DAYANANDA SAGAR UNIVERSITY



# A SOFTWARE ENGINEERING PRACTICE LAB REPORT

## ON

**“Payroll Management System”**

# BACHELOR OF TECHNOLOGY

## IN

**COMPUTER SCIENCE & ENGINEERING**

***Submitted by***

## **Disha Anand Madival (ENG18CS0093)**

**Aparna Duvvuri (ENG18CS0096)**

**Gauthami M (ENG18CS0104)**

**Jyothsna T(ENG18CS119)**

**V Semester, 2020**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF ENGINEERING**

**DAYANANDA SAGAR UNIVERSITY KUDLU GATE**

**BANGALORE – 560068**

**DAYANANDA SAGAR UNIVERSITY**

## School of Engineering, Kudlu Gate, Bangalore-560068



**CERTIFICATE**

***This is to certify that Mr/Ms. Disha Anand Madival bearing ENG18CS0093, has satisfactorily completed their Software Engineering Practice Lab Report as prescribed by the University for the Fifth semester B.Tech. Progrwam in Computer Science & Engineering during the year 2020 at the School of Engineering, Dayananda Sagar University, Bangalore***

Date:

Faculty in Charge

**Chairman**

**Department of Computer Science & Engineering**

**ACKNOWLEDGEMENT**

The success and final outcome of this software requirement document required a lot of guidance and assistance from many people and we are extremely privileged to have got this all through the completion of the project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Professor Dr. Bondu Venkateshwarluand Chairman Dr Sanjay Chitnis,for providing us an opportunity to do the Software Requirement Document and giving us all support and guidance which made me complete the project duly. We are extremely thankful to him for providing such a nice support and guidance, even though he has busy schedule managing the departmental affairs.

We owe our deep gratitude to our mentor Dr. Bondu Venkateshwarlu**,** who took keen interest on our Software Requirement Document and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system document.

We are thankful to, and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of the Computer Science Engineering department, which helped us in successfully completing our report. Also, we would like to extend our sincere esteems to all staff in laboratory for their timely support.

Student Names

1.Disha Anand Madival

2.Aparna Duvvuri

3.Gautami M

4.Jyothsna T

# Table of Contents

SL.NO Page No

|  |  |
| --- | --- |
| Cover Page | i |
| Certificate | ii |
| Acknowledgement | iii |
| Contents | iv |
| 1. Introduction | 6 |
| 1.1 Problem Statement | 6 |
| 2. Vision Document | 7 |
| 3. Glossary | 8 |
| 1. Supplementary Specification Document 2. Use-Case Diagrams | 9 |
| 5.1 Actors | 10 |
| 5.2 Use-Case | 10 |
| 5.3 Use-Case Diagram | 10 |
| * 1. Use Case Description      1. Use-Case Specification: Login | 11 |
| 5.4.2 Use-Case Specification: Enquire a Employee Information | 13 |
| 5.4.3 Use-Case Specification: List the records | 14 |
| 5.4.4 Use-Case Specification: Add record | 15 |
| 5.4.5 Use-Case Specification: Delete record | 16 |
| 5.4.6 Use-Case Specification: Payroll Generation | 18 |
| 5.5 Activity Diagram | 19 |

|  |  |  |
| --- | --- | --- |
| SL.NO |  | Page No |
| 6. | Design Model | 20 |
|  | 6.1 Class Diagram | 20 |
|  | 6.2 Sequence and Collaboration Diagrams | 21 |
| 7. | 6.3 State Chart Diagram for Title Class Deployment Model  7.1 Component Diagram | 28  29 |
|  | 7.2 Deployment Diagram | 29 |
| 8. | Conclusion | 30 |
| 9 | References | 30 |

# 1.Introduction

**1.1 Problem Statement**

The old system is outdated and no longer adequately manages the payroll process.The seemingly unending list of hurdles faced by an in-house payroll department eats valuable time that can be spent on other core HR activities.Hence there is a need of a computerized payroll management system which can mange the employee details and also compute their payment.The activities like seeing Employee Record , add Record, modify records, delete Record and finally receiving Pay Slip can be performed easily.

It also allows the user to maintain the information about the employee records, add new employee records, display employee records, edit the employee details, remove a particular employee record and also to see the details of all the existing employees.

To add an employee record the user has to give in the details like name, code, designation, age, salary and experience of the employee and give a ‘Y’ to save the record.

To edit the details of a particular employee you can give the employee code and make necessary changes to the data within that particular employee. Similarly you can delete the details of a particular employee, the user just needs to give the employee code.

The Payroll Management System can run on popular Operating system platforms like Windows, Linux . It can be easily extended with new functionality and other major updates to it.

1. **VISION DOCUMENT**

* Provides the searching facilities based on various factors. Such as Payroll, Salary, Appraisals, Working Points
* Payroll Management System also manage the Payments details online for Appraisals details, Working Points details, Payroll.

* It tracks all the information of Employee, Payments, Appraisals ect

* Manage the information of Employee

* Shows the information and description of the Payroll, Salary

* To increase efficiency of managing the Payroll, Employee

* It deals with monitoring the information and transactions of Appraisals.

* Manage the information of Payroll

* Editing, adding and updating of Records is improved which results in proper resource management of Payroll data.
* Manage the information of Appraisals

* Integration of all records of Working Points.

1. **GLOSSARY**

Keys is the word that will represent the different fields available in the payroll management system.The various keys used in our project include:

Emp\_id : the unique id assigned to each employee in the payroll management system.

Name : the name of the employee working in the organization

Designation : the work that a particular employee does in the organization

No\_of\_Years\_experience : the number of years of work experience that particular employee has.

Age : the age of the particular employee.

Salary : the amount paid in the organization for his/her work.

**Actors**

**Employee** : Interactive actor who uses the payroll management system to search for orders,details of the

stock such as price quantity etc, make addition to the existing stocks or simply view them stock.

**Manager** : Interactive actors responsible for maintenance of the payroll , keeps a check on the employees

as well as being ale to perform all actions performed by the employee

1. **SUPPLEMENT SPECIFICATION DOCUMENT**

**Objective**

The purpose of this document is to define the requirements of the Payroll management

system. This document lists the requirements that are not readily captured in the use-cases of

the use-case model. The supplementary specification and use-case model together capture a

complete set of requirements of the system.

**Scope**

This supplementary specification defines the non-functional requirements of the system

such as reliability, performance, supportability, and security as well as functional

requirements that are common across a number of use-cases.

**Common Functionalities**

• Manager must be able to modify the different employee details easily.

• If there is any change made to the database then it must be displayed while viewing

the contents.

**Usability**

The desktop user interface shall be any operating system (Linux,Windows,MAC etc)

**Reliability**

The system shall be 24 hours a day, 7 days a week and not more than 10% down time

**Security**

• The system must prevent any unauthorized person from viewing the information i.e. only

the correct username and password must be allowed.

• Only Manager can modify or make any changes to the employee’s details and the

reverse should not be.

1. **USE CASE DIAGRAM** 
   1. **Actors**

Actor is something external to the system and interacts with the system. Actor may be a

human being, device or some other software system.

For Payroll management system, actors might be;

• Employee

• Manager

* 1. **Use – Case**

A use-case represents sequence of actions performed by the system that yields an

observable result of value for a particular actor. Use-case represents a functional

requirement of a system.

