Object Oriented Programming Final Project

By: Hansel Faren (2501990350)

L2BC / Computer Science

Project Specification

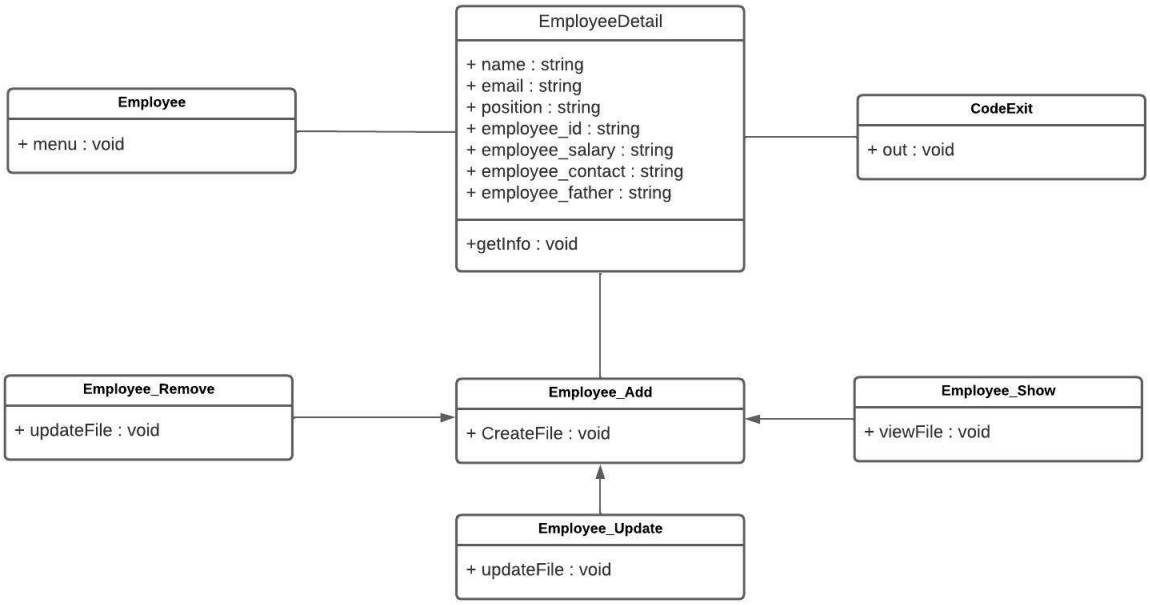
This project is a simple of program to store employee's data. Instead of using any data structures, I decided to use the write in file method where it's like other language, python. Why I used this method is because it's much simpler because the way they search the storage based on the ID, so they will search through in the file documents.

Features

There are only basic features in this program such as:

1. Add new employee
2. Preview existed employee
3. Delete existed employee
4. Update employee information

Solution Design



Classes

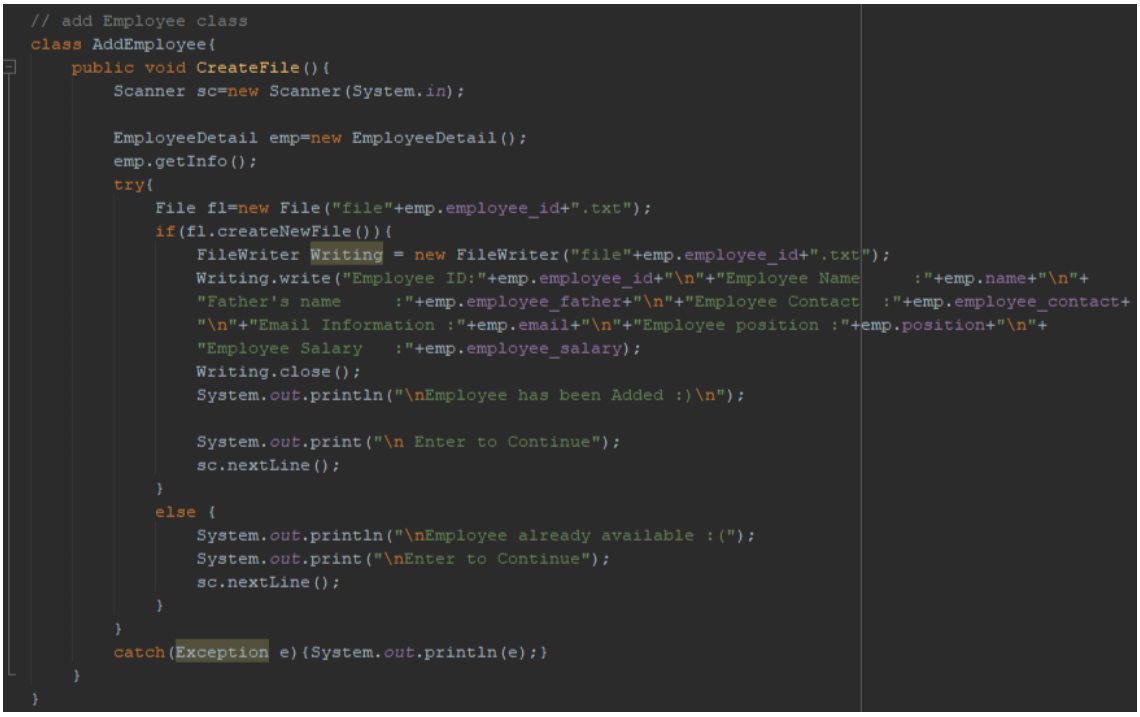
1. Menu. This class only shows the structure of how it will display on the screen.

```
// menu class
public class Employee{
    public void menu(){
        System.out.println("\t\t*****************************************");
        System.out.println("\t\t\t  EMPLOYEE MANAGEMENT SYSTEM");
        System.out.println("\t\t*****************************************");
        System.out.println("\n\nPress 1 : Add Employee");
        System.out.println("Press 2 : Employee Details ");
        System.out.println("Press 3 : Remove an Employee");
        System.out.println("Press 4 : Update Employee");
        System.out.println("Press 5 : Quit");
    }
}
```

