

PROJECT REPORT

Submitted by

<Neha Maurya><RA2211003011787>

Under the Guidance of

Dr. A. PANDIAN

Associate Professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE ENGINEERING



SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2023



**SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203**

BONAFIDE CERTIFICATE

Certified that this Project Report titled **“An Employ Management System”** the bonafide work done by **<Neha Maurya><RA2211003011787>** who completed the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

| | |
|---|--|
| SIGNATURE Dr. A.PANDIAN OODP – Course Faculty Associate Professor Department of Computing Technologies SRMIST | SIGNATURE Dr.M.Pushpalatha Professor & Head Department of Computing Technologies School of Computing SRMIST |
|---|--|

TABLE OF CONTENTS

| S.No | CONTENTS | PAGE NO |
|------|--------------------------|---------|
| 1. | Problem Statement | |
| 2. | Modules of Project | |
| 3. | Diagrams | |
| | a. Use case Diagram | |
| | b. Class Diagram | |
| | c. Sequence Diagram | |
| | d. Collaboration Diagram | |
| | e. State Chart Diagram | |
| | f. Activity Diagram | |
| | g. Package Diagram | |
| | h. Component Diagram | |
| | i. Deployment Diagram | |
| 4. | Code/Output Screenshots | |
| 5. | Conclusion and Results | |
| 6. | References | |

1.PROBLEM STATEMENT:-Employee management systems can be complex and challenging to implement and maintain.Employee management systems store sensitive employee data, such as personal information, job details, and performance records. Ensuring that this data is secure and protected from unauthorized access is critical. Employee management systems need to integrate with other HR systems, such as payroll, benefits, and time and attendance systems. Ensuring that these systems work together seamlessly can be a challenge.

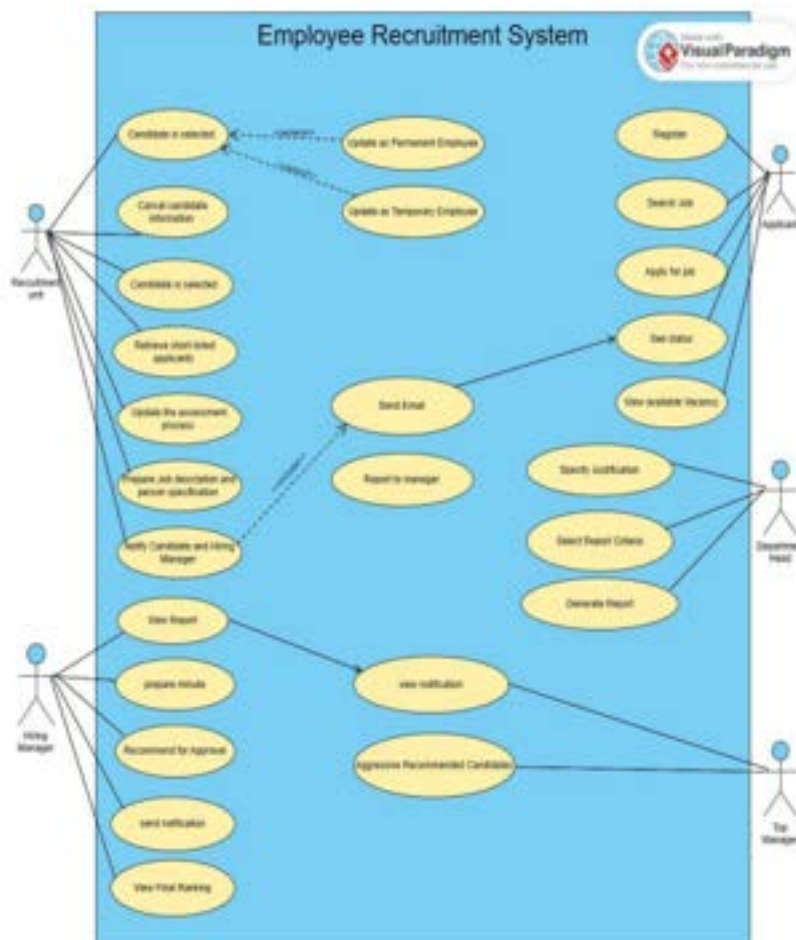
2.MODULES OF PROJECT:-An employee management system is a software solution that enables organizations to manage various aspects of their workforce, such as HR functions, payroll, performance management, employee benefits, time and attendance, and other administrative tasks. Employee management systems are essential for businesses of all sizes to ensure efficient and effective management of their workforce.