For payroll management system, we can find the following use-cases;

• Add Employee details

• Remove details

• See employee details

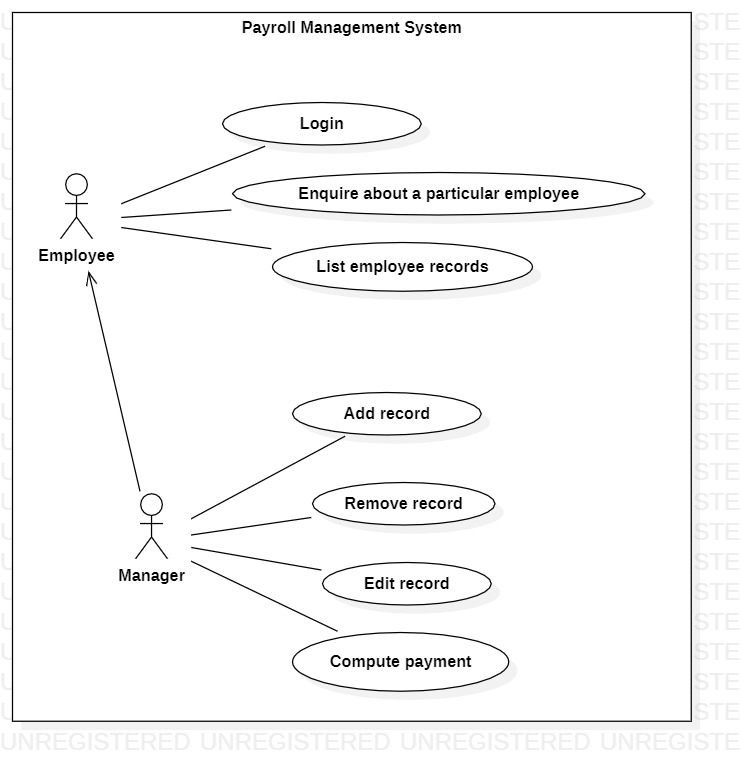
• Compute employee payments

• Edit employee details

• Manage employee records

• Enquire about a particular employee

**5.3 Use case diagram:**



**5.4 Use case description**

**5.4.1 Use-Case Specification: Login**

**Description**

A registered user can log in and, upon verification, can initiate subsequent actions.

**Flow of Events**

**Basic Flow**

1. Initiated when a registered user chooses to Login.
2. The system prompts for username and password.

1. The registered user enters a username and password and submits them.
2. The system authenticates the username and password combination.

5. The system authorizes the registered user according to the role to which the

registered user has been assigned.

1. The system displays the main page and awaits subsequent action.

**Alternative Flows**

• **Invalid Password**

7. The system displays the Authentication Failed message Which displays invalid

credentials please try again

**Preconditions**

User has an account with the system

**Post-conditions**

**Primary Success Post-condition**

The user is considered authenticated and is viewing the main page from which

additional actions can be initiated.

**Login Failure**

If the Login fails as described in any of the alternatives above, the message invalid

credentials please try again is displayed and the user is not considered authenticated.

**Notes**

1. So far, we are not doing much with roles.

2. The “invalid credentials please try again” above is vague; we need to come up with

how we report this to the user.

3. We need to talk to security people about how reasonable it is to lock the user account

after some number of failed attempts. If we keep that rule, we’ll need an Unlock

Account use case.

**5.4.2 Use-Case Specification: Enquiry about a particular Employee**

**Description**

A employee can enquire or fetch the details of another employee and can initiate subsequent actions.

**Flow of Events**

**Basic Flow**

1. Initiated when a user chooses Enquiry about a particular Employee.

2. The system prompts for Emp\_id

3. The user enters a Emp\_id

4. The system authenticates the Emp\_id

5.The system then displays the details of the requested employee

**Alternative Flows**

• **Invalid Emp\_id**

6.The system displays a error message asking user to enter a valid emp\_id

**Preconditions**

User record/details must already be existing

**Post-conditions**

**Primary Success Post-condition**

The user is considered authenticated and the fetching of detail of a particular employee is initiated.

**Failure**

If the Emp\_id is entered wrong and is not found in records, the message invalid

Emp\_id is displayed and the user search is not initiated .

**Notes**

1. So far, we are not doing much with roles.
2. The “invalid Emp\_id” above is vague

3. We need to talk to security people about how reasonable it is to lock the user account

after some number of failed attempts.

**5.4.3 Use-Case Specification: List the records**

**Description**

A user can view the records of employees in the organization through the payroll management system as a precursor to other actions.

**Flow of Events**

**Basic Flow**

1. Initiated when a user chooses to view employee records.
2. The system responds by displaying all of the details of the various employees in the system
3. The registered user selects 0 (zero) to get back to the home page.

4. The system displays the available functionalities along with the choice and the available action on

each choice.

**Preconditions**

The user has been authenticated.

**Post-conditions**

**Primary Success Post-condition**

The registered user is viewing a employee records .

**5.4.4 Use-Case Specification: Add record**

**Description**

This use-case starts when the user or the manger wants to make an addition for an item

Flow of Events

**Basic flow**

1. The system prompts the user to enter the information of the new employee which he wanted to.
2. The manager submits the information by giving a ‘yes’ while saving details

3. The system adds the choice and it will be reflected on the database while viewing

it again

**Special requirements**

Addition to the employee id can be made only by the Manager

**Pre-conditions**

The employee is viewing a particular id with a choice that is not currently available

**Post-conditions**

The choice is added and it is saved in the database in other words this will be reflected

**Notes**

So far there is no nice way to figure out what a choice has been added.15

**5.4.5 Use-Case Specification: Remove record**

**Description**

The manager can remove an existing employee with a particular id.

Flow of events

**Basic Flow**

1. The manager enters Emp\_id and submits .
2. System shows the choice is no longer present when the records are listed

Special requirements

The employee must be a manager to remove the details of the employee field

**Pre-conditions**

The employee is viewing a particular id with a choice that is needed to be removed.

**Post-conditions**

The previously existing record is no longer present

**5.4.5 Use-Case Specification: Edit record**

**Description**

The manager can edit an existing employee with a particular id.

Flow of events

**Basic Flow**

1. The manager enters Emp\_id and updates the details that had to be updated .

b .System shows the updated employee details when the records are listed

Special requirements

The employee must be a manager to editthe details of the employee field

**Pre-conditions**

The manager is viewing a particular id with a choice that is needed to be edit.

**Post-conditions**

The previously existing record is updated now.

**5.4.6 Use-Case Specification: Compute employee payment**

**Description**

The manager can compute payment of an existing employee with a particular id.

Flow of events

**Basic Flow**

a. The manager enters Emp\_id and name .

1. System asks for the number of days of work and the daily pay associated and the manager enters them

C The payment of the employee is displayed in the screen and hence saved in the record.

.

