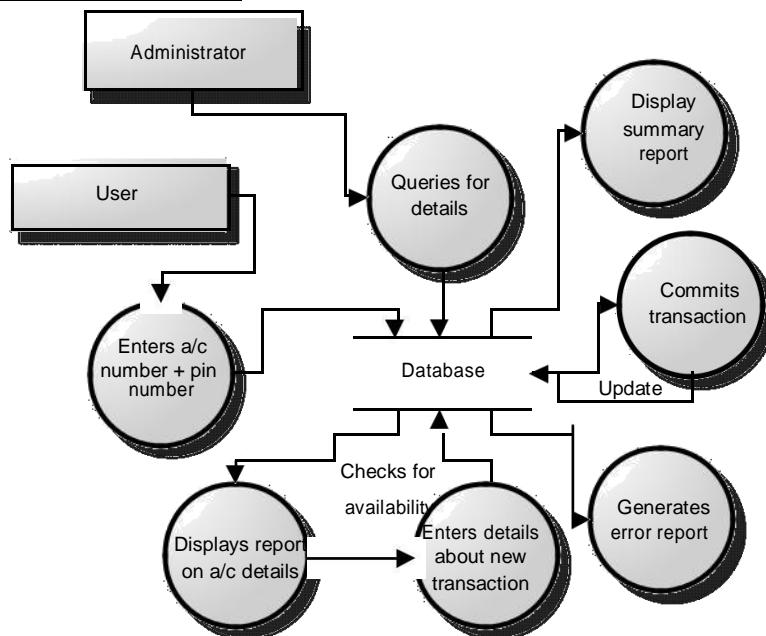
The Automated Banking System (ABS) database contains only one table. It correlates a unique card number, customer name, account type, pin number and the balance.

2.6 DATA STRUCTURES

2.6.1. ACCOUNT DETAILS

| FIELD NAME | TYP E | CONSTRAINTS |
|------------|------------|-------------|
| NAME | TEXT | |
| ACC_NO | AUTONUMBER | NOT NULL |
| AGE | NUMBER | |
| GENDER | TEXT | |
| EMAIL | TEXT | |
| PHONE_NO | NUMBER | |
| PASSWORD | TEXT | |
| BALANCE | NUMBER | |

2.7 DATA FLOW DIAGRAM:



PROCEDURE:

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction. The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community. So the purpose of the class diagram can be summarized as:

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

Class diagrams commonly contain the following things

- Classes
- Interfaces
- Collaborations
- Dependency, generalization and association relationships

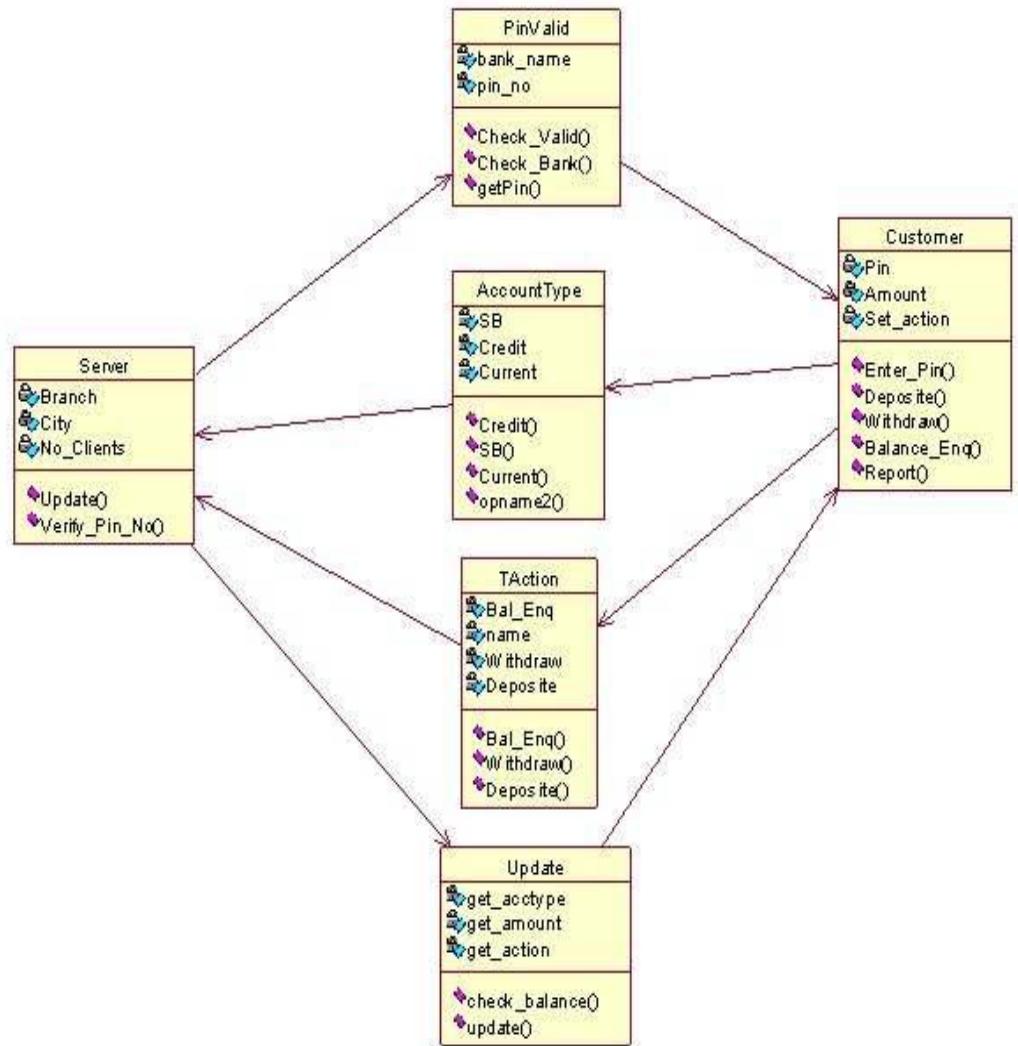
Steps :

Step1: First Classes are created.

Step2: Named as Pin Valid, Account Type, Transaction, Update, Server, Customer classes are created.

Step3: Appropriate relationships are provided between them as association.

DIAGRAM:



Inferences:

1. Understand the concept of classes
2. Identify classes and attributes and operations for a class
3. Model the class diagram for the system

Applications:

Online transaction
Online banking

B) NAME OF EXPERIMENT: Use case diagram for ATM System.

AIM: To design and implement ATM System through Use case Diagram.

Purpose:

The purpose of use case diagram is to capture the dynamic aspect of a system. Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified.

So in brief, the purposes of use case diagrams can be as follows:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interacting among the requirements are actors.

Procedure:

Step1: First an Actor is Created and named as User/Customer.

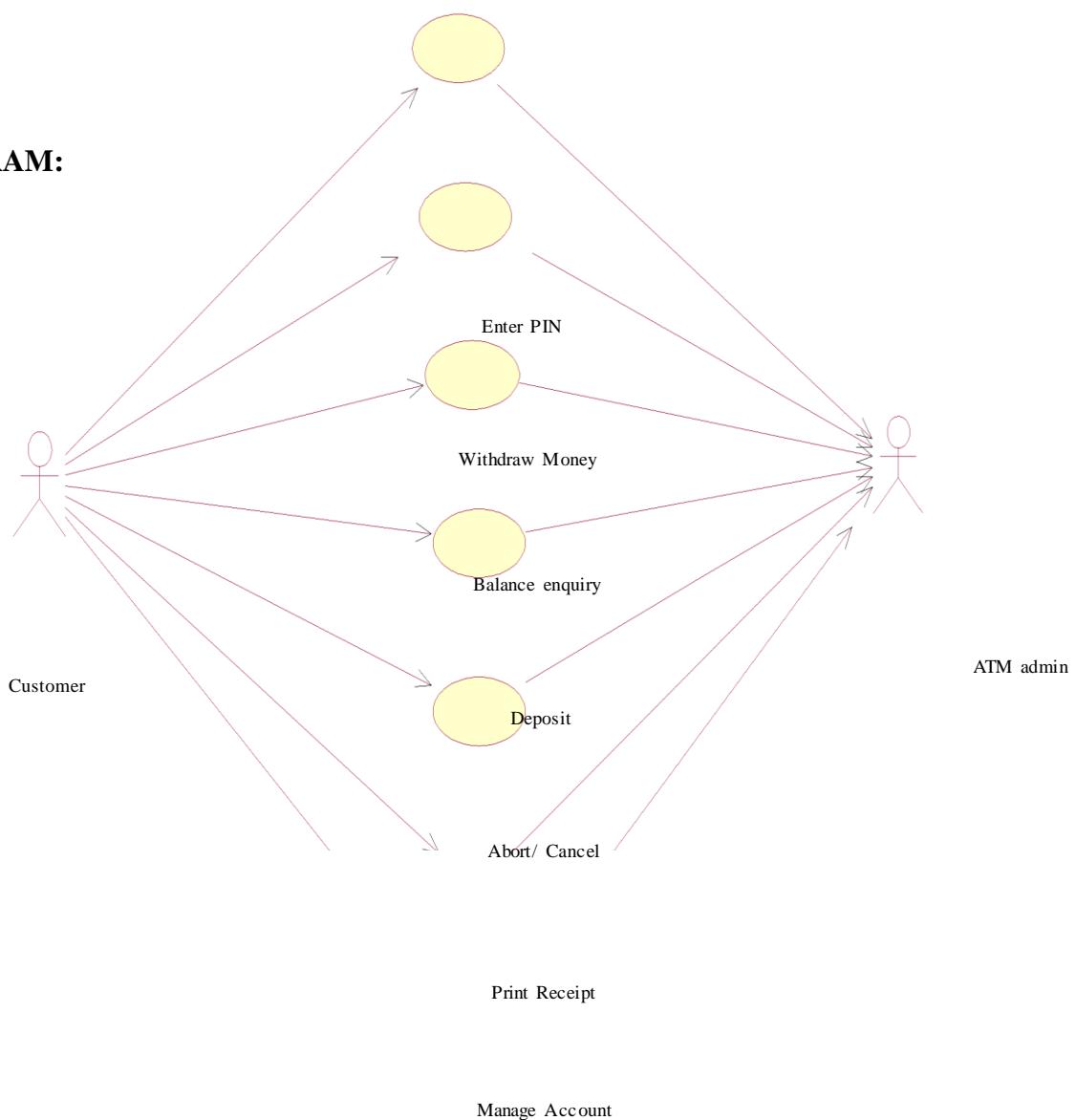
Step2: Secondly a system is created for ATM.

Step3: A use case Enter PIN, Withdraw money is created and connected with user as association relationship.

Step4: Similarly various use cases like Deposit money, Balance Enquiry, Manage Account etc. are created and appropriate relationships are associated with each of them.

Inferences:

1. Identification of use cases.
2. Identification of actors.

DIAGRAM:**Withdrawal UseCase**

A withdrawal transaction asks the customer to choose a type of account to withdraw from (e.g. checking) from a menu of possible accounts, and to choose an amount from a menu of possible amounts. The system verifies that it has sufficient money on hand to satisfy the request before sending the transaction to the bank. (If not, the customer is informed and asked to enter a different amount.) If the transaction is approved by the bank, the appropriate amount of cash is dispensed by the machine before it issues a receipt. A withdrawal transaction can be cancelled by the customer pressing the Cancel key any time prior to choosing the amount.

INTERACTION DIAGRAMS

We have two types of interaction diagrams in UML. One is sequence diagram and the other is a collaboration diagram. The sequence diagram captures the time sequence of message flow from one object to another and the collaboration diagram describes the organization of objects in a system taking part in the message flow.

So the following things are to be identified clearly before drawing the interaction diagram:

1. Objects taking part in the interaction.
2. Message flows among the objects.
3. The sequence in which the messages are flowing.
4. Object organization.

Purpose:

1. To capture dynamic behaviour of a system.
2. To describe the message flow in the system.
3. To describe structural organization of the objects.
4. To describe interaction among objects.

Contents of a Sequence Diagram

Objects
Focus of control
Messages
Life line

Contents of a Collaboration Diagram

Objects
Links
Messages

C) NAME OF EXPERIMENT: Sequence diagram for ATM System.

AIM: To design and implement ATM System through Sequence Diagram.

Procedure:-

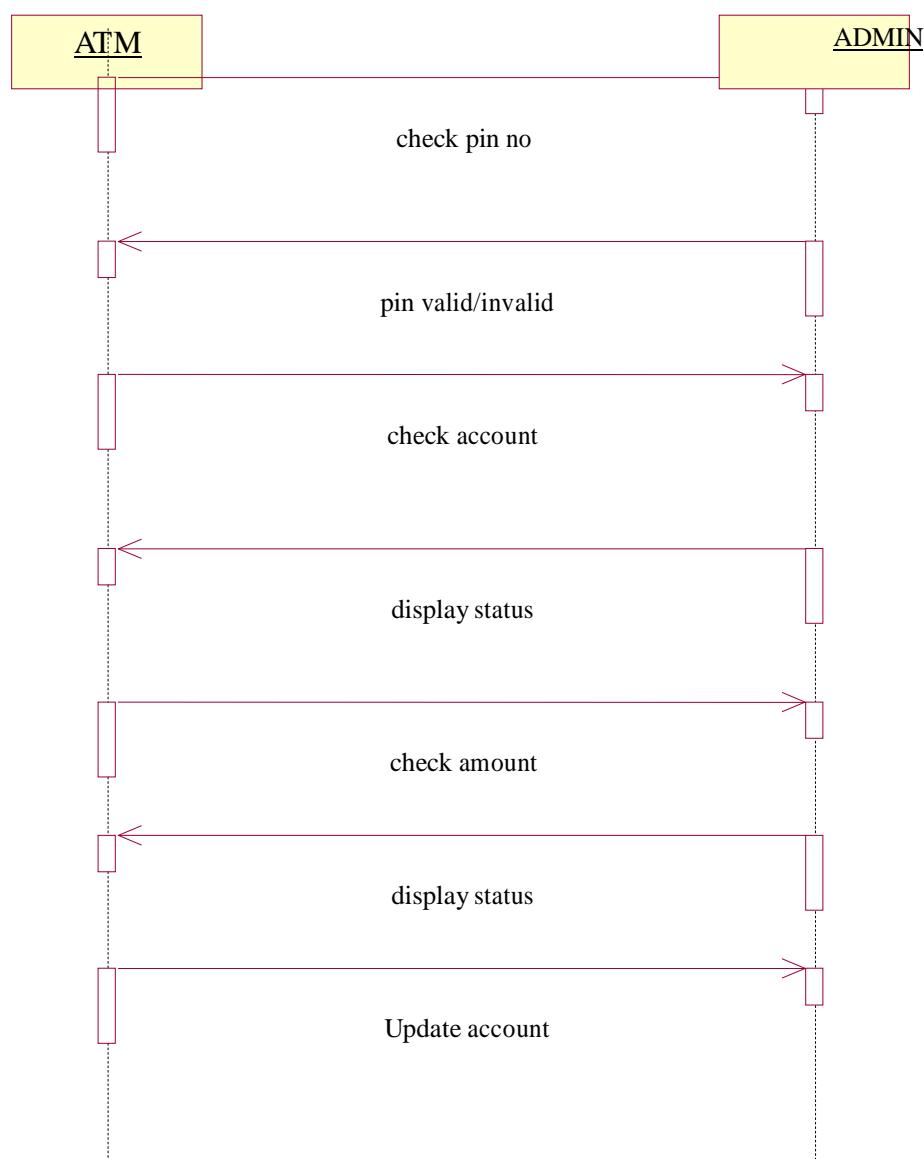
Step1: First An actor is created and named as user.

Step2: Secondly an object is created for Atm.

Step3: Timelines and lifelines are created automatically for them.

Step4: In sequence diagram interaction is done through time ordering of messages. So appropriate messages are passed between user and ATM is as shown in the figure.

DIAGRAM:



D) NAME OF EXPERIMENT: collaboration for ATM System.

AIM: To design and implement ATM System through Collaboration diagram.

Procedure:-

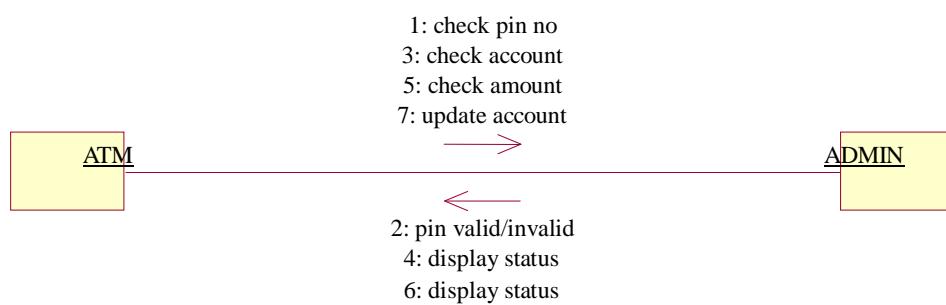
Step1: First an actor is created and named as user.

Step2: Secondly an object is created for ATM.

Step3: In collaboration diagram interaction is done through organization.

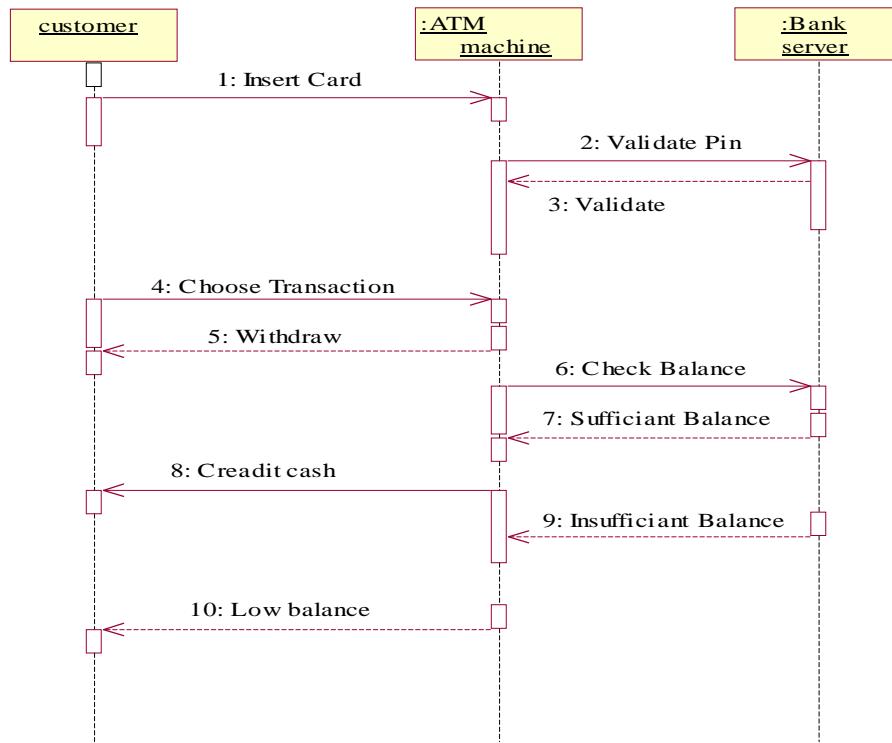
Step4: So appropriate messages are passed between user and ATM as shown in the figure.

