Introduction

The aim of this assignment is to create a system for Administrator and Employee in which admin performs and operations on add, search, edit, view, delete and calculate the each salary of peoples and employee only view on profile. Use Numeric Functions, and Work with OOPS Concept. The system allow different functions classified into menu driven program. We write a menu driven program to perform various calculation in which all the validation are used like a main menu, admin id password, employee id password and inside of program.

The project is to development on "Employee management system" using C++ code with object oriented concept and to introduce current technical issues in the field of object oriented programming (OOP).

The basic objective were:-

1. There are two modules –administrator and employee.
2. The application will be used by the administrator to add, search, edit, view, delete and calculate the each salary of peoples.
3. Employee can view only on personal information.
4. The system are more attractive, user friendly and use simple or basic functions.
5. Validation are used for novice employer to fill correct data.

Project Description

FUNCTIONALITY –

One of the most important modules of the Employee Management System is the LOGIN module. Each Employee of the system has its own user name and password. The administrator of the application has the highest access priority which enables him/her to perform all the functionalities that exist in the application. In addition to that, the administrator is able to register Employee. The Employee has the lowest access priority that only enables them to view their personal information and annual salary.

The administrator is allowed to:

- Register Employee as users of the system
- Add, search and edit personal information of Employee
- Delete record of any Employee

- Calculate salary package

The following details of each Employee are recorded:

- Identification Card (IC) Number
- Name
- Gender
- Qualification
- Designation
- Department
  (HR/MARKETING/PRODUCTION/QA)
- Date Joined
- Nationality
- Date Of Birth
- Marital Status

Each staff has a unique Identification Card (IC) Number and this will be automatically generated by the system.

Classification of designation for Employee is categorised as follows:

- Chief Executive Officer
- Managing Director
- Manager
- General Manager
- Assistant Manager
- Supervisor
- Labour

The category of designation and the related departments are determined by you in the system.

An Employee can view only his/her salary package.

The searching of any record in the system must be done by using the following categories:

- IC Number
- Designation
- Department

Design of the implemented code

Description: - Administrator has the highest access priority means he can do all the functionality given in the Employee Management system. He can on add, search, edit, view, delete and calculate the each salary of peoples. Also have highest priority is given to human resource person who can on add, search, edit, view, delete and calculate the each salary of peoples employee person he would not have any kind of access priority to the admin record and at last lowest access priority is given to the employee who can only view his own personal information.

Justification: - Administrator have highest priority to add, search, edit, view, delete and calculate the each salary of peoples. We know that every organization have generally admin of the any organization have more powerful persons and he has more capability to handle the organization so I have given top position to the admin. Second position is employee to view own personal information neither add, search, edit, view, delete and calculate the each salary of peoples and nor change the own ID Password.

Description and justification of the implementation codes in terms of object oriented Programming concepts:-

What is object?

Ans: - Basically object is instance of class where showing behavior and state and another way is This is the basic unit of object oriented programming. That is both data and function that operate on data are bundled as a unit called as object.

OOP includes the following features:-

1. Class
2. Object
3. Inheritance
4. Method
5. Encapsulation
6. Abstraction
7. Polymorphism

1. Class: - class is the collection of object or saying blueprint of object.

As example:-

```
class class-name
{
private:
variable declaration;
function declaration;
public:
variable declaration;
function declaration;
};
```

2. **Object**: - Object is defined as an instance of a class which has its own state and behavior. The set of values of the attributes of an object is called its state.