Special requirements

The employee must be a manager to compute payment of the employee

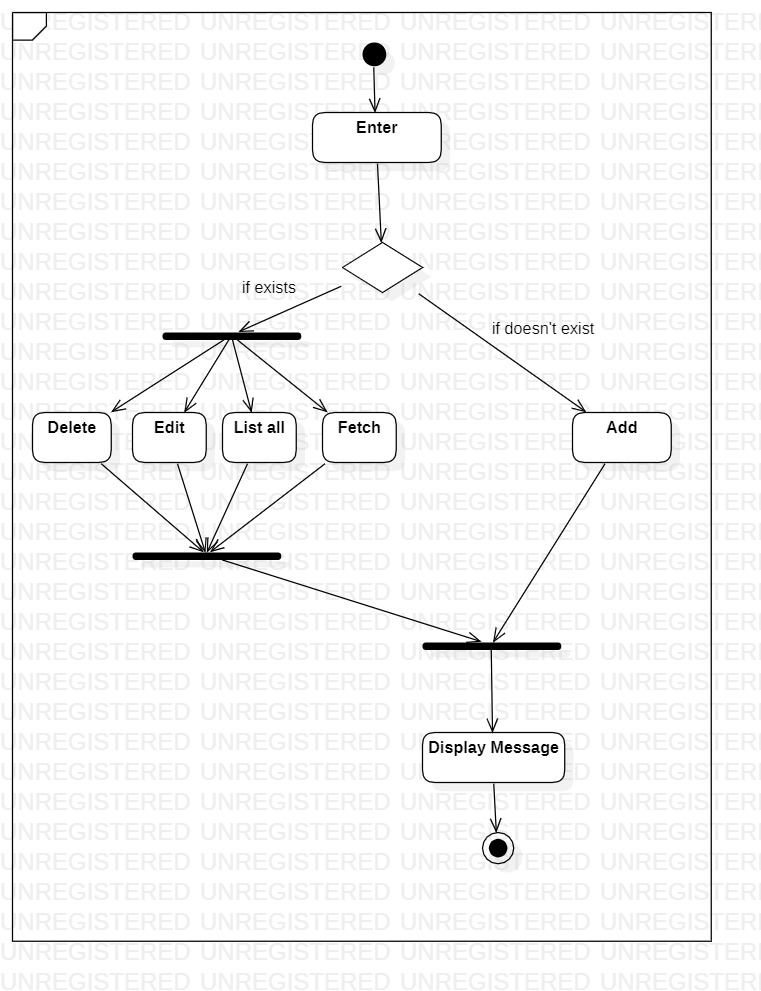
**Pre-conditions**

The employee is viewing a particular id with a choice whose pay needs to be computed.

**Post-conditions**

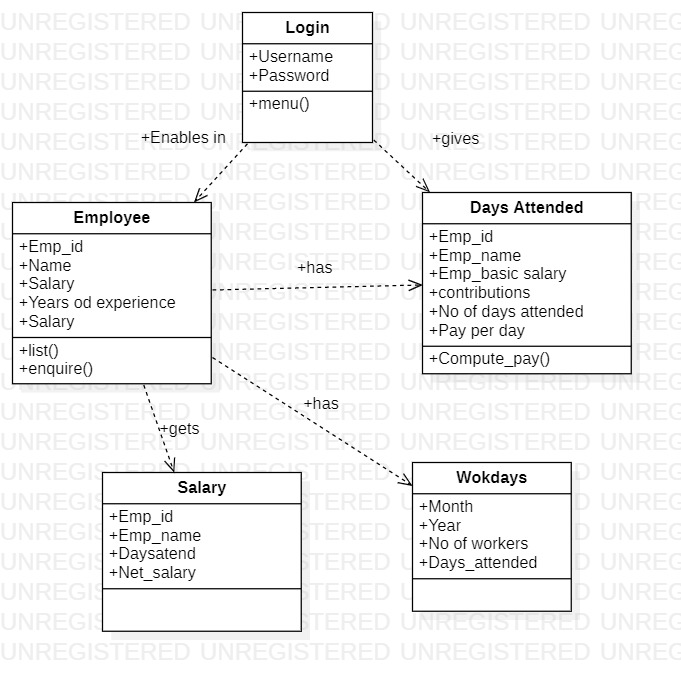
The payment of the employee is calculated and is saved in the records.