DIAGRAM:

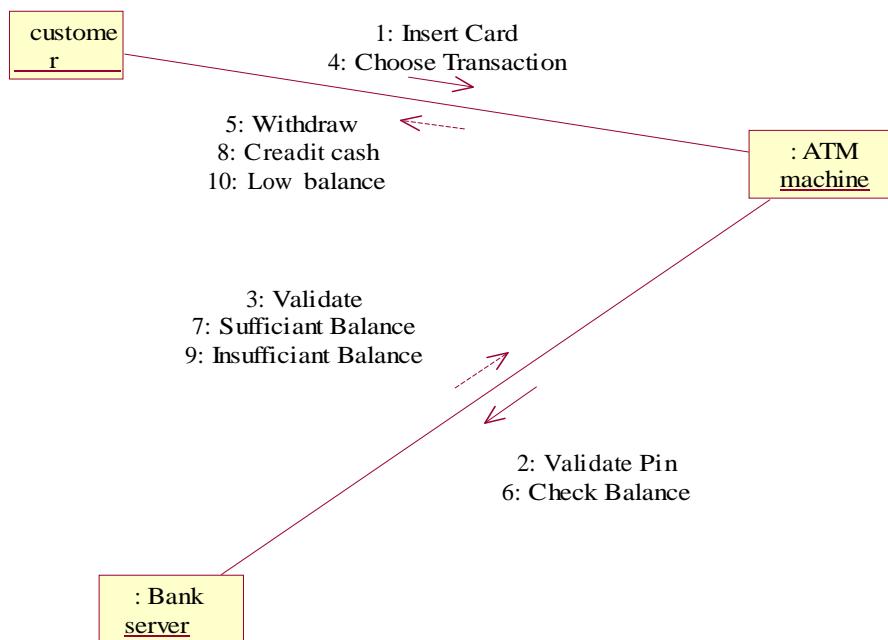


WITHDRAWL:

SEQUENCE DIAGRAM

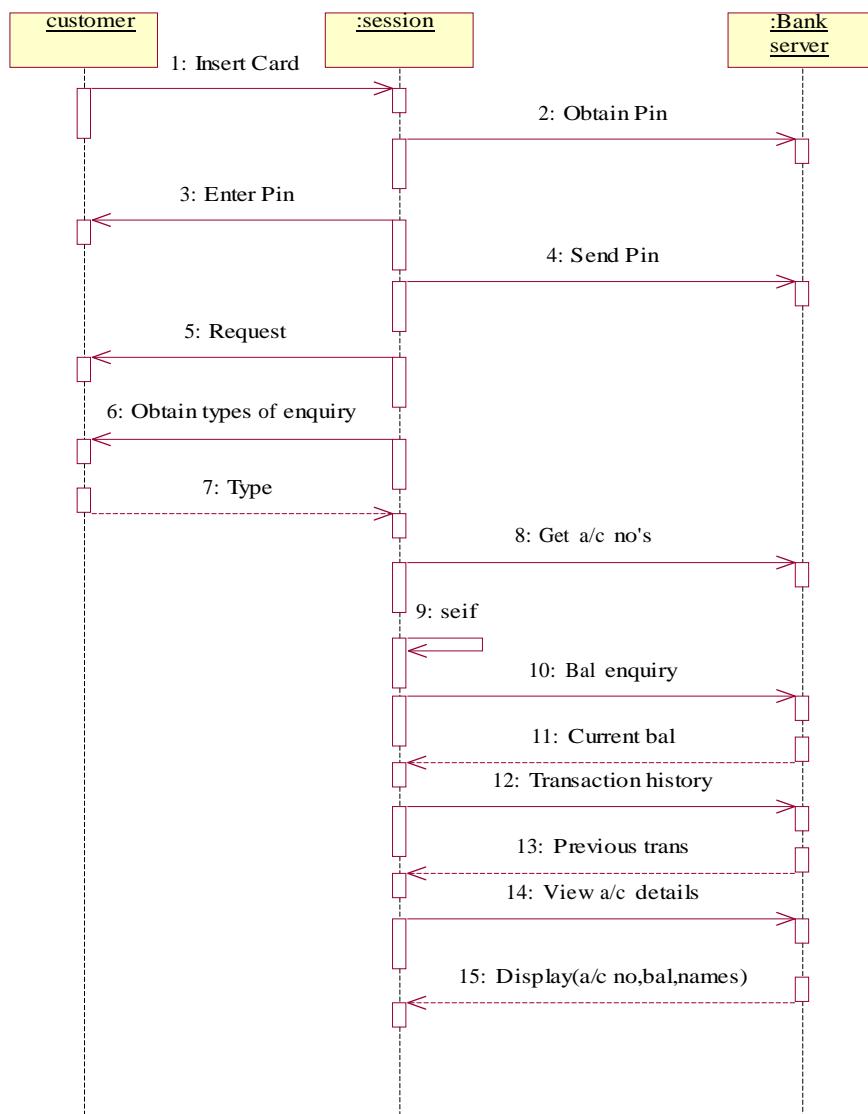


COLLABORATION DIAGRAM

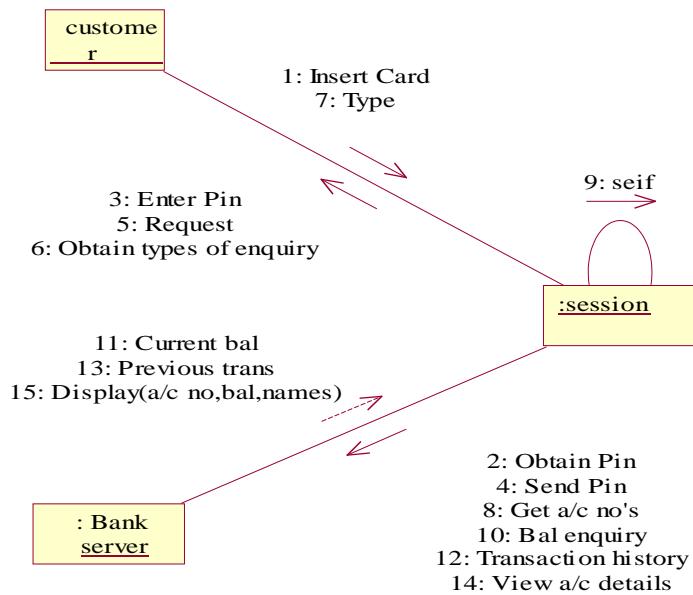


ENQUIRY:

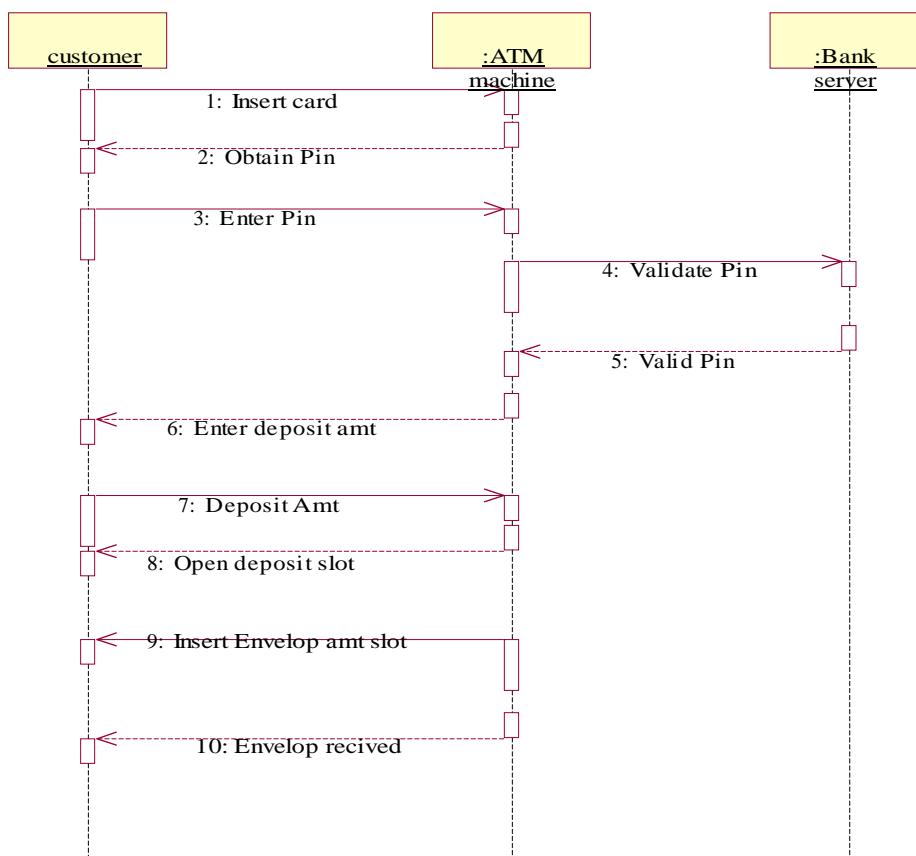
SEQUENCE DIAGRAM:



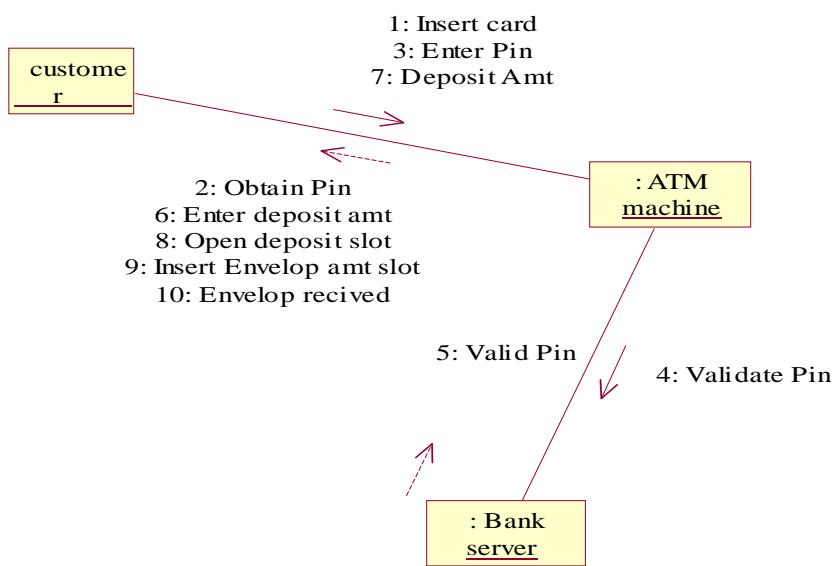
COLLABARATION DIAGRAM:

DEPOSITUseCase:

SEQUENCE DIAGRAM:



COLLABARATION DIAGRAM:

STATECHART Diagram :

State Chart diagram is used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. But the main purpose is to model reactive system.

Contents

- Simply state and composite states
- Transitions, including events and actions

E) NAME OF EXPERIMENT: State chart diagram for ATM System.

AIM: To design and implement ATM System through State Chart diagram.

Purpose:

Following are the main purposes of using State chart diagrams:

1. To model dynamic aspect of a system.
2. To model life time of a reactive system.
3. To describe different states of an object during its life time.
4. Defines a state machine to model states of an object.

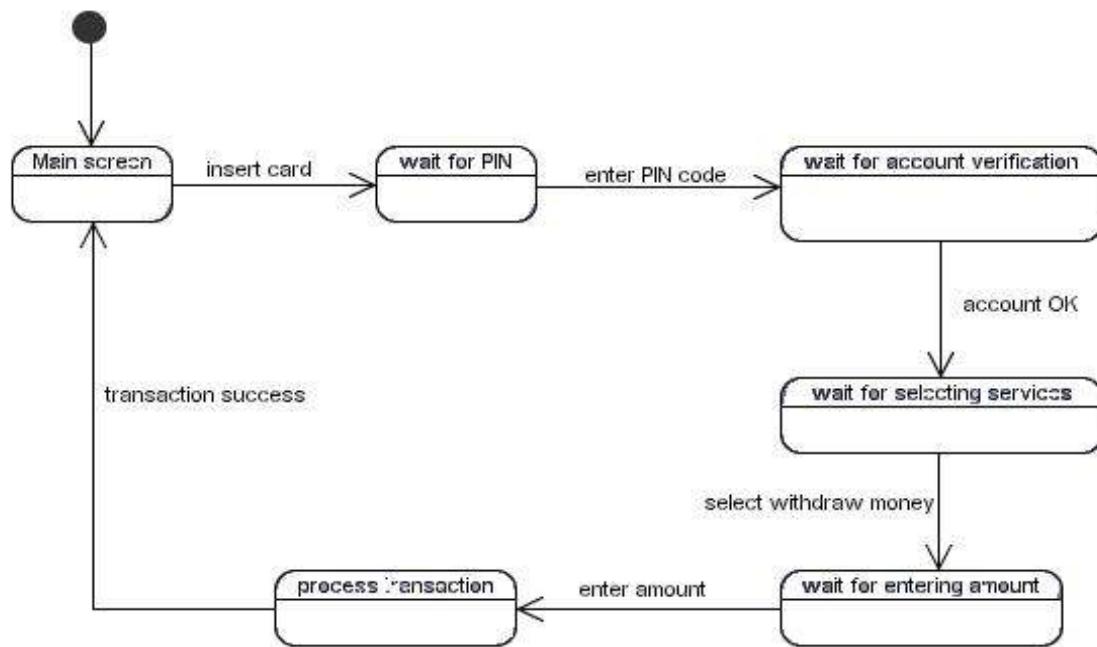
Procedure:-

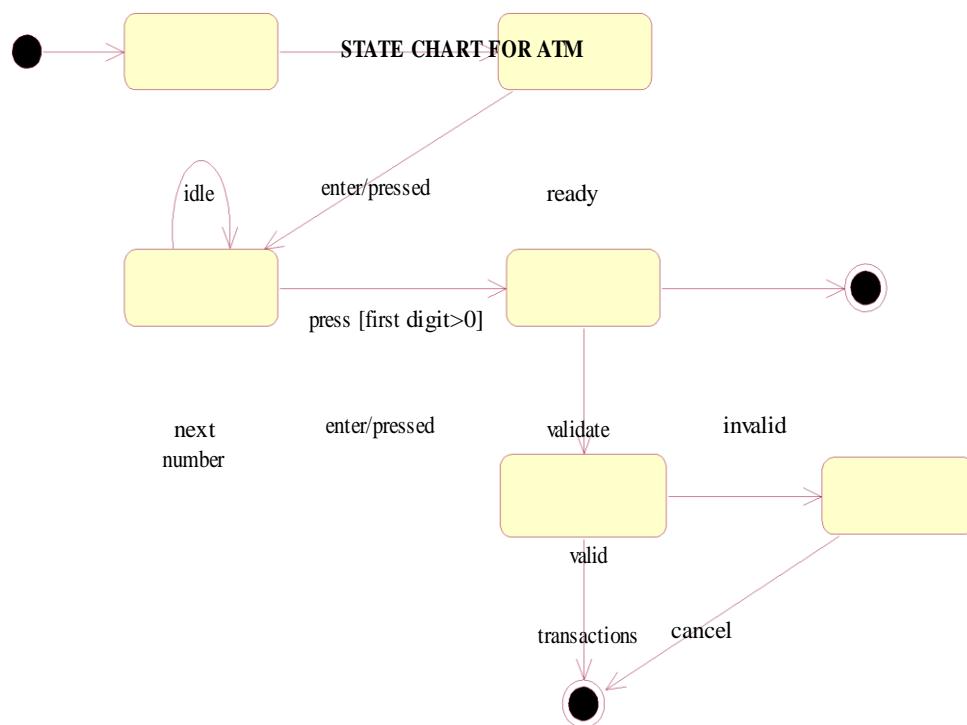
Step1: First after initial state control undergoes transition to ATM screen.

Step2: After inserting card it goes to the state wait for pin.

Step3: After entering pin it goes to the state account verification.

Step4: In this way it undergoes transitions to various states and finally reaches the ATM screen state as shown in the fig.

DIAGRAM:



Cancel pressed

F) NAME OF EXPERIMENT: Activity diagram for ATM System.

AIM: To design and implement ATM System through Activity Diagram.

THEORY: An activity diagram shows the flow from activity to activity .An activity is an ongoing non atomic execution within a state machine .Activities ultimately results in some action, which is made up of executable atomic computations. We can use these diagrams to model the dynamic aspects of a system.

Activity diagram is basically a flow chart to represent the flow form one activity to another . The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow by using elements like fork, join etc.

Contents

Initial/Final State, Activity, Fork & Join, Branch, Swim lanes

Fork

A fork represents the splitting of a single flow of control into two or more concurrent Flow of control. A fork may have one incoming transition and two or more outgoing transitions, each of which represents an independent flow of control. Below fork the activities associated with each of these path continues in parallel.

Join

A join represents the synchronization of two or more concurrent flows of control. A join may have two or more incoming transition and one outgoing transition. Above the join the activities associated with each of these paths continues in parallel.

Branching

A branch specifies alternate paths takes based on some Boolean expression Branch is represented by diamond Branch may have one incoming transition and two or more outgoing one on each outgoing transition, you place a Boolean expression shouldn't overlap but they should cover all possibilities.

Swim lane:

Swim lanes are useful when we model workflows of business processes to partition the activity states on an activity diagram into groups. Each group representing the business organization responsible for those activities, these groups are called Swim lanes .

Procedure:-

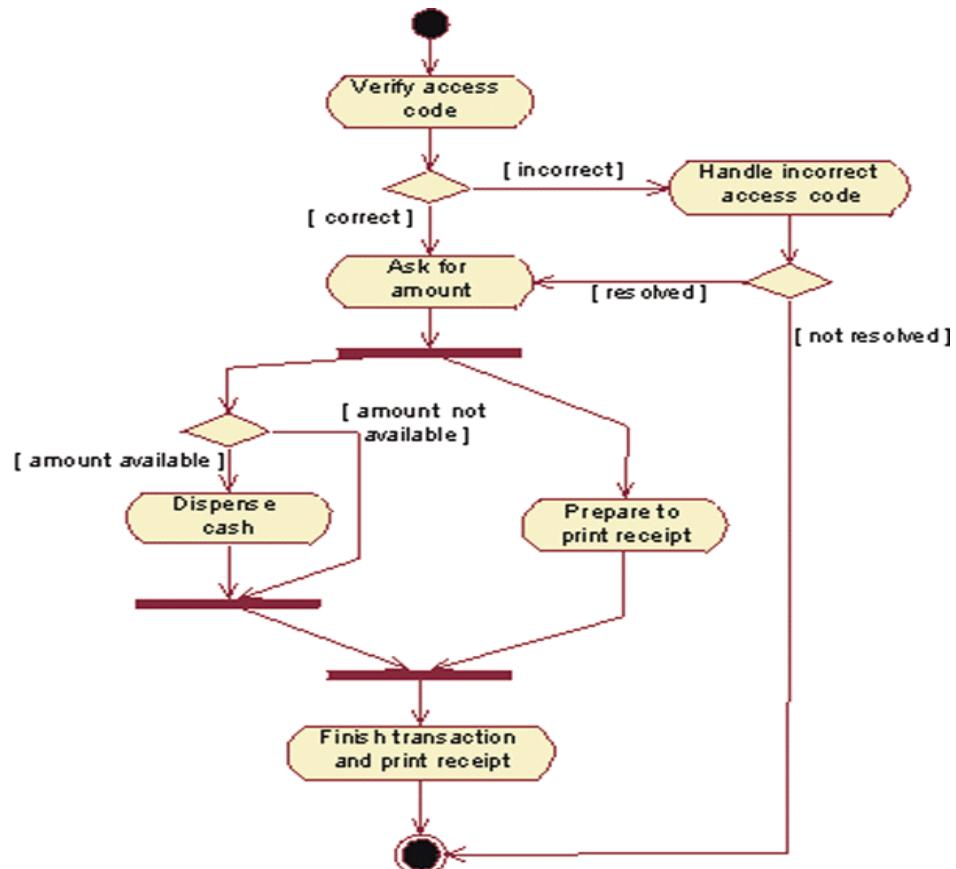
Step1: First initial state is created.

Step2: After that it goes to the action state insert card.

Step3: Next it undergoes transition to the state enter pin

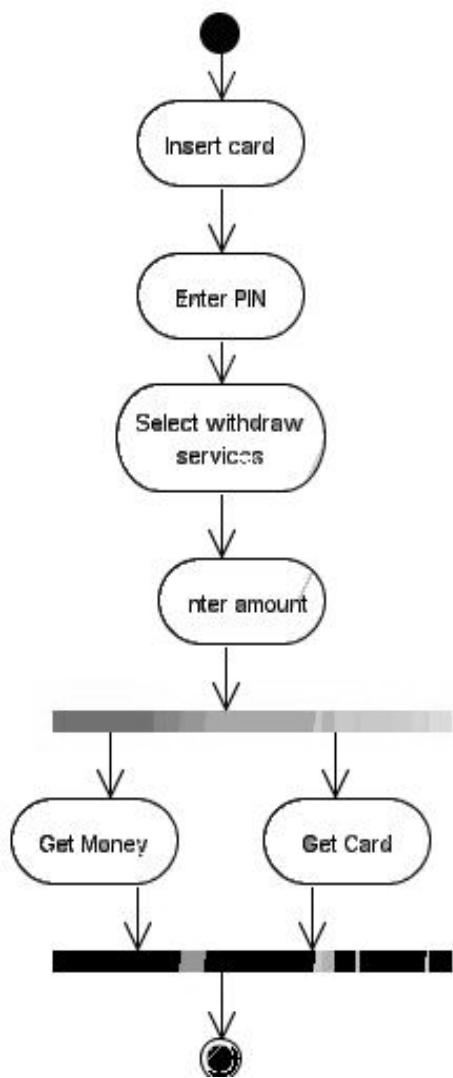
Step4: In this way it undergoes transitions to the various states.

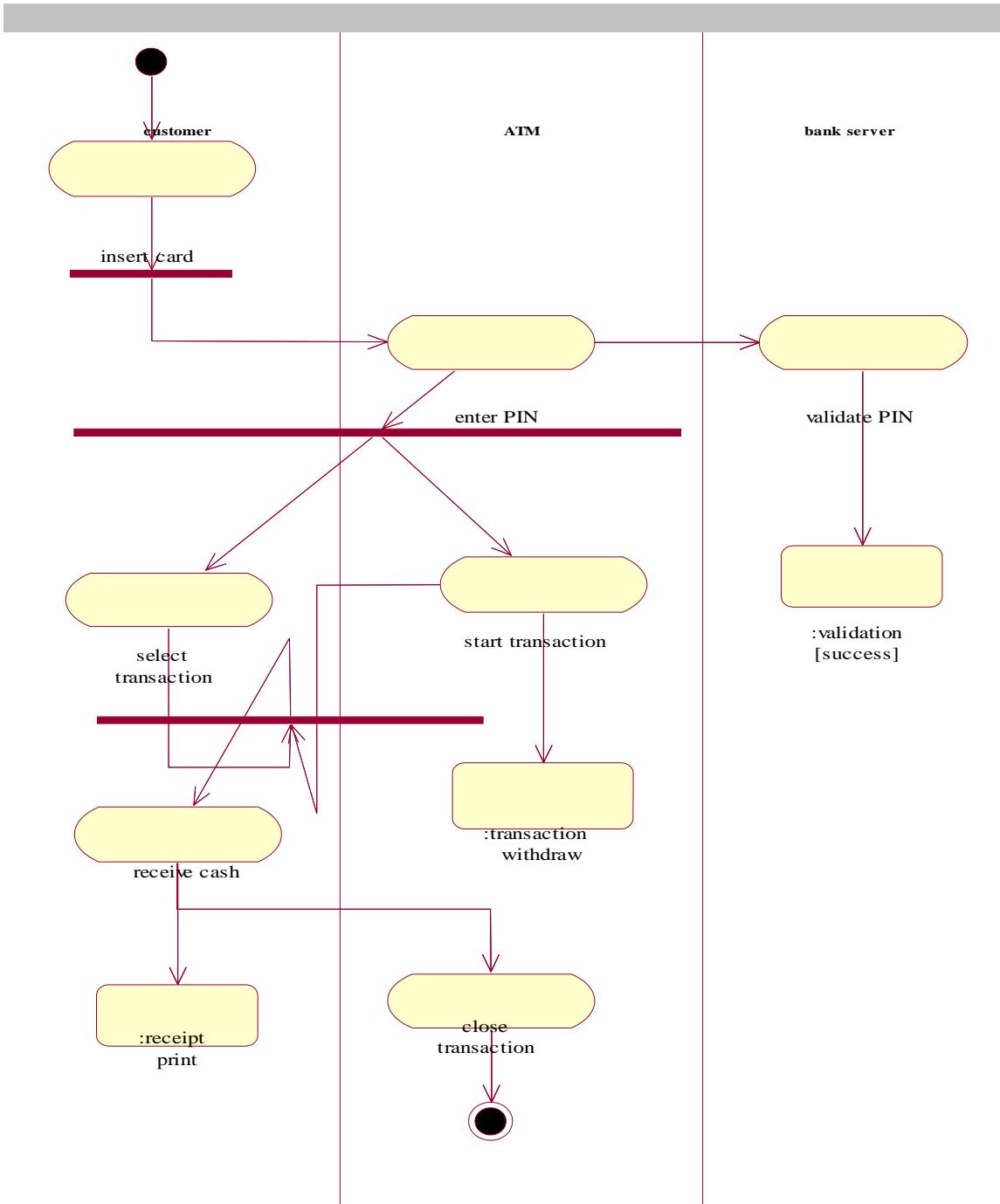
Step5: Use forking and joining wherever necessary.