3. **Inheritance:** - Parent child relationship among class or Inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class.

```
class employee
{
        public:
                int cHR;
                int  intempNo,num,type, countAg, countAgHR ;
                char uname[15],pword[8];
                char charUtype[8],charIcn[10], charName[20], charGender[5],
charDesignation[15], charDepartment[20], charDateofjoined[10], charNationality[15],
charDateofbirth[10], charMarritalstatus[2];



public:
        bool valid_Date(char* a);
        bool valid_string(char* a);
        bool valid_alpha(char* a);
        void getdata();
        void putdata(char *);
        void loginemployee(void);
        void headdata(void);
   void logoutAll(void);
```

```
    void showbanner(void);
        int getemployeeNo();



};

class E_M_S: public employee
{
public:
        void enterRecords(void);
        void viewRecords(void);
        void searcHRecords(void);
        void editRecords(void);
        void deleteRecords(void);
        int searchChoice(void);
        int viewChoice(void);
        int autoGenerate(void);
        void loginHR(void);
        void adminEmp();
        int readSectionHR(void);
};
```

There are different type of inheritance: -

A). **Single Inheritance**: - Type of inheritance in which a derived class inherits from a single base class.

B). **Multiple Inheritance:** - Type of inheritance in which a derived class inherits from multiple base classes.

C). **Multilevel Inheritance**: - Type of inheritance in which a class inherits from a class which itself inherits from another class thus forming a chain like structure.

D). **Hybrid Inheritance:** -it is the combination of all above inheritance.

4. **Method**: - A method in object-oriented programming is a subroutine associated with an object of a class that forms its interface through which the outside members of the class can access its private members.

5. **Encapsulation**: - Encapsulation is the method of combining the data and functions inside a class. This hides the data from being accessed from outside a class directly, only through the functions inside the class is able to access the information.

6. **Abstraction: -** Data abstraction refers to, providing only essential information to the outside world and hiding their background details.

7. **Polymorphism: -** Polymorphism (many forms) is the ability of an object or reference to take many different forms at different instances and they are related by inheritance.

Ex. Administrator and employee method

```
void employee::getdata(void)
{
username: cout<<"Enter User Name: ";
            cin>>uname;
                if(!info.valid_alpha(uname))
                goto username;

password: cout<<"Enter password:  ";
                cin>>pword;
                if(strlen(pword)<6)
                {
                        cout<<"password is too short. "<<endl;
                        goto password;
                }
                fflush(stdin);
}
```

```
void Admin::enterRecordsHR()
{
        fstream fileHR;
   int numHR;
        cout<<"\n\n\t\tEnter how many Employee Details you want to enter :";
        cin>>numHR;

        fileHR.open("HRemployee",ios::out|ios::app);
        for(int i=1; i<= numHR; i++)
```

```
        {
                cout<<"\n\tEnter the information of Employee #"<<i<<endl;
                        cout<<"\n\t\tYour IC No# Employee"<<++info.countAgHR<<endl;
                        info.intempNo=info.countAgHR;
                info.getdata();
                fileHR.write((char*) &info, sizeof(info));
        }
        fileHR.close();
}
```

There are two types of polymorphism:-

a) Compile time polymorphism (1.overloading of function 2. Overloading of operators)

