Ex no:

SOFTWARE PERSONNEL MANAGEMENT SYSTEM

Date:

AIM:

To implement a software for software personnel management system

(I) PROBLEM STATEMENT:

Software personnel management system allows employees to record time card electronically and automatically generates pay slips based on number of hours worked and total amount of sales. The system will run on individual employee desktops where the employee can access and edit only their personal details. The system will maintain information on the employee in the company in order to calculate the payroll. The employees will also be able to know from the system, the number of hours worked per day and total of all hours spent on a project and total pay received year-to-date etc. Payroll administrators keep track of all the information including adding new employees, deleting employees, and edit information and run reports. The system will generate records and performance report of the employees.

(II)SOFTWARE REQUIREMENT SPECIFICATION:

1.0 INTRODUCTION

The Software Personnel Management system is an interface between Employee and the Administrator responsible for generation of payment slip. It aims at improving the efficiency in the generation of Pay slip and reduces the complexities involved in it to the maximum possible extent.

1.1 PURPOSE

If the entire process of Software personnel management is done in a manual manner then it would more time for pay slip generation process. Considering the fact that the number of employee is increasing every year, a maintenance system is essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process.

1.2 SCOPE

- Software system allows Administrator to manage its employee in a better way.

- When needed, it will take just a few second to find out the background of an employee and his/her contribution to the organization, it will also facilitate keeping all the records of employee.

- So all the information about an employee will be available in a few seconds, it will also make it very easy to generate statistical data or custom data, line finding a certain set of employee.

1.3 DEFINITIONS, ACRONYMS AND THE ABBREVIATIONS

- **ADMINISTRATOR**
  Refers to the super user who is maintaining the employee details.
- **Employee**
  One who works for a software company.
- **SPMS**

  Refers to this Software personnel management system.

- **HTML**

  Markup Language used for creating web pages.

- **J2EE**

  Java 2 Enterprise Edition is a programming platform java platform for developing and running distributed java applications.

- **HTTP**

  Hyper Text Transfer Protocol.

1.4 REFERENCES

IEEE Software Requirement Specification format.

1.5 TECHNOLOGIES TO BE USED
   • HTML
   • JSP
   • Javascript
   • Java
   • XML
   • AJAX

1.6 TOOLS TO BE USED
        • Eclipse IDE (Integrated Development Environment)
        • Rational Rose tool (for developing UML Patterns)

1.7 OVERVIEW
SRS includes two sections overall description and specific requirements

Overall Description will describe major role of the system components and inter-connections.

Specific Requirements will describe roles & functions of the actors.

2.0 OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE
        The SPMS acts as an interface between the 'ADMINISTRATOR' and the 'employee'. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which to manage the software personnel.

2.2 SOFTWARE INTERFACE
        • **Front End Client** - The applicant and Administrator online interface is built using

        JSP and HTML. The ADMINISTRATOR's local interface is built using Java.

        • **Web Server** – Apache Tomcat application server(Oracle Corporation).

        • **Back End** – Oracle 11g database.

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

2.4 SYSTEM FUNCTIONS

Payment Slip
The payment module greatly reduces the workload of the ADMINISTRATOR department by automating the payroll process, allowing ADMINISTRATOR to ensure the payroll functions are completed on time and without errors. The payroll class automatically calculates payment amounts and various deductions such as income tax before generating pay checks and employee tax reports.

View Salary
The employee views the salary details efficiently from the SPMS. The employees will also be able to know from the system, the number of hours worked per day and total of all hours spent on a project and total pay received year-to-date etc.

2.5 USER CHARACTERISTICS

- **Employee**

  These are the person who desires to view the salary details.

- **Administrator**

  Administrator has the certain privileges to generate pay slip for the employee.

- **Database manager**

  DB manager stores all the data related to Employee and Administrator.

2.6 CONSTRAINTS

- The administrator requires a system to monitor information of the employee.