3.DIAGRAMS:-

USE CASE DIAGRAM

A use case diagram for a recruitment system would depict the different actors and use cases involved in the process of recruiting and hiring new employees. The main actors in a recruitment system are typically the job candidates, the recruiters or hiring managers, and the human resources (HR) department. Other potential actors could include external job boards or recruiters, interviewers, and reference checkers.

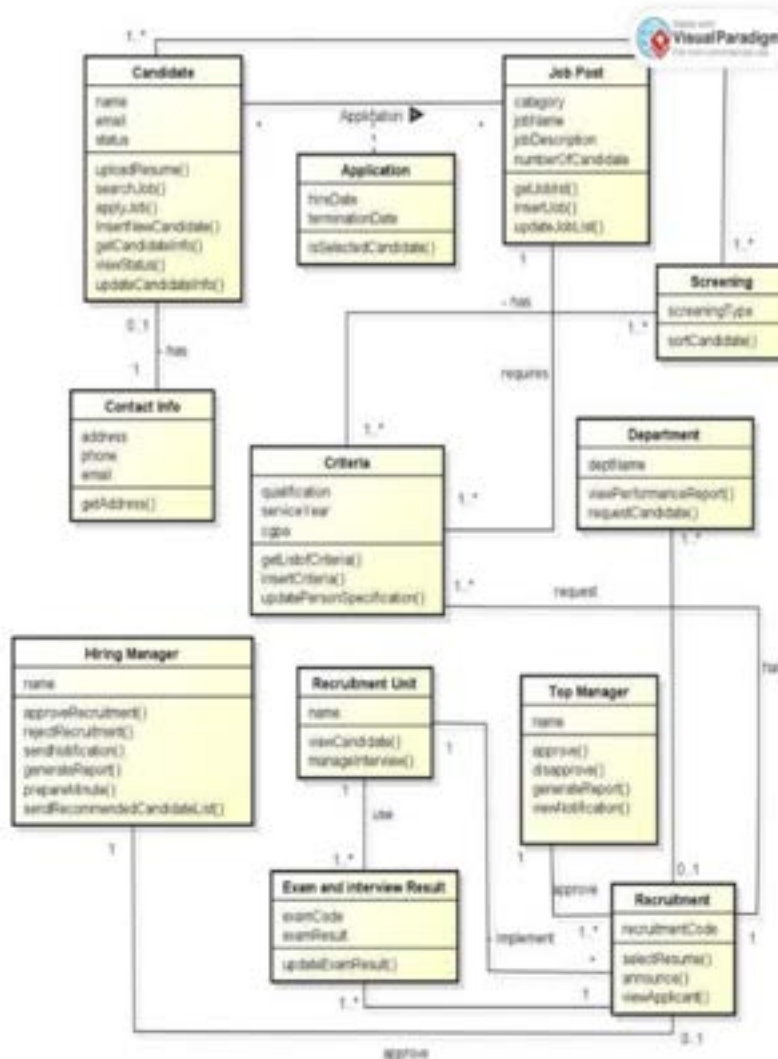
USE CASE DIAGRAM



B. CLASS DIAGRAM

A class diagram is a type of UML diagram that represents the structure of a system by showing its classes, attributes, operations, and relationships between objects. Overall, this class diagram provides a high-level overview of the key classes and relationships in a recruitment system.

CLASS DIAGRAM



C. SEQUENCE DIAGRAM

Sequence diagram is a type of UML diagram which showcases the interaction between the objects in a system. In the employment recruitment system, sequence diagrams help to illustrate the candidates experience, previous position so that proper selection for appropriate role for a position is done.