5.5 Activity Diagram



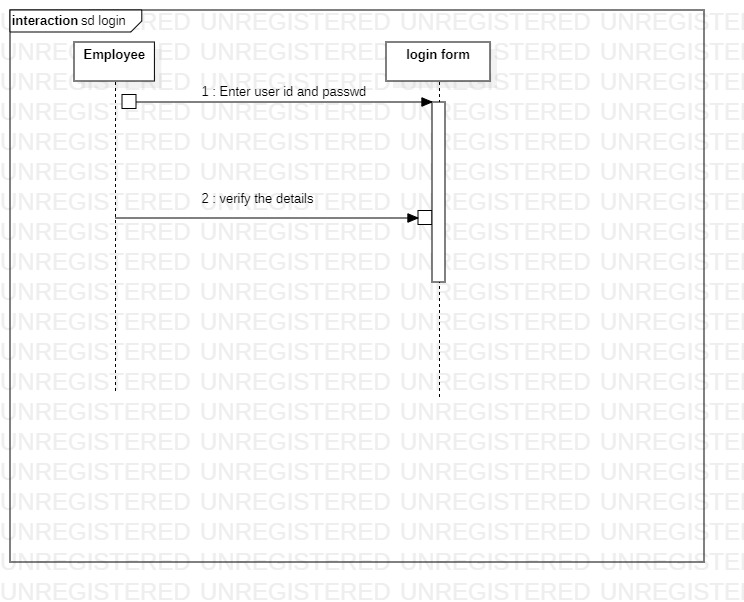
1. **Design Diagram**

**6.1 Class Diagram**

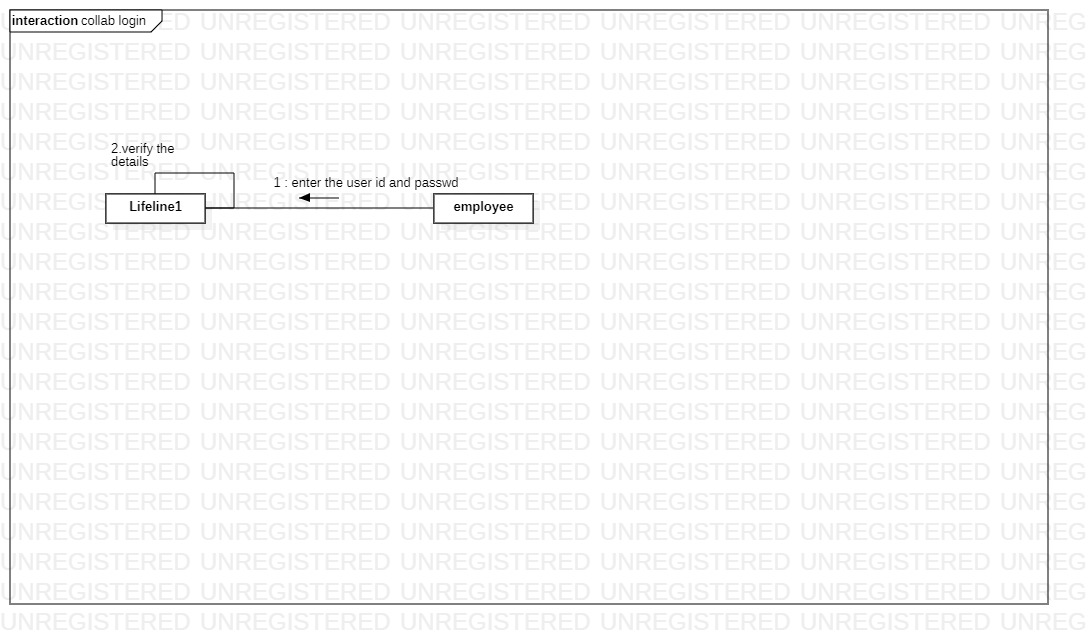
****

**6.2 Sequence and Collaboration Diagrams**

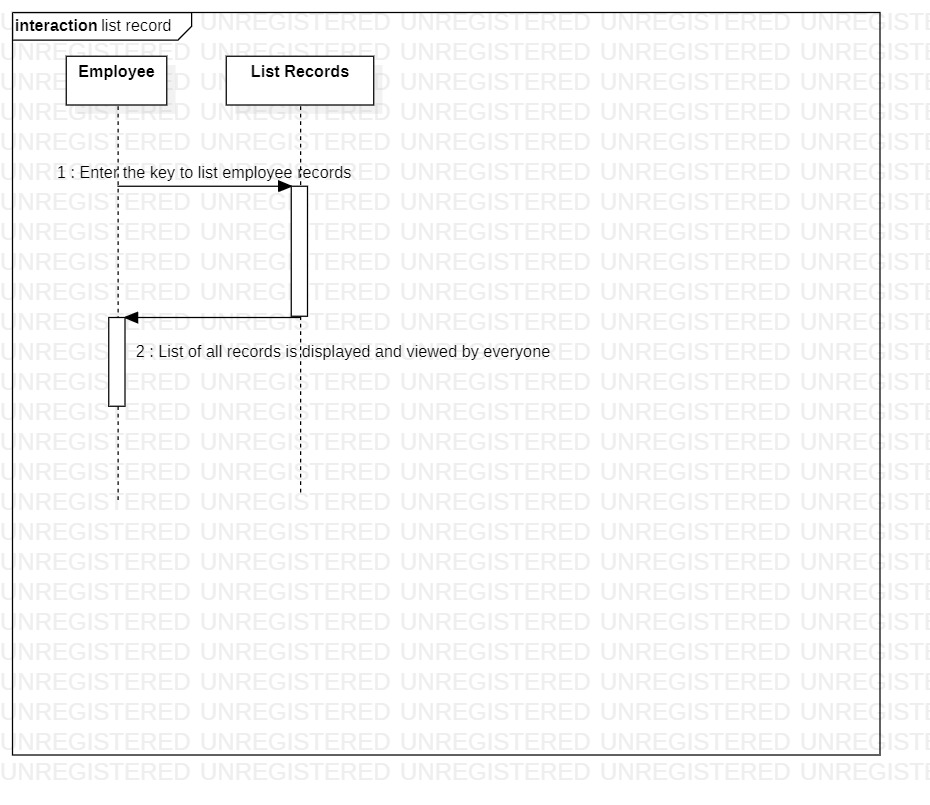
**6.2.1 Sequence Diagram for Login Use-case**

****

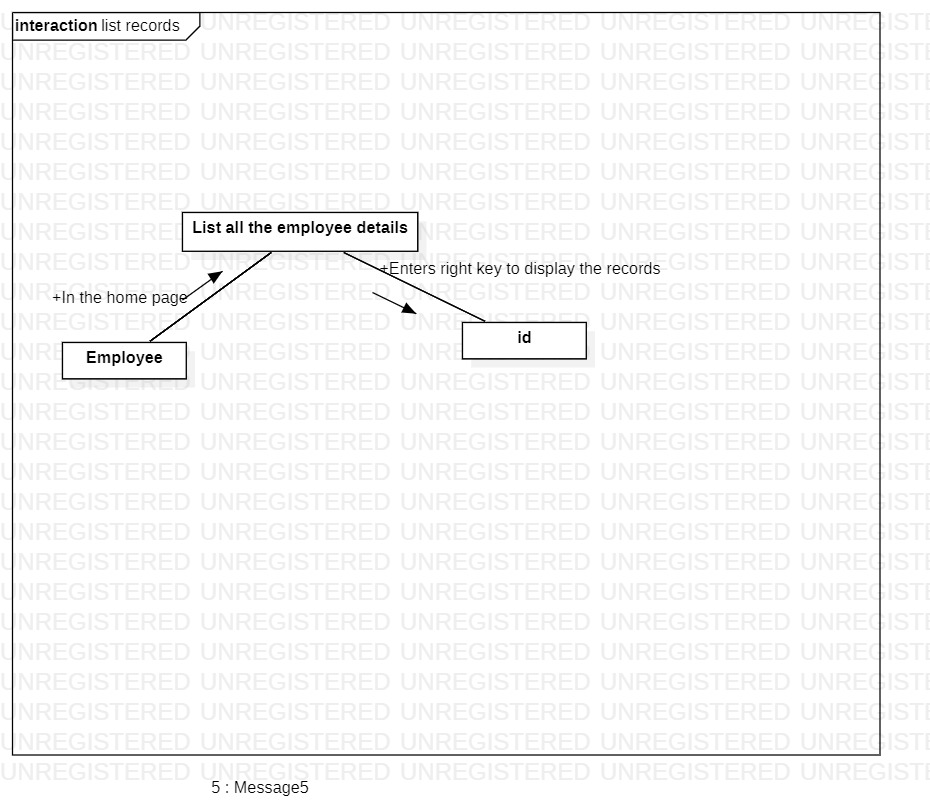
**6.2.2 Collaboration diagram for login:**

****

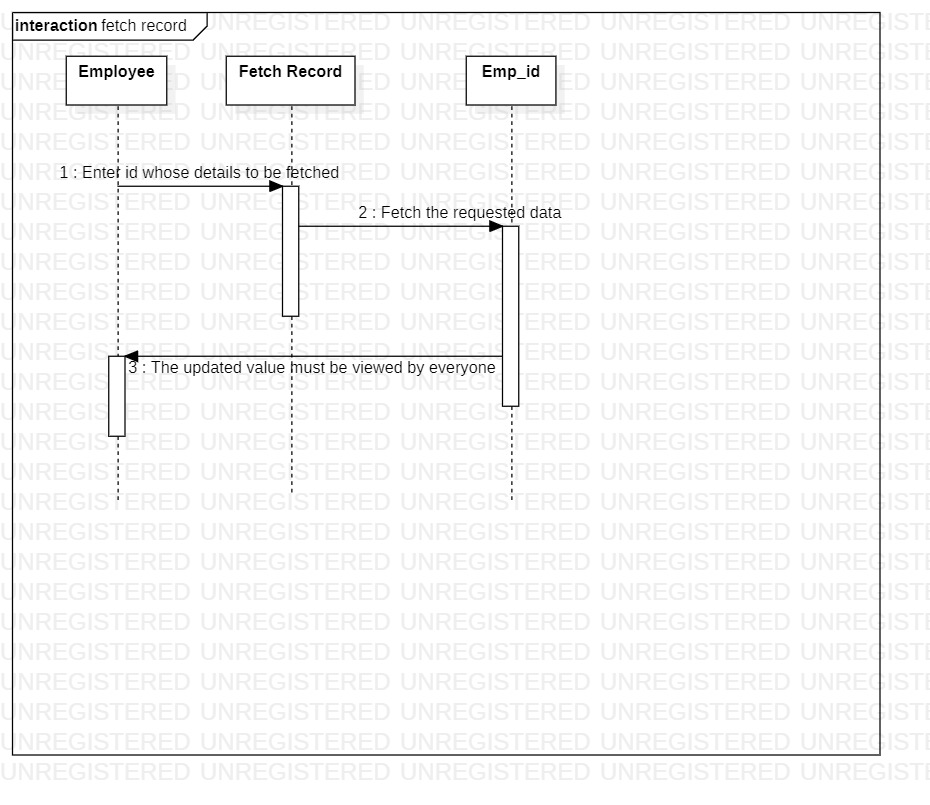
**6.2.3 Sequence diagram for Listing records**