2.7 ASSUMPTIONS AND DEPENDENCIES

- The employee and Administrator must have basic knowledge of computers and English

  Language.

(III) USECASE DIAGRAM:

The Software personnel management system use cases are:

1. Login

2. Job Assigned

3. View Salary

4. View Employee details

5. Generate payment slip

6. Create DB

7. Update DB

8. Delete DB

ACTORS INVOLVED:

1. Employee

2. Administrator

3. Database Manager

USE-CASE NAME: LOGIN
The Employee login to the system to view the salary details

USE-CASE NAME: JOB ASSIGNED
The employee views the job assigned to him/ her by the Administrator.

USE-CASE NAME: VIEW SALARY

The employee views the salary details efficiently from the SPMS. The employees will also be able to know the number of hours worked per day and total of all hours spent on a project and total pay received year-to-date etc.

.

USE-CASE NAME: VIEW EMPLOYEE DETAILS

The Administrator views the details of the employee for the payroll process

USE-CASE NAME: GENERATE PAYMENT SLIP

The Administrator generates the pay slip based on the details of the no of hours/ no of days worked by the employee.

USE-CASE NAME: CREATE DB

The database manager creates individual database tables for the employees

USE-CASE NAME: UPDATE DB

When an employee information changes the database manager updates individual database tables for the employees.

USE-CASE NAME: DELETE DB

When an employee relieves/terminated the database manager deletes individual database tables for the employees.
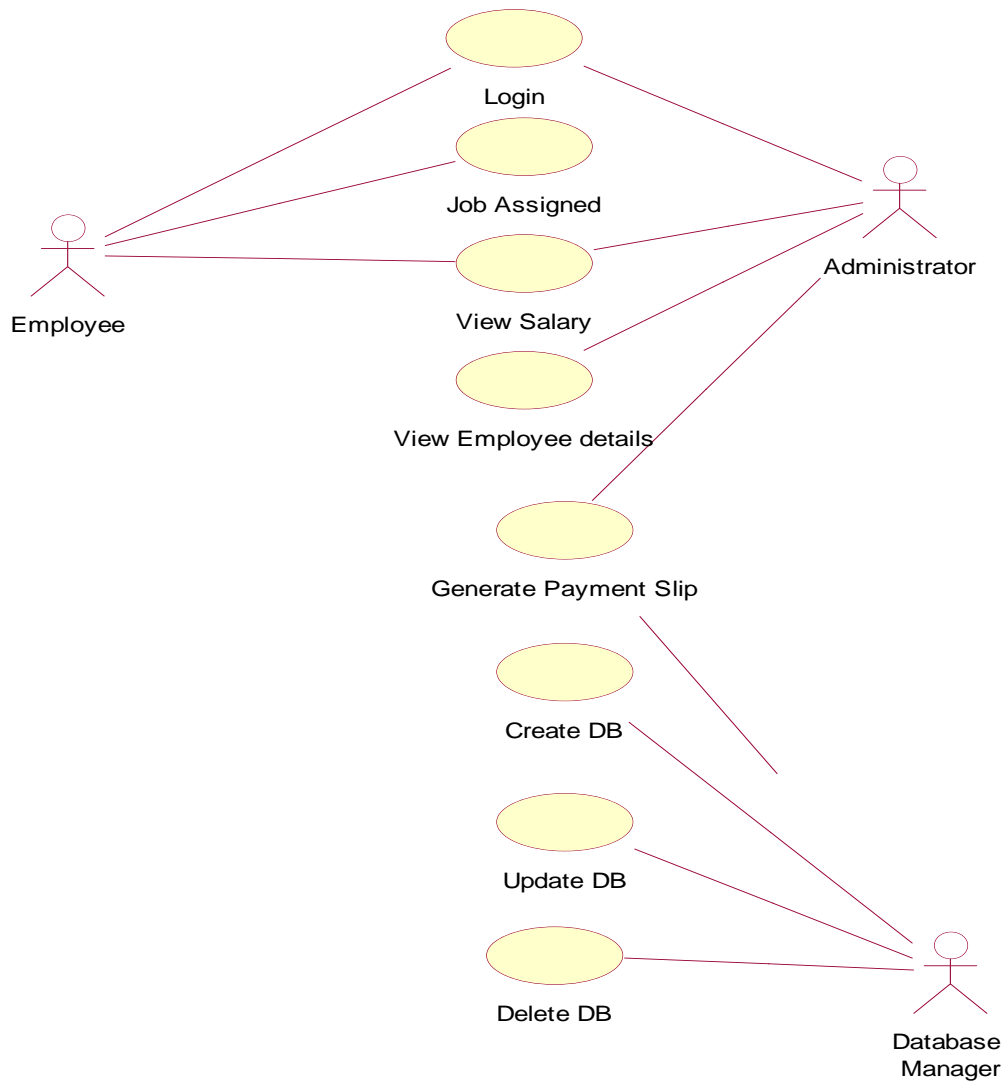
Fig.3. USE CASE DIAGRAM FOR SOFTWARE PERSONNEL MANAGEMENT SYSTEM