DIAGRAM:**Activity diagram for Transactions:**

:

Activity diagram for Withdraw



**Inferences:**

1. Identify the action states of the objects .
2. Understand the transitions and events for various objects.

G) NAME OF EXPERIMENT: Component diagram for ATM System.

AIM: To design and implement Component diagram for ATM System.

THEORY:

Component diagrams are used to model physical aspects of a system. Physical aspects are the elements like executable, libraries, files, documents etc. which resides in a node. So component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Purpose:

Component diagrams can be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

Before drawing a component diagram the following artifacts are to be identified clearly:

- Files used in the system.
- Libraries and other artifacts relevant to the application.
- Relationships among the artifacts.
- Now after identifying the artifacts the following points needs to be followed:
 - Use a meaningful name to identify the component for which the diagram is to be drawn.
 - Prepare a mental layout before producing using tools.
 - Use notes for clarifying important points.

Contents

Components, Interfaces, Relationships

Procedure:-

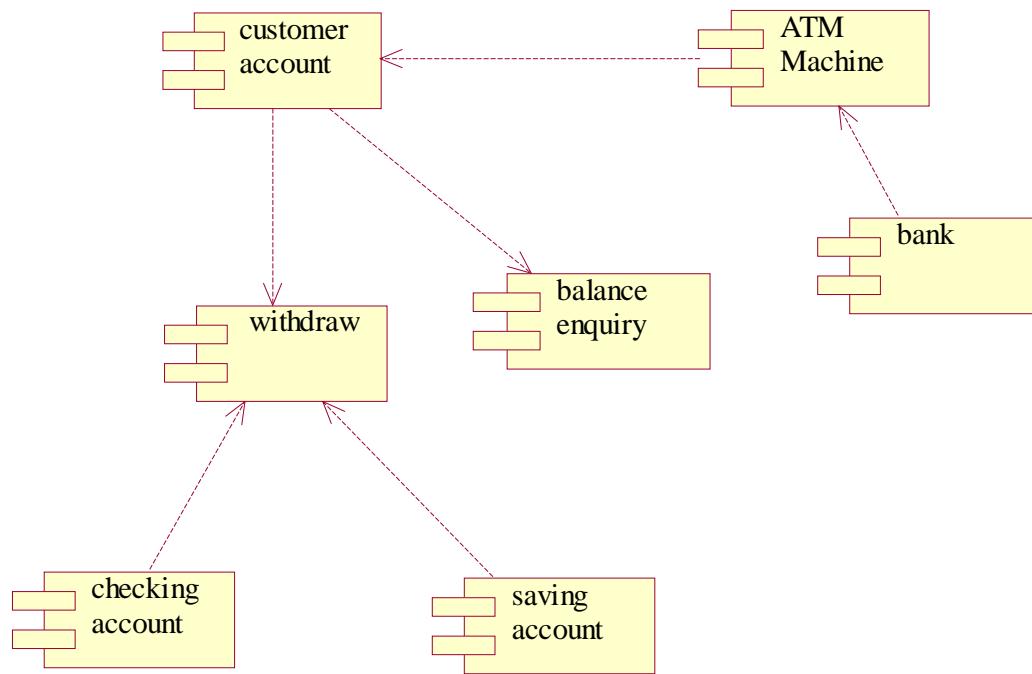
Step1: First user component is created.

Step2: ATM system package is created.

Step3: In it various components such as withdraw money, deposit money, check balance, transfer money etc. are created.

Step4: Association relationship is established between user and other components.

COMPONENT DIAGRAM:



H) NAME OF EXPERIMENT: Deployment diagram for ATM System.

AIM: To design and implement ATM System through Deployment diagram.

Purpose:

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

Contents: Nodes, Dependency & Association relationships

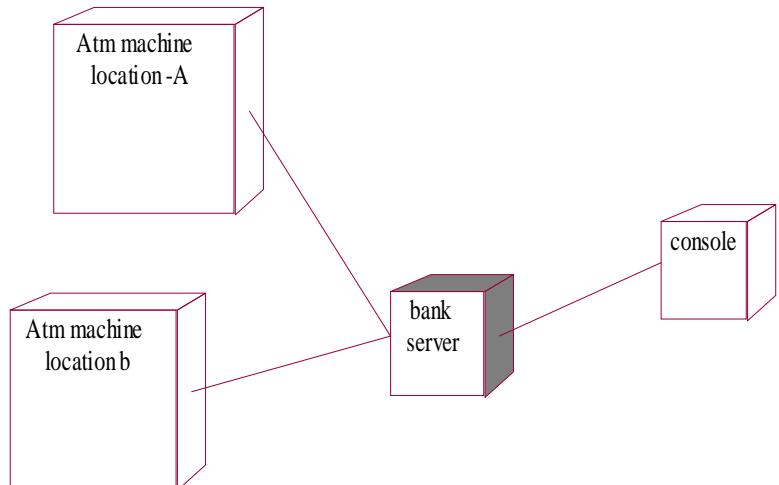
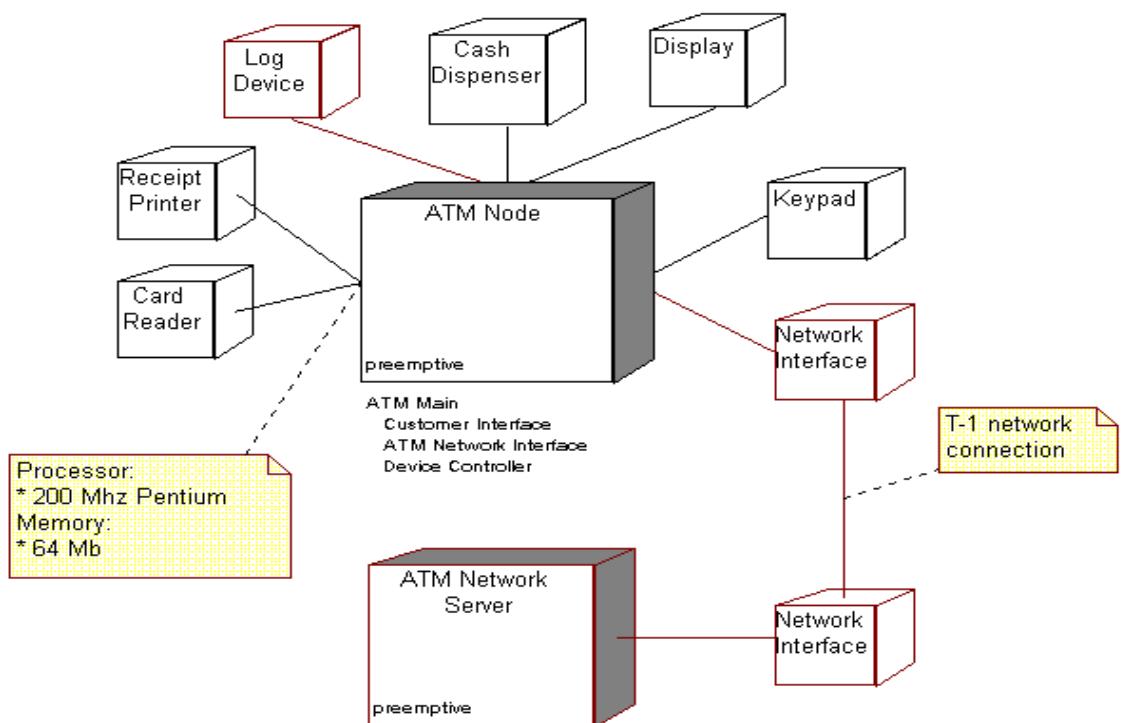
Procedure:-

Step1: First user node is created

Step2: various nodes withdraw money, deposit money, and check balance, transfer money etc. are created.

Step4: Association relationship is established between user and other nodes.

Step5: Dependency is established between deposit money and check balance.



I) NAME OF EXPERIMENT: Test Design for ATM System

AIM: To design and Write the test cases for ATM System

PURPOSE:

Test Design is creating a set of inputs for given software that will provide a set of expected outputs.

The idea is to ensure that the system is working good enough and it can be released with as few

Problems as possible for the average user

Test Design for ATM Machine

1. Machine is accepting ATM card
2. Machine is rejecting expired card
3. Successful entry of PIN number
4. Unsuccessful operation due to enter wrong PIN number 3 times
5. Successful selection of language
6. Successful selection of account type
7. Unsuccessful operation due to invalid account type
8. Successful selection of amount to be withdraw
9. Successful withdrawal.
10. Expected message due to amount is greater than day limit
11. Unsuccessful withdraw operation due to lack of money in ATM
12. Expected message
Due to amount to withdraw is greater than possible balance.
13. Unsuccessful withdraw operation due to click cancel after insert card

Write the Test Cases for ATM

Test Cases For Atm Machine

1. Successful inspection of ATM card
2. Un successful operation due to insert card in wrong angle
3. Un successful operation due to invalid account Ex:other bank card or time expired Card
4. Successful entry of PIN number
- 5.un successful operation due to enter wrong PIN number 3times
6. Successful selection of language
7. Successful selection of account type
8. un successful operation due to invalid account type
10. Successful selection of withdrawl operation
11. Successful selection of amount to be withdrawl
12. Successful withdrawl operation
13. Unsuccessful withdrawl operation due to wrong denominations
14. Unsuccessful withdrawl operation due to amount is greater than day limit
15. Unsuccessful withdrawl operation due to lack of money in ATM
16. Unsuccessful withdrawl operation due to amount is greater than possible balance

17. Unsuccessful withdrawl operation due to transactions is greater than day limit
18. Unsuccessful withdrawl operation due to click cancel after insert card
19. Unsuccessful withdrawl operation due to click cancel after insert card & pin number
20. Unsuccessful withdrawl operation due to click cancel after insert card , pin number

2.8 RESULT

Thus the requirements involved in developing an Automated Banking System was completed successfully

Contents

CLASS DIAGRAMS

1. Electronic Cash Counter
2. Online Library Management
3. Railway Reservation System
4. University Registration System
5. Online library system
6. A university is implementing a new student information system. How would you design a class diagram to represent the various entities such as students, courses, professors, and departments, along with their attributes and relationships?
7. Employee Management

USE CASE DIAGRAMS

1. Library Management System
2. Online Railway Reservation System
3. Employee Management
4. Banking system
5. A hospital is redesigning its patient management system. Develop a use case diagram to identify and illustrate the different actors such as patients, doctors, nurses, and administrators, along with the various use cases they are involved in, such as scheduling appointments, recording medical history, and managing inventory.
6. Online banking system

SEQUENCE DIAGRAMS

1. ATM System
2. Deposit Use Case
3. Online shopping
4. A customer is making a purchase on an e-commerce website. Create a sequence diagram illustrating the interactions between the customer, the website interface, the shopping cart, and the payment gateway.
5. Employee Management
6. Imagine a scenario where an online shopping application is being designed. Create a sequence diagram and a collaboration diagram to illustrate the interface between the user, the shopping cart, and the payment system during a typical transaction. Analyse the benefits of using these diagrams in the design process..

ACTIVITY DIAGRAM

1. Unified Library System
2. Online Railway Reservation System
3. Data Warehouse Management System
4. ATM transaction process
5. Employee Management

COMMUNICATION DIAGRAM

1. Library Management System
2. Railway Reservation System

COLLABORATION DIAGRAMS

1. Messaging Application
2. Employee Management
3. Imagine a scenario where an online shopping application is being designed. Create a sequence diagram and a collaboration on diagram to illustrate the interface between the user, the shopping cart, and the payment system during a typical transaction. Analyse the benefits of using these diagrams in the design process.

INFORMATION FLOW DIAGRAM

1. Library Management System

COMPONENT DIAGRAM

1. ATM
2. Online banking system
3. Simple email system
4. A software company is developing a mobile application that integrates with various third-party services such as payment gateways, social media platforms, and analytics tools. Design a component diagram to represent the different components of the mobile application and their dependencies on external services and libraries.
5. Library Management System
6. Online Railway Reservation System

STATE CHART

1. ATM System
2. Railway Reservation System
3. Library Management System

OBJECT DIAGRAM

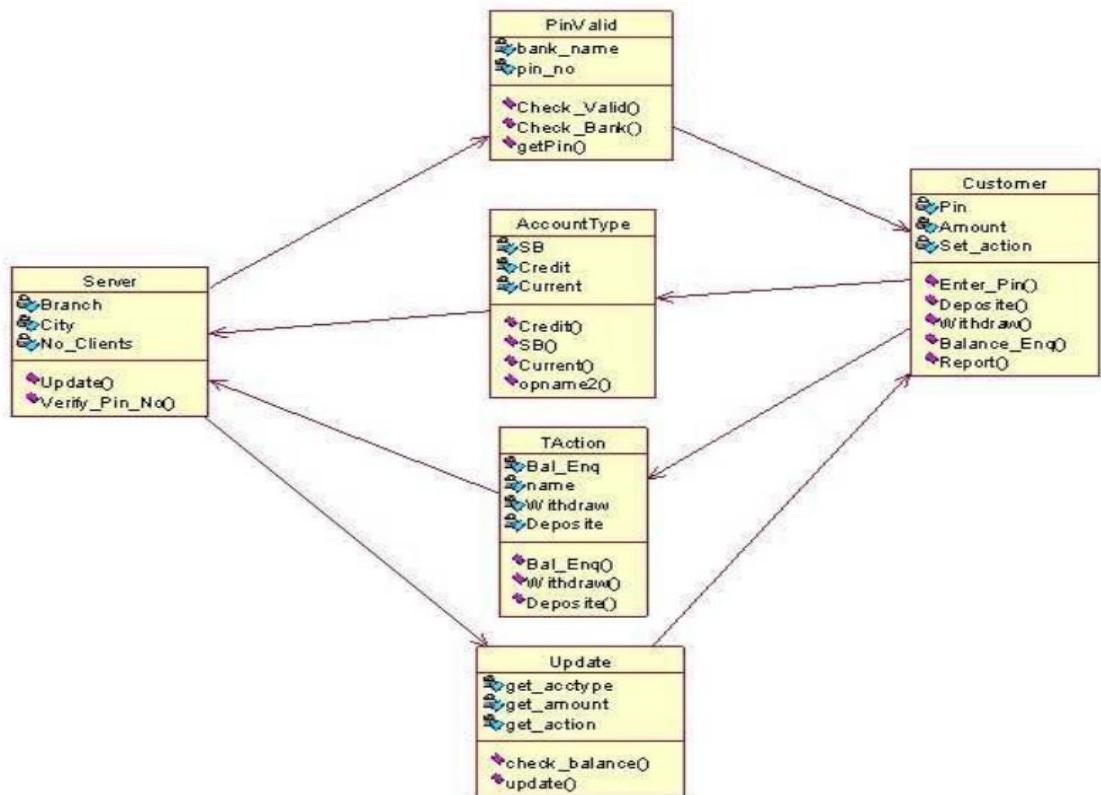
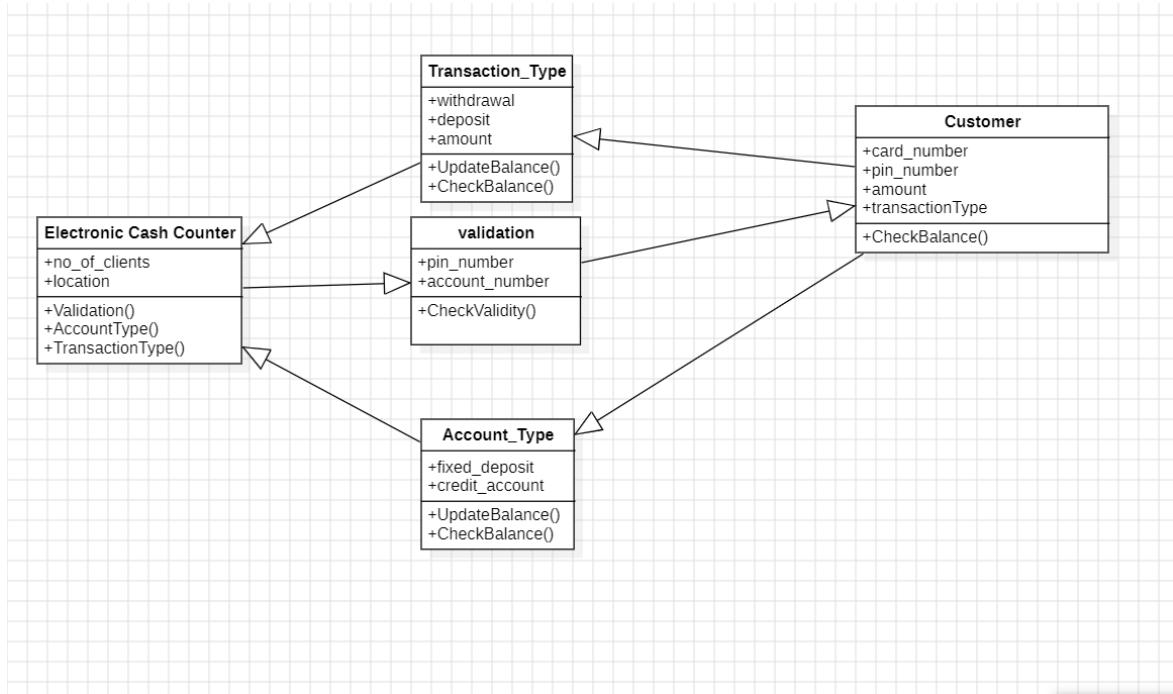
1. Car Rental System
2. Railway Reservation System
3. Library Management System

INTERACTION SEQUENCE

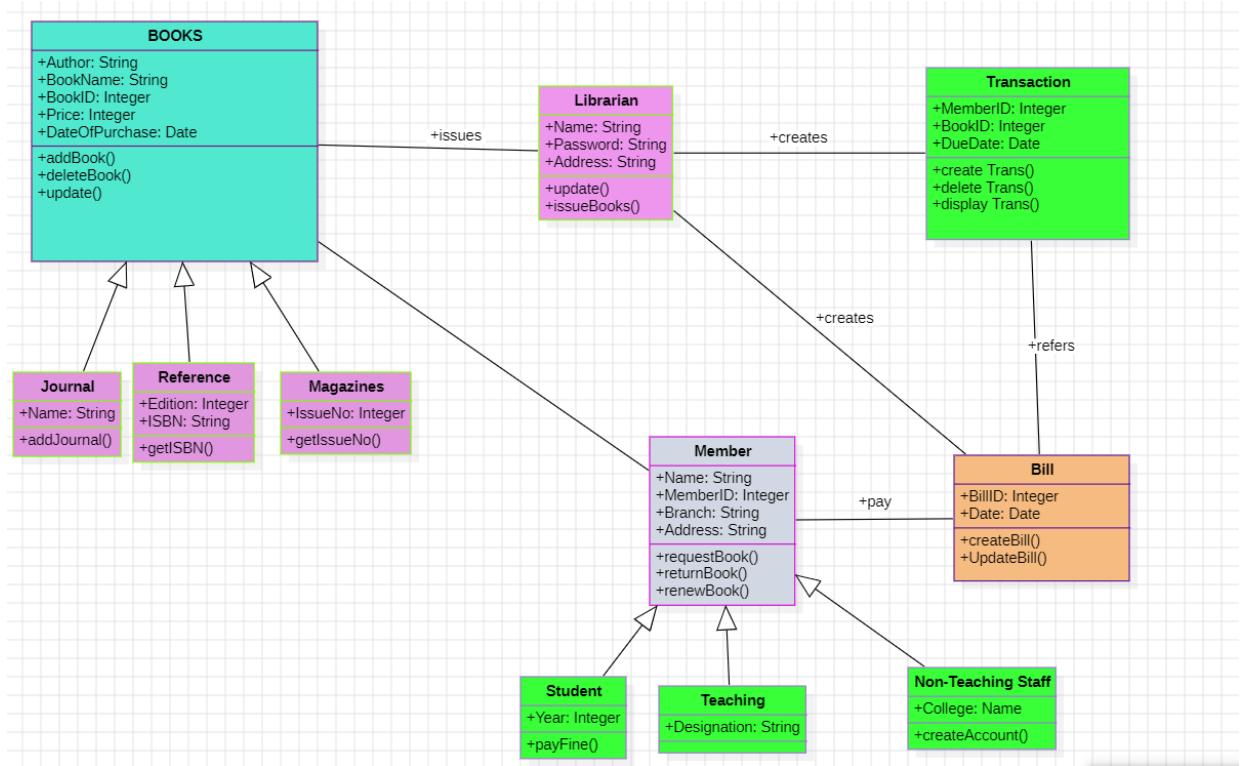
1. Railway Reservation System
2. Library Management System

