



Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

22CS604 – OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

Submitted by

Class : III CSE - A



SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008

Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

22CS604– OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

Practical record submitted by

Name :

Register No. :

Class : III CSE A

Degree & Branch : B.E CSE

BONAFIDE CERTIFICATE

This is to certify that this is a bonafide record of work done by _____

(Register No.: _____) during the academic year 2024 – 2025.

Faculty In-charge

Head of the Department

Submitted for the University practical examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

INDEX

S. No	Date	Title of the Experiment	Page No.	Marks
1	09.01.2025	Identify a Software System and document the Software Requirement Specification for the identified system.		
2	23.01.2025	Sketch the class diagrams to identify and describe key concepts and their relationships.		
3	29.01.2025	Identify Use Cases and develop the Use Case model.		
4	04.02.2025	Identify the conceptual classes and develop a Domain Model with Class Diagrams.		
5	13.02.2025	Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams.		
6	19.02.2025	Sketch the Activity and State Diagrams for an identified application.		
7	03.03.2025	Sketch the UML package diagram to show the User Interface, Domain objects and Technical services.		
8	08.03.2025	Sketch the component diagram assuming that you will build your system by reusing existing components along with few new components.		
9	14.03.2025	Sketch the deployment diagrams to model the runtime architecture of your application.		
10	20.03.2025	Apply appropriate design patterns to improve the reusability and maintainability of the software system.		



SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade
Kuniamuthur, Coimbatore – 641008
Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

22CS604 – OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

Record of laboratory work

EVEN SEMESTER - 2024-2025

CONTINUOUS EVALUATION SHEET

REFERENCES RUBRICS TABLE

Criteria	Range of Marks			
	Excellent	Good	Average	Below Average
Aim & Algorithm (20)	18-20	14-17	10-13	0-9
Coding (30)	27-30	21-26	15-20	0-14
Compilation and Debugging (30)	27-30	21-26	15-20	0-14
Execution and Result (10)	9-10	7-8	5-6	0-4
Documentation (10)	9-10	7-8	5-6	0-4
Overall Marks	90-100	70-85	50-68	0-45



SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institution | Approved by AICTE | Affiliated to Anna University | Accredited by NAAC with A++ Grade

Kuniamuthur, Coimbatore – 641008

Phone : (0422)-2678001 (7 Lines) | Email : info@skcet.ac.in | Website : www.skcet.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

22CS604 – OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

Record of laboratory work

EVEN SEMESTER - 2024-2025

CONTINUOUS EVALUATION SHEET

Criteria	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8	Exp 9	Exp 10
Aim & Algorithm (20)										
Coding (30)										
Compilation and Debugging (30)										
Execution and Result (10)										
Documentation (10)										
Total										

Faculty Signature

EX.NO:01 DATE:	BANKING MANAGEMENT SYSTEM
---------------------------------	----------------------------------

AIM

To create an automated system to manage bank operations, including account management, transaction processing, and report generation.

PROBLEM STATEMENT

The **Banking Management System** is designed to automate essential banking processes, ensuring efficiency, accuracy, and security in handling transactions and customer records. The system enables online registration, fund transfers, loan management, and real-time balance inquiries. By integrating customer and administrator functionalities into a single platform, the system minimizes manual efforts and streamlines banking operations.

(I) SOFTWARE REQUIREMENT SPECIFICATION

INTRODUCTION

The **Banking Management System (BMS)** provides an interface between customers and the bank, focusing on improving the speed, security, and ease of banking operations.

PURPOSE

The BMS offers a faster, more user-friendly banking experience. It minimizes manual intervention and helps banks scale operations efficiently as customer numbers grow.

SCOPE

- Online interface for customers to manage accounts and transactions.
- Secure, real-time updates for users and administrators.
- Integration with payment systems for seamless processing.
- Automation of reporting and auditing processes.

DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS

- **Admin:** Bank staff managing system operations.
- **Customer:** An individual holding an account.
- **BMS:** Bank Management System.
- **HTTP:** Hypertext Transfer Protocol.
- **TCP/IP:** Transmission Control Protocol/Internet Protocol.

REFERENCES

- IEEE Software Requirement Specification format.
- Banking Compliance Guidelines.

TECHNOLOGIES TO BE USED

- **Frontend:** HTML, CSS, JavaScript.
- **Backend:** Java (Spring Framework).
- **Database:** MySQL.

TOOLS TO BE USED

- Eclipse IDE (Integrated Development Environment)
- Apache Tomcat Server.

OVERVIEW

The **SRS** includes two sections: overall description and specific requirements.

1. **Overall Description:** Explains the components and interconnections of the system.

2. **Specific Requirements:** Details the roles and functions of actors.

(II) OVERALL DESCRIPTION

PRODUCT PERSPECTIVE

The BMS is an integrated solution for both customers and bank staff. It enables secure transactions, efficient operations, and reliable data handling.

SOFTWARE INTERFACE

- **Frontend:** HTML and JSP for customer and admin interfaces.
- **Backend:** Java for business logic.
- **Database:** MySQL for storing account and transaction data.

HARDWARE INTERFACE

Server requirements: Minimum 16GB RAM, 500GB SSD.

SYSTEM FUNCTIONS

- Secure account registration and management.
- Processing transactions in real time.
- Notifications via SMS or email.
- Generation of detailed reports.

USER CHARACTERISTICS

- **Customer:** Interacts via a simple interface for account and transaction management.
- **Admin:** Manages system operations and monitors reports.

CONSTRAINTS

- Requires stable internet connectivity.
- Must comply with PCI DSS for data security.

ASSUMPTIONS AND DEPENDENCIES

- Users need basic computer knowledge.
- External payment gateway integration is required.

(III) USECASE DIAGRAM

The **Banking Management System** use cases are:

1. Login
2. Account Registration
3. Balance Inquiry
4. Fund Transfer
5. Loan Application
6. Approve Loan
7. Transaction History

ACTORS INVOLVED:

1. Customer
2. Administrator
3. External Payment Gateway

USE-CASE NAME

1. LOGIN

Description: Customers and administrators log in to access their respective functionalities.

2. ACCOUNT REGISTRATION

Description: A new customer registers by entering personal and

account details.

3. BALANCE INQUIRY

Description: Customers check their current account balance in real-time.

4. FUND TRANSFER

Description: Customers transfer funds between accounts securely.

5. LOAN APPLICATION

Description: Customers apply for loans, which are reviewed by the administrator.

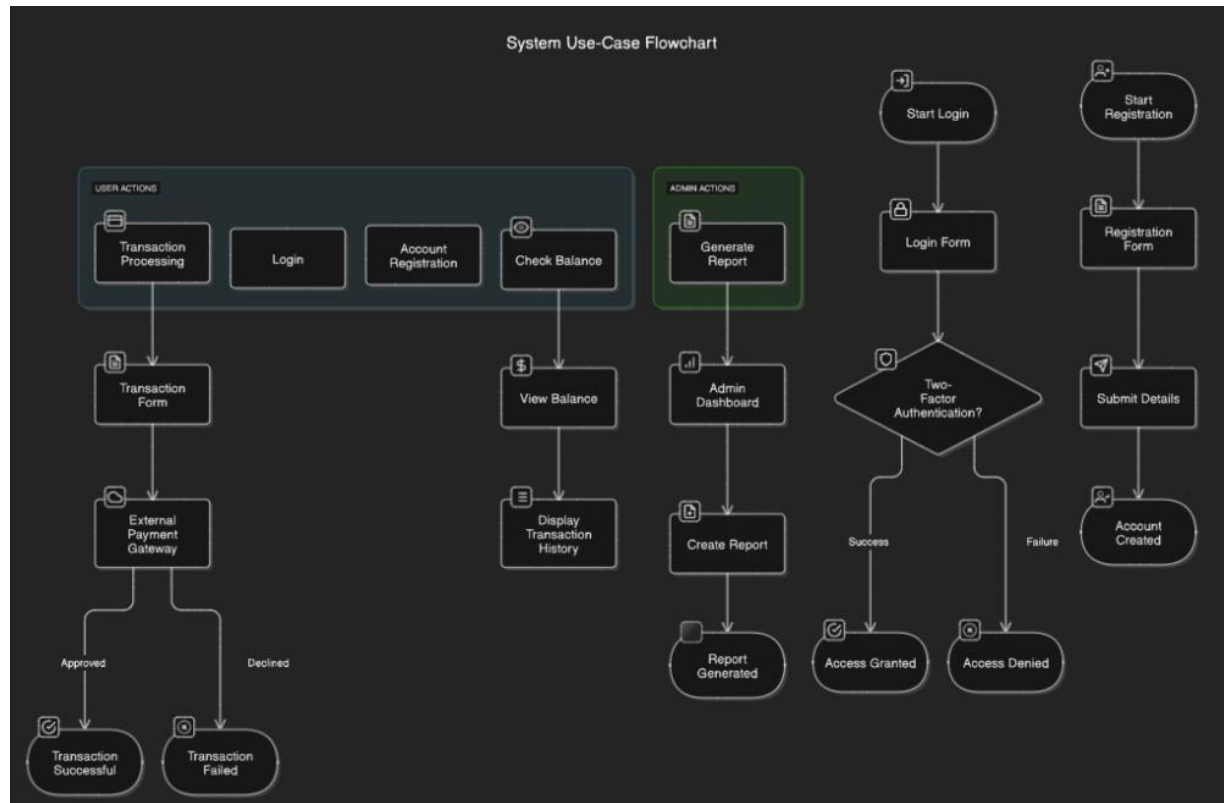
6. APPROVE LOAN

Description: Administrators review and approve/reject loan applications.

7. TRANSACTION HISTORY

Description: Customers view a detailed history of their past transactions.

USE CASE DIAGRAM



(IV) IMPLEMENTATION OF USER INTERFACE LAYER

MEMBER LOGIN

Username

Password

☒ Remember me ☐ Forget password?

LOGIN

RESULT

The Bank Management System simplifies banking operations, enhancing efficiency and user satisfaction. Its modules are ready for deployment and implementation.

Ex No: 2	Class Diagram and Association
Date:	

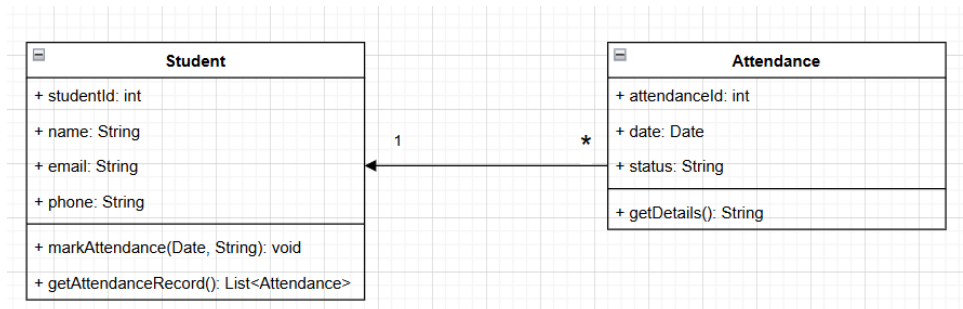
AIM:

To demonstrate a class Diagram with different types of associations or relations.