2. EmployeeDetail. In this class, it declares the variable for the information that the employee needs to fill.

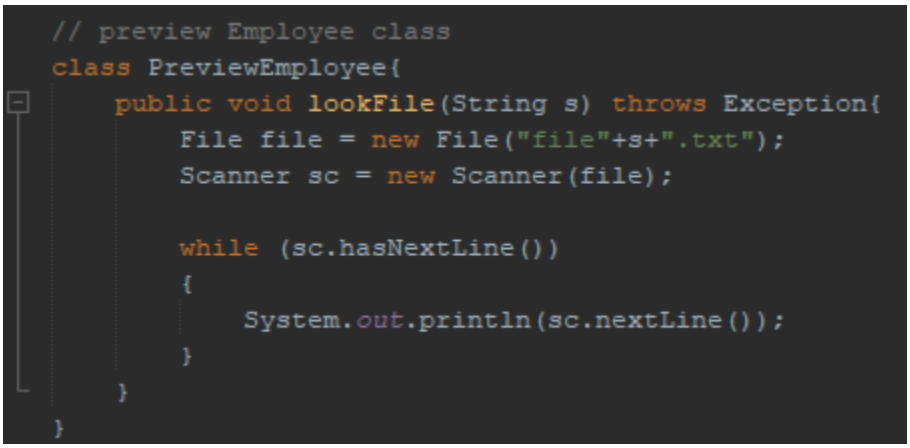
```
// Employee detail class
class EmployeeDetail{
    String name;
    String email;
    String position;
    String employee_id;
    String employee_salary;
    String employee_contact;
    String employee_father;
    public void getInfo(){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter Employee's name       : ");
        name = sc.nextLine();
        System.out.print("Enter Father's name        : ");
        employee_father = sc.nextLine();
        System.out.print("Enter Employee's ID        : ");
        employee_id = sc.nextLine();
        System.out.print("Enter Employee's Email ID  : ");
        email = sc.nextLine();
        System.out.print("Enter Employee's Position  : ");
        position = sc.nextLine();
        System.out.print("Enter Employee contact Info : ");
        employee_contact = sc.nextLine();
        System.out.print("Enter Employee's Salary    : ");
        employee_salary = sc.nextLine();
    }
}
```

3. AddEmployee. This class responsible for storing the employee's data in the from of text file.



```
// add Employee class
class AddEmployee{
    public void CreateFile(){
        Scanner sc=new Scanner(System.in);

        EmployeeDetail emp=new EmployeeDetail();
        emp.getInfo();
        try{
            File fl=new File("file"+emp.employee_id+".txt");
            if(fl.createNewFile()){
                FileWriter Writing = new FileWriter("file"+emp.employee_id+".txt");
                Writing.write("Employee ID:"+emp.employee_id+"\n"+"Employee Name      :"+emp.name+"\n"+
                "Father's name      :"+emp.employee_father+"\n"+"Employee Contact   :"+emp.employee_contact+
                "\n"+"Email Information :"+emp.email+"\n"+"Employee position :"+emp.position+"\n"+
                "Employee Salary    :"+emp.employee_salary);
                Writing.close();
                System.out.println("\nEmployee has been Added :)\n");

                System.out.print("\n Enter to Continue");
                sc.nextLine();
            }
            else {
                System.out.println("\nEmployee already available :(");
                System.out.print("\nEnter to Continue");
                sc.nextLine();
            }
        }
        catch(Exception e){System.out.println(e);}
    }
}
```