****

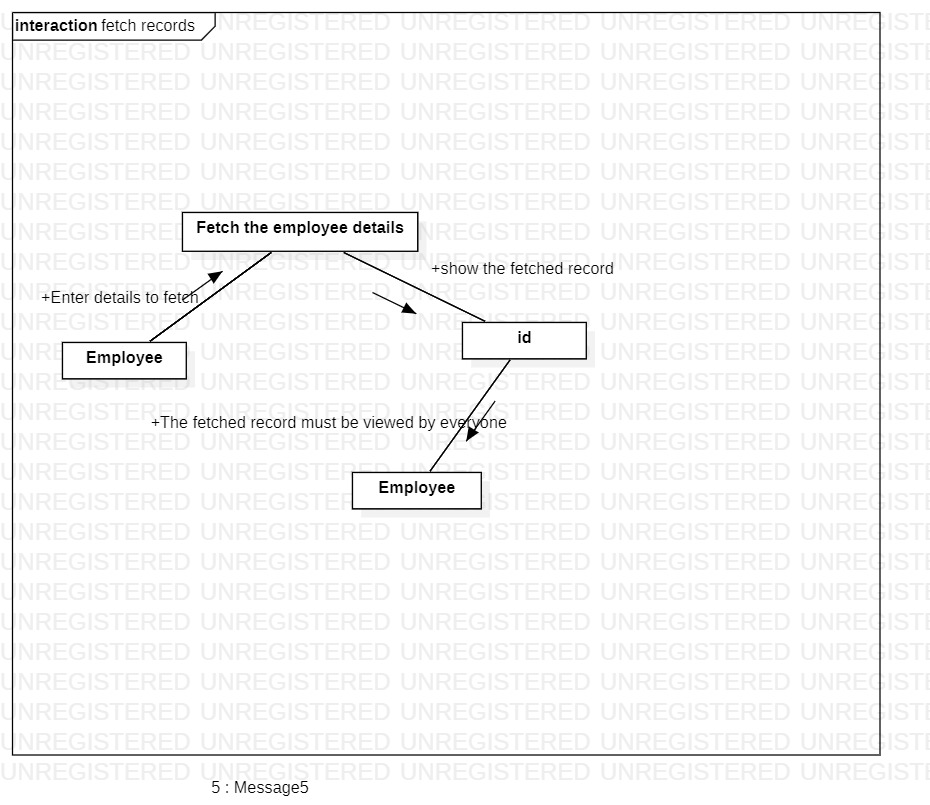
**6.2.4 Collaboration diagram for Listing records**

****

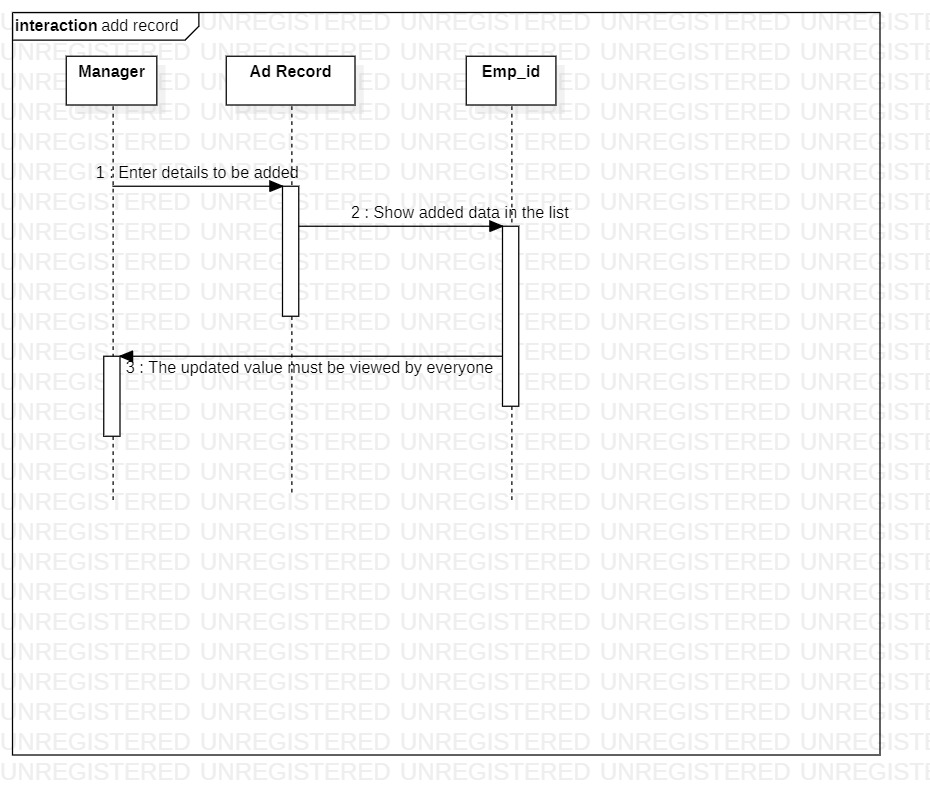
**6.2.5 Sequence diagram for Fetching records**

****

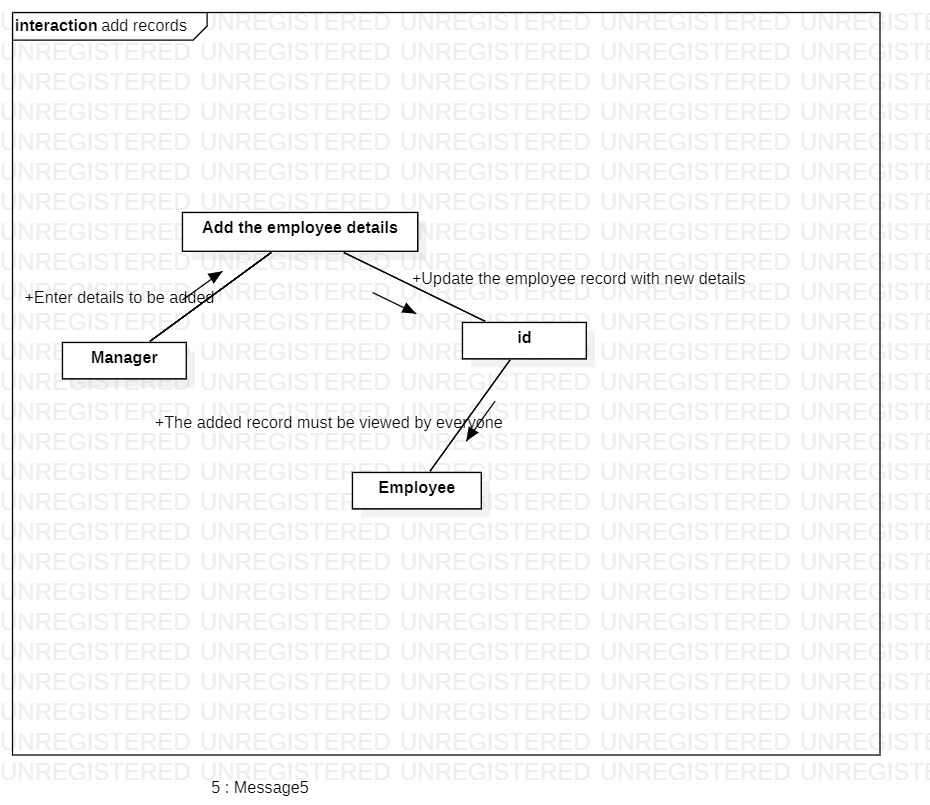
**6.2.6 Collaboration diagram for Fetching records**

