ABSTRACT

Employees are the backbone of any company, therefore, their management plays a major role in deciding the success of an organization. Employees Management Software makes it easy for the employer to keep track of all records. This software allows the administrator to edit employees, add new employees, transfer/promote/terminate employees. Each employee in the database is associated with a position that can be added and edited when the need arises. Employees can be transferred between positions easily without having to retype back their information in the database. You can check to see if there are duplicate positions/employees in the database. Most of all, the employer can assign tasks to employees and assess their progress in order to keep track of employee performance.

This system brings about an easy way of maintaining the details of employees working in any organization. It is simple to understand and can be used by anyone who is not even familiar with simple employees system. It is user-friendly and just asks the user to follow step by step operations by giving easy to follow options. It is fast and can perform many operations for a company. The goal of this project is to design and develop an employee management system to fill existing gaps in the electronic management of employees.

TABLE OF CONTENTS

1.INTRODUCTION

In this world of growing technologies everything has been computerized. With large number of work opportunities the Human workforce has increased. Thus there is a need of a system which can handle the data of such a large number of Employees in an organization. This project simplifies the task of maintain records because of its user friendly nature.

1.1 PROBLEM STATEMENT

The employee Management system is an application that enables users to create and store Employee Records. The application also provides facilities of a payroll system which enables the user to generate Pay slips too. This application is helpful to the department of the organization which maintains data of employees related to an organization.

1.2 OBJECTIVES OF PROJECT

In this world of growing technology everything has been computerized. With large number of work opportunities the Human workforce has increased. Thus there is a need of a system which can handle the data of such a large number of Employees in an organization. This project simplifies the task of maintaining records because of its user friendly nature.

2.SYSTEM REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS

Provides the searching facilities based on various factors. Such as Employee,Salary,Contact Number,Designation.

It tracks all the information of Employee details.Manage the information of Employee.Shows the information and description of the Employee. To increase efficiency of managing the Employee details.

Manage the information of Employee Editing, adding and updating of Employee details such as EmployeeID,Salary,Contact Number,Designation is improved which results in proper resource management of Employee data.

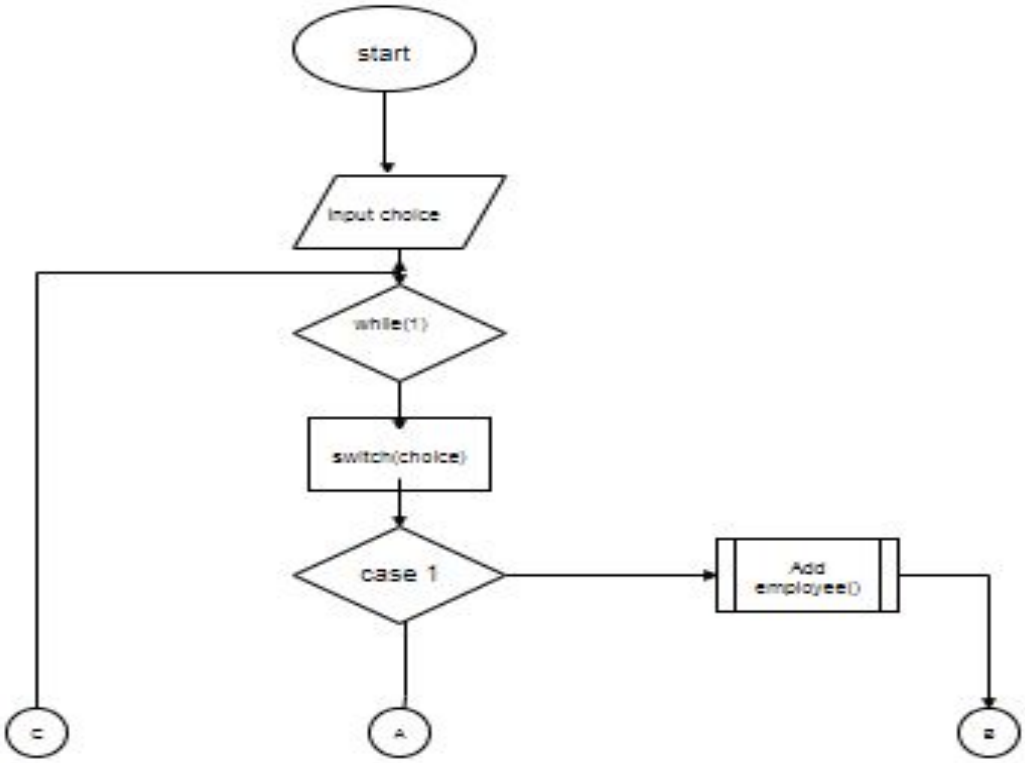
2.2 SOFTWARE AND HARDWARE REQUIREMENT

Software Requirements:

- CodeBlocks
- MinGW 32
- Windows
- XAMPP Control Panel
- Apache
- MySQL
- DevC++
-

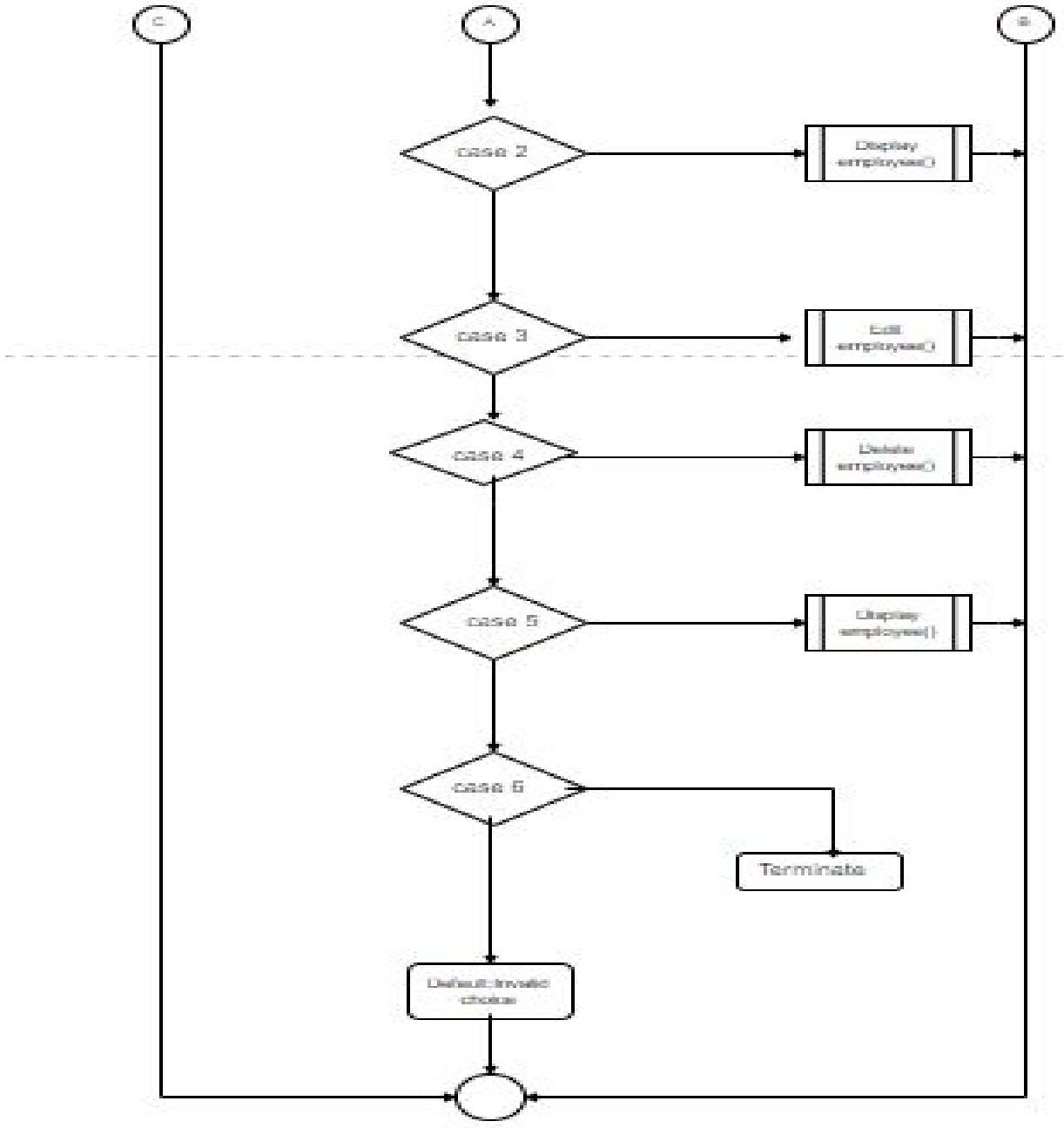
Hardware Requirements:

- Computer or laptop



3.SYSTEM DESIGN

3.1 ARCHITECTURE/ DATA FLOW DIAGRAM


C
A
B
case 2
Display employee()
case 3
Edit employee()
case 4
Delete employee()
case 5
Display employee()
case 6
Terminate
Default: Invalid choice

3.2 MODULES

class employee();

void show();

void add();

void editemployee();

void deleteemployee();

void searchid();

void user();

int main();

4.SYSTEM IMPLEMENTATION

4.1 MODULE DESCRIPTION

class employee();

class employee is the most important module in this project. void show(); void add(); void editemployee(); void deleteemployee(); void searchid(); are the methods in this class. Each method is performing specific task. All the methods are in public access mode. Some of the methods are called from main function and few are called inside class from another member functions.

void show();

This function is called from the main function. This displays all the available operations which user can perform in this system. The options available are adding employee , display records, deleting employees, modify the details and updating details .

void add();

This function is to add new employee's details into database. It will ask user to enter the details of new employee such as ID number, Name, Salary, Contact number. It will be stored in the database .

void editemployee();

This module is for editing the details of the existing records. Whenever user wants to modify the details of already existing employee details. This method calls the method SearchID(). User can enter details of that particular student with specified ID with the proper ID. If the ID is not found, It displays the message tells “Not found!”

void deleteemployee();

This module performs deleting of any record in database. It will delete the record or details of a employee associated with the specified ID number. It will take employee ID as input. If the details are present in the database having specified ID then it will delete that particular record. If any error comes during this operation, then error message will be displayed. If the ID is not present in the database then it will display the message telling ID not found.

void searchid();

This method is called inside the class. It will ask the user to insert the employee ID. This method search for the entered ID in the database. It will return a Boolean value True when ID is present, False in case the ID is not found.

void user();

This function calls most of the method listed above. It will perform switch condition on the user input choice. According to the choice of user it will call the methods performing various operations.

int main();

This is the main function which calls the function void user().

4.2 PSEUDO CODE

```
#include <iostream>

#include<windows.h>

#include<winsock.h>

#include<mysql.h>

#include <sstream>

#include<stdio.h>


using namespace std;


// Global Variable

int qstate;

MYSQL* conn;

MYSQL_ROW row;

MYSQL_RES* res;

// Global Variable End


class employee

{
```

```
public:
   employee();
   void show();
   void add();
   void editemployee();
   void deleteemployee();
   void searchid();
};



employee::employee()
{
  conn = mysql_init(0);
   if (conn)
   {
      cout << "Database Connected" << endl;



   }
   else {
      cout << "Failed To Connect!" << mysql_errno(conn) << endl;
   }
   conn = mysql_real_connect(conn, "localhost", "root", "", "employee", 0, NULL, 0);
   if (conn)
   {
```

```
        cout << "Database Connected To MySql" << conn << endl;


    }
    else
        cout << "Failed To Connect!" << mysql_errno(conn) << endl;
}



void employee::add()
{



        string eid;
        string name;
                string phno;
        float salary;
        string desig;

        cout << "Enter name of Employee: ";
        cin >> name;

        cout << "Enter Employee ID : ";
        cin >> eid;

        cout << "Enter Employee Contact number: ";
```

```
cin >> phno;

cout<<"Enter Employee Salary : ";
cin >> salary;

cout << " Enter Employee Designation : ";
cin >> desig;

stringstream ss;
ss << "INSERT INTO details(name, eid, phno, salary, desig) VALUES('" << name <<"','"<< eid <<"','"<< phno <<"'," << salary <<",'"<< desig <<"')";

string query = ss.str();
const char* q = query.c_str();
qstate = mysql_query(conn, q);

if(qstate == 0){
   cout << "Record Inserted" << endl;
}else{
   cout << "Insert Error" << mysql_error(conn) << endl;
}

system("pause");
system("cls");
```

```
}


void employee::editemployee()
{
	string eid;
	float salary;
	cout << "Enter EmployeeID : ";
	cin >> eid;
	cout << "Enter Employee Salary : ";
	cin >> salary;

	stringstream ss;
	ss << "UPDATE details SET salary = salary +" << salary << " WHERE eid = '" << eid << "'";

	string query = ss.str();
	const char* q = query.c_str();
	qstate = mysql_query(conn, q);

	 if(qstate == 0){
	   cout << "Record Updated" << endl;
	}else{
	   cout << "Insert Error" << mysql_error(conn) << endl;
	}
```

```
        system("pause");
        system("cls");

}

void employee :: searchid()
{
        string eid;
        cout << "Enter Employee ID : ";
        cin >> eid;

        stringstream ss;
        ss << "select * from details  WHERE eid = '" << eid << "'";

        string query = ss.str();

        const char* q = query.c_str();
        qstate = mysql_query(conn, q);
        if(!qstate){
           res = mysql_store_result(conn);

           cout<<" Name\t |\tEID\t |\tPhone Number\t |\tSalary\t |\tDesignation\n";
           while(row = mysql_fetch_row(res)){
```