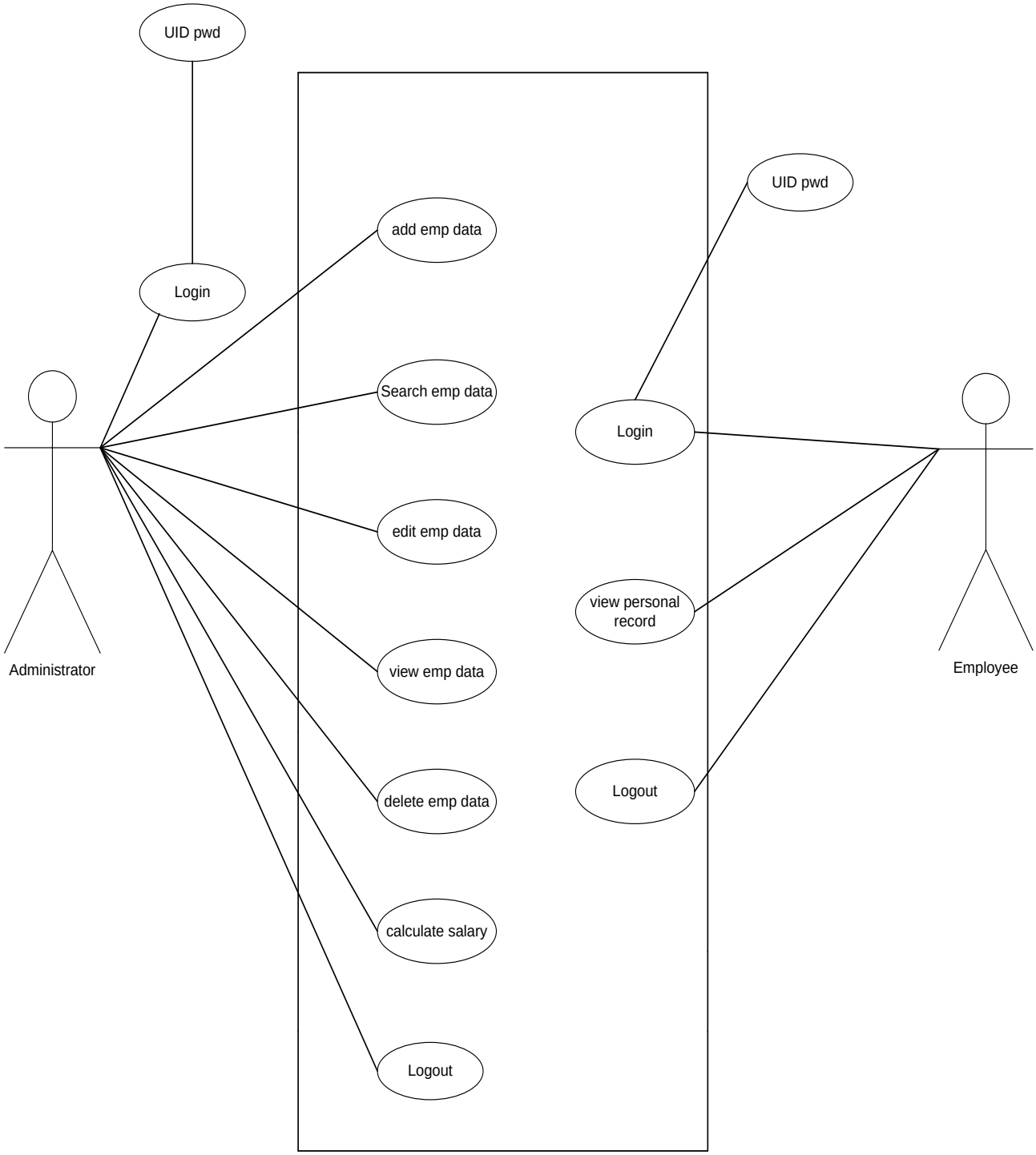
b) Run time polymorphism (virtual functions)

Validation codes applied into implementation code:-

1. cout<<"password is too short. "<<endl; (employee)
2. cout<<"Enter a valid gender "<<endl; (employee)
3. cout<<"Please Enter Valid Designation"; (employee)
4. cout<<"Please Enter Valid Department"; (employee)
5. cout<<"\n\nSorry !! Your Username & password are not matching."; (employee)
6. cout<<"Enter right choice"; (employee)
7. cout<<"\n\tYour choice is Invalid" <<endl;
8. cout<<"\n\n\tPlease enter right choice!"<<endl;
9. cout<<"\n\n\tSorry !! Your Username & password are not matching.";
10. cout<<"\n\tYour choice is Invalid" <<endl; (Admin)
11. cerr <<"File could not be opened";(Admin)
12. cout<<"\n\t\t\tPlease enter right choice!"<<endl; (Admin)
13. cout<<"Wrong ID or password"<<endl; (Admin)
14. cout<<endl<<"\t\t**** INVALID Entry! Please Re-enter ****"<<endl;(Admin)
15. cout<<endl<<"\t\t**** INVALID Date! ****"<<endl; (Admin)

Diagrams

Use case:-



Class diagram:-