****

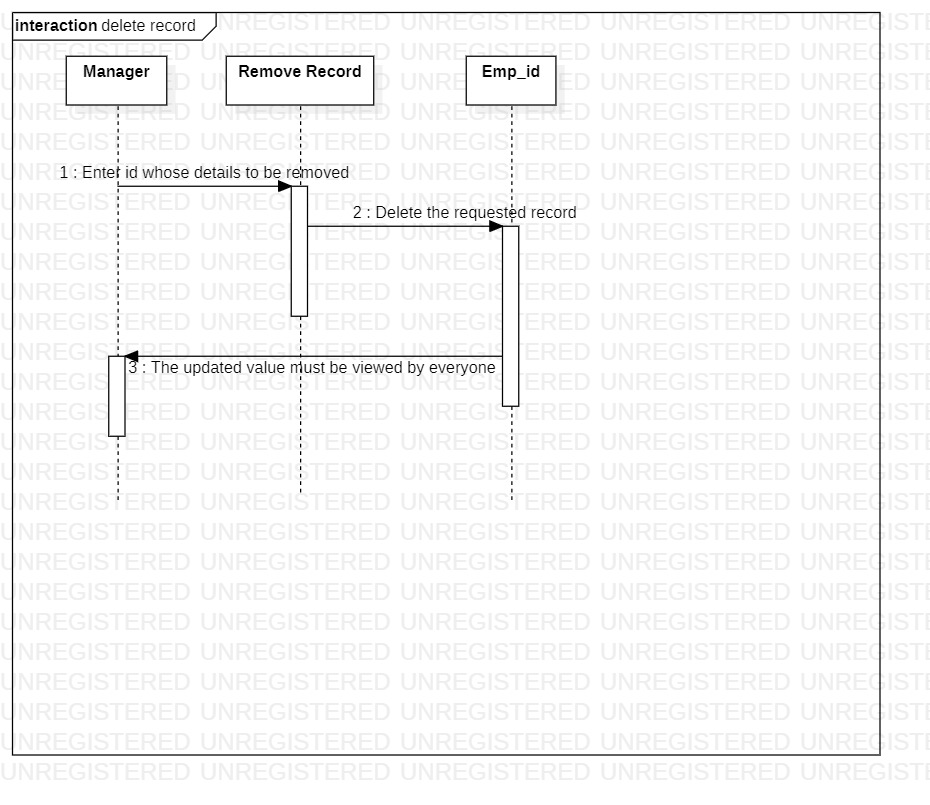
**6.2.7. Sequence diagram for Adding records**

****

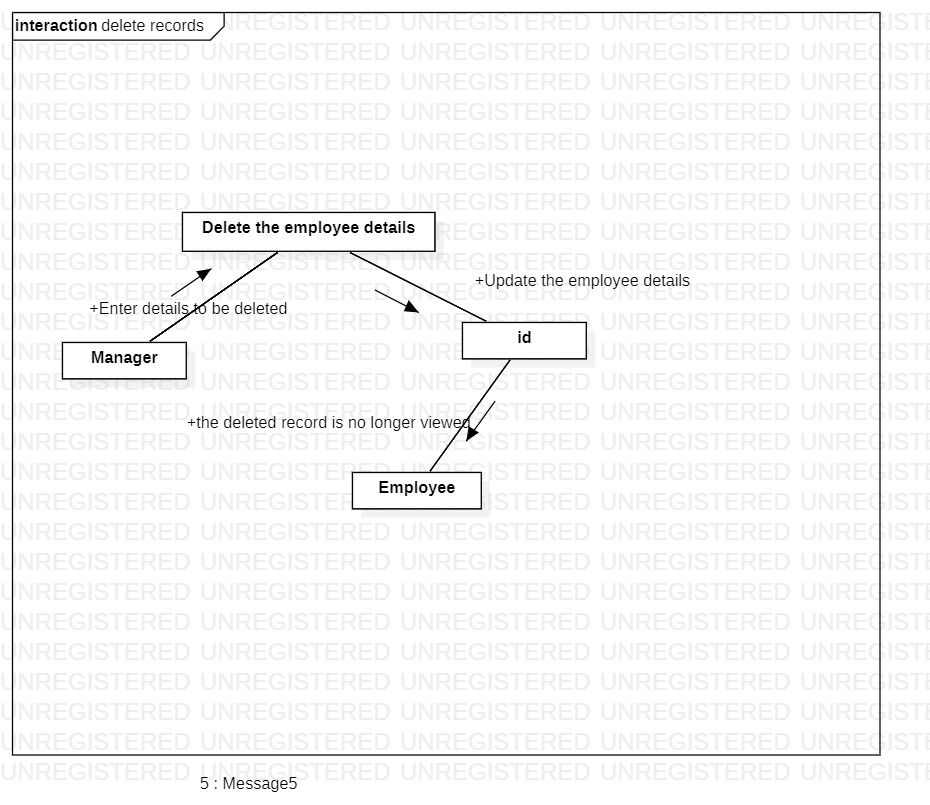
**6.2.8 Collaboration diagram for Adding records**

****

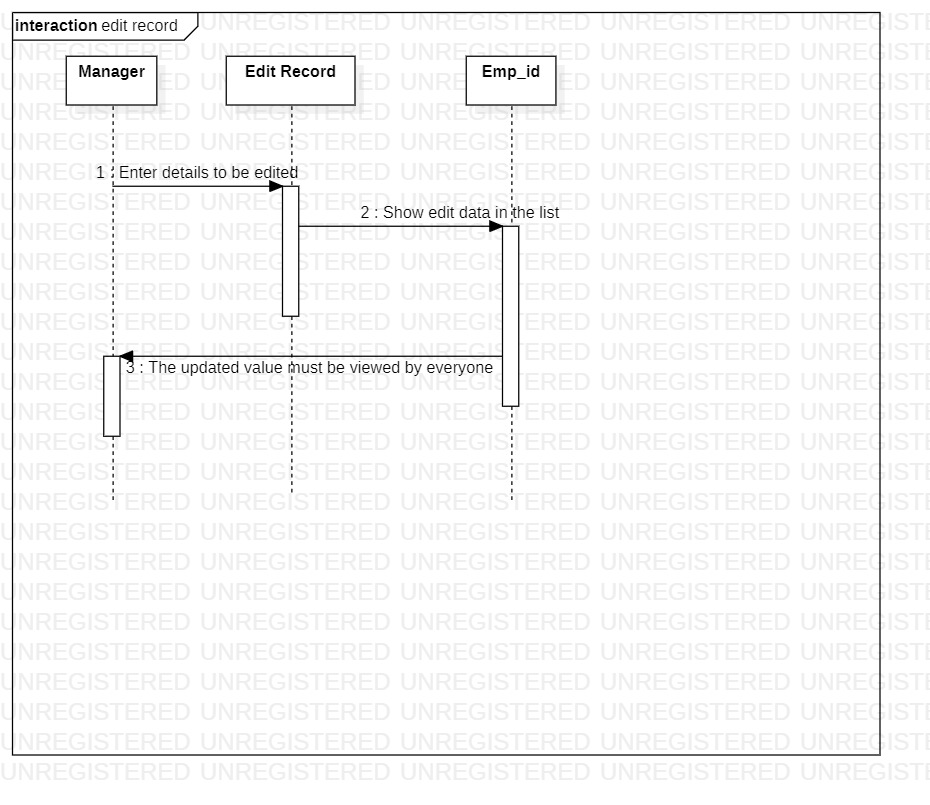
**6.2.9 Sequence diagram for Deleting records**