(IV) ACTIVITY DIAGRAM:

The activity diagram notation is an action, partition, fork 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 data flow.
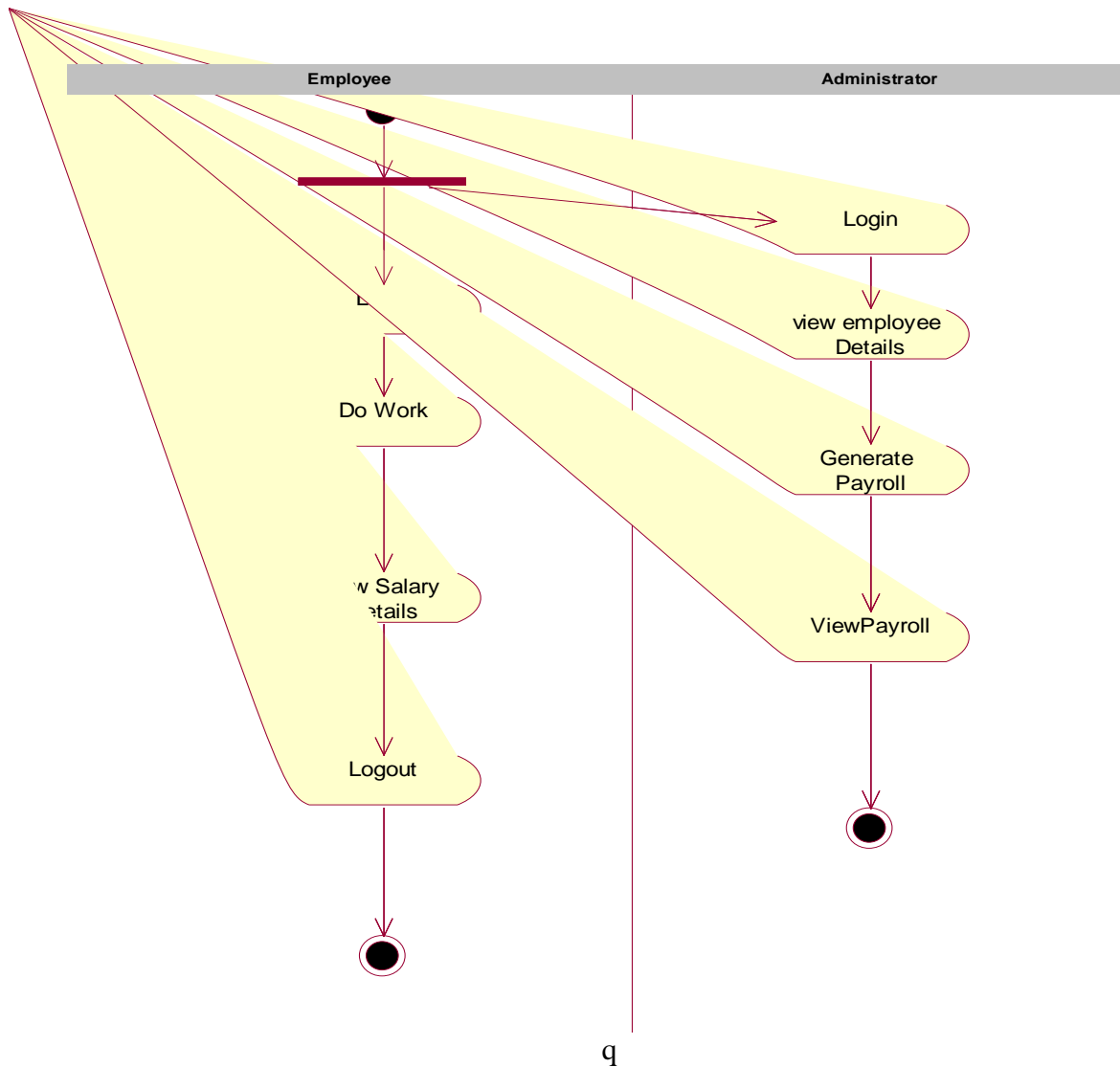
Fig.4. ACTIVITY DIAGRAM FOR SOFTWARE PERSONNEL MANAGEMENT SYSTEM