Association:

An association diagram, commonly referred to as a class diagram, is a visual tool used in software engineering and system design to illustrate the connections among different classes within an object-oriented system.

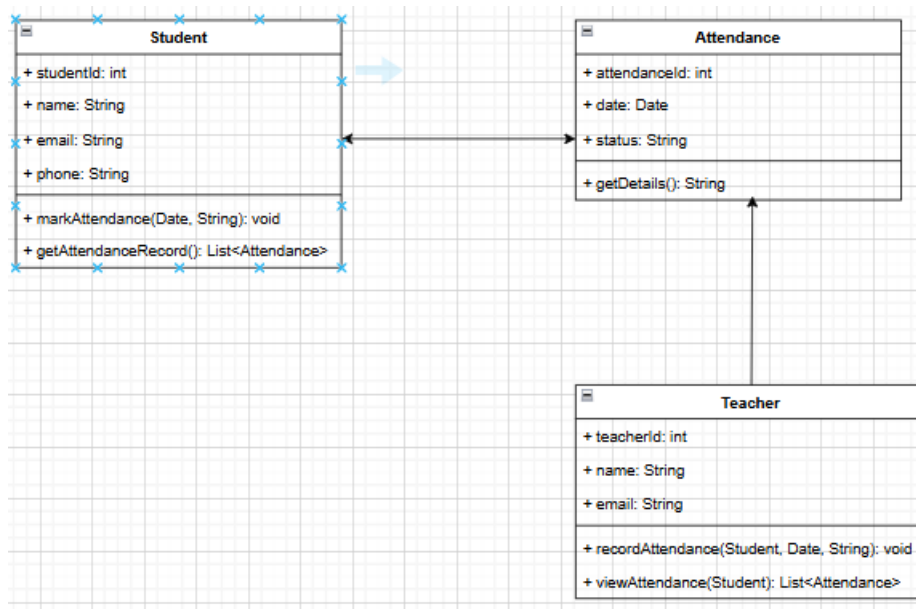
Ex: In the Student Attendance Management System, the Student class and Attendance class are associated together.



Directed Association:

Directed association relationships are associations with navigability in a single direction. They signify that control moves from one classifier to another, such as from a teacher to the attendance records.

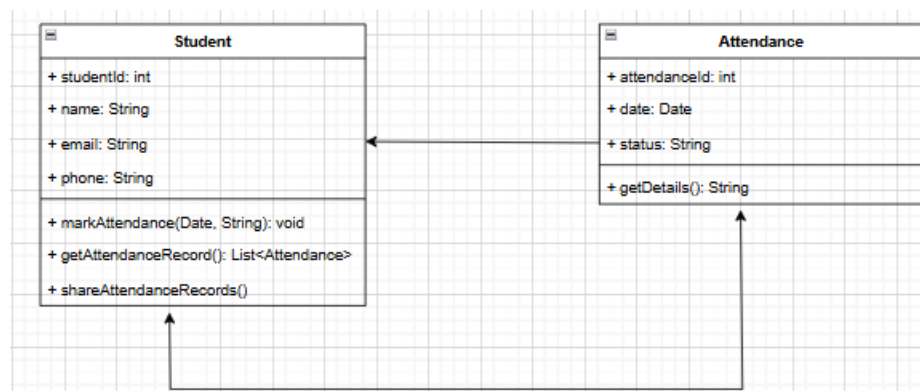
Ex: In the Student Attendance Management System, the Teacher class is directed toward the Attendance class.



Reflexive Association:

A reflexive association, or self-association, emerges when a class establishes a connection with itself. It is useful to represent relationships where an object engages with other objects of the same kind.

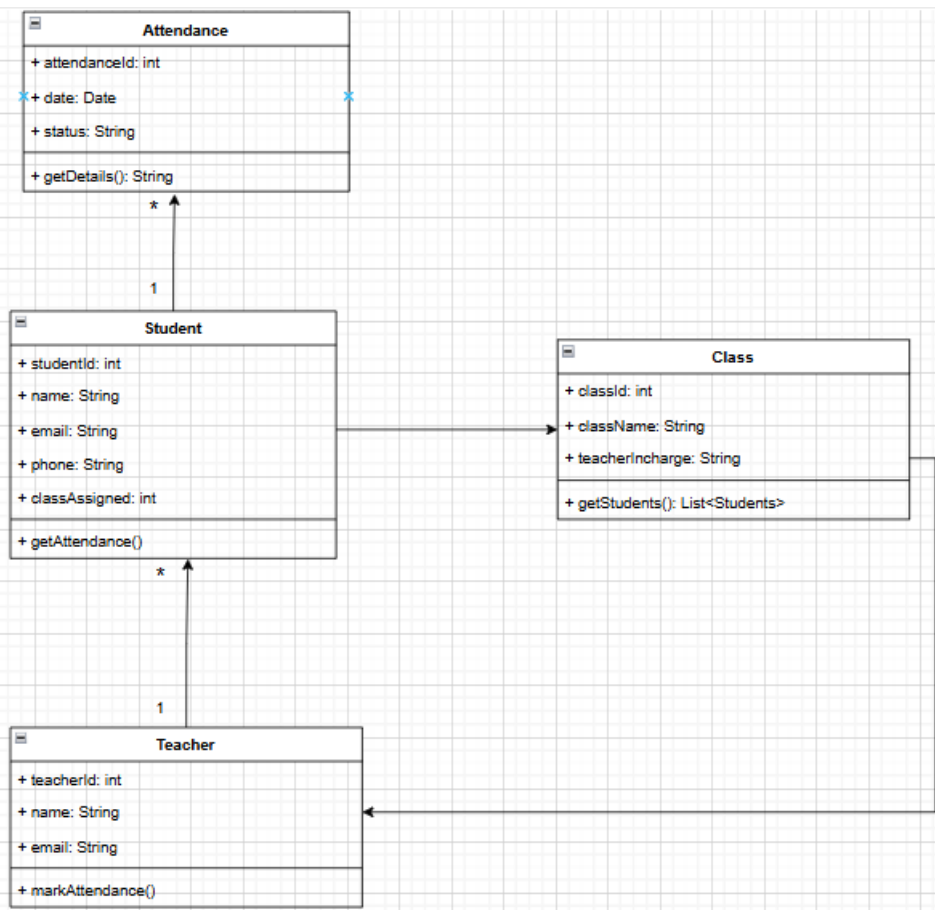
Ex: In the Student Attendance Management System, the Student class can interact with other Student objects, such as sharing group attendance records.



Multiplicity:

Multiplicity defines the numerical extent of a relationship connecting two classes. It shows how many instances of one class relate to instances of another.

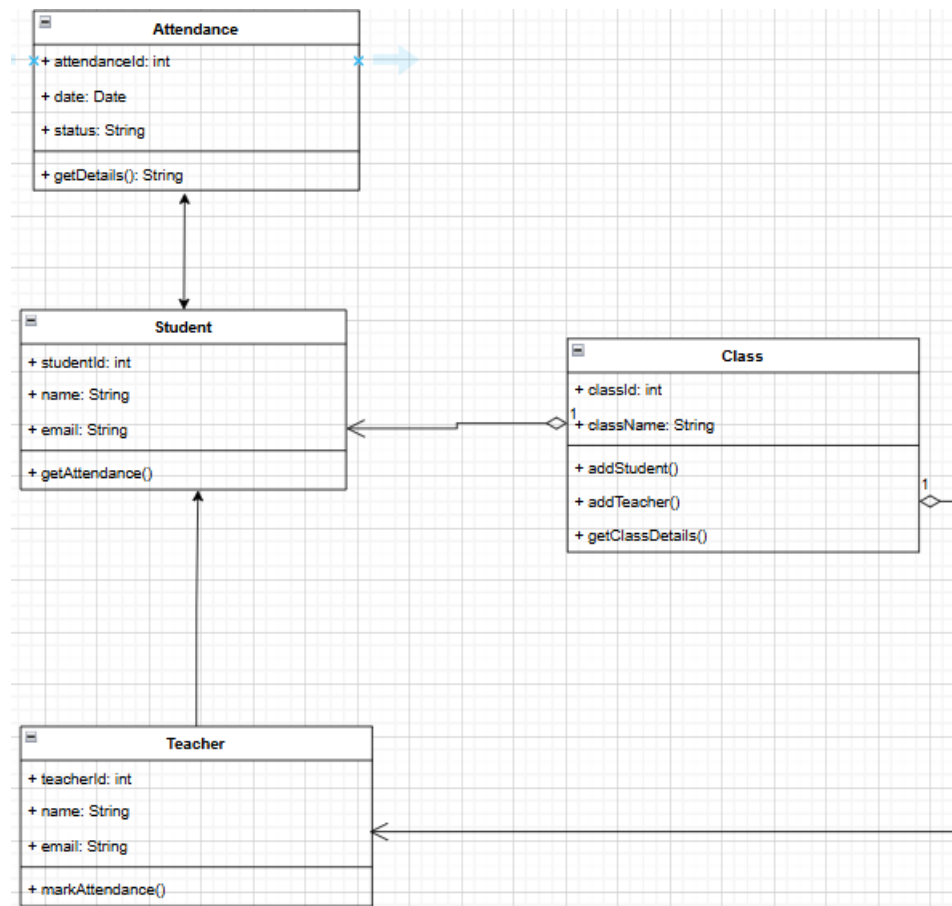
Ex: One Teacher can mark attendance for multiple Students, and each Student belongs to one specific Class.



Aggregation:

Aggregation symbolizes a connection where a part-whole relationship exists, but the child element is independent of the parent.

Ex: In the Student Attendance Management System, the Class aggregates Students and Teachers.



Composition:

Composition illustrates a whole-part relationship where the parts cannot exist independently of the whole.

Ex: In the Student Attendance Management System, Attendance Records cannot exist without the Class they are associated with.\

Generalization:

Generalization occurs when one entity is the parent, and another is the child, inheriting its properties and methods.

Ex: The Person class acts as a parent for the Teacher and Student classes, inheriting common attributes like Name and ID.

Classes, Methods, and Attributes:

1.Student:

A student is an individual associated with the attendance system.

- **Methods:**
 - markAttendance(): Allows the student to mark their attendance.
 - viewAttendance(): Allows the student to check their attendance.
- **Attributes:**
 - Name, ID, Class, Section.

2.Teacher:

A teacher is responsible for managing attendance records.

- **Methods:**
 - markAttendanceForClass(): Marks attendance for all students in a class.
 - viewClassAttendance(): Views attendance records for a class.
- **Attributes:**
 - Name, ID, Department.

3.

Attendance:

Represents the attendance records for students.

- **Methods:**
 - addAttendance(): Adds a new attendance record.
 - getAttendanceByDate(): Retrieves attendance for a specific date.
- **Attributes:**
 - Date, Student ID, Status (Present/Absent).