D. COLLABORATION DIAGRAM

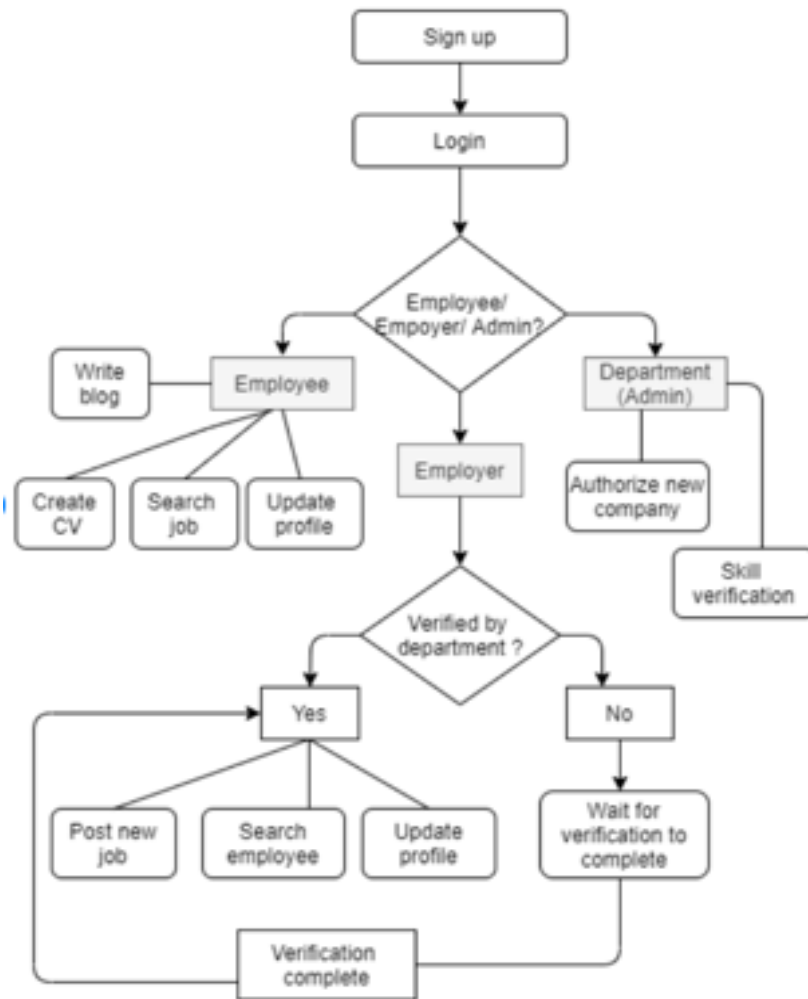
A collaboration diagram is used to visualize the iteration between the objects and actors within a system. It can be used to model the communication among objects and actors. The employment recruitment/management system helps to enter the candidate details like age, experience and position, and also helps in allocating an employment id which makes it easier to search, edit or delete any employee details.