(V) CLASS DIAGRAM:

The class diagram is referred as object modeling in the 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.

The Software Personnel Management system class diagram consists of four classes

1. Employee class
2. Administrator class
3. Database Manager class
4. Payment class

1. EMPLOYEE CLASS

It consists of seven attributes and two operations. The attributes are empid, empname, emppassword, address, mobile number, date, Hours Worked. The operations of this class are Login( ) and viewsalary( ).

2. ADMINISTRATOR CLASS

It consists of attributes Adminid, Adminname and Adminpassword. The operations are login ( ), Generate payroll ( ), view payroll ( ) and viewemployeedetail ( ).

3. DATABASE MANAGER CLASS

The attributes of this class are DBmanagerid, DBmanagername( ) and DBmanagerpassword. The operation are create( ), update( ),delete( ) and display payroll( ).

4. PAYMENT CLASS

The attributes of this class are paymentid, empid, date, Basic pay, HRA, DA, PF, Netpay and Gross pay. The operation are calculatesalary( ) and GenerateSlip( ).
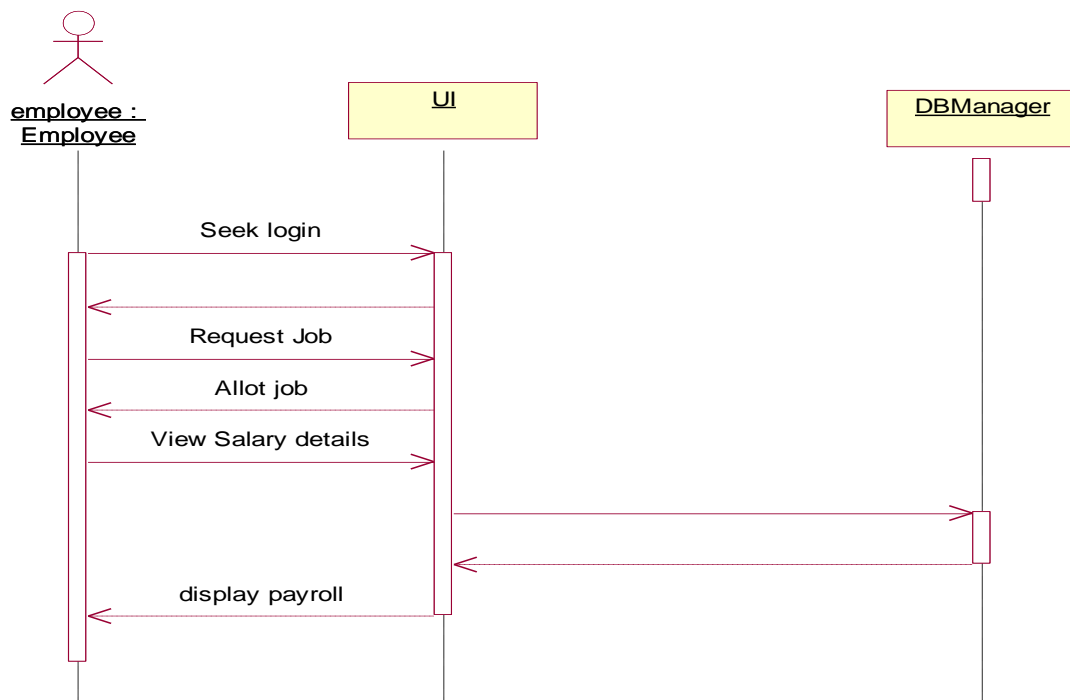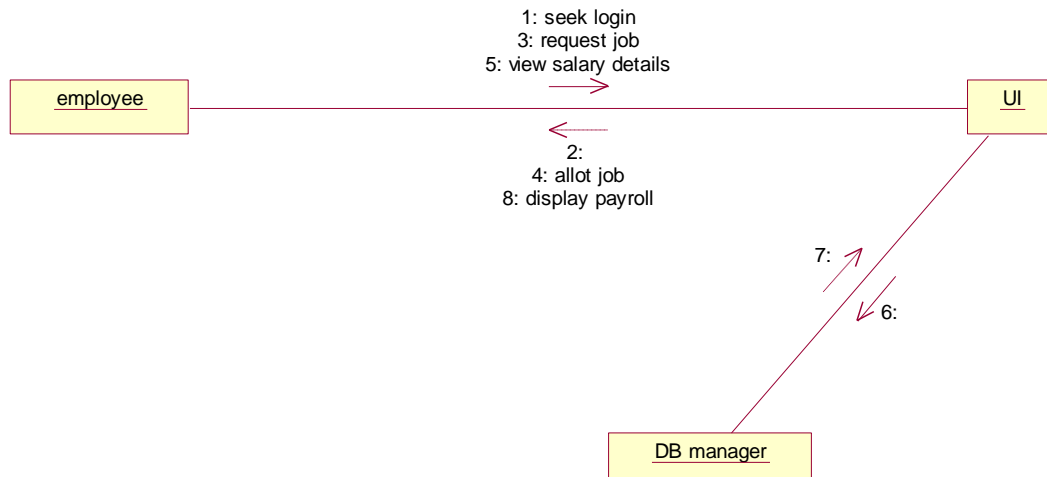
Adminstrator

🔒Adminid : int
🔒Adminname : string
🔒Adminpassword : string

◆login()
◆Generatepayroll()
◆viewpayrolll()
◆viewemployeedetail()

Employee

🔒empid : int
🔒empname : string
🔒emppassword : string
🔒address : string
🔒mobilenumber : int
🔒date : date
🔒HoursWorked : int

◆login()
◆viewsalary()

DatabaseManager

🔒DBmanagerid : int
🔒DBmanagername : string
🔒DBmanagerpassword : string

◆Create()
◆Update()
◆delete()
◆displaypayroll()

Payment

🔒paymentid : int
🔒empid : int
🔒date : date
🔒Basicpay : int
🔒HRA : int
🔒DA : int
🔒PF : int
🔒Netpay : int
🔒Grosspay : int

◆CalculateSalary()
◆GenerateSlip()

Fig.5. CLASS DIAGRAM FOR SOFTWARE PERSONNEL MANAGEMENT SYSTEM

(VI) 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.
- An event also is considered to be any action by an object that sends information.
- The event line represents a message sent from one object to another, in which the "form" object is requesting an operation be performed by the "to" object.
- The "to" object performs the operation using a method that the class contains.
- It is also represented by the order in which things occur and how the objects in the system send message to one another.
- The sequence diagram for each USE-CASE that exists when a user administrator, check status and new registration about passport automation system are given.