CLASS DIAGRAMS

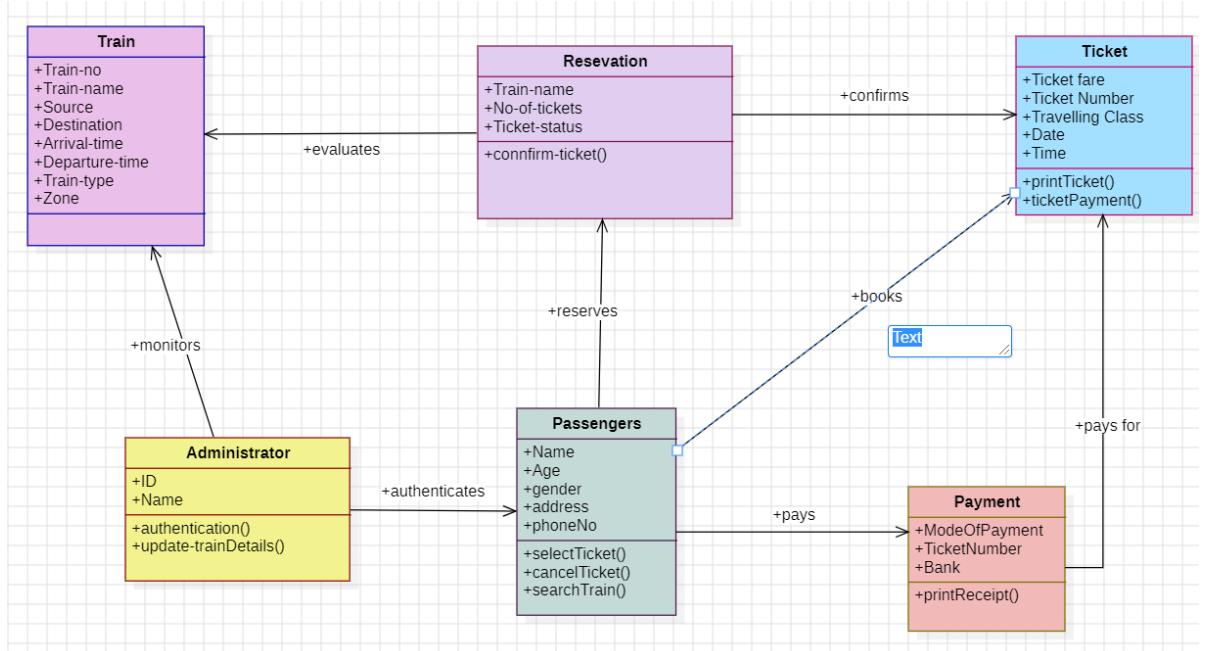
1. Electronic Cash Counter



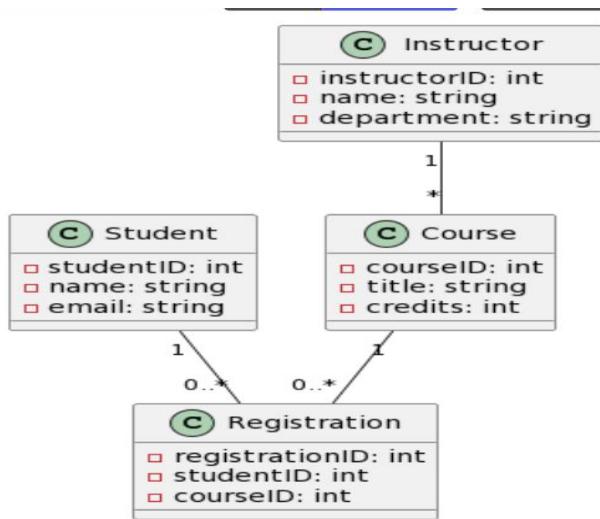
2. Online Library Management



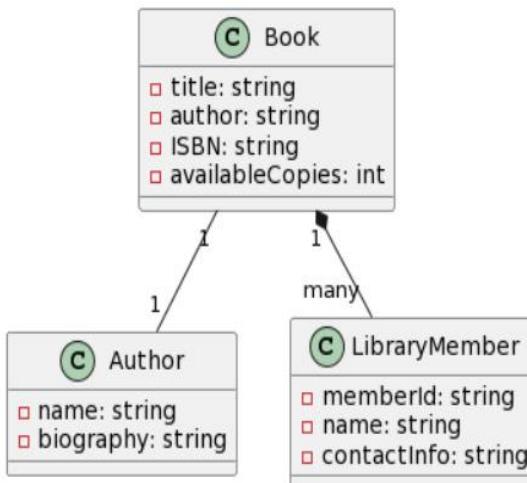
3. Railway Reservation System



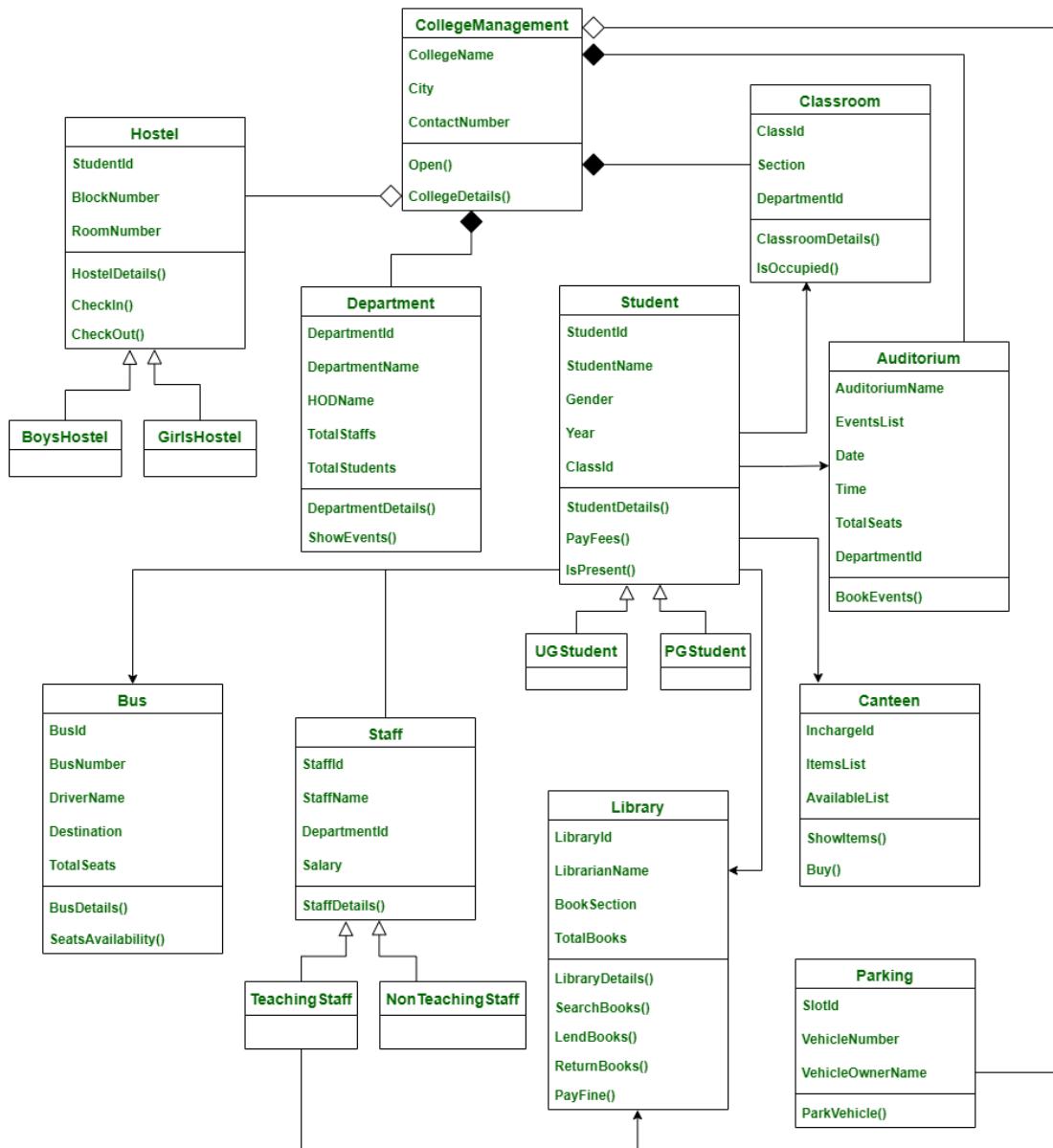
4. University Registration System



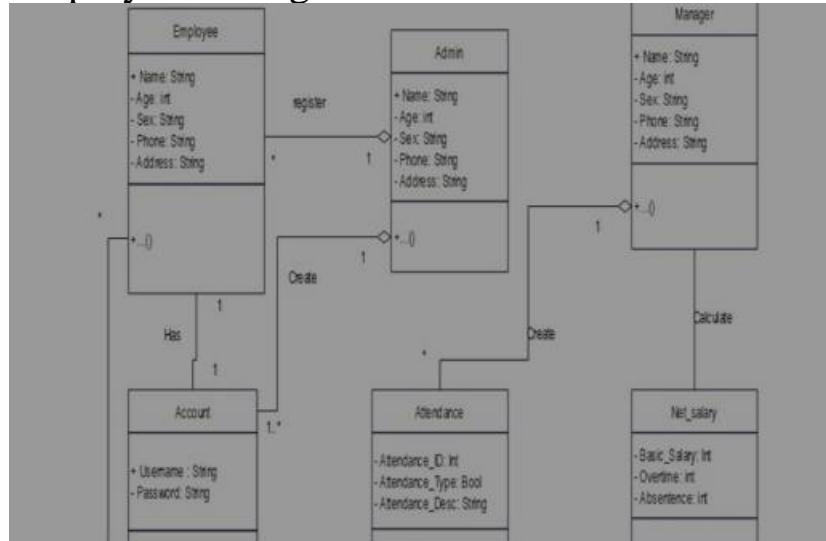
5. Online library system



6. A university is implementing a new student information system. How would you design a class diagram to represent the various entities such as students, courses, professors, and departments, along with their attributes and relationships?