COLLABORATION DIAGRAM



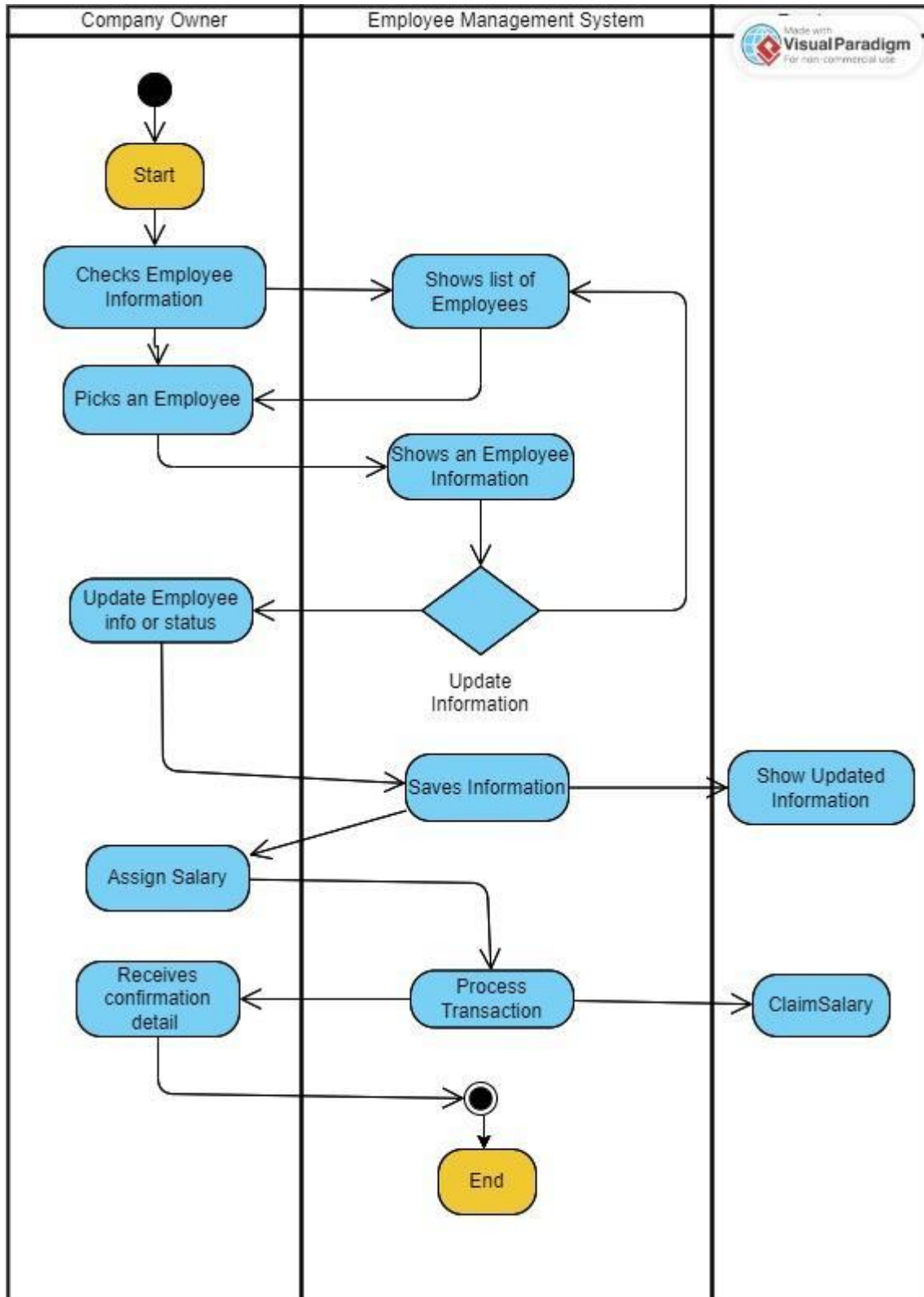
E. STATE CHART DIAGRAM

A state diagram is a type of UML diagram that shows the various states and transitions of an object or system. For an employee requirement system, a state diagram could represent the different states that an employee goes through during the hiring process.