****

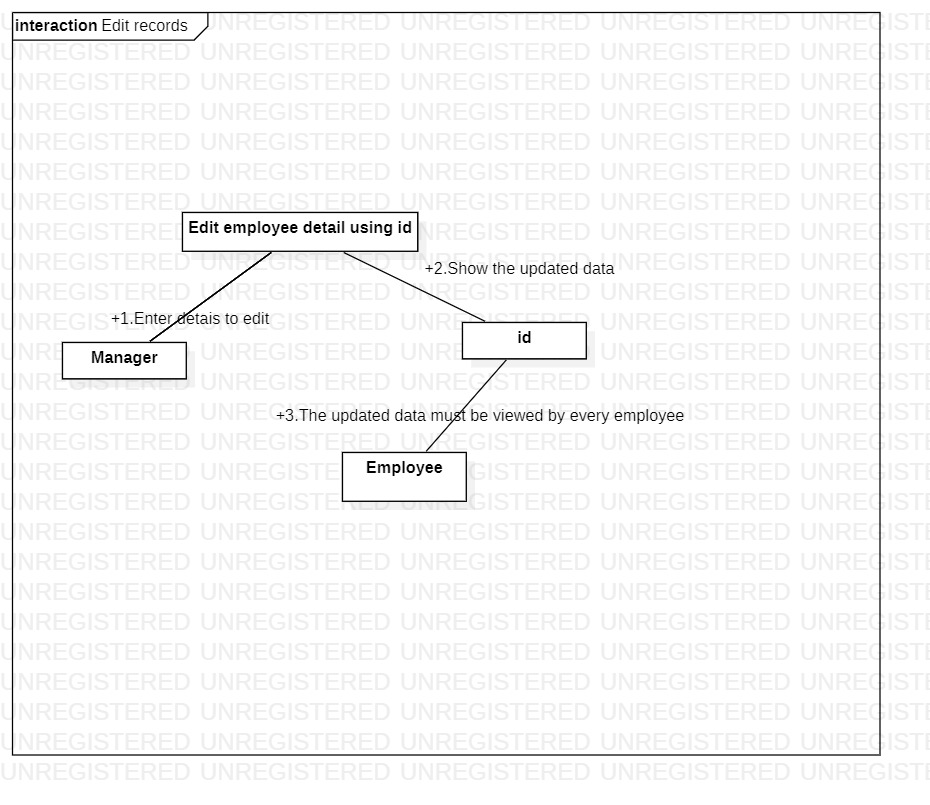
**6.2.10 Collaboration diagram for Deleting records**

****

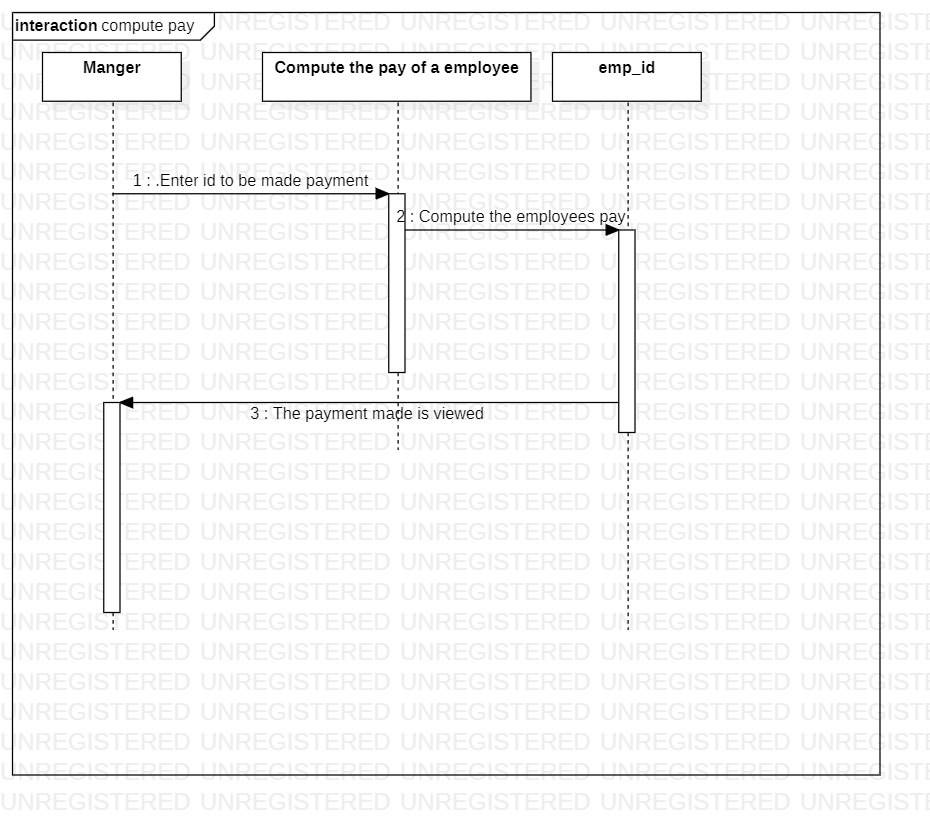
**6.2.11 Sequence diagram for Editing records**