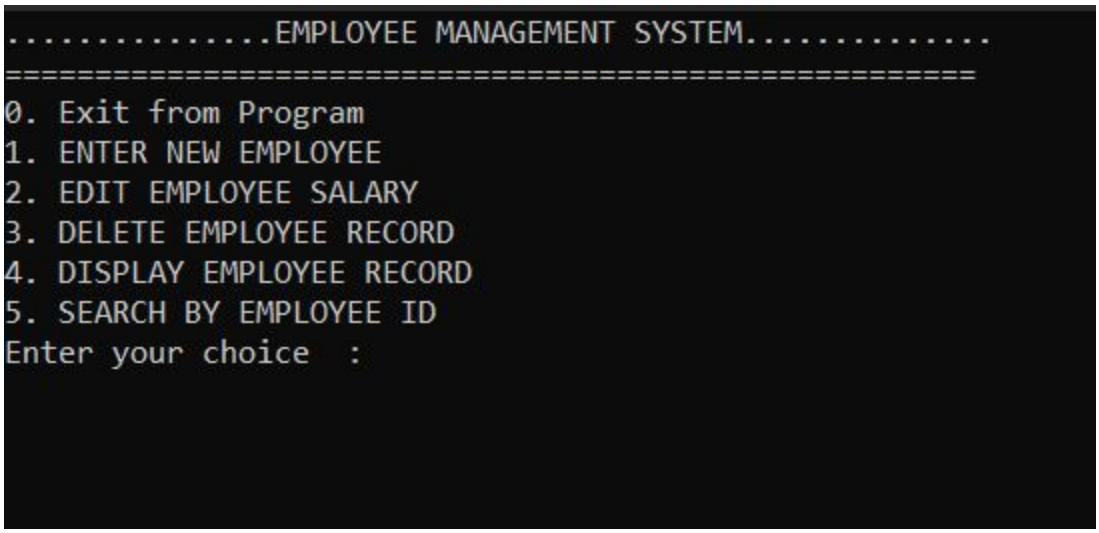
```
            cout << row[0] <<"\t | \t" << row[1] <<"\t | \t" << row[2] <<"\t\t | \t" << row[3] <<"\t | \t" << row[4]  <<endl << endl;

          }

       }




   system("pause");

   system("cls");




}

void employee::show()

{

   qstate = mysql_query(conn, "SELECT name, eid, phno, salary , desig FROM details");


   if(!qstate){

         res = mysql_store_result(conn);

         cout<<" Name\t |\tEmployeeID\t |\tPhNo\t |\tSalary\t |\tDesignation\n";

         while(row = mysql_fetch_row(res)){

            cout << row[0] <<"\t | \t\t" << row[1] <<"\t | \t" << row[2] <<"\t | \t" << row[3] <<"\t | \t" << row[4]  <<endl << endl;

          }

       }
```