4. PreviewEmployee. This class to show the data of the selected employee that is existed.



```
// preview Employee class
class PreviewEmployee{
    public void lookFile(String s) throws Exception{
        File file = new File("file"+s+".txt");
        Scanner sc = new Scanner(file);

        while (sc.hasNextLine())
        {
            System.out.println(sc.nextLine());
        }
    }
}
```

5. RemoveEmployee. For removing the data that is stored, this class works for it where if you want to delete the existed data.

```
// delete Employee class
class RemoveEmployee{
    public void deleteFile(String ID){

        File file = new File("file"+ID+".txt");
        if(file.exists())
        {
            if(file.delete());
            {
                System.out.println("\nEmployee has been removed Successfully");
            }
        }else{
            System.out.println("\nEmployee does not exists :( ");
        }
    }
}
```

6. Update Employee. In order to update the information about the data that is exist, this function will get the job done.

```
// update Employee class
class UpdateEmployee{
    public void updateFile(String s,String o,String n) throws IOException
    {
        File file = new File("file"+s+".txt");
        Scanner sc = new Scanner(file);
        String FileContext="";
        while (sc.hasNextLine())
            {
                FileContext =FileContext+"\n"+sc.nextLine();
            }
        FileWriter Writing = new FileWriter("file"+s+".txt");
        FileContext = FileContext.replaceAll(o,n);
        Writing.write(FileContext);
        Writing.close();
    }
}
```

7. CodeExit. This class is to exterminate the program that is running endlessly in the program.

```
// Exit program class
class CodeExit{
    public void quit(){
    System.out.println("\n*****************************************");
    System.out.println("Thanks for Using us");
    System.out.println("*****************************************");
    System.exit(0);
    }
}
```

Driver

As in the driver class, this file will be the main function from the class function.

```
public class Driver extends Employee{
    public static void main(String args[]){
        System.out.print("\033[H\033[2J"); //clear the screen

        Scanner sc=new Scanner(System.in);
        PreviewEmployee preview =new PreviewEmployee(); // calling the preview class

        int i=0;

        Employee obj1 = new Employee(); // calling the class function
        obj1.menu();
```

I start off with importing the employee class which is I use the “extends” command. Next is write the scan command so the user can input in to the program and calling the preview and also menu class.

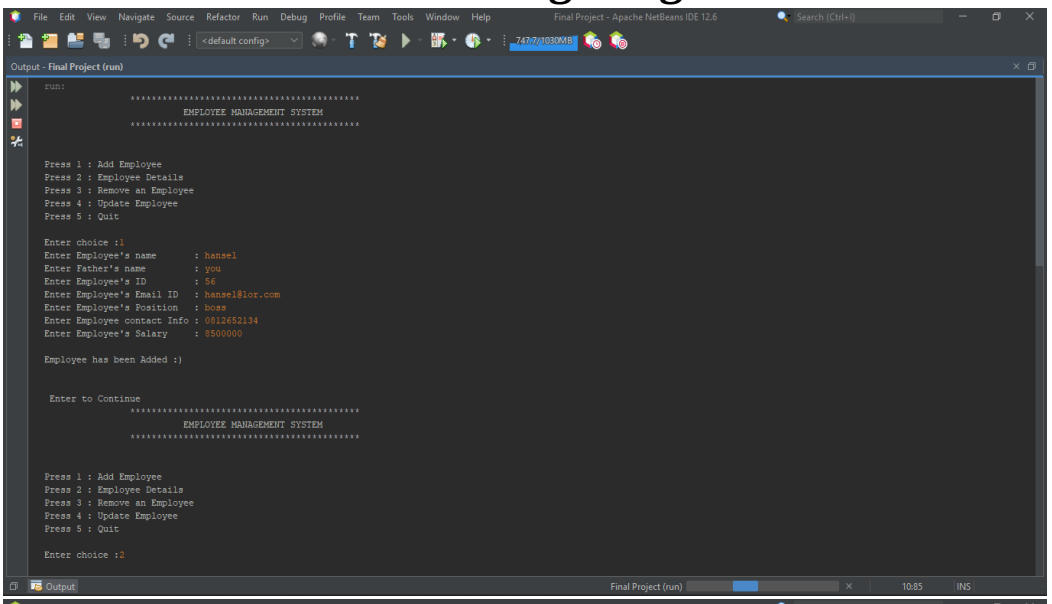
```
    while(i<6) // starting the loop
    {

        System.out.print("\nEnter choice :");
        i=Integer.parseInt(sc.nextLine());
```