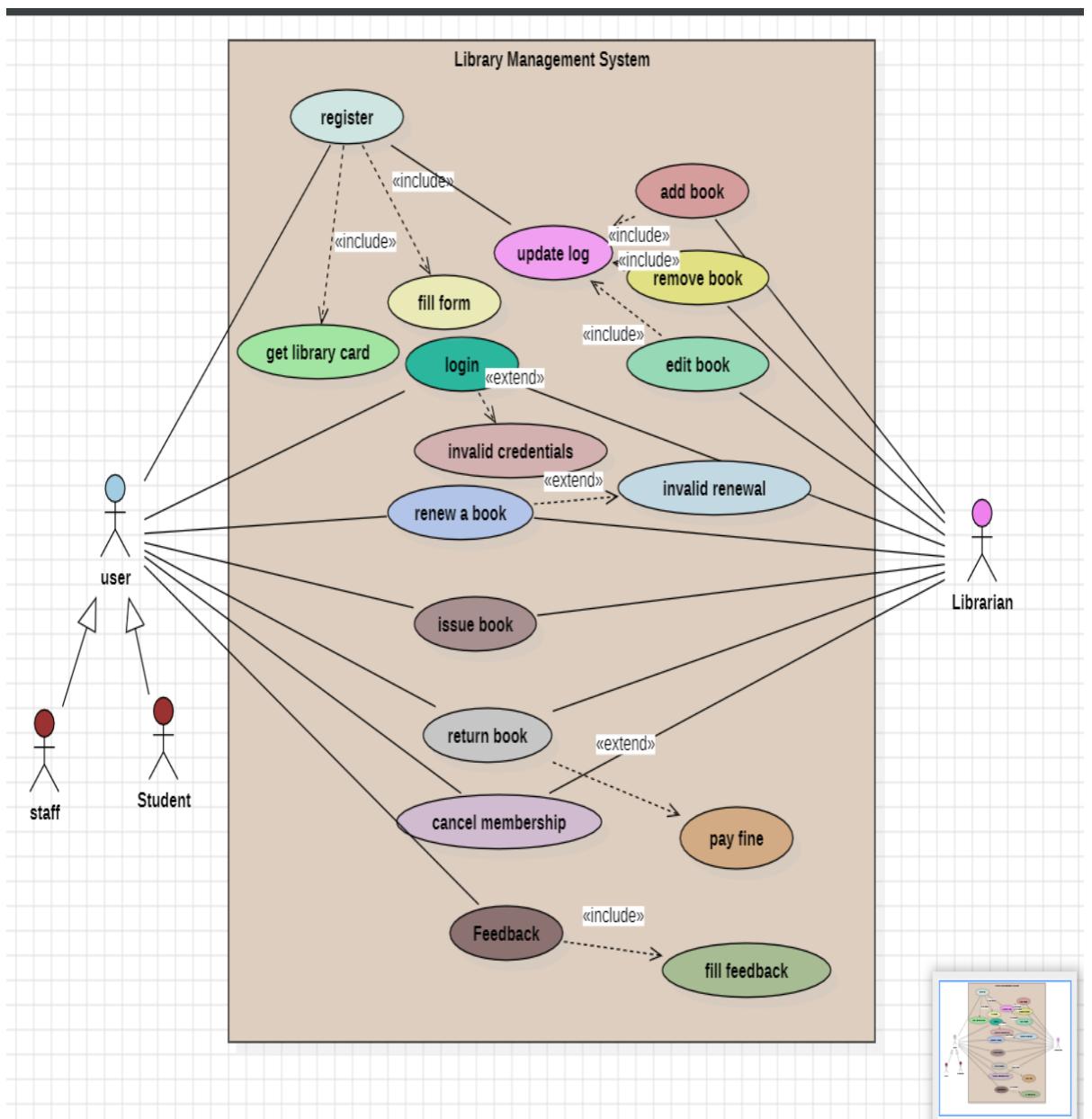


7. Employee Management

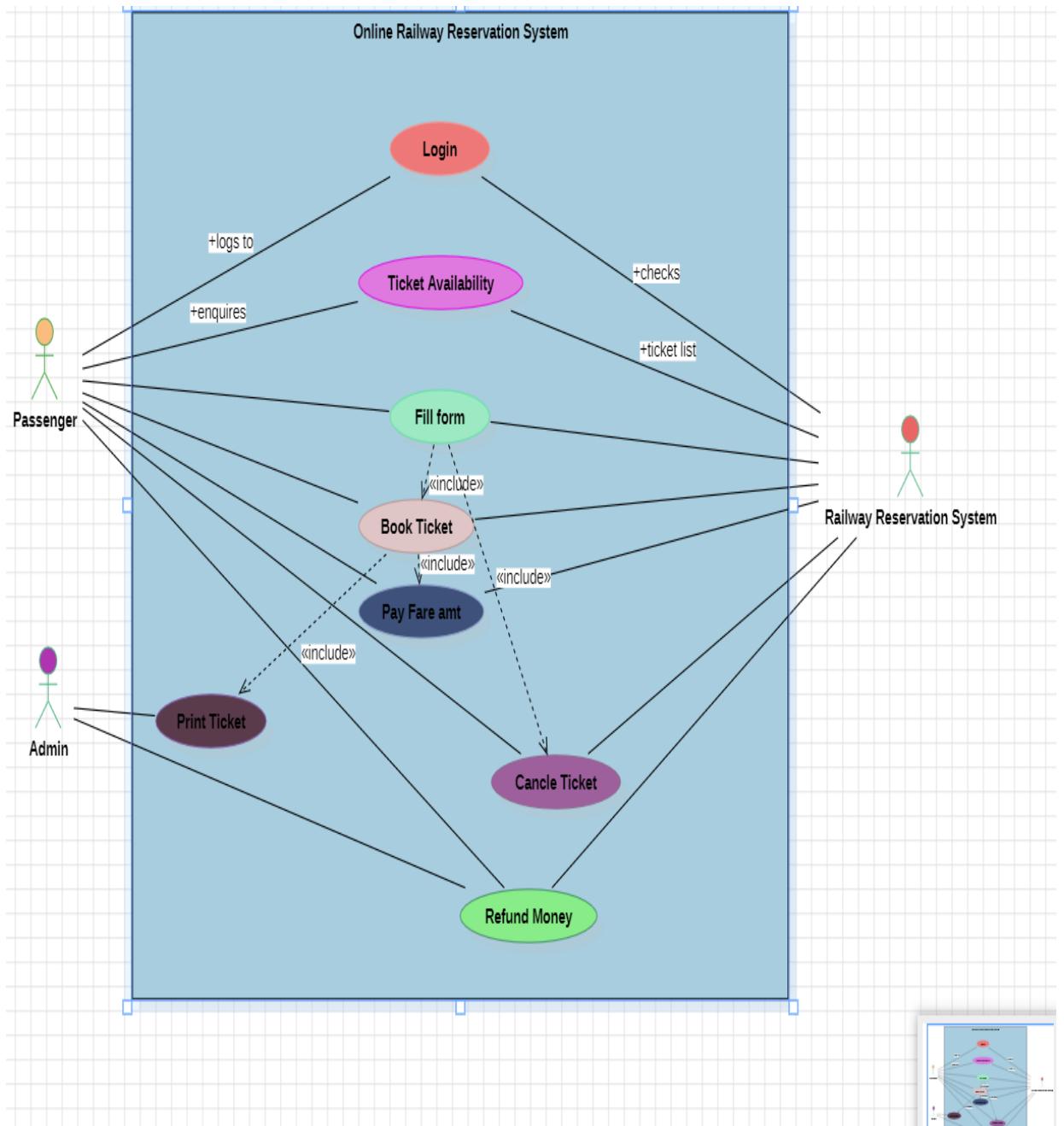


USE CASE DIAGRAMS

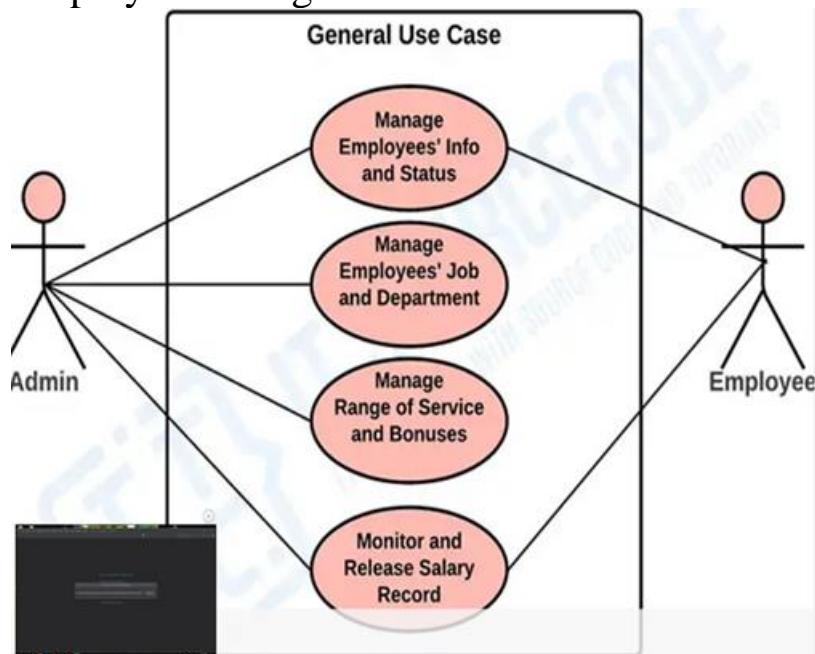
1. Library Management System



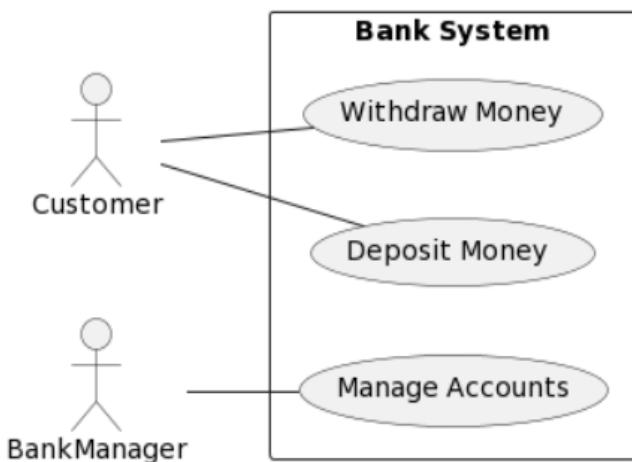
2. Online Railway Reservation System



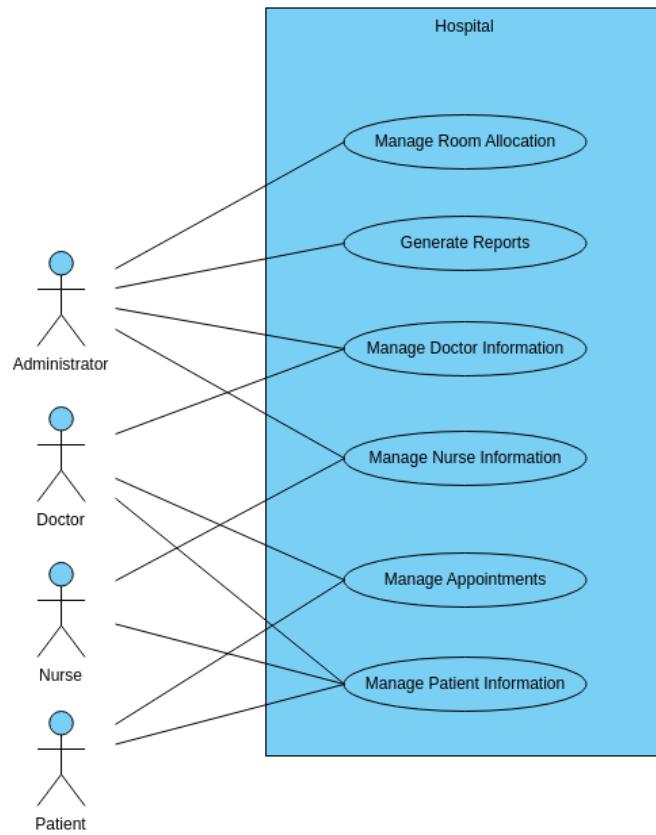
3. Employee Management



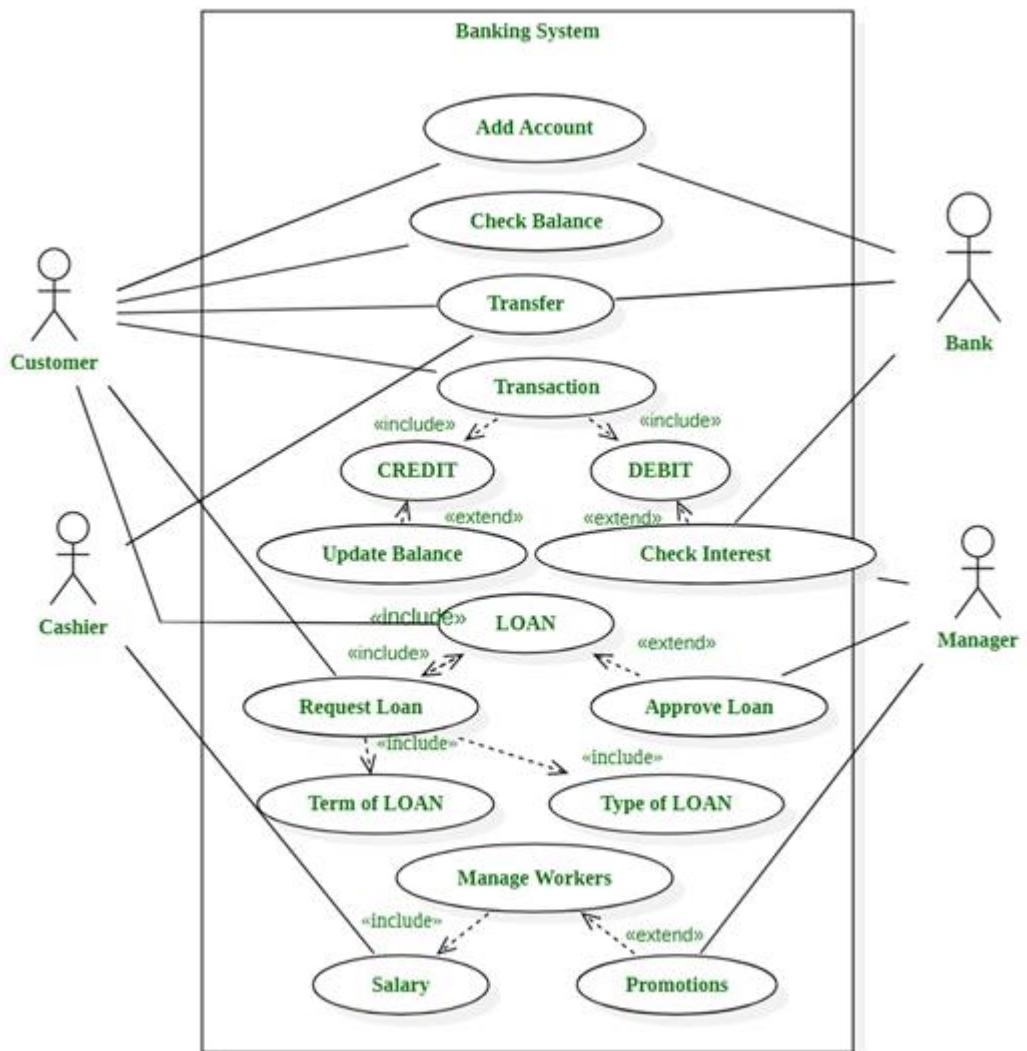
4. Banking system



5. A hospital is redesigning its patient management system. Develop a use case diagram to identify and illustrate the different actors such as patients, doctors, nurses, and administrators, along with the various use cases they are involved in, such as scheduling appointments, recording medical history, and managing inventory.

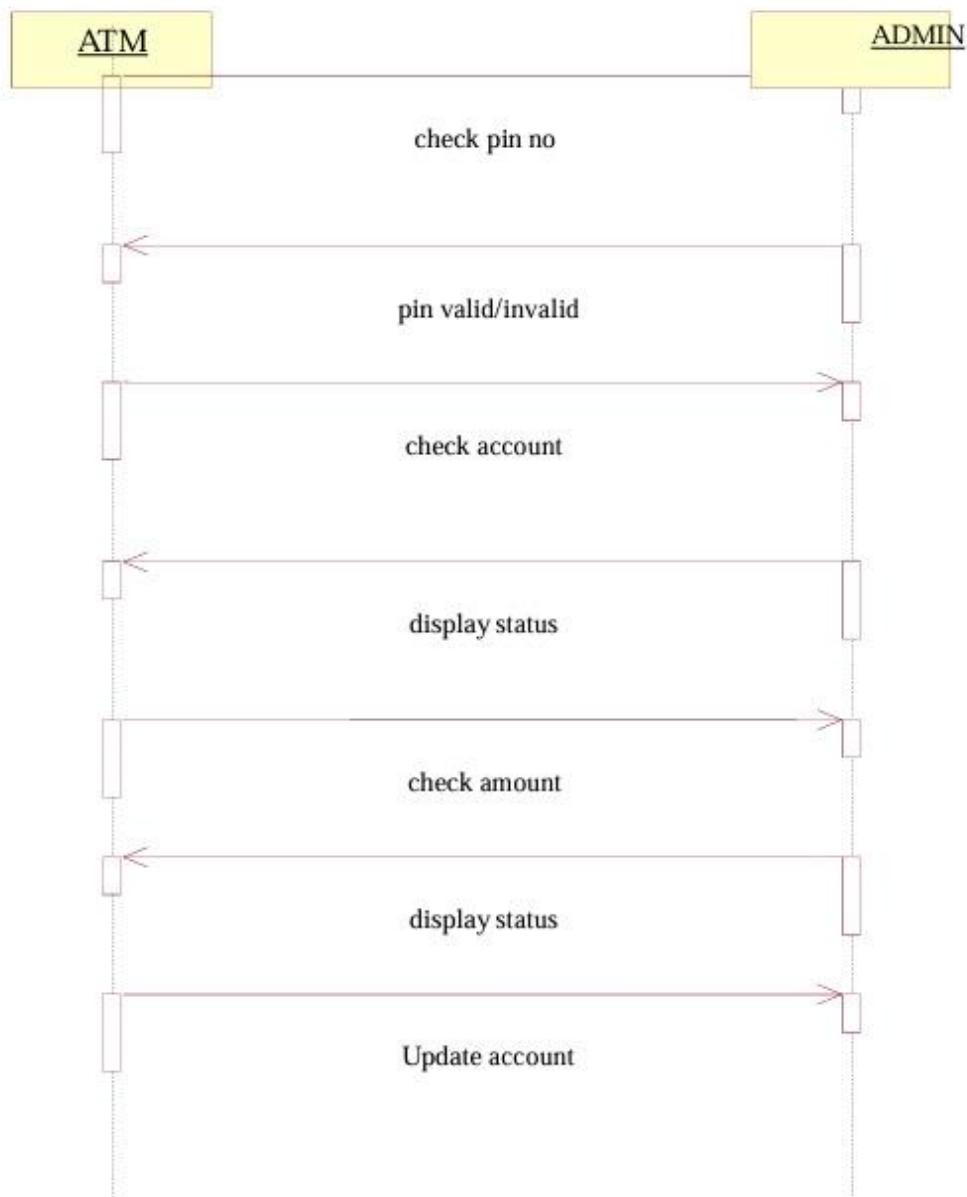


6. Online banking system

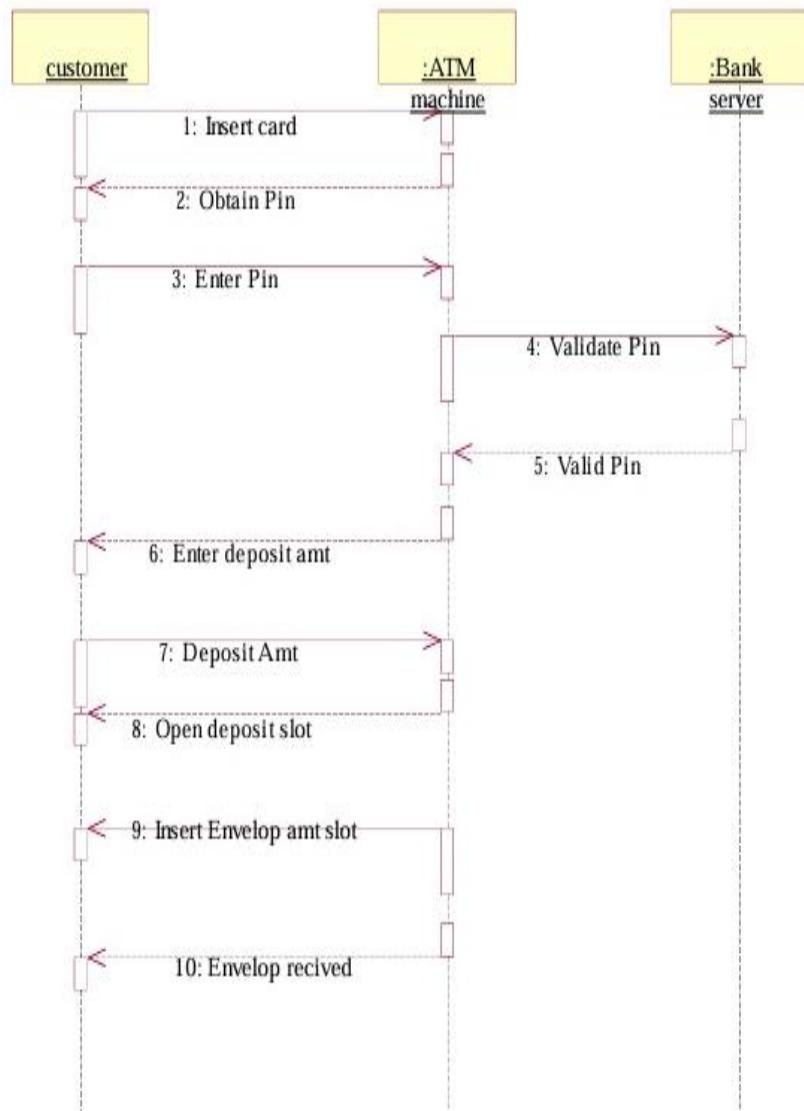


SEQUENCE DIAGRAMS

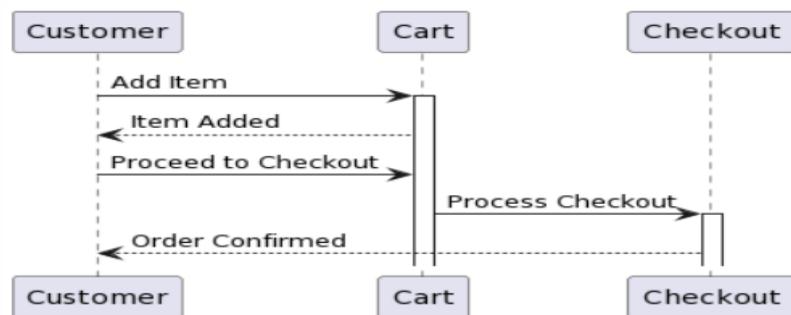
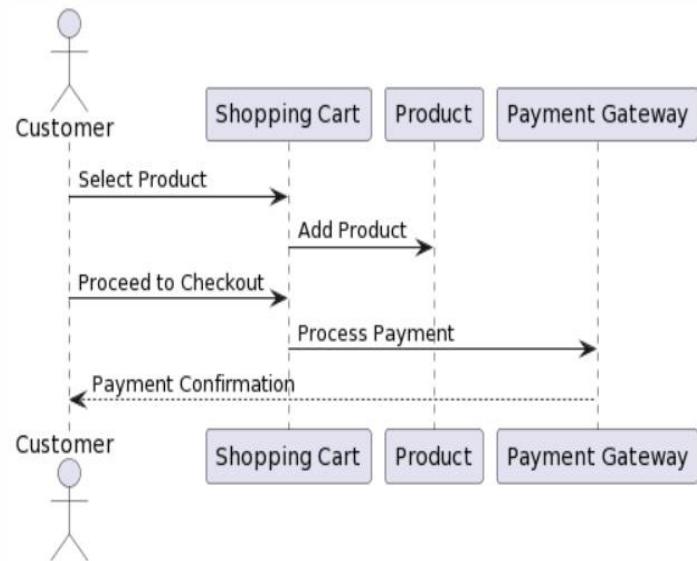
1. ATM System



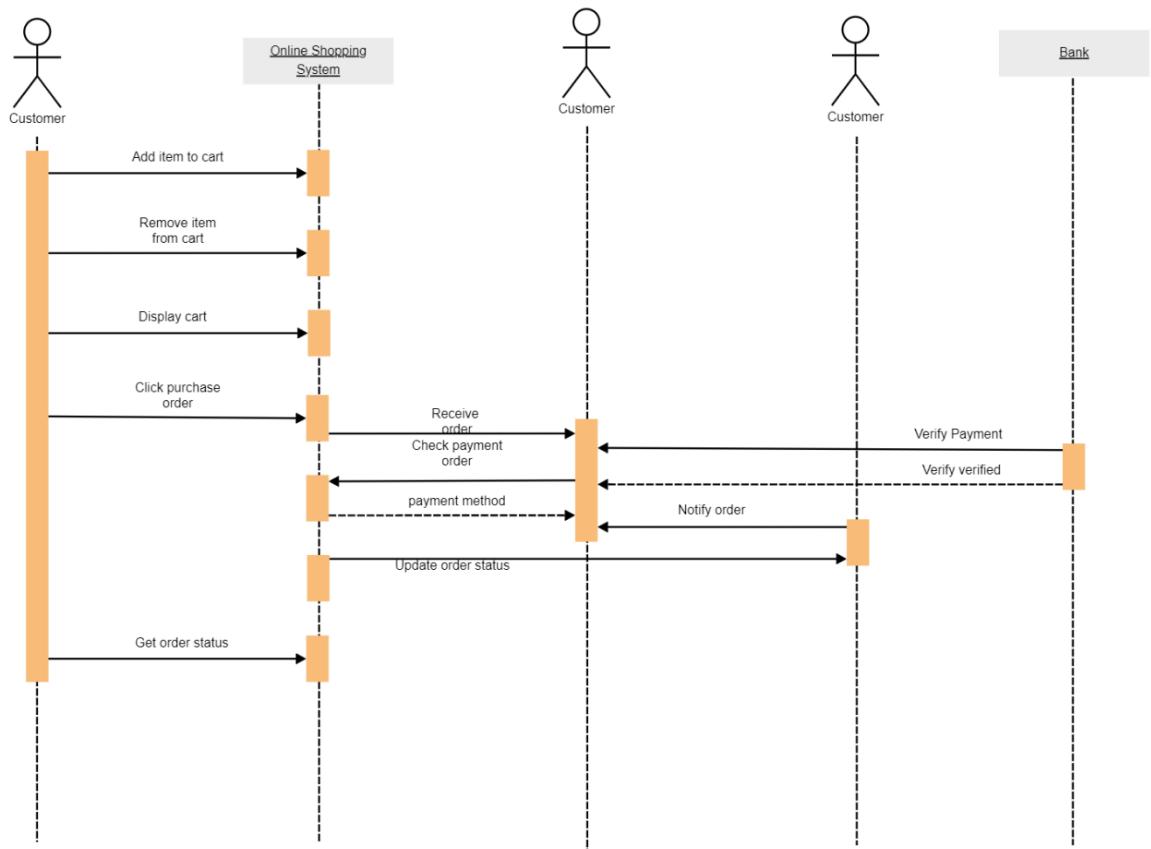
2. Deposit Use Case



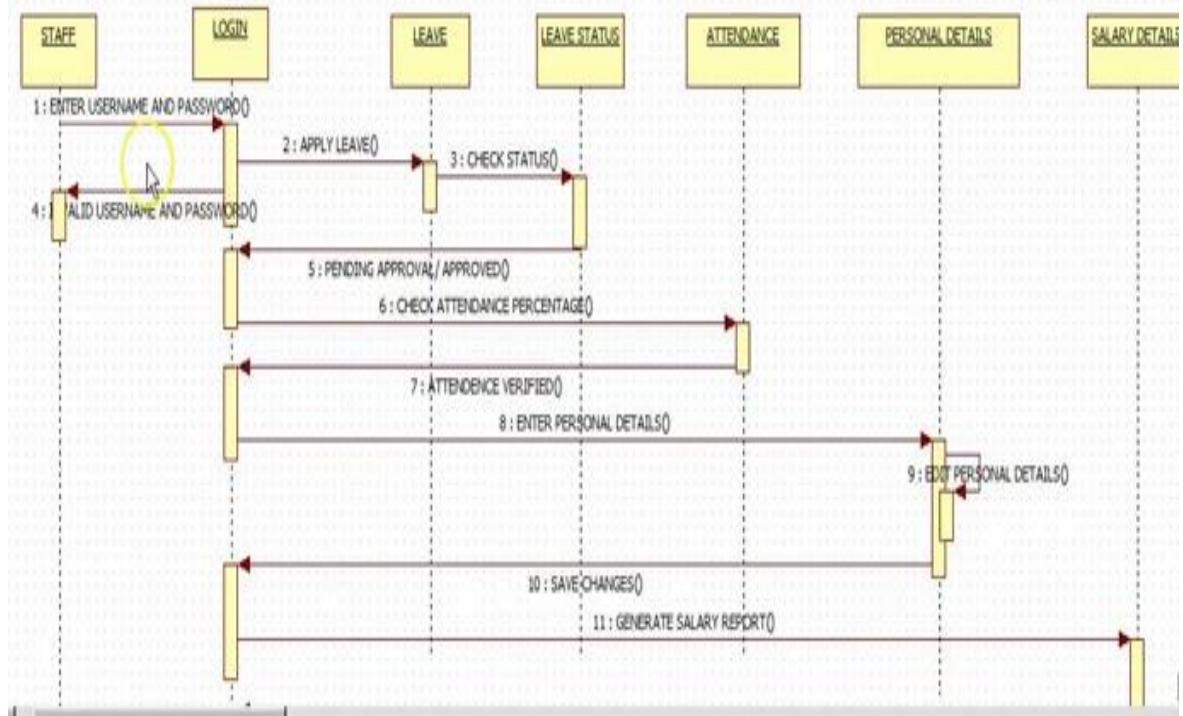
3. Online shopping



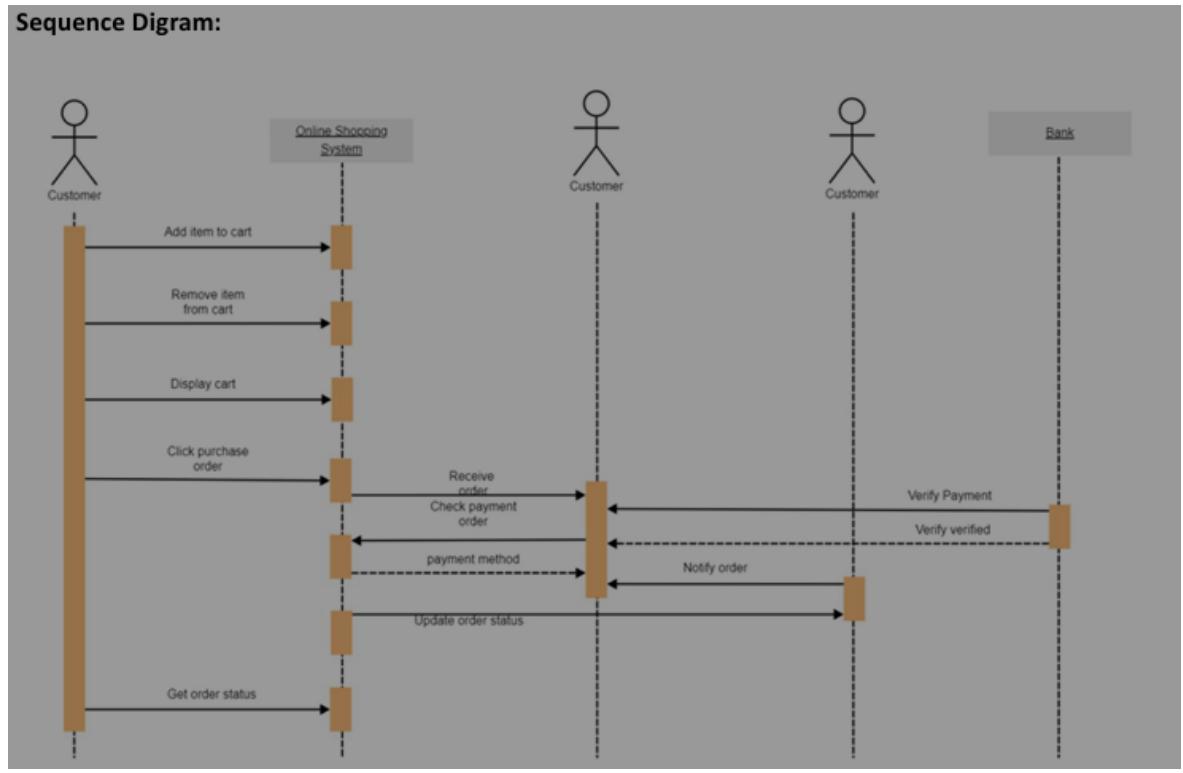
4. A customer is making a purchase on an e-commerce website. Create a sequence diagram illustrating the interactions between the customer, the website interface, the shopping cart, and the payment gateway.