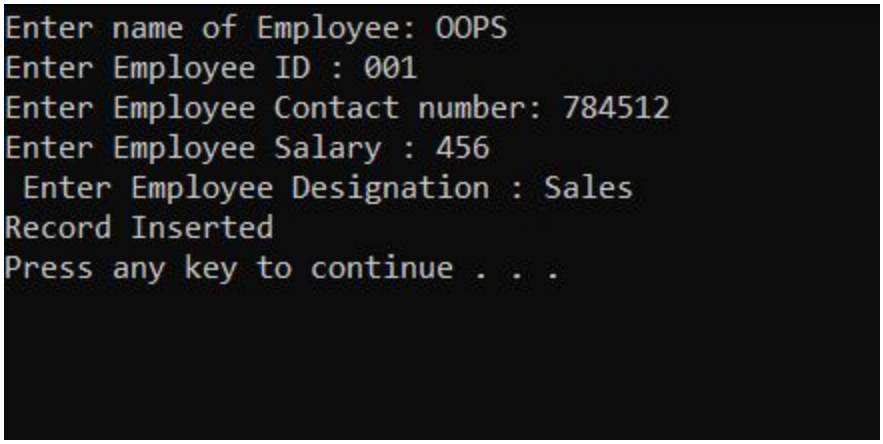
```
}



void employee::deleteemployee()


{
   string id;
   cout<<"Enter Employee Id to delete: ";
   cin>>id;




   string query = "SELECT eid FROM details;";




      query = "delete from details where eid = '"+id+"';";
      const char* q = query.c_str();
      qstate = mysql_query(conn, q);


      if (!qstate)
      {
         cout << "Successfully Deleted From Item Record." << endl;
      }
```

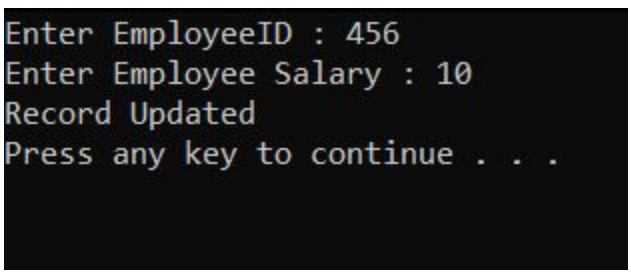
```
        else
        {
           cout << "Failed To Delete!" << endl;
        }

}

void user()
{
            int ch;
            do
            {
                        system("cls");
                        cout<<".............EMPLOYEE MANAGEMENT SYSTEM.............\n";
                        cout<<"=====================================================\n";
                        cout<<"0. Exit from Program\n";
                        cout<<"1. ENTER NEW EMPLOYEE\n";
                        cout<<"2. EDIT EMPLOYEE SALARY\n";
                        cout<<"3. DELETE EMPLOYEE RECORD\n";
                        cout<<"4. DISPLAY EMPLOYEE RECORD\n";
                        cout<<"5. SEARCH BY EMPLOYEE ID\n";
                        cout<<"Enter your choice  : ";
                        cin>>ch;
                        system("cls");
                        switch(ch)
```

```
                {
                        case 1: s.add(); break;
                        case 2: s.editemployee(); break;
                        case 3: s.deleteemployee(); break;
                        case 4: s.show(); break;
                        case 5: s.searchid(); break;
                }
                system("pause");
        }while(ch);
}



int main()
{
    user();
    return 0;
}
```

5. OUTPUT SCREENSHOTS







```
 Name    |      EmployeeID       |      PhNo     |      Salary   |      Designation
1        |                       |               |      0        |

OOPS     |             001       |      784512   |      460      |      Sales

1        |             1         |      1        |      1        |      1

2        |             2         |      2        |      4        |      2

3        |             3         |      3        |      3        |      3

5        |             5         |      5        |      5        |      5

6        |             6         |      6        |      6        |      6

45       |             85        |      741      |      96       |      74

9        |             9         |      9        |      9        |      9

Press any key to continue . . .
```

```
Enter Employee ID : 001
 Name    |      EID      |      Phone Number     |      Salary   |      Designation
OOPS     |      001      |      784512           |      460      |      Sales

Press any key to continue . . .
```

6.CONCLUSION

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

A description of the background and context of the project and its relation to work already done in the area. The description of Purpose, Scope, and applicability.

We define the problem on which we are working in the project. We describe the requirement Specifications of the system and the actions that can be done on these things. We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system. We included features and operations in detail, including screen layouts. We designed user interface and security issues related to system. Finally the system is implemented and tested according to test cases.