EMPLOYEE:

Fig.6. 1.SEQUENCE DIAGRAM FOR EMPLOYEE

1: seek login
3: request job
5: view salary details

employee → UI

2:
4: allot job
8: display payroll

7:

6:

DB manager

Fig.6. 2.COLLABORATION DIAGRAM FOR EMPLOYEE

: Administrator | UI | Payment | Database Manager

Seek Login

View Employee Details

Display Employee Detail

Generate Payslip

Calculate Salary

return salary

Display Payroll
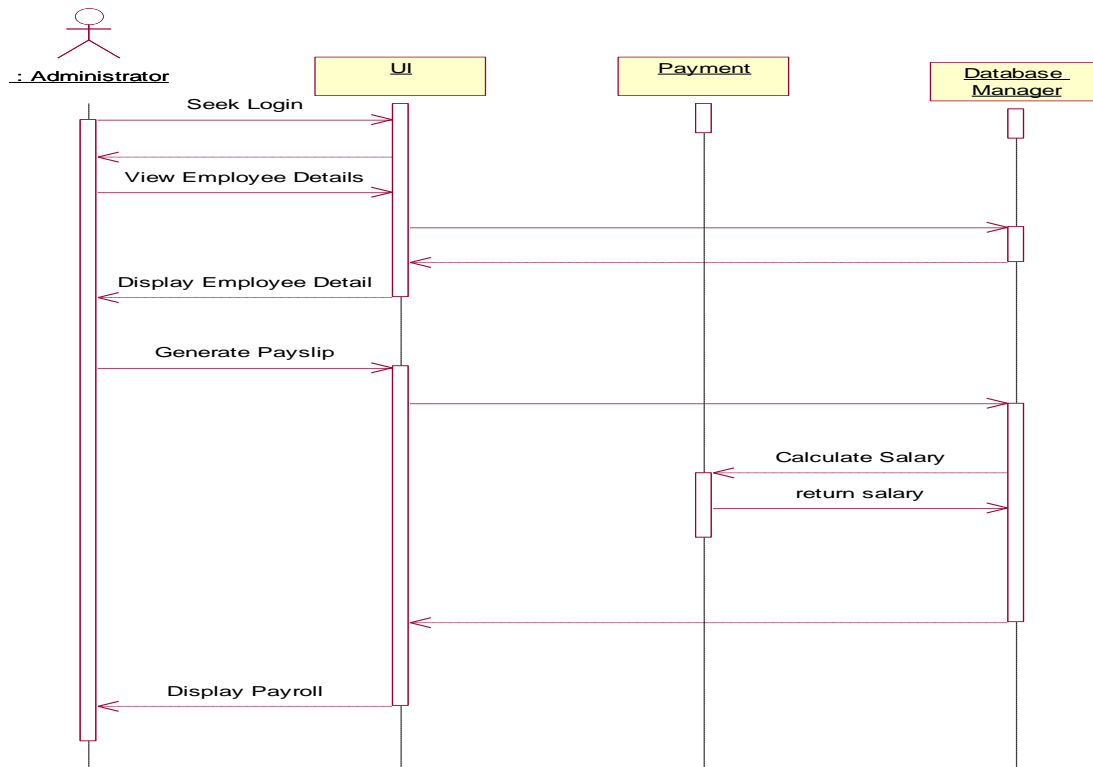
Fig.6. 3.SEQUENCE DIAGRAM FOR ADMINISTRATOR

1: seek login
3: view employee details
7: generate payslip

administrator

UI

2:
6: display employee details
12: display payroll

5:
11:

4:
8:

10: return salary

payment

DB manager

9: calculate salary
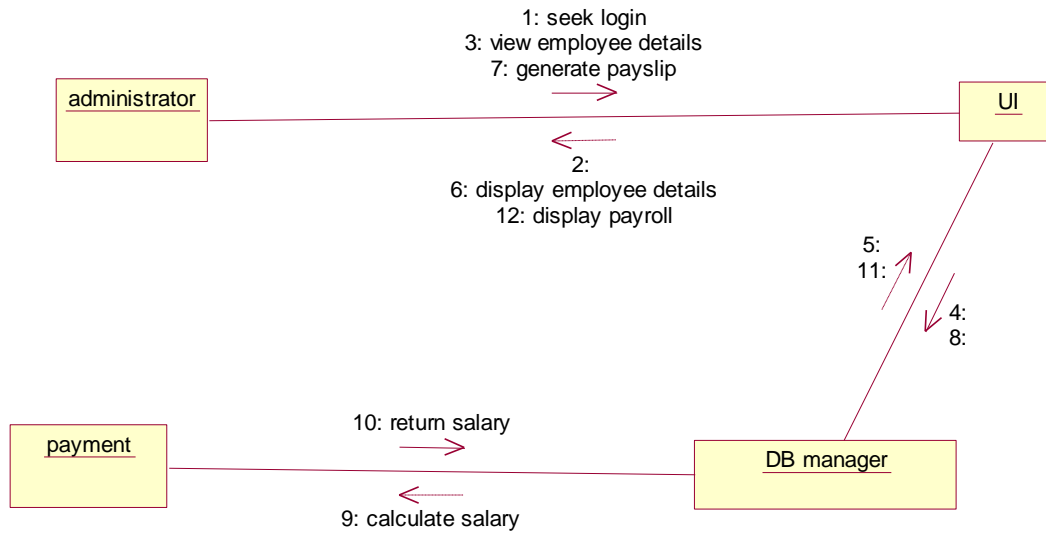
Fig.6. 4.COLLABORATION DIAGRAM FOR EMPLOYEE

(VII) 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.

Login → View Employee Detail → Generate Payroll → calculate Salary

Display Payroll
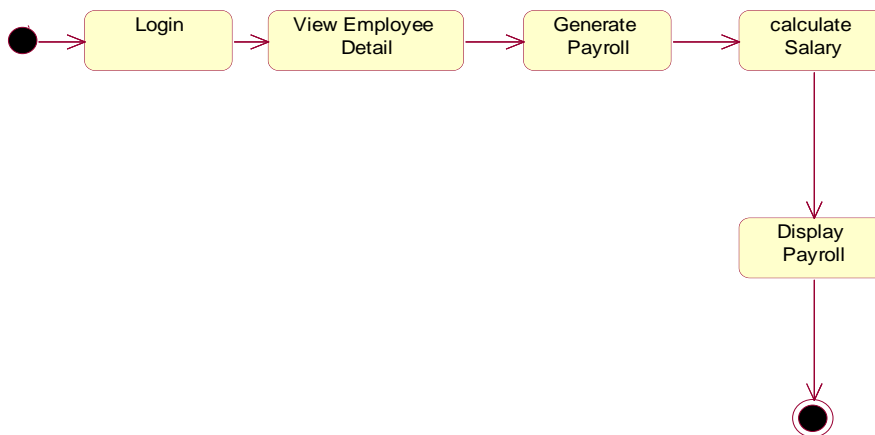
Fig.7.STATE TRANSITION DIAGRAM FOR SOFTWARE PERSONNEL MANAGEMENT SYSTEM