5. Employee Management



6. Imagine a scenario where an online shopping application on is being designed. Create a sequence diagram and a collaboration on diagram to illustrate the interface between the user, the shopping cart, and the payment system during a typical transaction. Analyse the benefits of using these diagrams in the design process.



ACTIVITY DIAGRAM

1. Unified Library System

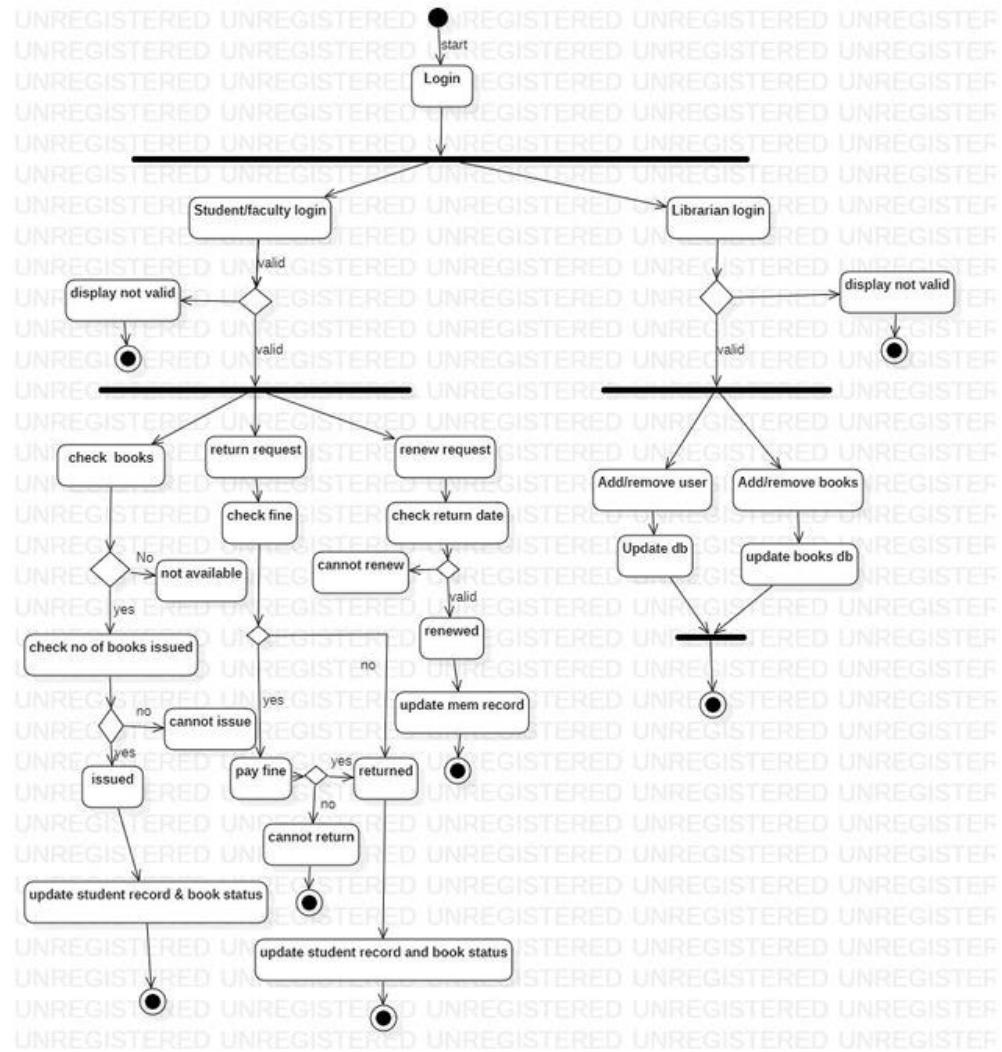
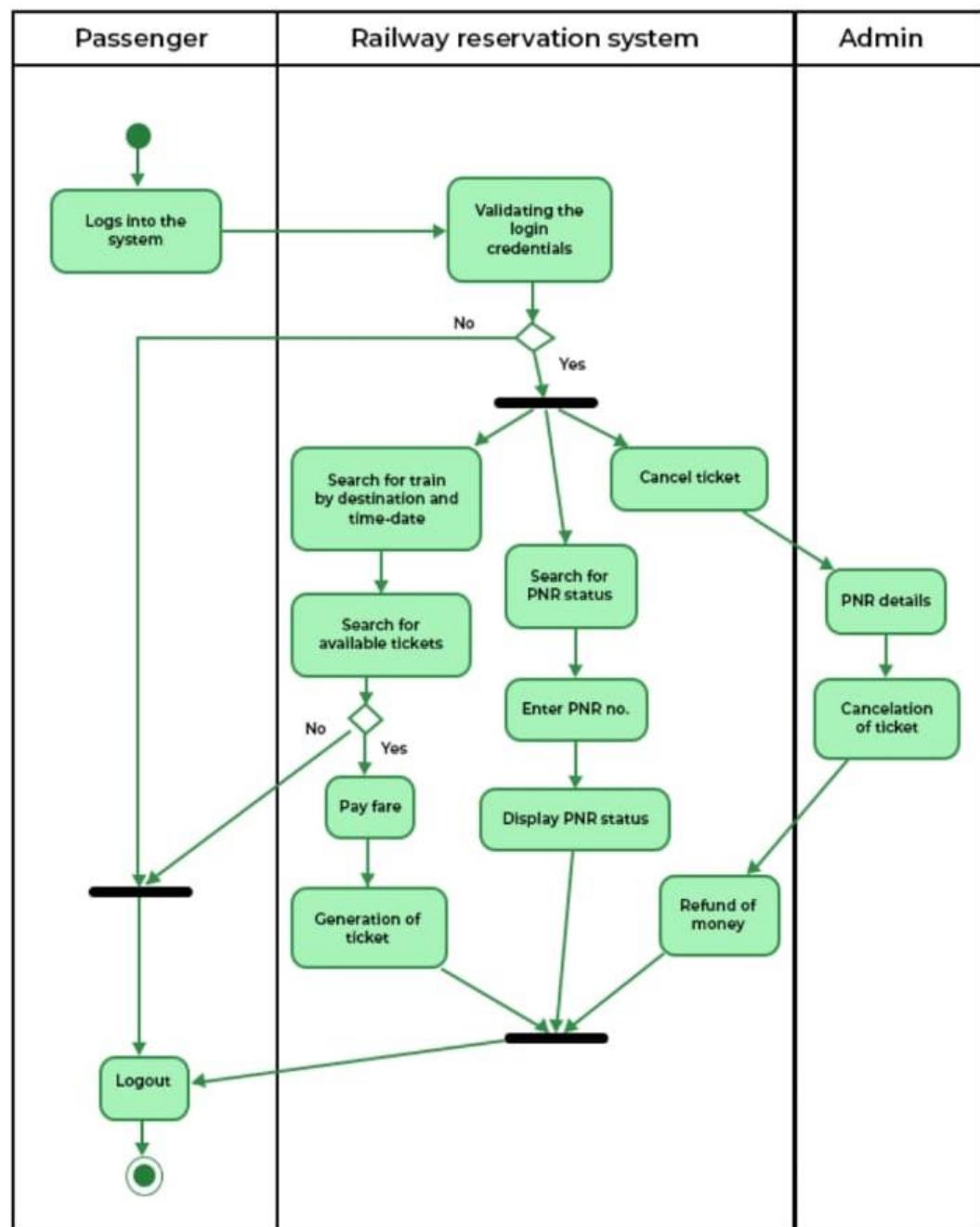
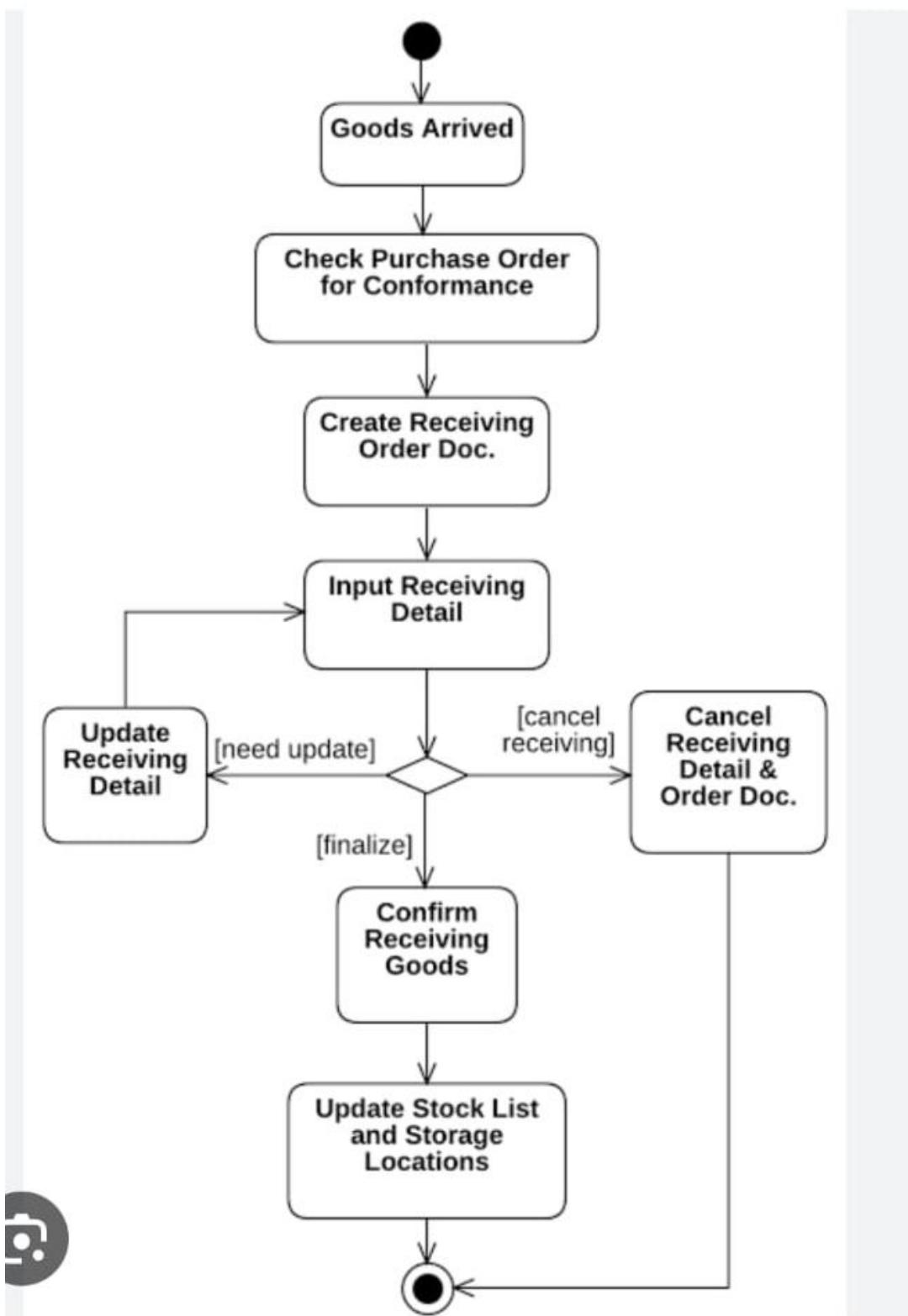


Fig:1: Activity diagram for Unified Library System

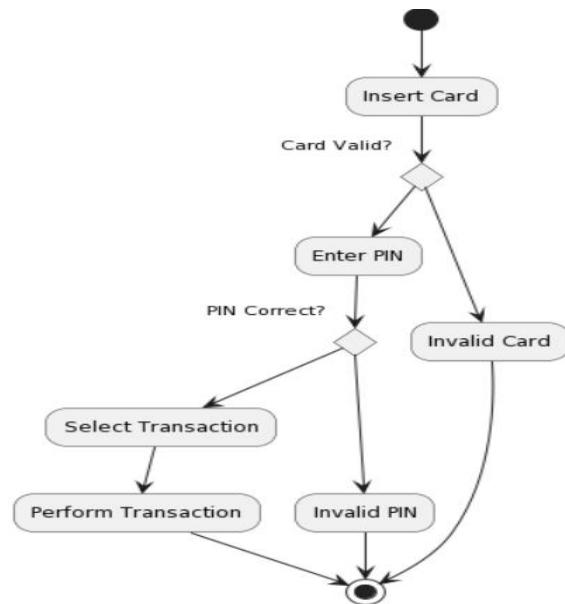
2. Online Railway Reservation System



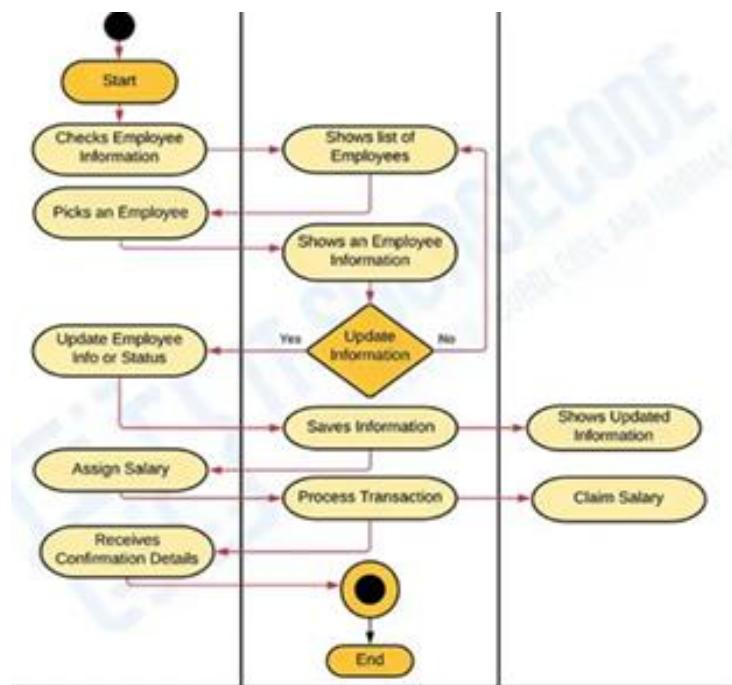
3. Data Warehouse Management System



4. ATM transaction process

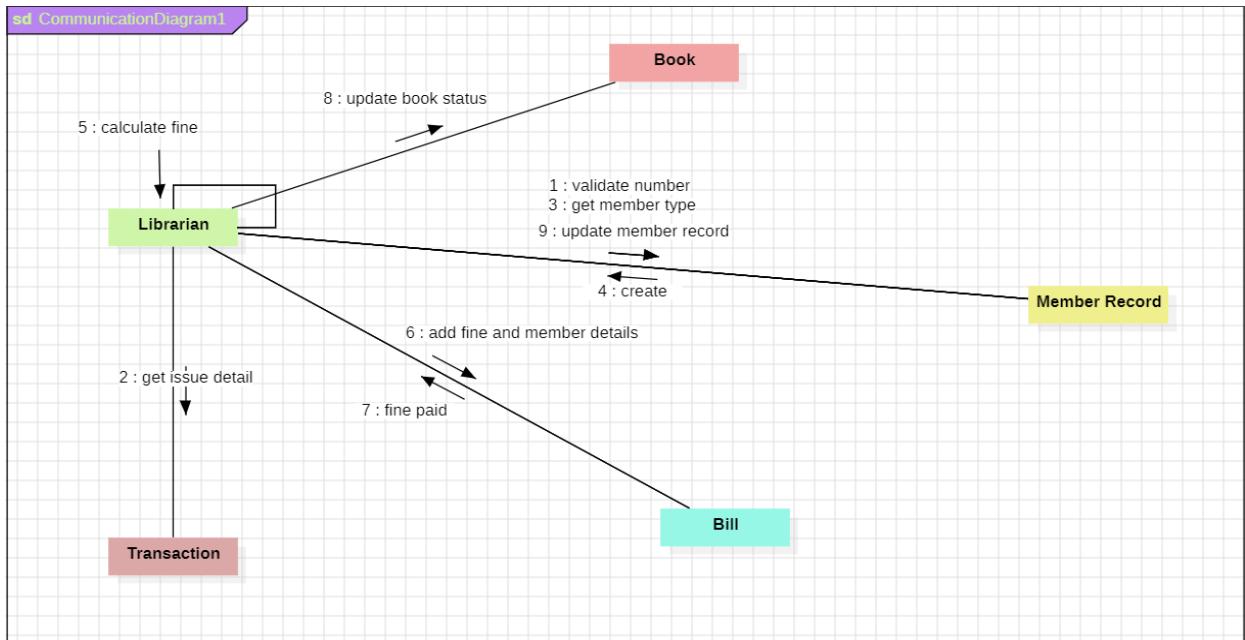


5. Employee Management

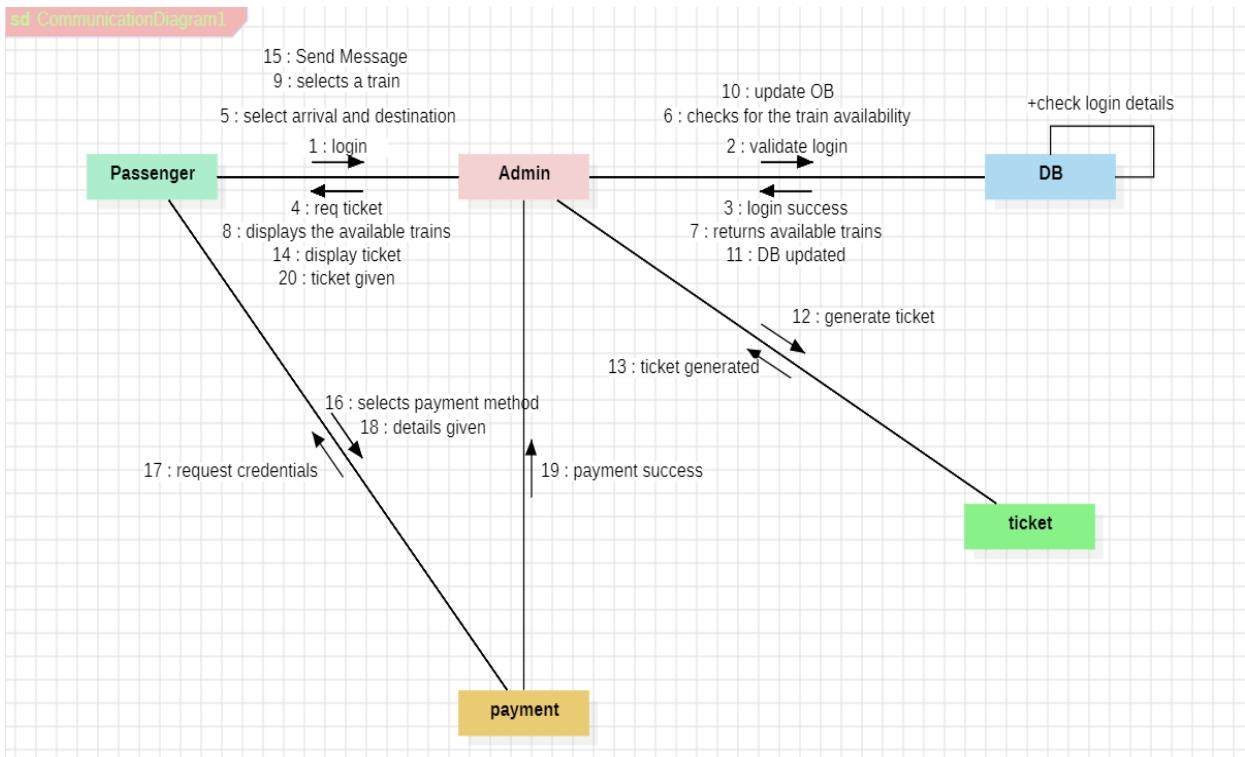


COMMUNICATION DIAGRAM

1. Library Management System

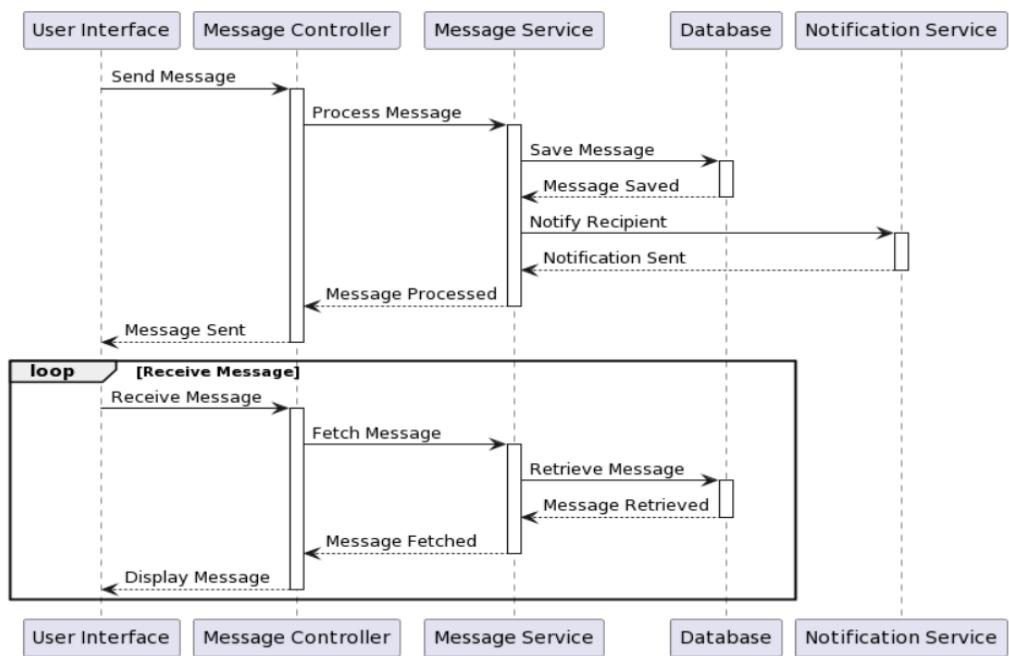
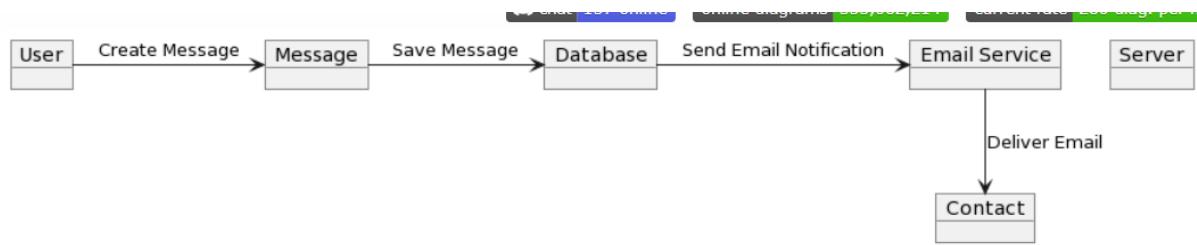