****

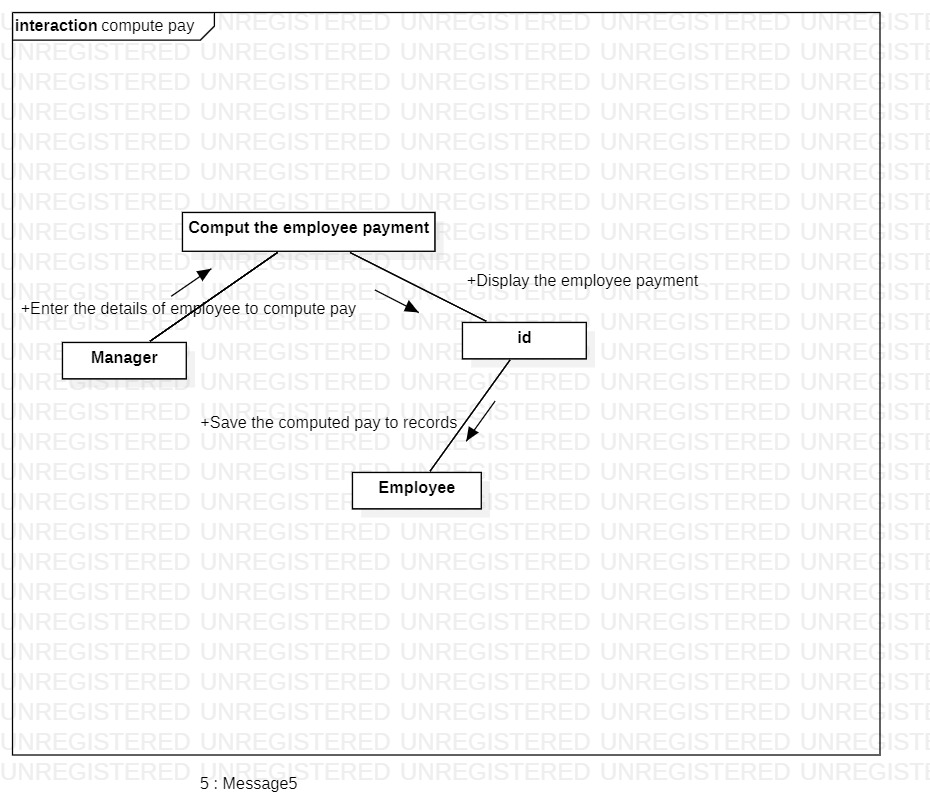
**6.2.12Collaboration diagram for Editing records**

****

**6.2.13 Sequence diagram for Computing payment**

****

**6.2.14 Collaboration diagram for Computing payment**

****

**STATE START DIAGRAM FOR EMP\_ID**

Insert()

cout<<"\n Do you wish to save the above details(y/n) : ";

cin>>ch;

if(ch=='y')

{

fa=fopen("details.txt","a");

fprintf(fa,"%s %s %s %d %d %d \n",e.name,e.code,e.des,e.age,e.exp,e.sal);

fclose(fa);

}

Del()

ff=fopen("details.txt","r");

fg=fopen("temp.txt","w");

while(fscanf(ff,"%s %s %s %d %d %d \n",e.name,e.code,e.des,&e.age,&e.exp,&e.sal)!=EOF)

{

if(strcmp(ccode,e.code)!=0)

{

fprintf(fg,"%s %s %s %d %d %d \n",e.name,e.code,e.des,e.age,e.exp,e.sal);

}

}

char s[]="details.txt";

char t[]="temp.txt";

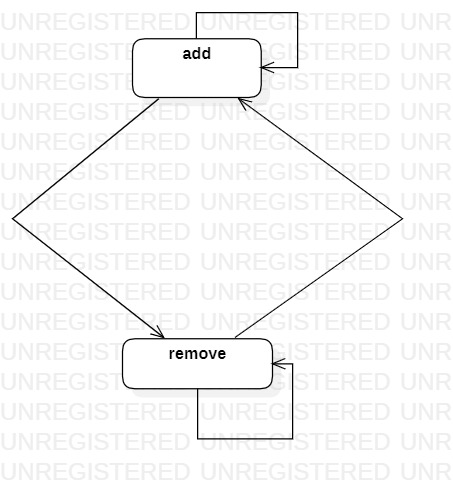
fclose(ff);

fclose(fg);

int a = remove(s);

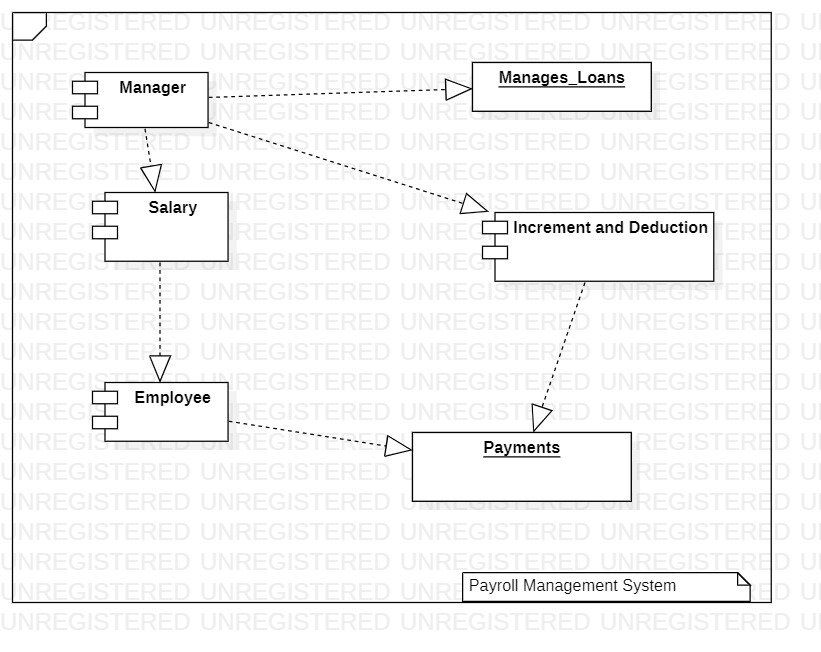
int b = rename(t,s);

getch();

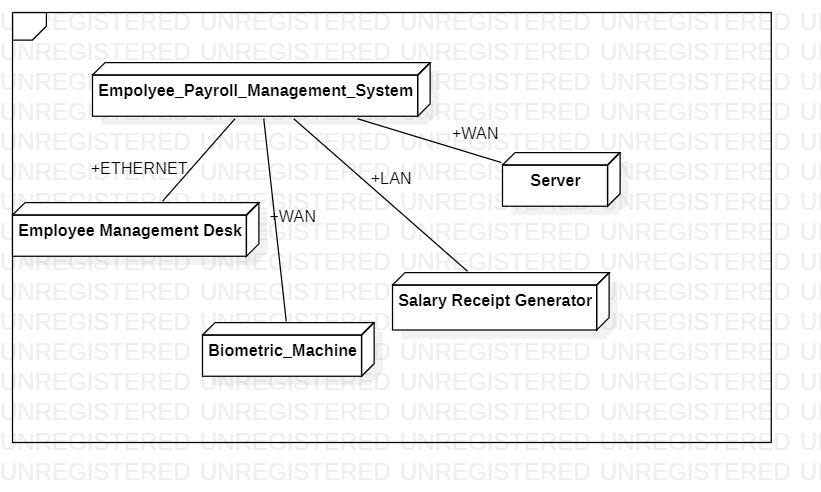


**7 DEPLOYMENT DIAGRAM**

**7.1 COMPONENT DIAGRAM**



**7.2 DEPLOYMENT MODEL**



**8 .CONCLUSION**

“Payroll Management System” software developed for a company has been designed to achieve maximum efficiency and reduce the time taken to handle the Payroll activity. It is designed to replace an existing manual record system thereby reducing time taken for calculations and for storing data. The system uses Asp .Net as front end and Microsoft SQL as a backend for the database. The system is strong enough to withstand regressive daily operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.

**9.REFRENCES**

<https://www.academia.edu/34061794/Introduction_to_Payroll_Management_System_Project_Payroll_Management_System_document>

<https://ijisrt.com/wp-content/uploads/2018/04/Employee-Payroll-Management-System-2-1.pdf>