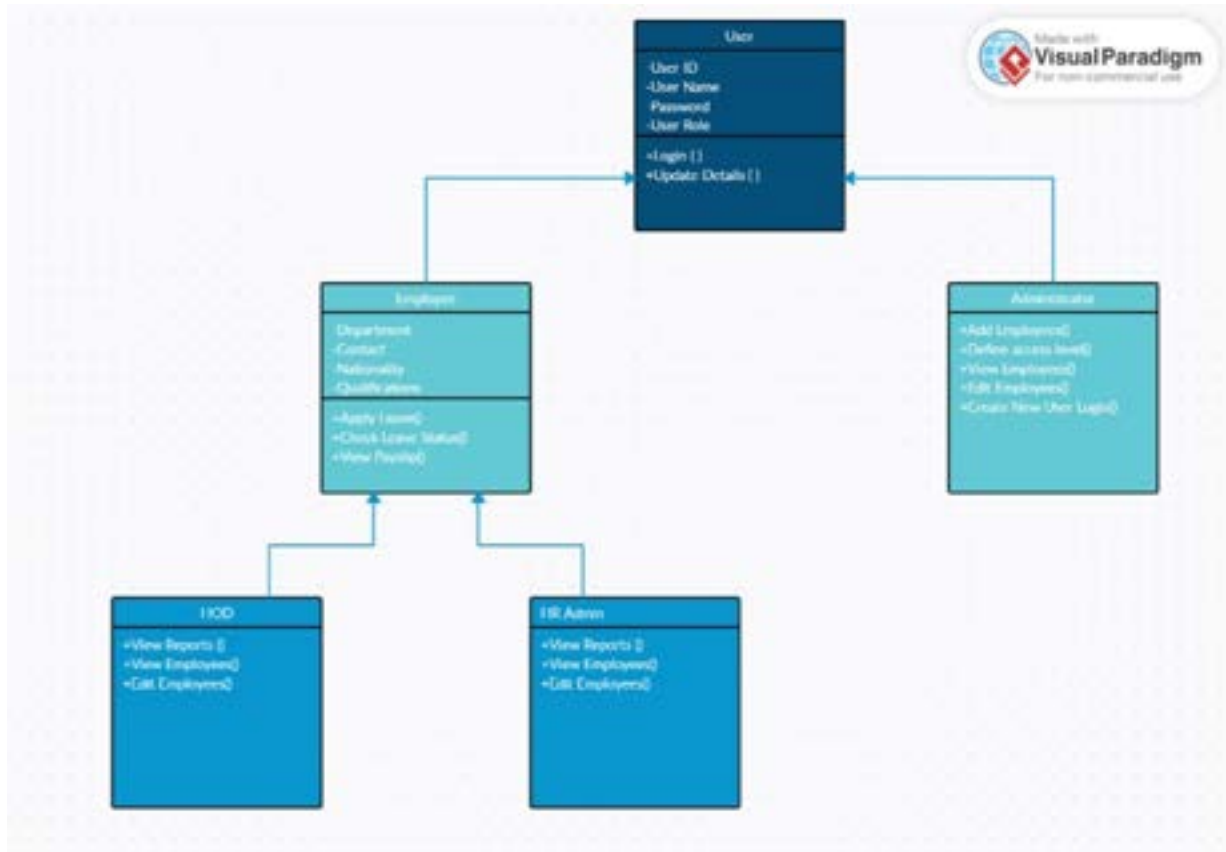
F. ACTIVITY DIAGRAM

_This activity diagram provides a visual representation of the steps involved in the employee requirement process, which can help to identify potential bottlenecks or inefficiencies in the process and ensure that it is optimized for maximum efficiency and effectiveness.



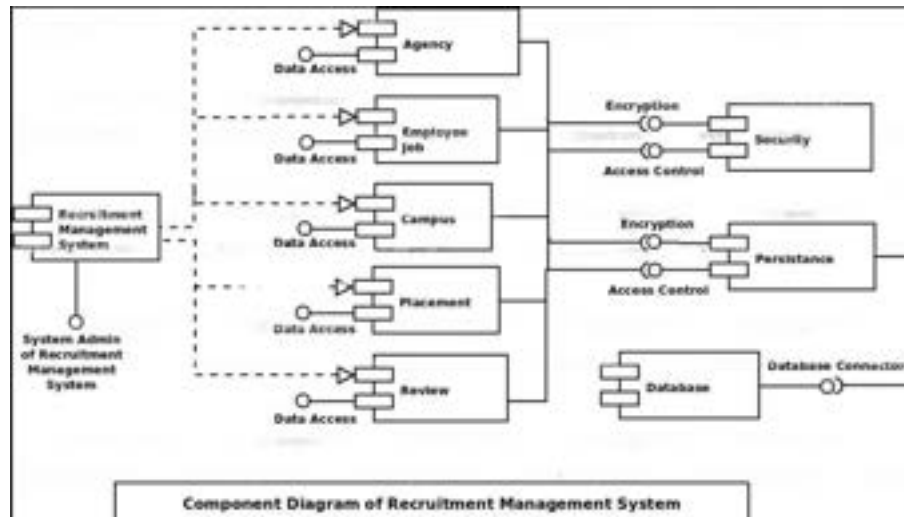
G. PACKAGE DIAGRAM

A package diagram showcases the dependencies between packages in a software system. In the context of an employment management system, a package diagram can be used to show different packages or modules in the system and their relationship with each other. User interface package is used to showcase the screen and forms used by the employee to enter their details and hence interact with the administrators.