2. Railway Reservation System

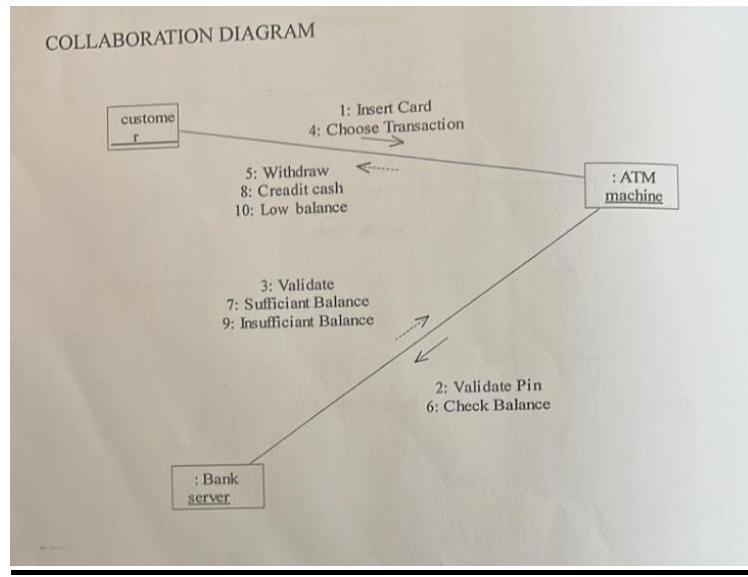


COLLABORATION DIAGRAMS

1. Messaging Application

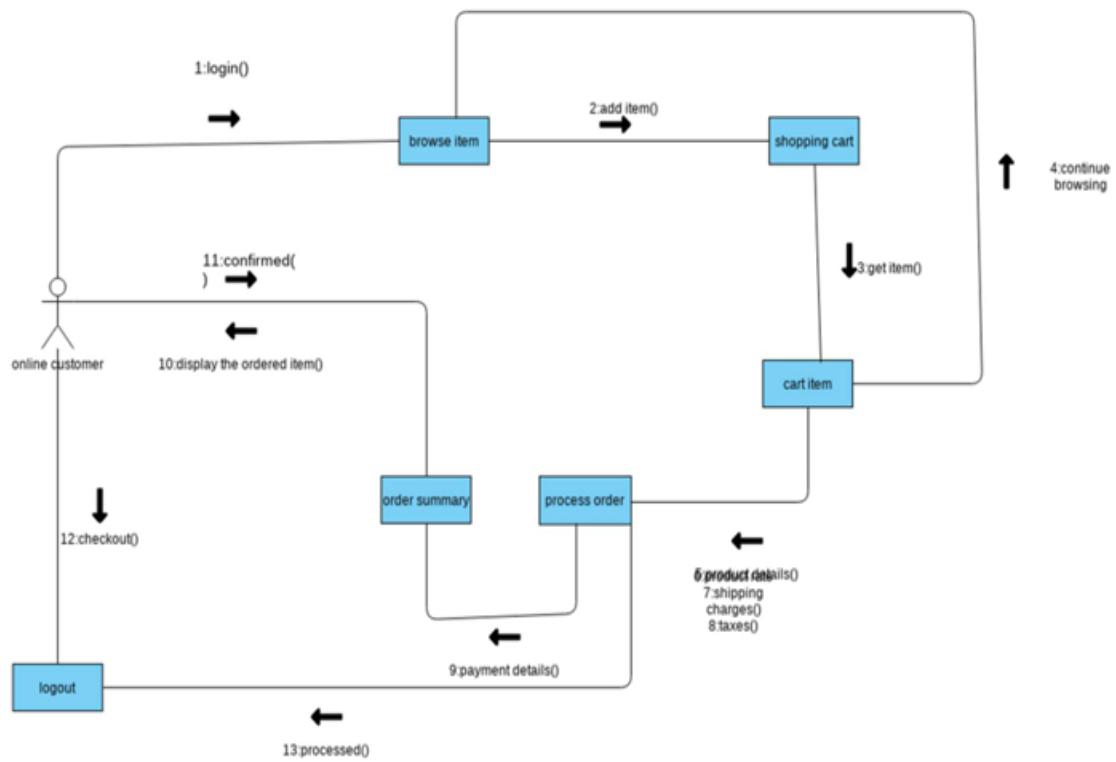


2. Employee Management



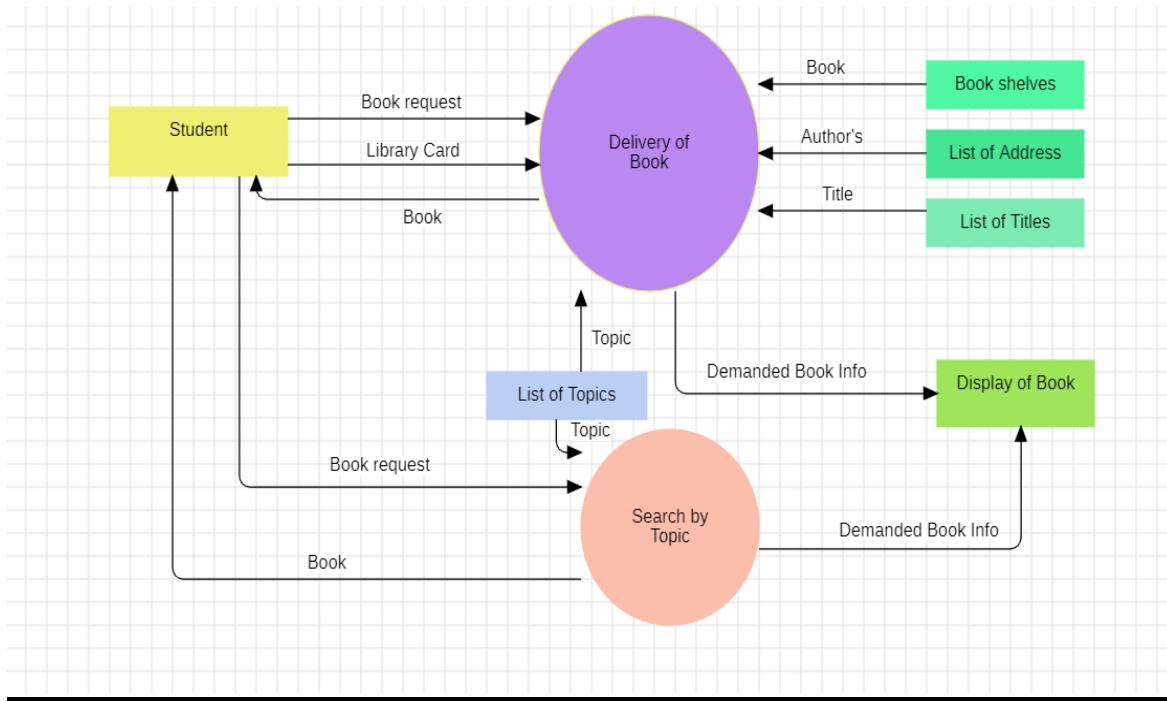
3. Imagine a scenario where an online shopping application on is being designed. Create a sequence diagram and a collaboration on diagram to illustrate the interface between the user, the shopping cart, and the payment system during a typical transaction. Analyse the benefits of using these diagrams in the design process.

Collaboration Diagram:



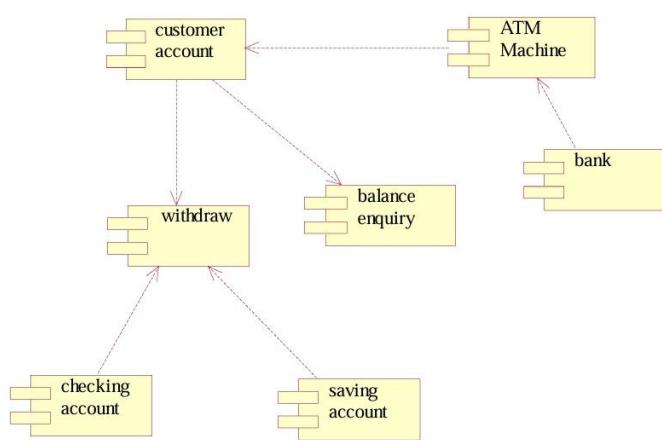
INFORMATION FLOW DIAGRAM

1. Library Management System

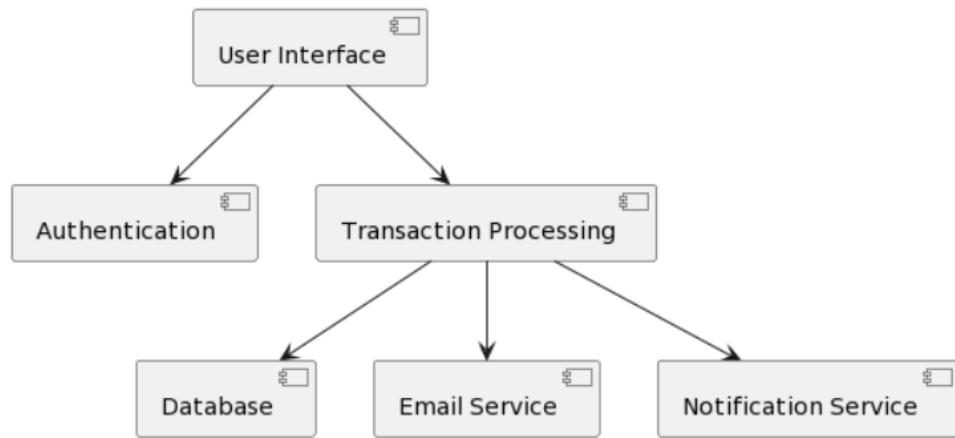


COMPONENT DIAGRAM

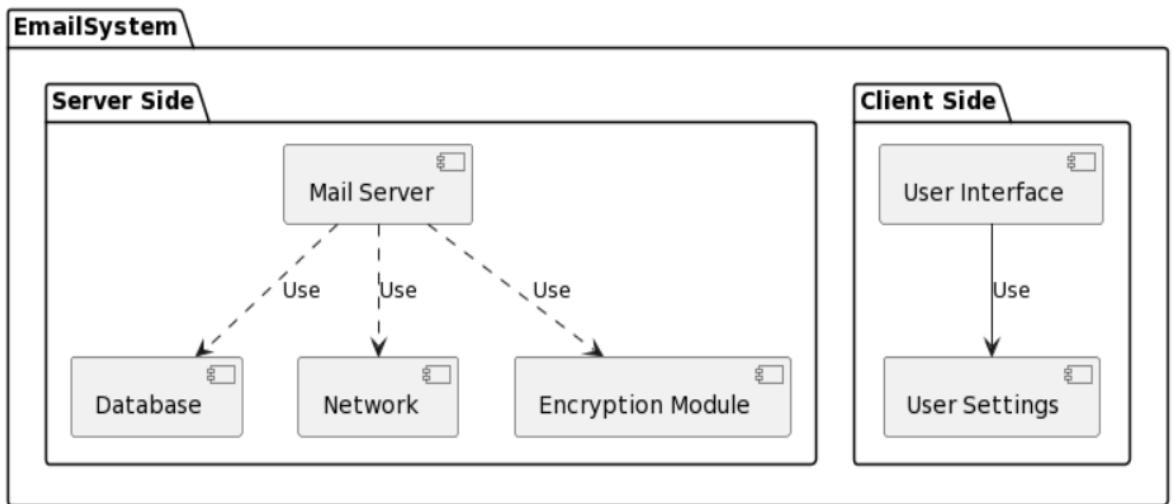
1. ATM



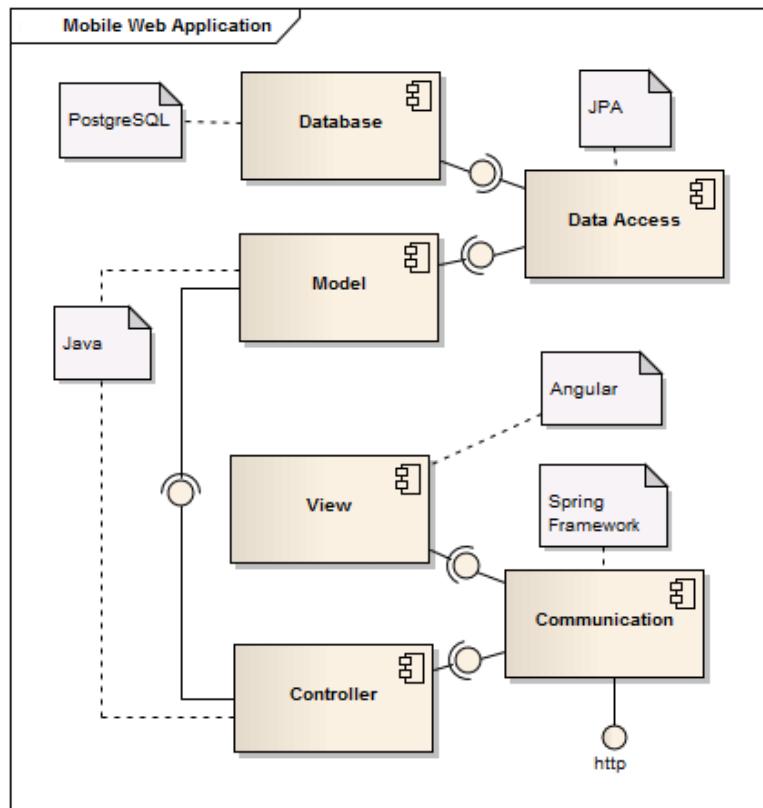
2. Online banking system



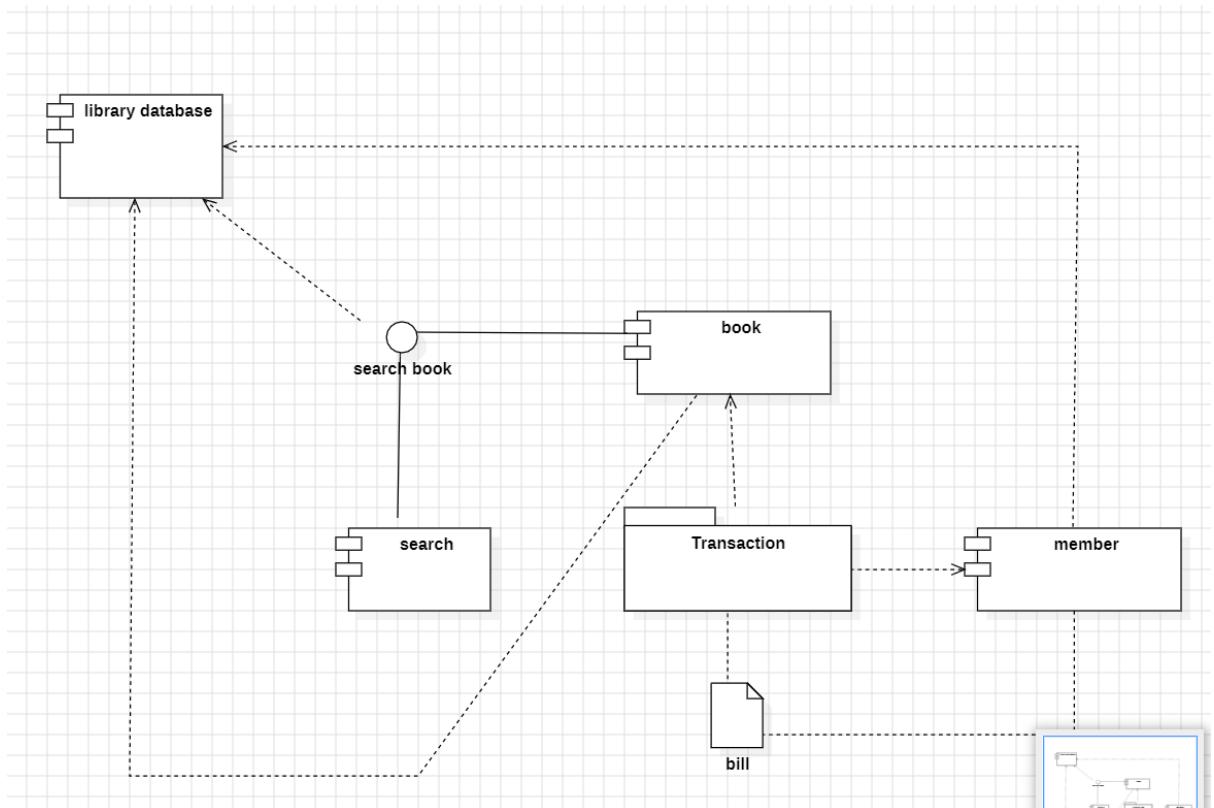
3. Simple email system



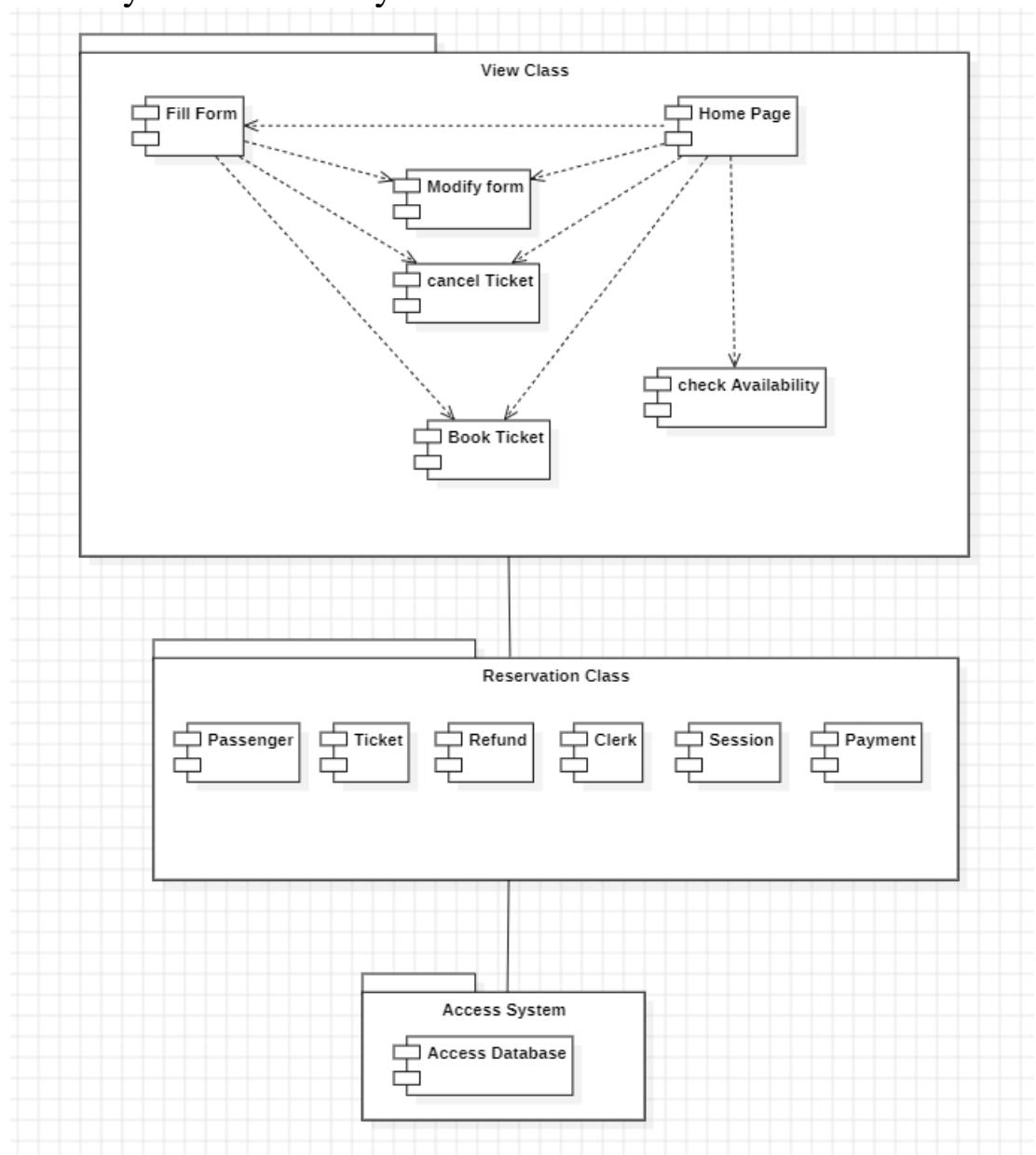
4. A software company is developing a mobile application that integrates with various third-party services such as payment gateways, social media platforms, and analytics tools. Design a component diagram to represent the different components of the mobile application and their dependencies on external services and libraries.