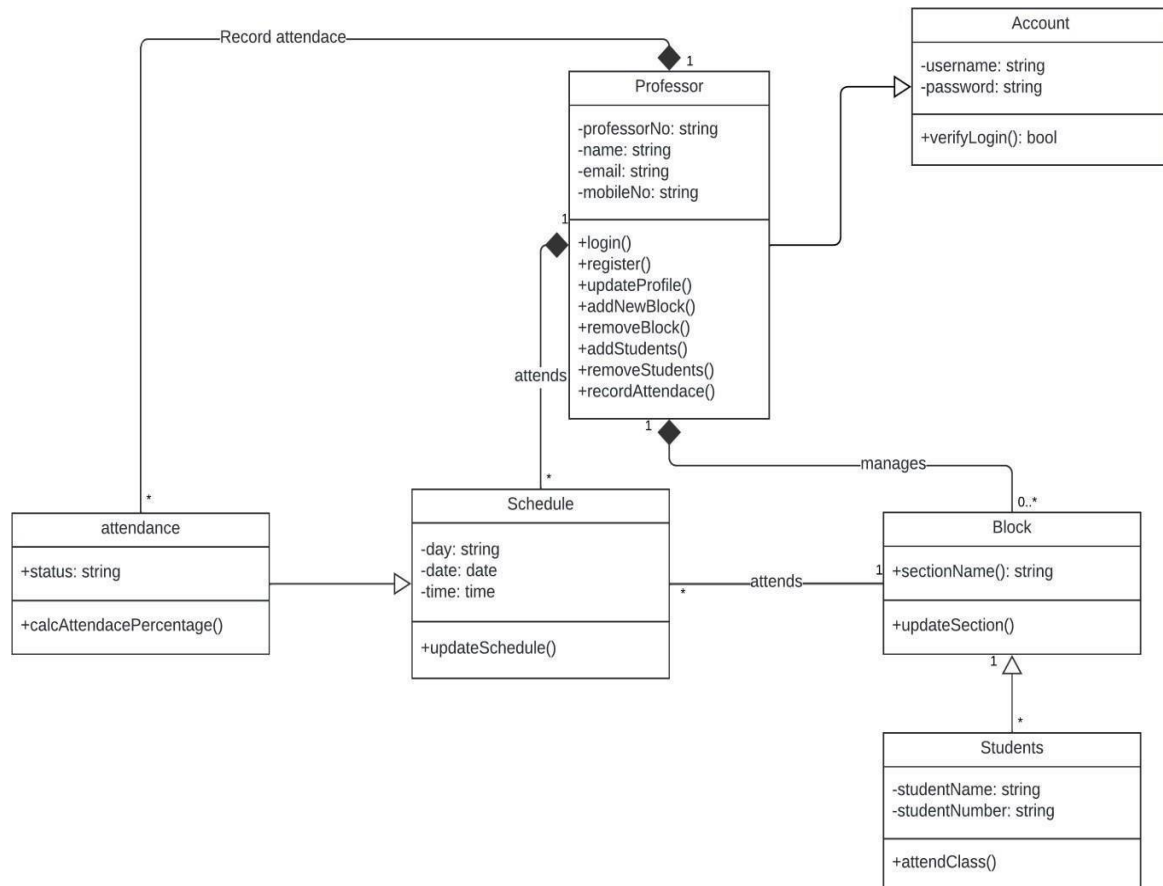
4.

Class:

Represents a group of students and their associated teacher.

- **Methods:**
 - assignTeacher(): Assigns a teacher to the class.
 - getStudents(): Retrieves the list of students in the class.
- **Attributes:**
 - Class Name, Teacher, Students.

Class Diagram



RESULT:

Hence, the Student Attendance Management System class diagram and relationships between the classes were successfully demonstrated

EX.NO:03

DATE:

**IDENTIFY USECASES AND DEVELOP THE
USECASE MODEL**

AIM

To identify the UseCases and develop the UseCase Model for Payroll Management System.

PROCEDURE

Step 1: Identify the System and Boundaries

Define the Payroll Management System for which the use case modeling is to be developed.

Step 2: Identify Actors

Identify all the actors (users, external systems, etc.) interacting with the system.

Step 3: List Use Cases

Identify the main tasks or interactions that the actors have with the system. Each task becomes a use case.

Step 4: Describe Use Cases

Write a brief description for each use case explaining its purpose and steps involved.

Step 5: Create Use Case Diagram

Visualize the relationships between actors and use cases using a use case diagram.

USECASE MODEL

A use case model is a visual representation and description of the functional requirements of the Payroll Management System from the perspective of its users, known as actors. It is used to capture and communicate interactions between users and the system.

Actors:

- **Employee** – Can view salary details and request leave.
- **HR Manager** – Manages employee records, salaries, and approvals.
- **Admin** – Responsible for payroll calculations and system maintenance.
- **Bank System** – Handles salary disbursements.

Include Relationship:

- Payroll processing includes salary calculation.
- Salary disbursement includes tax deduction.

Extends Relationship:

- Leave approval extends from employee request.
- Payroll correction extends from salary disbursement.

3. USECASE SCENARIO FOR PAYROLL MANAGEMENT SYSTEM

1. Employee Payroll Processing:

- The system computes employee salaries based on working hours, overtime, and deductions.
- Tax and provident fund deductions are applied before final salary computation.

2. Salary Disbursement:

- After processing, salaries are transferred to employees' bank accounts.
- The system generates salary slips and notifies employees.

3. Leave and Attendance Management:

- Employees can request leave through the system.
- HR managers review and approve/reject requests.

4. Employee Record Management:

- HR managers can add, update, or remove employee records.
- The system maintains employment history and payroll details.

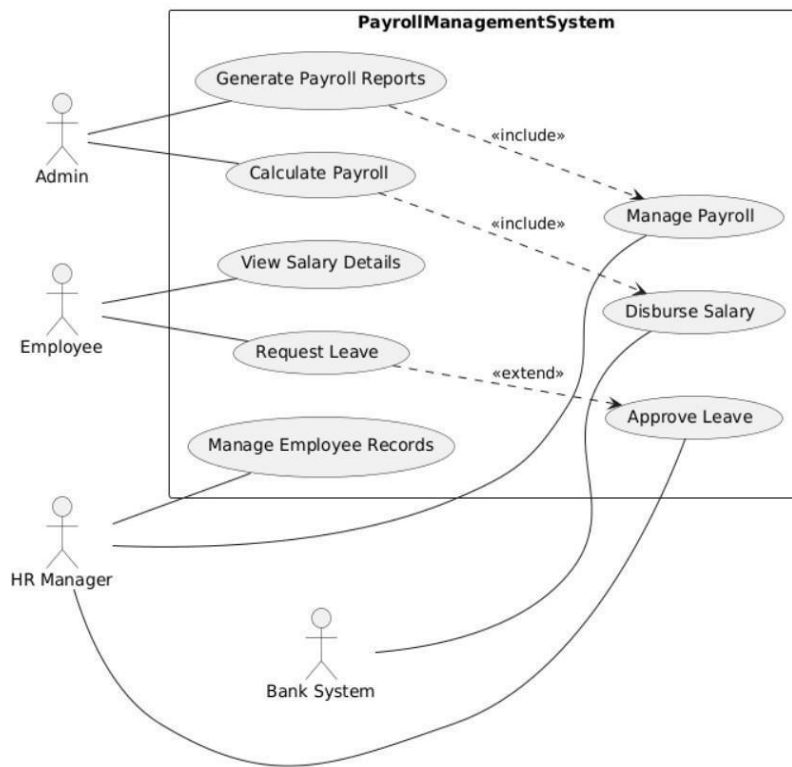
5. Tax and Deduction Management:

- The system calculates tax deductions and generates reports.
- Employees can view their tax details.

6. Payroll Reports Generation:

- The system generates reports on payroll, taxes, and expenses.
- Admins can export reports for auditing purposes

USECASE DIAGRAM FOR PAYROLL MANAGEMENT SYSTEM



RESULT

Thus, the Use Cases are identified and the UseCase Model is developed for the Payroll Management .

EX.NO:04

DATE:

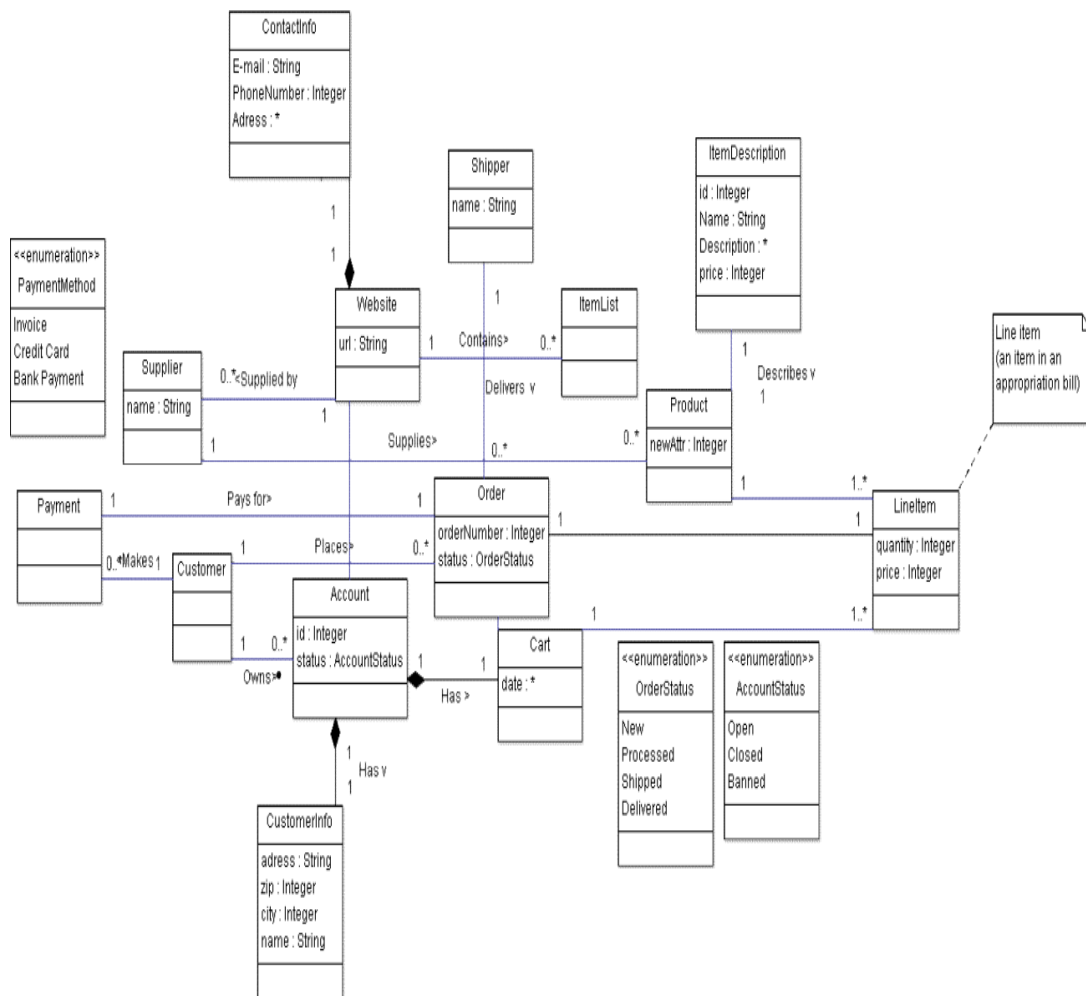
DOMAIN MODEL WITH DIAGRAM

OBJECTIVE:

To design a new online marketplace platform and identify the conceptual classes of the platform and create a comprehensive domain model with class diagram.