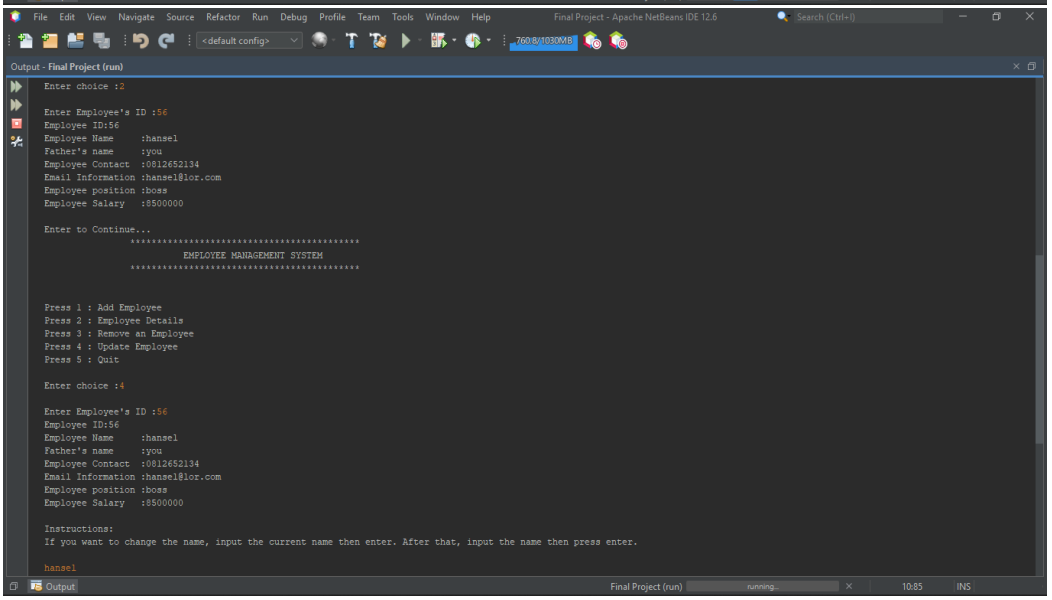
By using the while loop, the program will run endlessly so that the user can always use the program until input the terminate function.

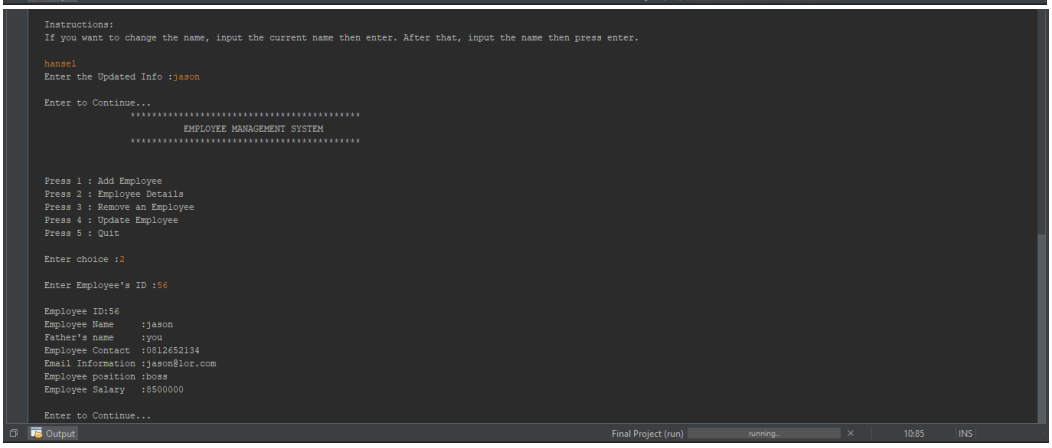
```
switch(i){ // to switch statements
    case 1:
    {
        AddEmployee ep =new AddEmployee(); // calling the create class function and run the function
        ep.CreateFile();

        System.out.print("\033[H\033[2J");
        objl.menu();
        break;
    }
    case 2:
    {
        System.out.print("\nEnter Employee's ID :");
        String s=sc.nextLine();
        try
        {
            preview.lookFile(s);}
            catch(Exception e){System.out.println(e);}

            System.out.print("\nEnter to Continue...");
            sc.nextLine();
            System.out.print("\033[H\033[2J");
            objl.menu();
            break;
        }
```

Using the switch function is to make the program can run different function based on the user input.

Working Program







References:

https://www.programiz.com/java-programming/examples/create-and-write-to-file#:~:text=java%20File%20file%20%3D%20new%20File,file%20to%20the%20specified%20path.
https://www.w3schools.com/java/java_switch.asp
https://www.w3schools.com/java/java_try_catch.asp