5. Library Management System

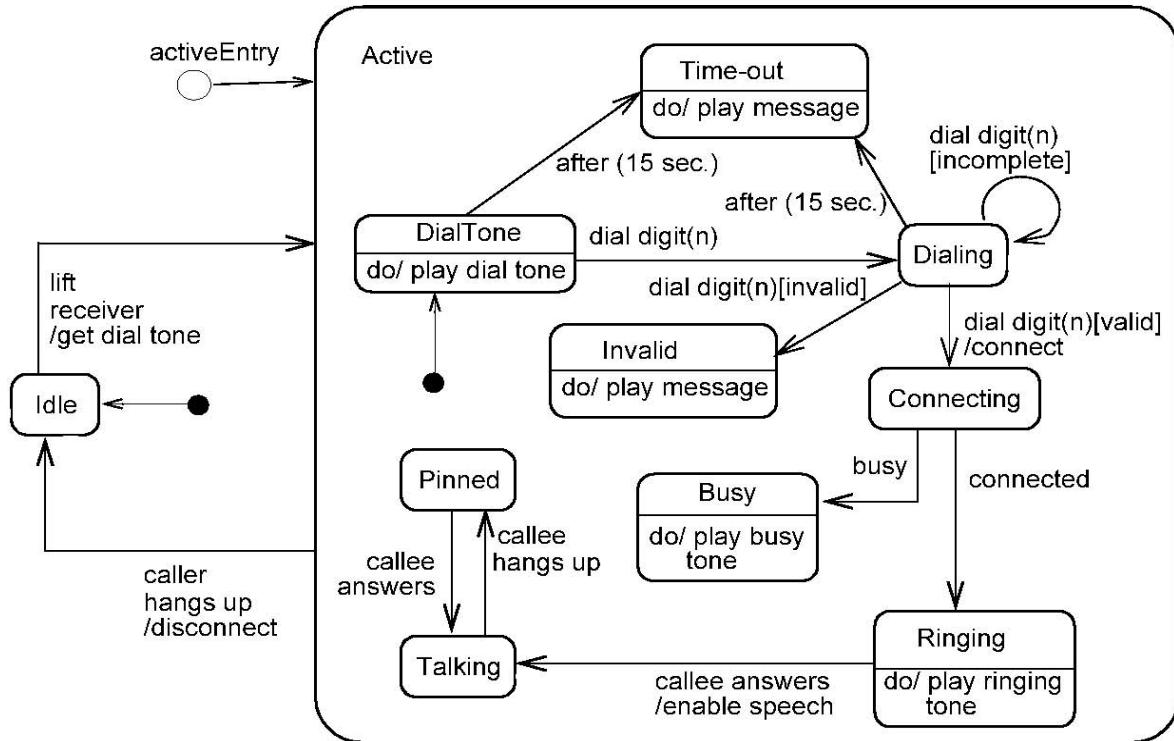


6. Railway Reservation System

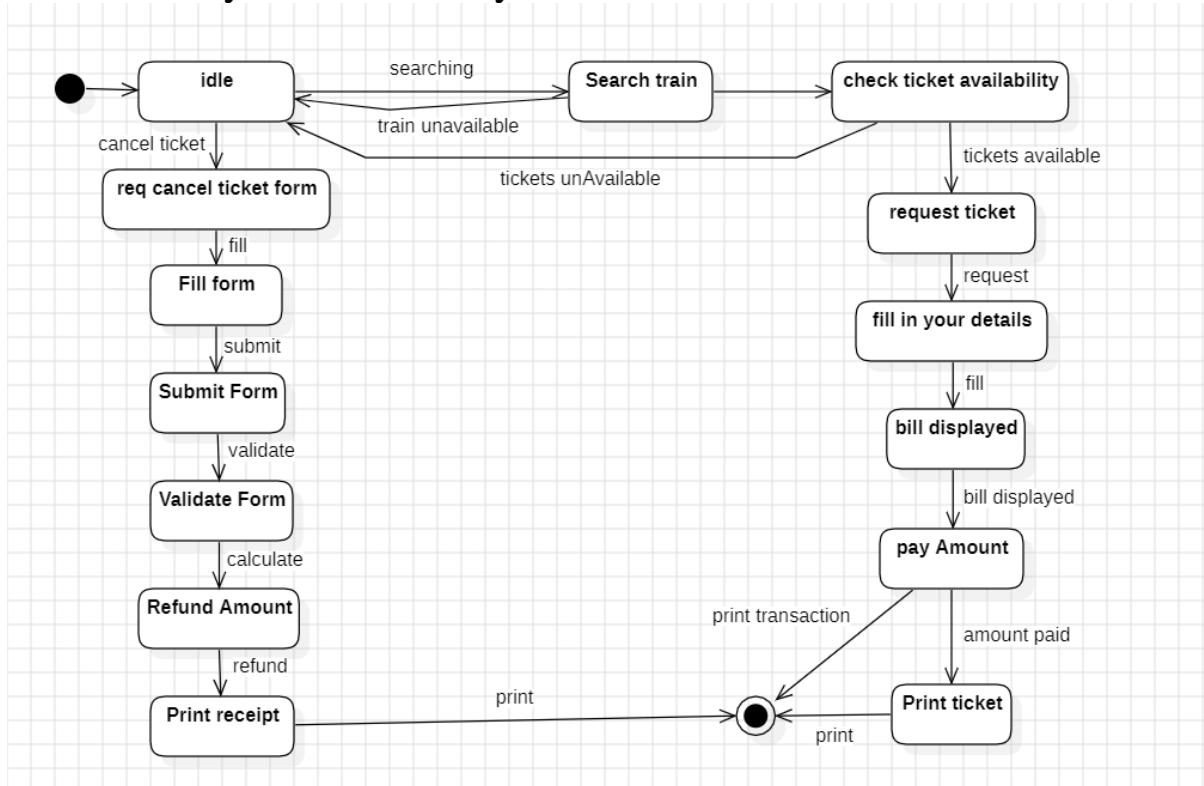


STATE CHART

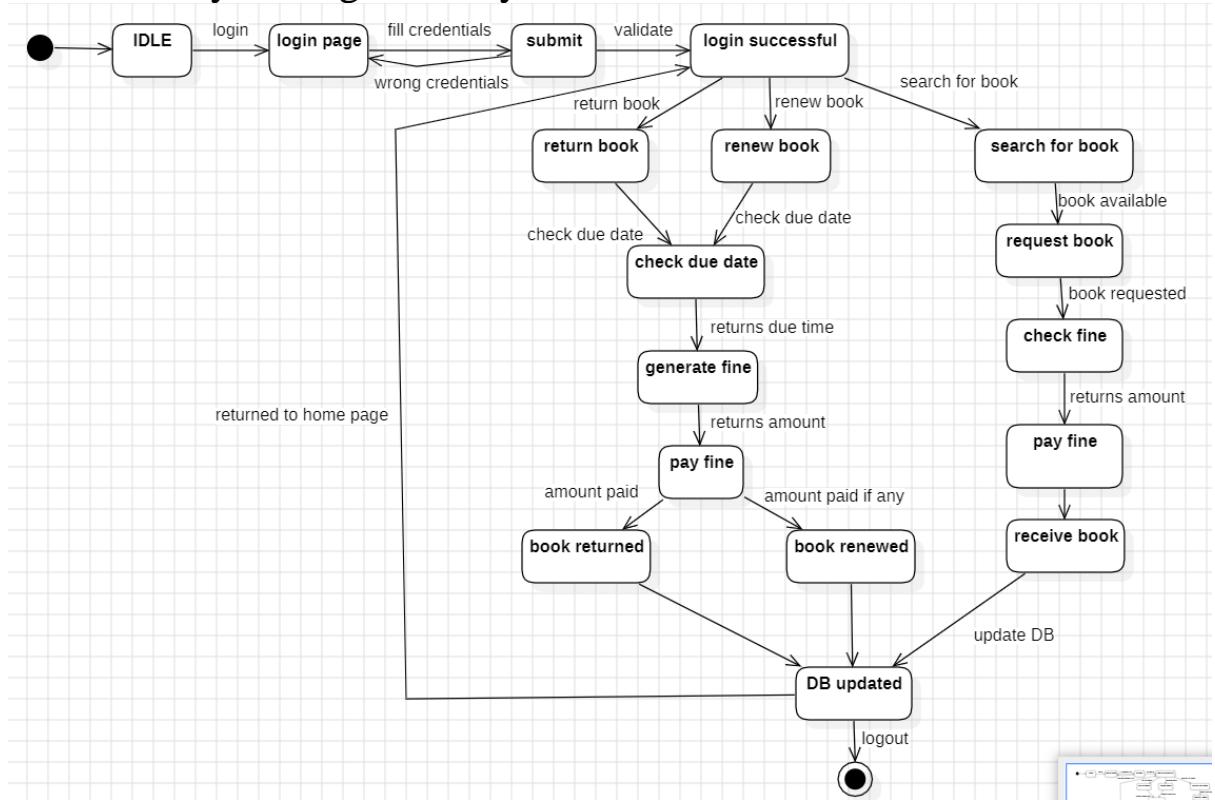
1. ATM System



2. Railway Reservation System

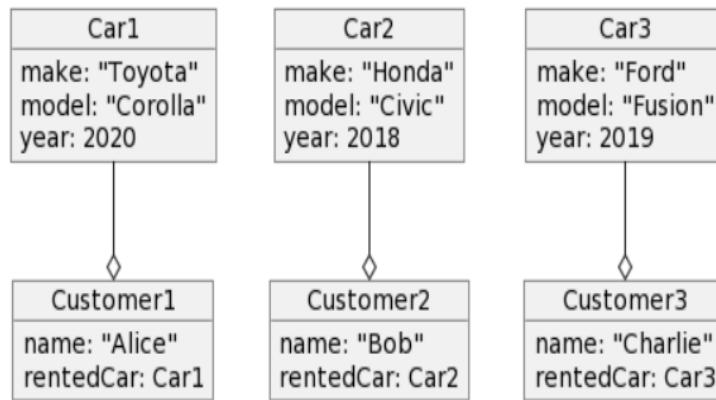


3. Library Management System

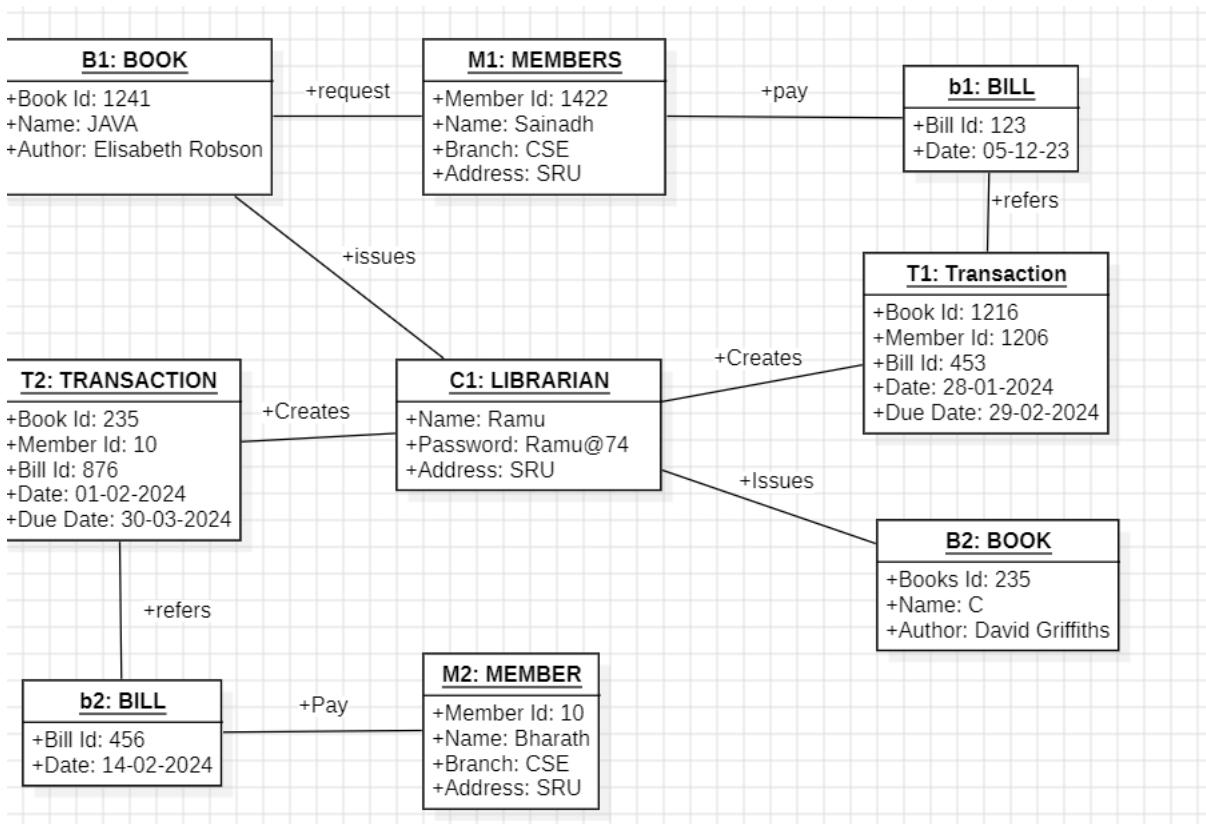


OBJECT DIAGRAM

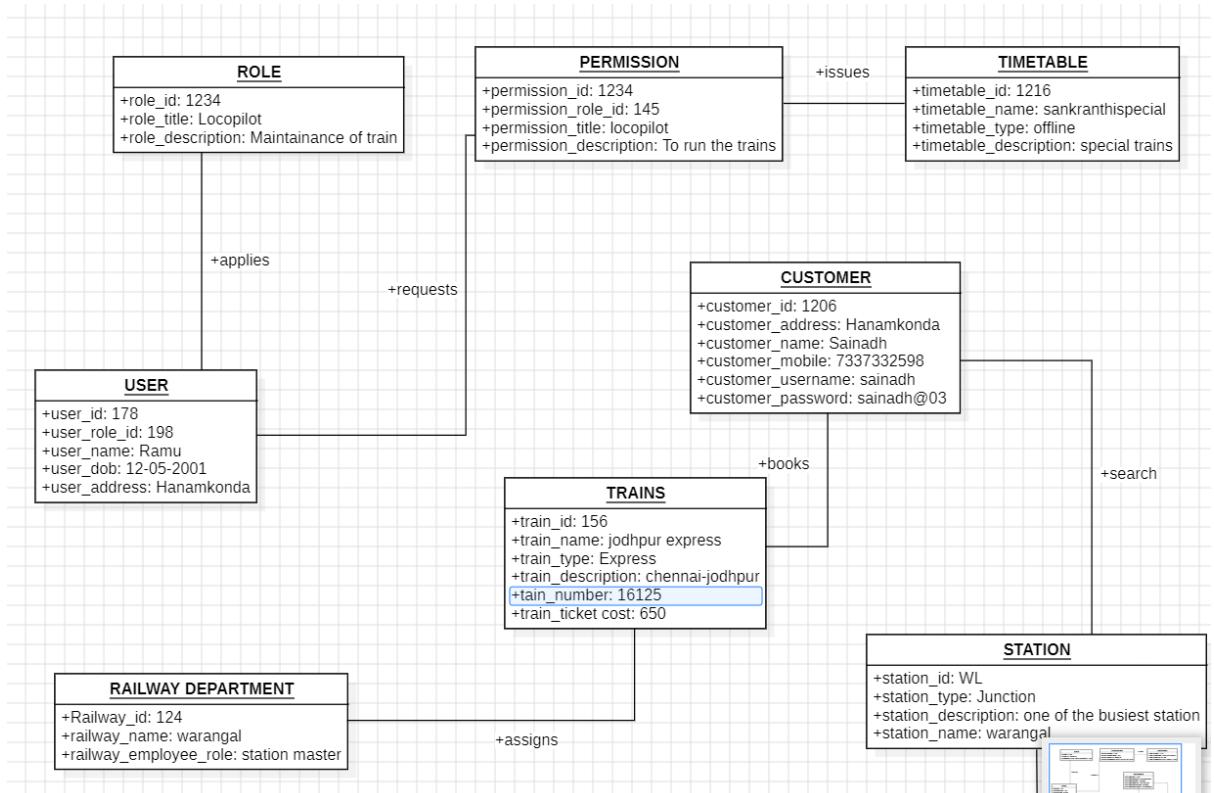
1. Car Rental System



2. Library Management System

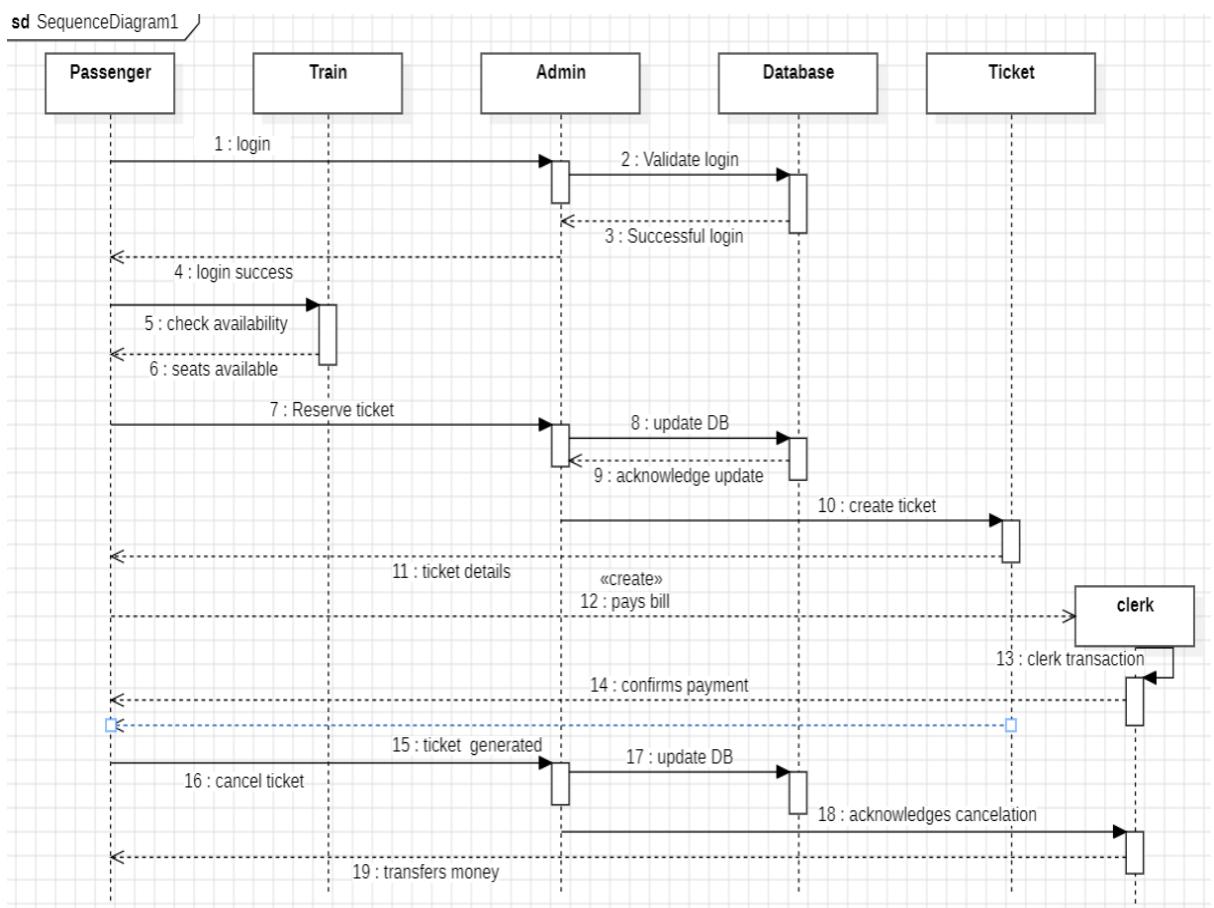


3. Railway Reservation System



INTERACTION SEQUENCE

1. Railway Reservation System



2. Library Management System