Problem Statement:

Each online store has an admin who controls one or many items in the website. Customers can place orders and make payments. Each order has shipping information.



Result:

Thus the UML domain model with class diagram was created for online market place application

EX NO.5	Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams
DATE:	

AIM:

To draw the diagrams [use case, activity, sequence, collaboration, class] for the E-ticketing system.

HARDWARE REQUIREMENTS:

Intel Pentium Processor 3

SOFTWARE REQUIREMENTS:

Rational rose / Visual Basic

PROJECT DESCRIPTION:

This software is designed for supporting the computerized e-ticketing. This is widely used by the passenger for reserving the tickets for their travel. This E-ticketing is organized by the central system. The information is provided from the railway reservation system

SEQUENCE DIAGRAM:

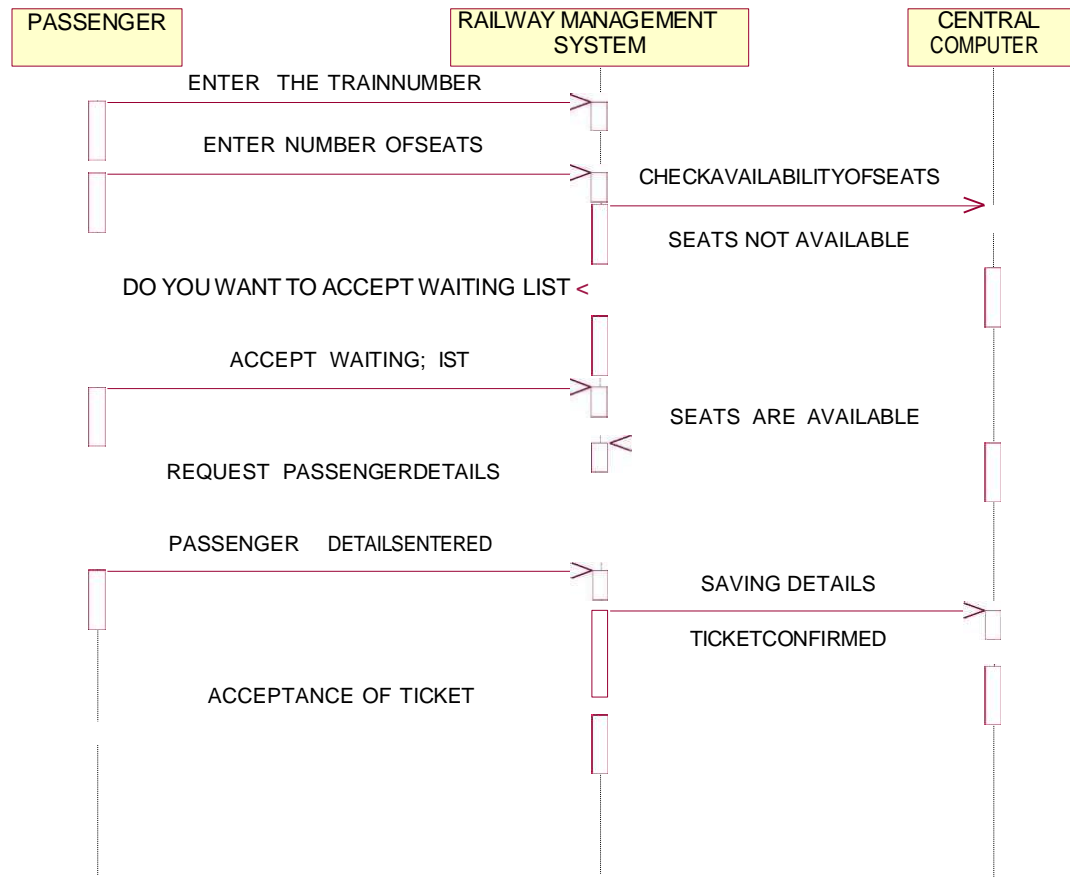
This diagram consists of the objects, messages and return messages.

Object: Passenger, Railway reservation system, Central computer

COLLABORATION DIAGRAM:

This diagram contains the objects and actors. This will be obtained by the completion of the sequence diagram and pressing the F5 key.

SEQUENCE DIAGRAM:



COLLABORATION DIAGRAM:

- 1: ENTER THE TRAIN NUMBER
- 2: ENTER NUMBER OF SEATS
- 6: ACCEPT WAITING LIST
- 9: PASSENGER DETAILS ENTERED

PASSENG
ER

- 5: DO YOU WANT TO ACCEPT WAITING LIST
- 8: REQUEST PASSENGER DETAILS
- 12: ACCEPTANCE OF TICKET

RAILWAYMANAGEMENT
SYSTEM

- 4: SEATS NOT AVAILABLE
- 7: SEATS ARE AVAILABLE
- 11: TICKET CONFIRMED

- 3: CHECK AVAILABILITY OF SEATS
- 10: SAVING DETAILS

RESULT:

Thus the diagrams [sequence, collaboration] for the E-ticketing has been designed, executed and output is verified.

EX NO.6	Draw the Activity and State Diagrams for given application.
DATE:	a) Stock MaintenanceSystem. b) Passport Automationsystem.

PASSPORT AUTOMATION SYSTEM

AIM:

To draw the diagrams [activity, state] for the Passport automation system.

HARDWARE REQUIREMENTS:

Intel Pentium Processor 3

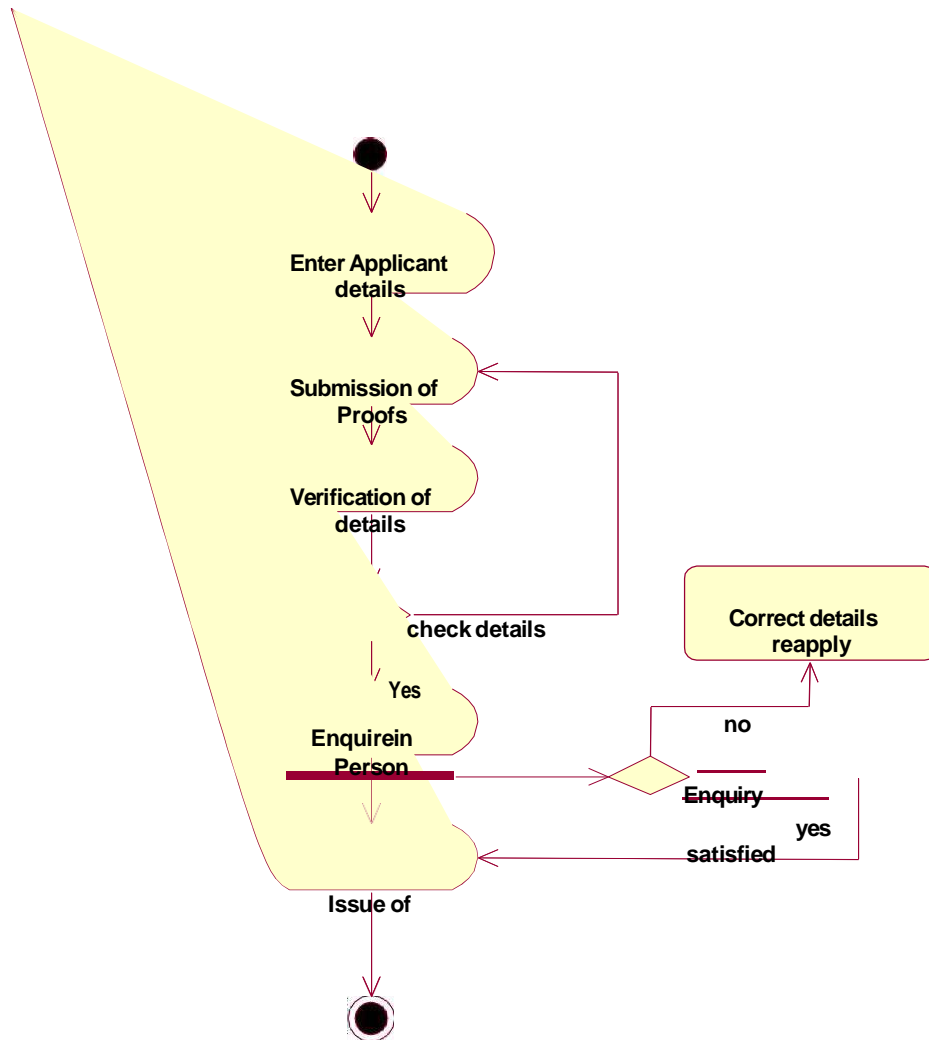
SOFTWARE REQUIREMENTS:

AGROUML

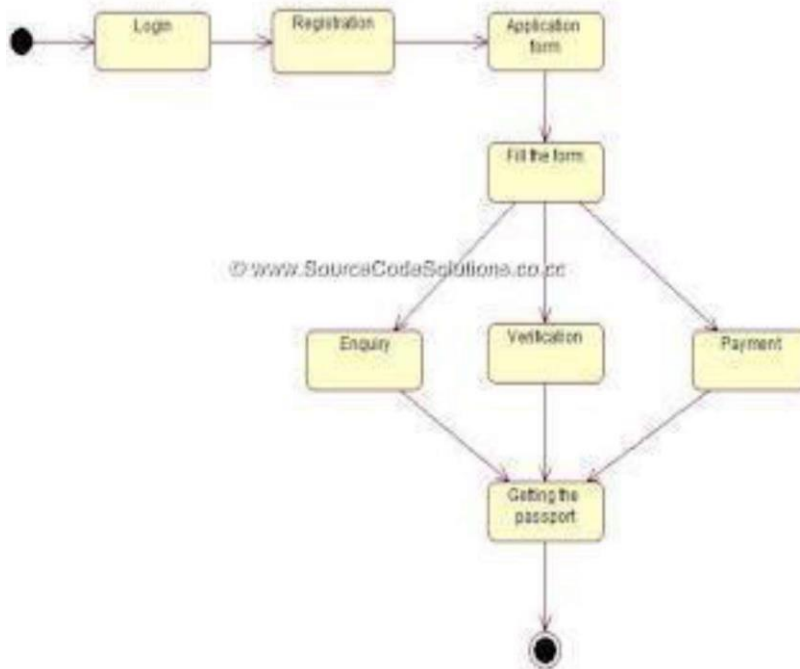
PROJECT DESCRIPTION:

This software is designed for the verification of the passport details of the applicant by the central computer. The details regarding the passport will be provided to the central computer and the computer will verify the details of applicant and provide approval to the office. Then the passport will issue from the office to the applicant.

ACTIVITY DIAGRAM:



STATE DIAGRAM:



RESULT:

Thus the diagrams [Use case, class, activity, sequence, collaboration] for the passport automation system has been designed, executed and output is verified.