(VIII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed
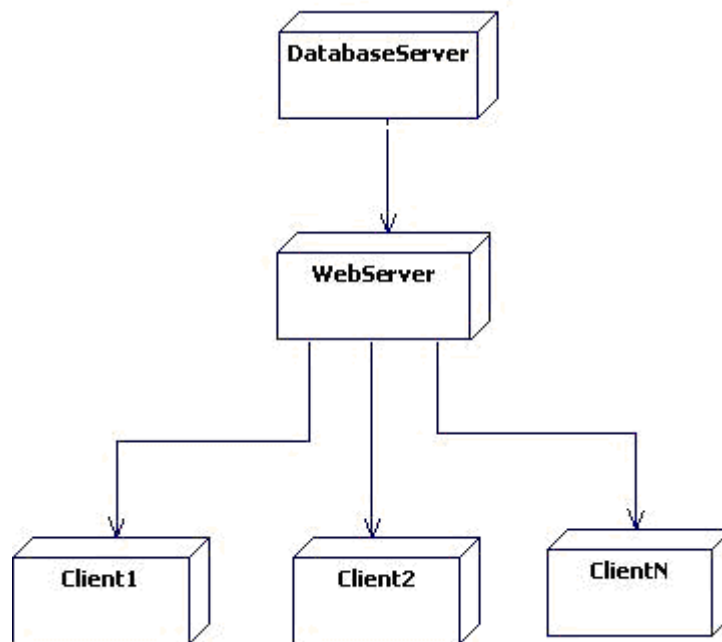
Fig.8.1.DEPLOYMENT DIAGRAM FOR SOFTWARE PERSONNEL MANAGEMENT SYSTEM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships among components in a system.
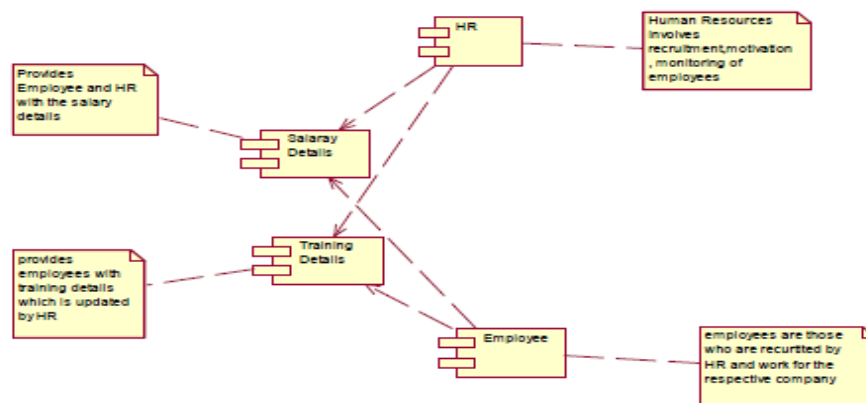
Fig.8.2.COMPONENT DIAGRAM FOR SOFTWARE PERSONNEL MANAGEMENT SYSTEM

(IX) IMPLEMENTATION OF DOMAIN OBJECTS LAYER AND TECHNICAL SERVICE LAYER

//Source file: employee.java

```java
public class employee
{
  private int emp_id;
  private string emp_name;
  private string adrr1;
  private string addr2;
  private string addr3;
  public administrator1 theAdministrator1;

  /**
  @roseuid 515D15470203
   */
  public employee()
  {

  }

  /**
  @roseuid 515D1509029F
   */
  public void login()
  {

  }

  /**
  @roseuid 515D150D031C
   */
  public void view()
  {

  }
}
```
//Source file: administrator1.java
```java
public class administrator1
{
```

```
    private int admin_id;
    private string admin_name;
    private string admin_pwd;
    public employee theEmployee;

    /**
    @roseuid 515D154701A5
     */
    public administrator1()
    {

    }

    /**
    @roseuid 515D130F030D
     */
    public void login()
    {

    }

    /**
    @roseuid 515D13140271
     */
    public void generatePayroll()
    {

    }

    /**
    @roseuid 515D131F0251
     */
    public void viewPayroll()
    {

    }
}
```

(IX) IMPLEMENTATION OF USER INTERFACE LAYER

Fig.11. Generate Payroll

RESULT:

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