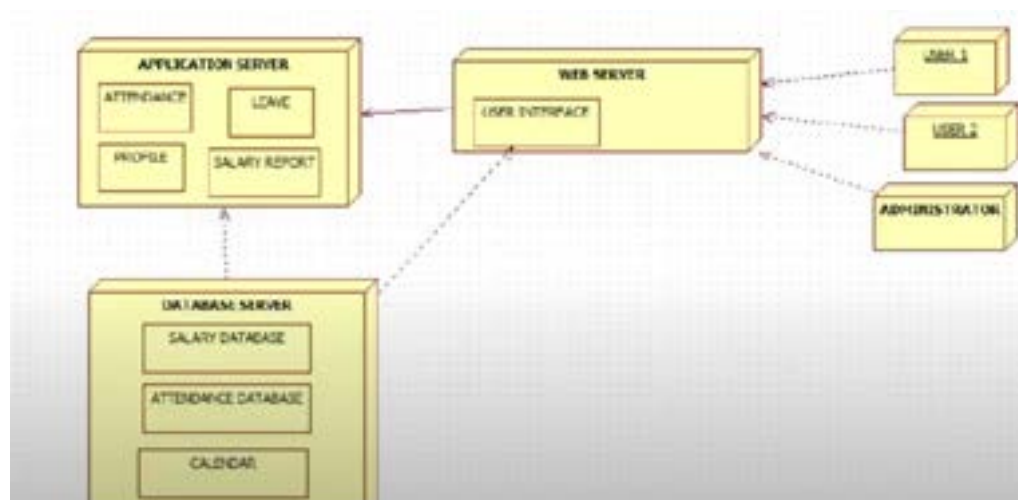
H. COMPONENT DIAGRAM

Component diagram is used to showcase the components of a system and the dependencies between them. In the case of an employment management system, a component diagram might include user interface, database, external system and many more. It would help to provide a high level view of the system architecture and help identify the key components and their relationships.



I. DEPLOYMENT DIAGRAM

A deployment diagram is used to showcase the components of a software system which are deployed and executed on hardware components. In the context of an employment management system, a deployment diagram may be used to illustrate the various components on different hardware nodes.



4.CODE/OUTPUT SCREENSHOTS:-

CODE

```
#include<iostream>
#include <bits/stdc++.h>
#define max 20
using namespace std;
struct employee {
    string name;
    long int code;
    string designation;
    int exp;
    int age;
};

int num;
void showMenu();
employee emp[max], tempemp[max],
    sortemp[max], sortemp1[max];
void build()
{
    cout << "Build The Table\n";
    cout << "Maximum Entries can be "
        << max << "\n";

    cout << "Enter the number of "
        << "Entries required ";
    cin >> num;

    if (num > 20) {
        cout << "Maximum number of "
            << "Entries are 20\n";
        num = 20;
    }
    cout << "Enter the following data:\n";

    for (int i = 0; i < num; i++) {
        cout << "Name ";
        cin >> emp[i].name;

        cout << "Employee ID ";
        cin >> emp[i].code;

        cout << "Designation ";
        cin >> emp[i].designation;

        cout << "Experience ";
        cin >> emp[i].exp;

        cout << "Age ";
        cin >> emp[i].age;
    }
    showMenu();
}
```

```

}
void insert()
{
    if (num < max) {
        int i = num;
        num++;

        cout << "Enter the information "
              << "of the Employee\n";
        cout << "Name ";
        cin >> emp[i].name;

        cout << "Employee ID ";
        cin >> emp[i].code;

        cout << "Designation ";
        cin >> emp[i].designation;

        cout << "Experience ";
        cin >> emp[i].exp;

        cout << "Age ";
        cin >> emp[i].age;
    }
    else {
        cout << "Employee Table Full\n";
    }

    showMenu();
}
void deleteIndex(int i)
{
    for (int j = i; j < num - 1; j++) {
        emp[j].name = emp[j + 1].name;
        emp[j].code = emp[j + 1].code;
        emp[j].designation
            = emp[j + 1].designation;
        emp[j].exp = emp[j + 1].exp;
        emp[j].age = emp[j + 1].age;
    }
    return;
}
void deleteRecord()
{
    cout << "Enter the Employee ID "
          << "to Delete Record ";
    int code;
    cin >> code;
    for (int i = 0; i < num; i++) {
        if (emp[i].code == code) {
            deleteIndex(i);
            num--;
            break;
        }
    }
    showMenu();
}