EX NO.7	Draw the UML package diagram to show the User Interface, Domain objects and Technical services.
DATE:	

AIM: To draw the UML Package Diagram for Ticket Reservation System and Bank Management System

HARDWARE REQUIREMENTS:

Intel Pentium Processor 3

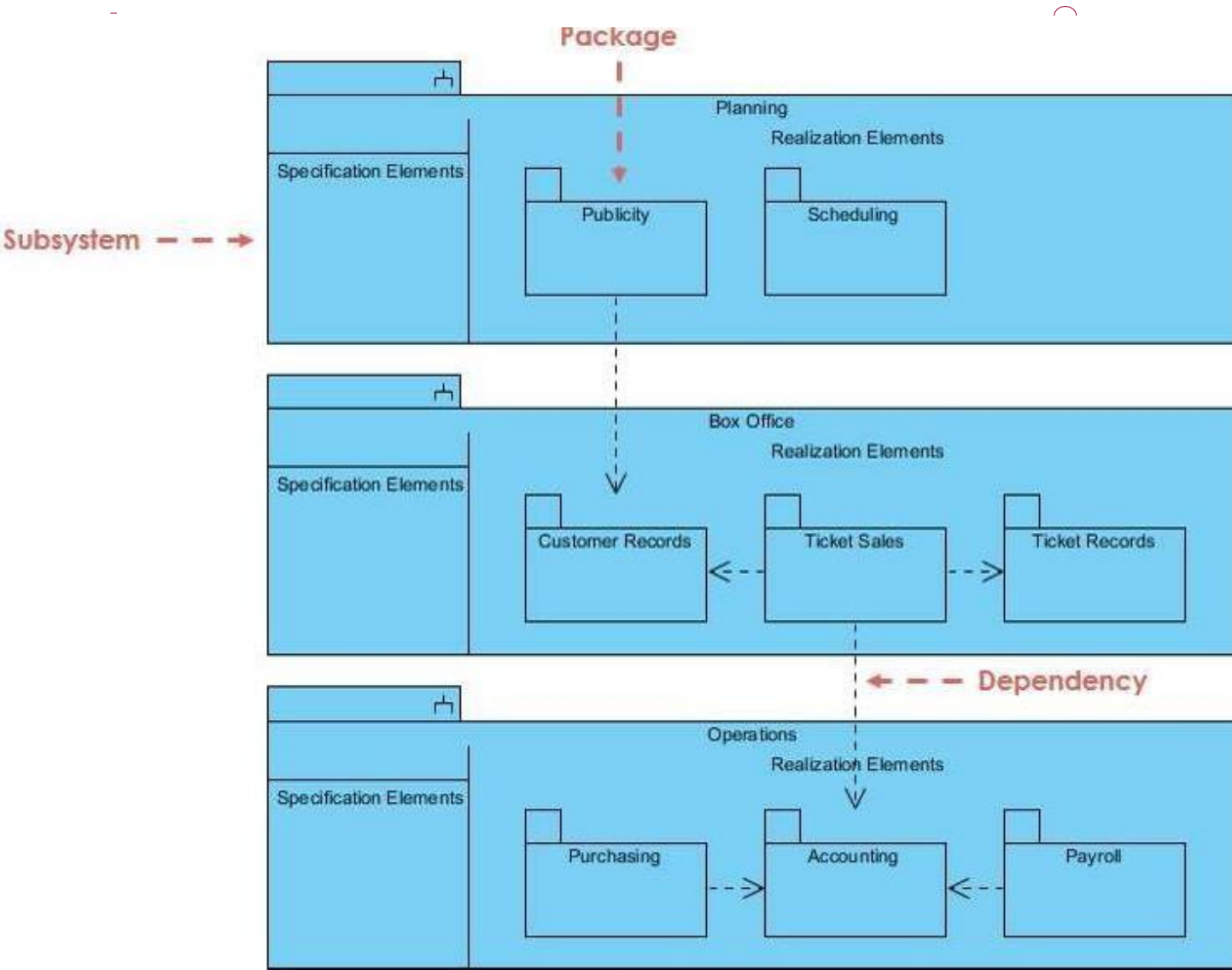
SOFTWARE REQUIREMENTS:

AGROUML

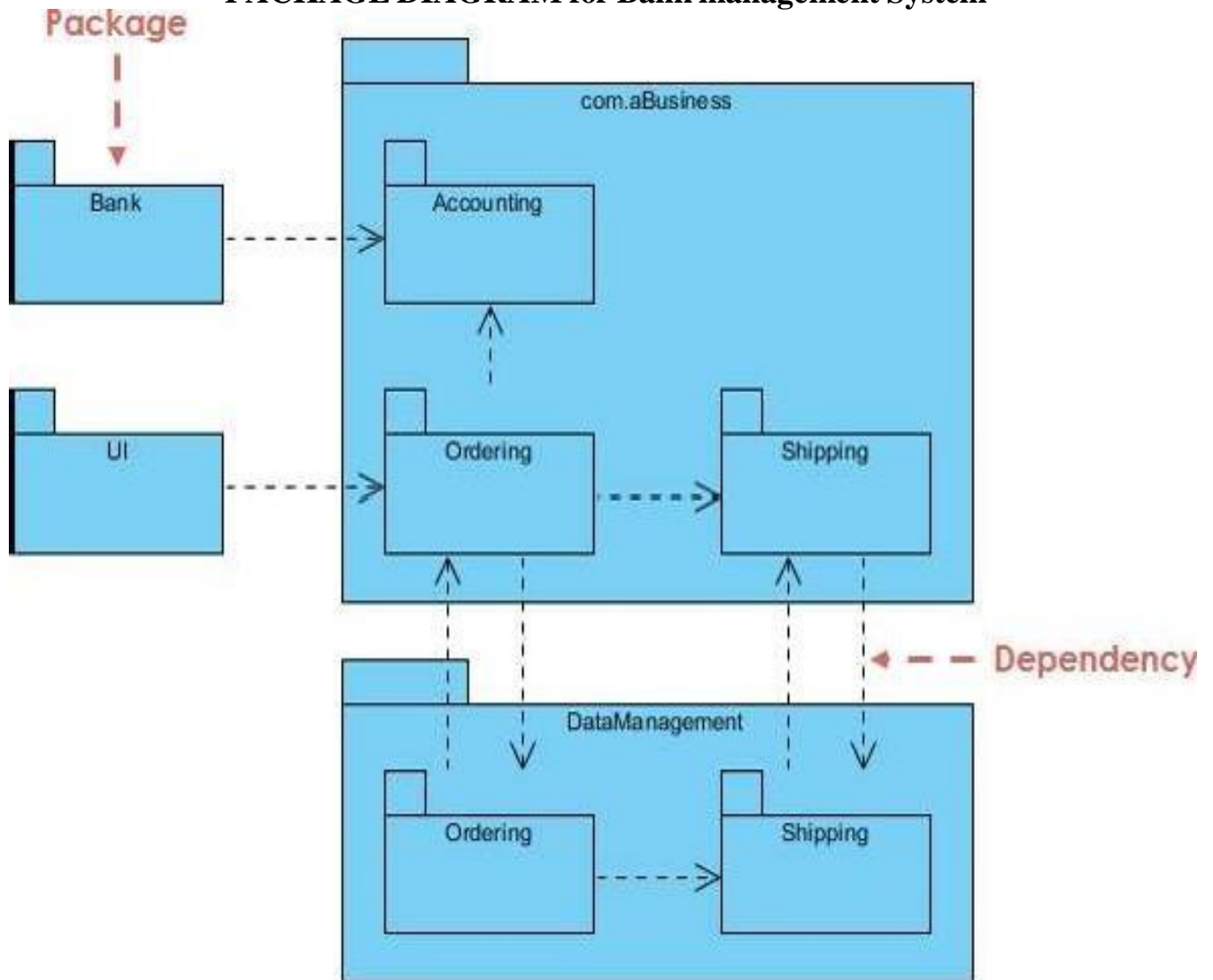
PROJECT DESCRIPTION:

A package diagram is represented as a folder shown as a large rectangle with a top attached to its upper left corner. A package may contain both sub ordinate package and ordinary model elements. All uml models and diagrams are organized into package. A package diagram in unified modeling language that depicts the dependencies between the packages that make up a model. A Package Diagram (PD) shows a grouping of elements in the OO model, and is a Cradle extension to UML. PDs can be used to show groups of classes in Class Diagrams (CDs), groups of components or processes in Component Diagrams (CPDs), or groups of processors in Deployment Diagrams (DPDs). There are three types of layer. They are a. User interface layer b. Domain layer c. Technical services layer

PACKAGE DIAGRAM for Ticket Reservation System



PACKAGE DIAGRAM for Bank management System



RESULT:

Thus the UML Package diagrams for the Ticket Reservation system and Bank management system has been designed, executed and output is verified

EX NO.8	Draw component diagram assuming that you will build your system by
DATE:	reusing existing components along with few new components.

AIM:

To draw the component diagrams for Ticket Reservation System and Bank Management System

HARDWARE REQUIREMENTS:

Intel Pentium Processor 3

SOFTWARE REQUIREMENTS:

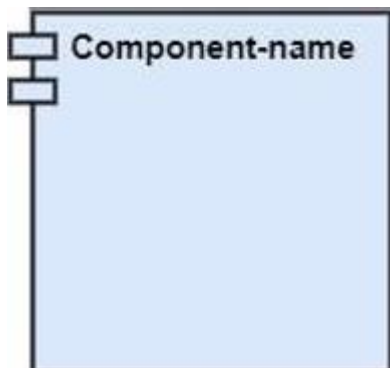
AGROUML

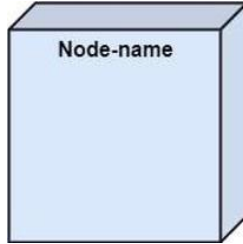
PROJECT DESCRIPTION:

A component diagram is used to break down a large object-oriented system into the smaller components, so as to make them more manageable. It models the physical view of a system such as executables, files, libraries, etc. that resides within the node. It visualizes the relationships as well as the organization between the components present in the system. It helps in forming an executable system. A component is a single unit of the system, which is replaceable and executable. The implementation details of a component are hidden, and it necessitates an interface to execute a function. It is like a black box whose behavior is explained by the provided and required interfaces.

Notation of a Component Diagram

a) A component





Purpose of a Component Diagram

Since it is a special kind of a UML diagram, it holds distinct purposes. It describes all the individual components that are used to make the functionalities, but not the functionalities of the system. It visualizes the physical components inside the system. The components can be a library, packages, files, etc.

The component diagram also describes the static view of a system, which includes the organization of components at a particular instant. The collection of component diagrams represents a whole system.

The main purpose of the component diagram are enlisted below:

1. It envisions each component of a system.
2. It constructs the executable by incorporating forward and reverse engineering.

Component DIAGRAM for Ticket Reservation System

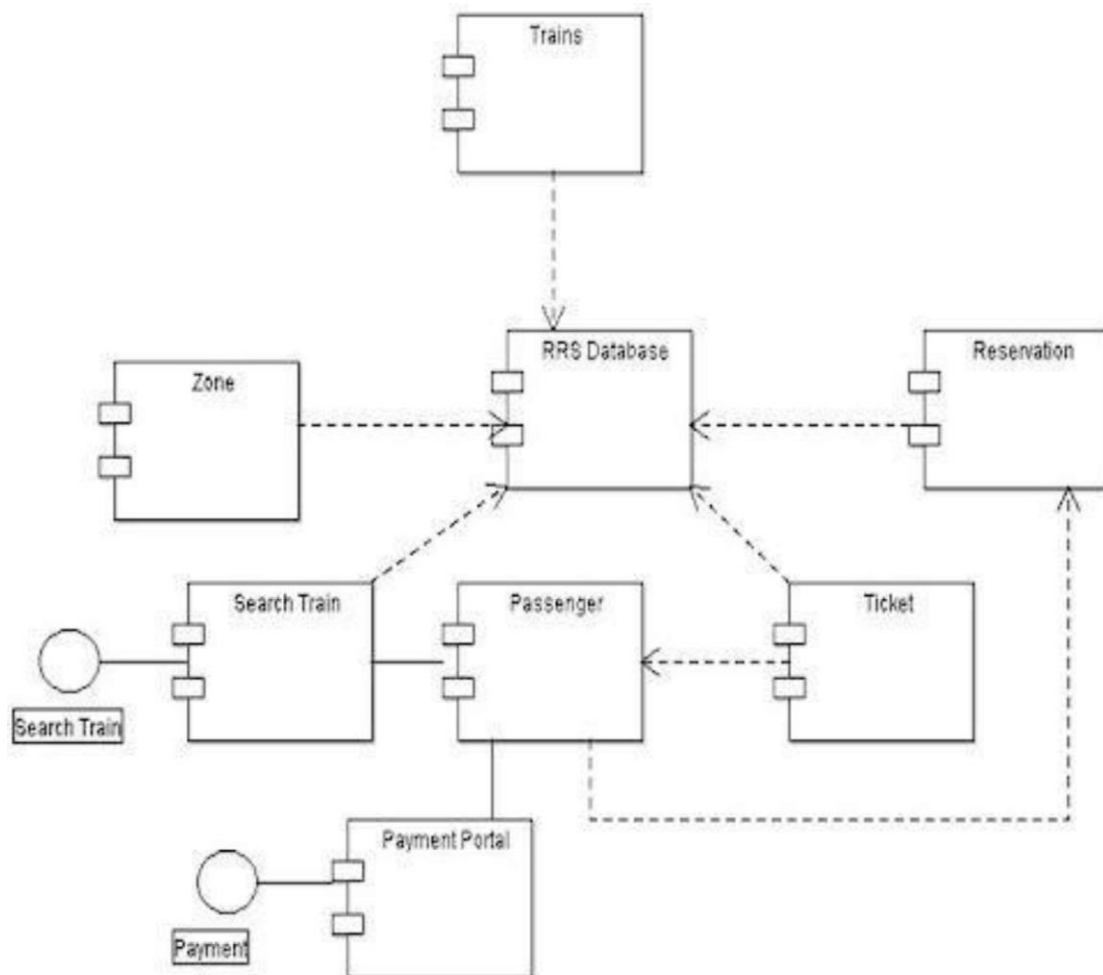
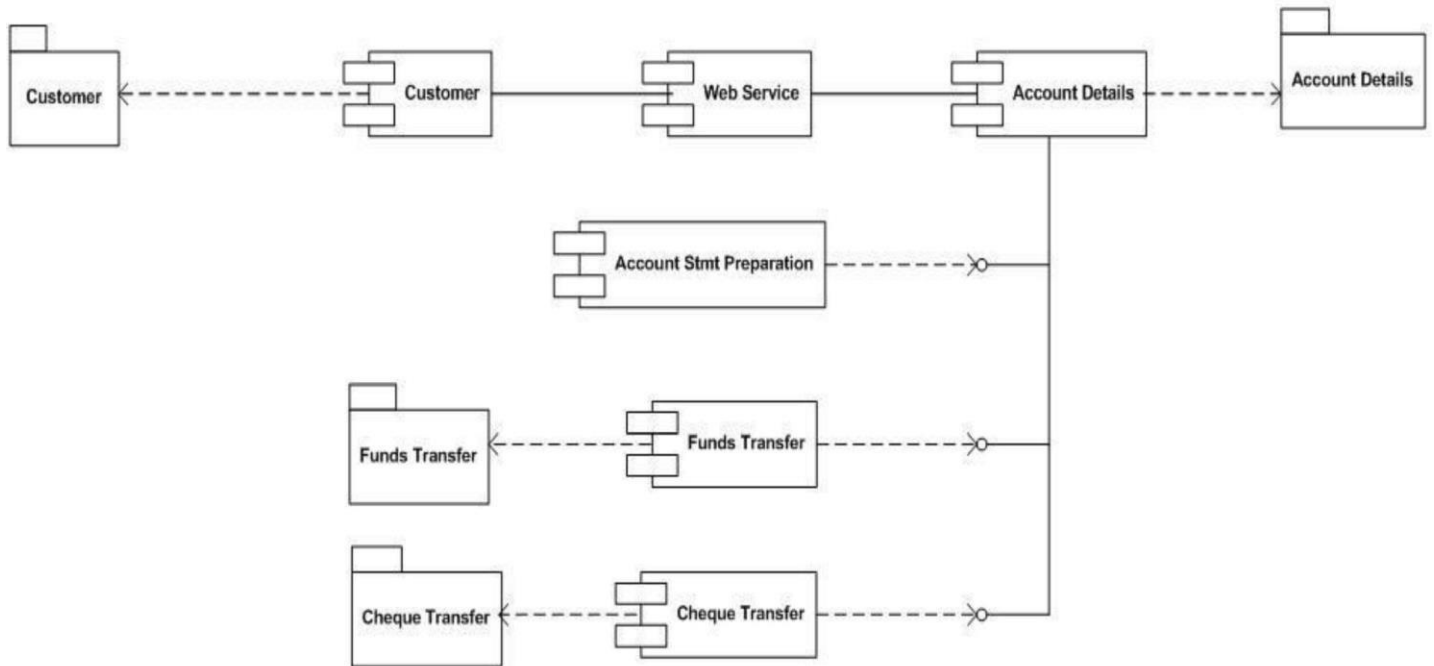


fig. Component Diagram for Railway Reservation System

Component DIAGRAM for Bank management System

Component diagram for Internet banking



RESULT:

Thus the Component diagrams for the Ticket Reservation system and Bank management system has been designed, executed and output is verified

EX NO.9	Draw deployment diagrams to model the runtime architecture of your application
DATE:	

AIM:

To draw the Deployment Diagrams for Ticket Reservation System and Bank Management System

HARDWARE REQUIREMENTS:

Intel Pentium Processor 3

SOFTWARE REQUIREMENTS:

AGROUML

PROJECT DESCRIPTION:

The deployment diagram visualizes the physical hardware on which the software will be deployed. It portrays the static deployment view of a system. It involves the nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

Purpose of Deployment Diagram

The main purpose of the deployment diagram is to represent how software is installed on the hardware component. It depicts in what manner a software interacts with hardware to perform its execution.

Both the deployment diagram and the component diagram are closely interrelated to each other as they focus on software and hardware components. The component diagram represents the components of a system, whereas the deployment diagram describes how they are actually deployed on the hardware.

The deployment diagram does not focus on the logical components of the system, but it put its attention on the hardware topology.

Following are the purposes of deployment diagram enlisted

below: To envision the hardware topology of the system.

To represent the hardware components on which the software components are installed.

To describe the processing of nodes at the runtime.