```

```

void searchRecord()
{
    cout << "Enter the Employee "
          << " ID to Search Record ";

    int code;
    cin >> code;
    for (int i = 0; i < num; i++) {
        if (emp[i].code == code) {
            cout << "Name "
                  << emp[i].name << "\n";

            cout << "Employee ID "
                  << emp[i].code << "\n";

            cout << "Designation "
                  << emp[i].designation << "\n";

            cout << "Experience "
                  << emp[i].exp << "\n";

            cout << "Age "
                  << emp[i].age << "\n";
            break;
        }
    }
    showMenu();
}

void showMenu()
{
    cout << "-----"
          << "Employee"
          << " Management System"
          << "-----\n\n";

    cout << "Available Options:\n\n";
    cout << "Build Table      (1)\n";
    cout << "Insert New Entry  (2)\n";
    cout << "Delete Entry      (3)\n";
    cout << "Search a Record   (4)\n";
    cout << "Exit              (5)\n";

    int option;
    cin >> option;
    if (option == 1) {
        build();
    }
    else if (option == 2) {
        insert();
    }
    else if (option == 3) {
        deleteRecord();
    }
    else if (option == 4) {
        searchRecord();
    }
    else if (option == 5) {

```

```

        return;
    }
    else {
        cout << "Expected Options"
              << " are 1/2/3/4/5";
        showMenu();
    }
}
int main()
{
    showMenu();
    return 0;
}

```

OUTPUT

Output

Available Options:

| | |
|------------------|-----|
| Build Table | (1) |
| Insert New Entry | (2) |
| Delete Entry | (3) |
| Search a Record | (4) |
| Exit | (5) |

1

Build The Table

Maximum Entries can be 20

Enter the number of Entries required 2

Enter the following data:

Name Avani

Employee ID 500

Designation manager

Experience 5

Age 19

Name Marushka

Employee ID 600

5.CONCLUSION:-

In conclusion, an employee recruitment system is an important tool for organizations looking to streamline and automate their recruitment processes. The system enables HR personnel, recruiters, and hiring managers to manage the recruitment process more efficiently and effectively, while providing a positive candidate experience.

The successful implementation of an employee recruitment system requires a structured approach that includes gathering requirements, analyzing and designing the system, implementing the system, and deploying and maintaining it. UML diagrams can be used to visualize the system design and to communicate the system's functionality to stakeholders. The outcome of an employee recruitment system project should be a system that enables the organization to hire the best candidates quickly and efficiently, while providing a positive candidate experience and reducing the costs and time associated with traditional recruitment methods. The system can be applied in different industries and sectors, and can be customized to meet the specific needs of the organization.

Overall, an employee recruitment system can benefit organizations by improving the quality of hires, reducing the time and costs associated with recruitment, and creating a positive candidate experience.

6.REFERENCES:-

1. Mathur, R., & Gupta, S. (2016). Design of an online recruitment system using UML diagrams. *International Journal of Advanced Research in Computer Science*, 7(4), 129-133.
2. Zarei, B., Rezaei, M. H., & Alavi, M. (2016). Developing a web-based recruitment system using UML and RUP methodologies. *Journal of Software Engineering and Applications*, 9(8), 401-415.