Symbol and notation of Deployment diagram

The deployment diagram consist of the following

notations: A component

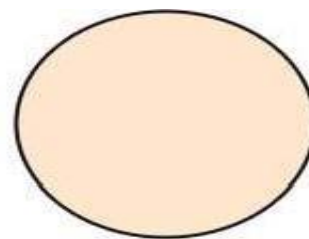
An

artifact

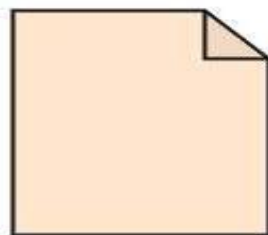
An

interface

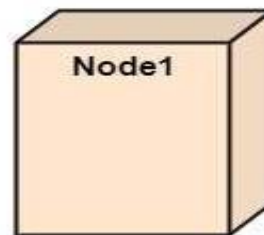
A node



Interface1

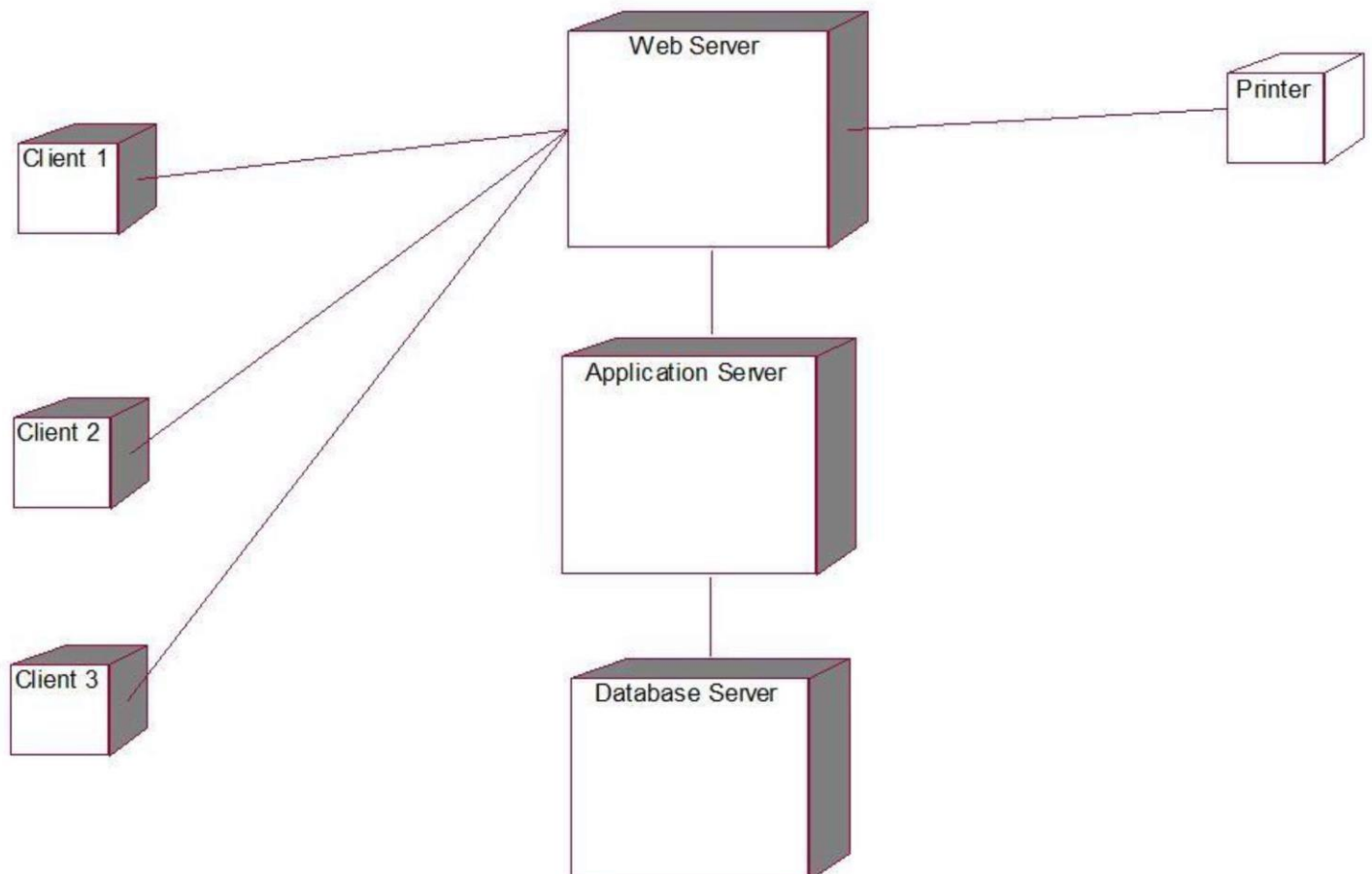


Artifact1

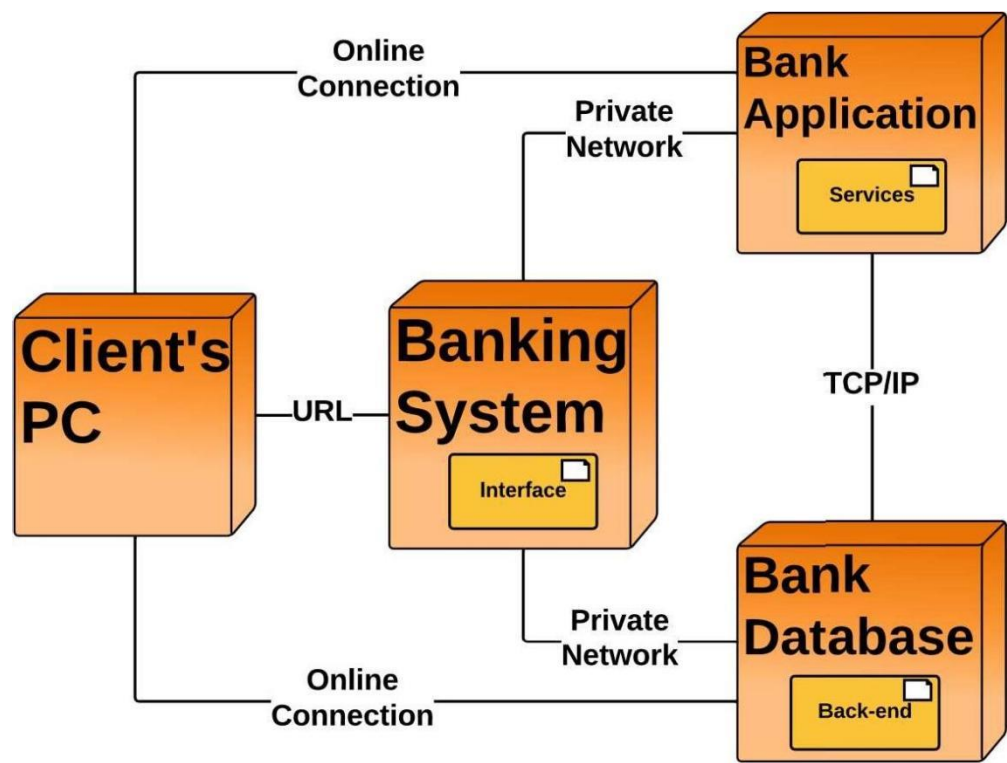


Node1

Deployment Diagram for Railway Reservation System



Deployment Diagram for Bank Management System



RESULT:

Thus the Deployment diagrams for the Ticket Reservation system and Bank management system has been designed, executed and output is verified

EX NO.10

DATE:

Apply appropriate design patterns to improve the reusability and maintainability of the software system

SOFTWARE PERSONNEL MANAGEMENT SYSTEM AIM:

To implement a software for software personnel management system

PROBLEMSTATEMENT:

Human Resource management system project involves new and/or system upgrades of software of send to capture information relating to the hiring termination payment and management of employee. He uses system to plan and analyze all components and performance of metrics driven human resource functions, including recruitment, attendance, compensation, benefits and education. Human resources management systems should align for maximum operating efficiency with financial accounting operations customer relationship management, security and business lines as organization.

(I) SOFTWARE REQUIREMENT

SPECIFICATION:

2.1 SOFTWAREINTERFACE

• **Front End Client**- The applicant and Administrator online interface is built using JSP and HTML. The HR's local interface is built using Java.

Server - Glassfish application server (SQLCorporation).

Back End SQL database.

2.2 HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

(II) USECASE DIAGRAM:

The HR of an organization involves recruitment training, monitoring and motivation of an employee. The HR also involves gives salary as observed in the payroll sheet. The employee undergoes training, receives the salary, gives the expected performance and manages time in order to complete a given task within the required period.

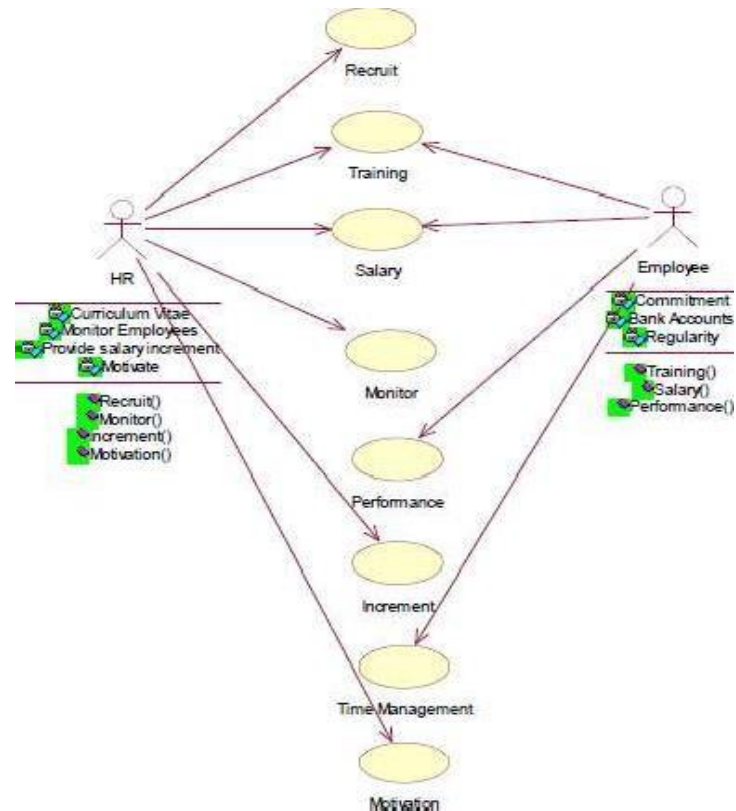


Fig.3. USE CASE DIAGRAM

(III) ACTIVITYDIAGRAM:

The activity diagram action, partition, for k join and object node. Most of the notation is self-explanatory, two subtle points. Once an action finished, there is an automatic outgoing transaction. The diagram can show both control flow and dataflow.

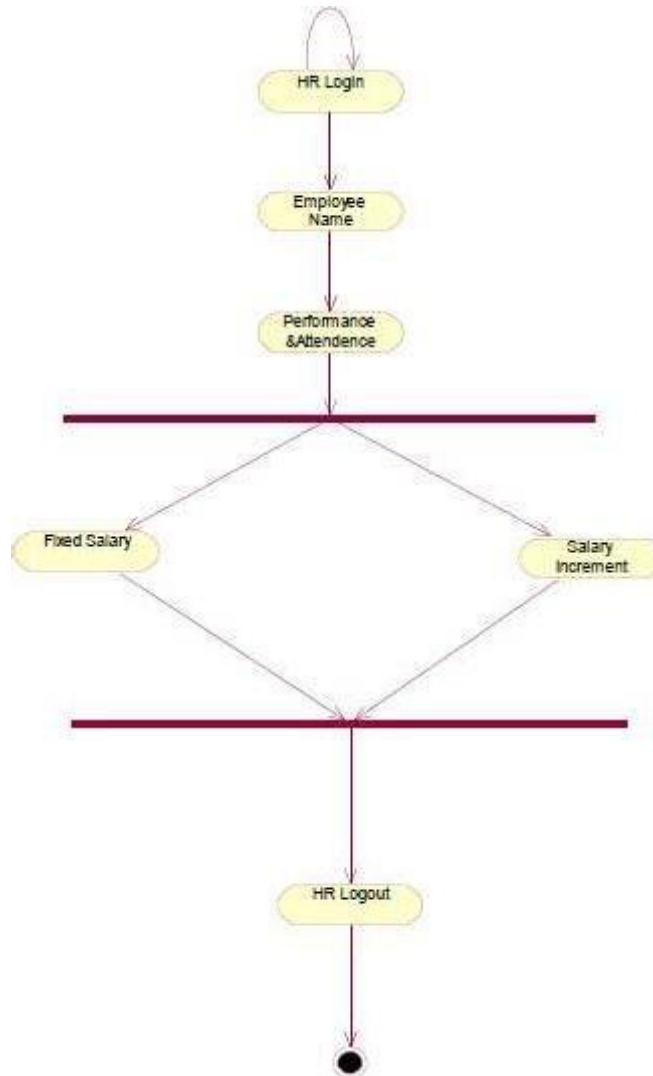


Fig.4. ACTIVITY DIAGRAM

(IV) CLASSDIAGRAM:

The class diagram, also referred to as object modeling is the main static analysis

diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

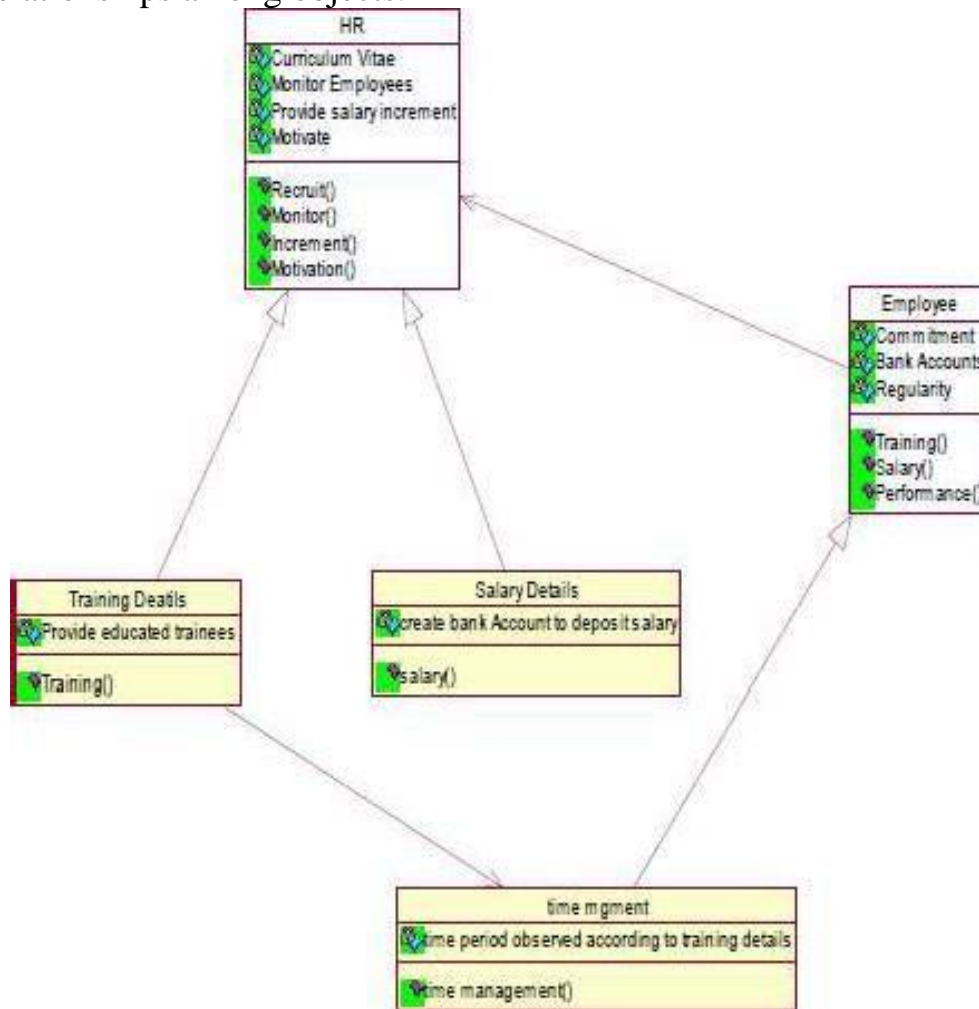


Fig.5. CLASS DIAGRAM

(V) INTERACTION DIAGRAM:

A sequence diagram represents the sequence and interactions of a given USE- CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

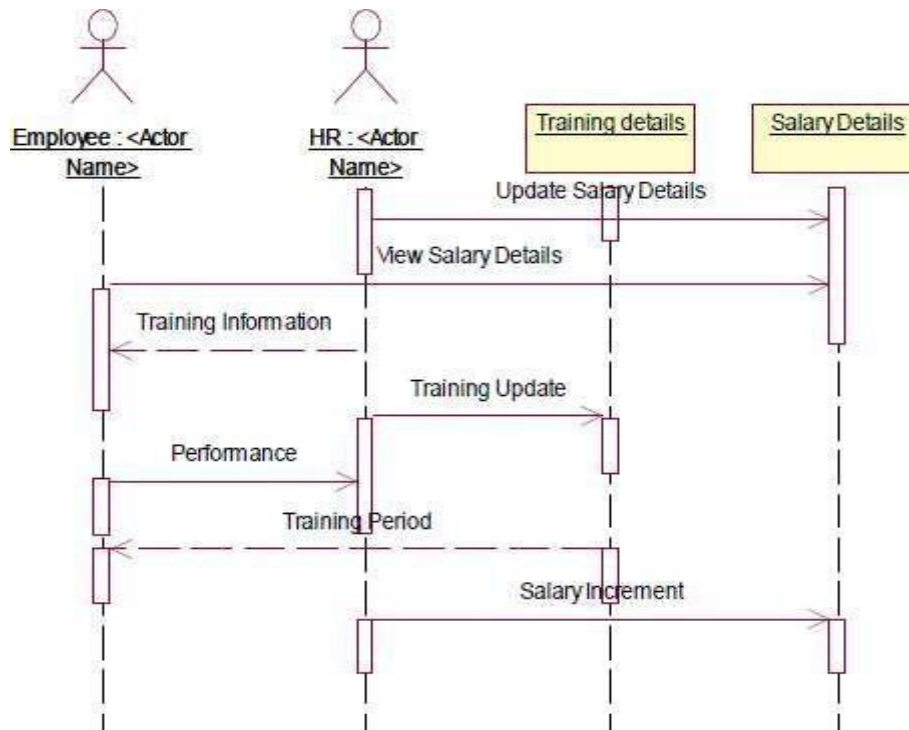


Fig.6.1.SEQUENCE DIAGRAM



Fig.6.2.COLLABORATION DIAGRAM

(VI) State Transition Diagram

States of object are represented as rectangle with round corner, the transaction between the different states. A transition is a relationship between two state that indicates that when an event occur the object moves from the prior state to the subsequent.

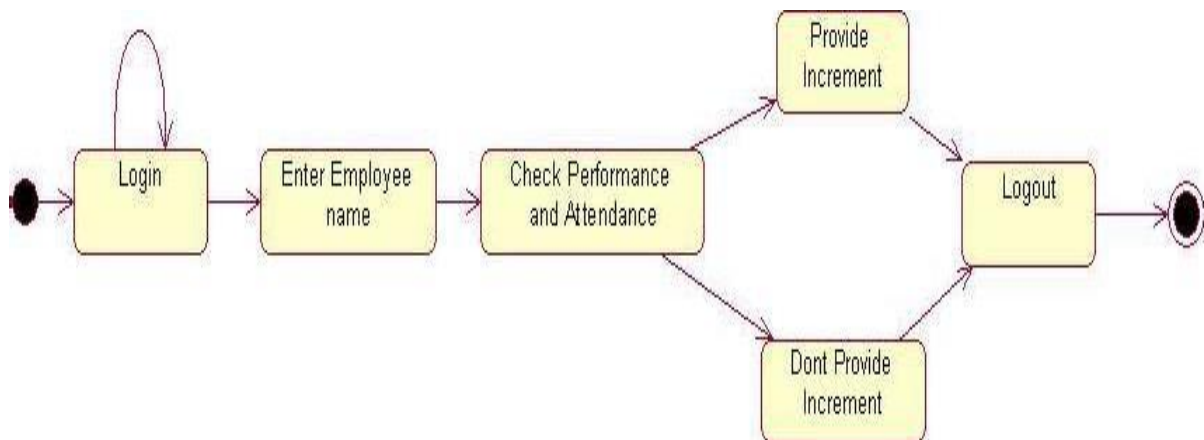


Fig.6.3 STATE TRANSITION DIAGRAM

(VII) DEPLOYMENT DIAGRAM AND COMPONENTDIAGRAM

HR recruits employee for a company employee recruited by HR goes under training before actually working. Training period is given to the employee with the training details. The salary details for the employee are provided.

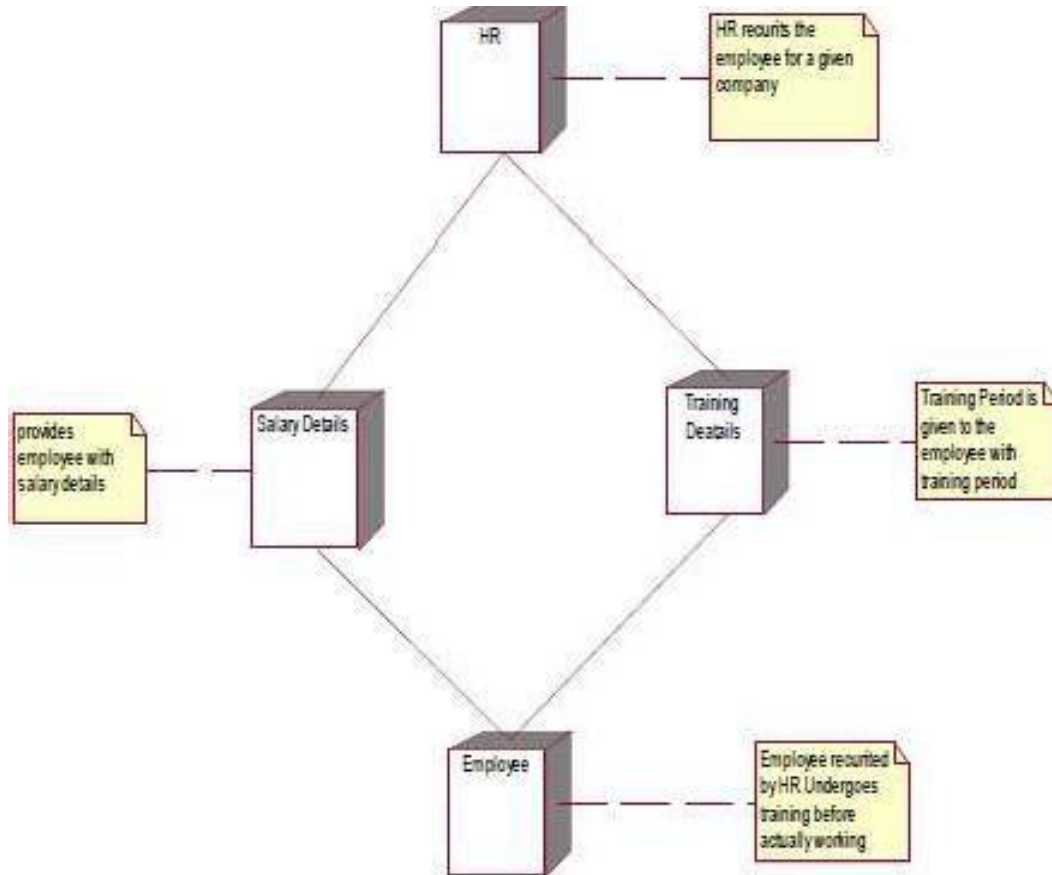


Fig.8.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

The HR recruits, motivate and monitor the employee, HR also update the salary details and training details for reference. The employee are those who are recruited by HR and work for the company. The training details provide employees with training details which is updated by HR

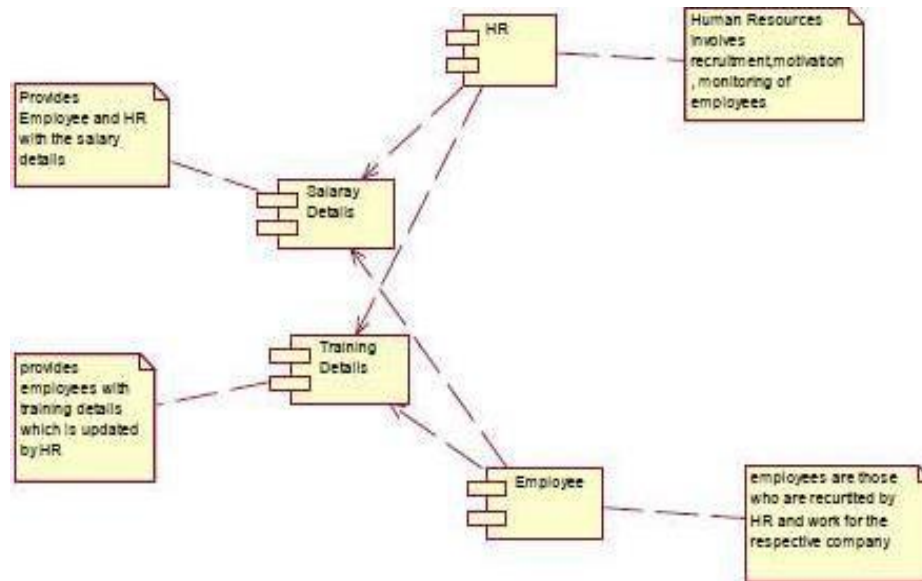


Fig.8.2.COMPONENT DIAGRAM

RESULT:

Thus the project for software personnel management system has been successfully executed and codes are generated.