EMPLOYEE RECORD SYSTEM

Submitted By

ABHISHEK SHARMA
University Roll No. 12019009001127
Department of Computer Science
UEM, Kolkata

ADITYA JAISWAL
University Roll No. 12019009023073
Department of Computer Science
UEM, Kolkata

AKASH SARKAR
University Roll No. 12019009023099
Department of Computer Science
UEM, Kolkata

Paper Code: PCCCS481

Paper Name: Object Oriented Programming Using Java

Department of Computer Science

UEM, Kolkata

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA
University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700 156

Certificate

This is to certify that the project titled **"Employee Record System using JAVA"** submitted by,
ABHISHEK SHARMA [12019009001127]
ADITYA JAISWAL [12019009023073]
and **AKASH SARKAR [12019009023099]**
students of University of Engineering & Management, Kolkata, i partial fulfilment of requirement of the degree of Bachelor of Computer Science, is a bonafide work carried out by them under the supervision and guidance of **Prof. Sudeshna Kundu Mondal** during the **4th Semester of academic session of 2020-21**. The content of this report has not been submitted to any other institute or university. I am glad to inform you that, the work is entirely original and it's performance is found to be quite satisfactory.

Prof. Sudeshna Kundu Mondal
Department of Computer Science
University of Engineering & Management, Kolkata

Acknowledgement

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our guide **Prof. SUDESHNA KUNDU MONDAL** of the Department of Computer Science, UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to Prof. Sukalyan Goswami, HOD, Computer Science, UEM, Kolkata and all other departmental faculties for their ever-present assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

ABHISHEK SHARMA

ADITYA JAISWAL

AKASH SARKAR

Table of Contents

EMPLOYEE RECORD SYSTEM

AN OBJECT ORIENTED PROGRAMMING PROJECT USING JAVA

Abstract

Employee Record System using JAVA is one the smartest and innovative projects in terms of automating the record system of various organizations and institutions. In this Employee Record System we have implemented a code base which will store the data of the employees of an organization or any other institution. Storing information and records of the employees have been a tough work for manual users, hence to make it simple this JAVA empowered system is imposed to make the tough job simple.

To implement this project we have used Visual Studio code and JDK 15, to compile and run the program. Visual Studio code is a versatile IDE which has lot more facilities than any other IDEs and also the latest version of Java Development Kit 15, which help the system to empower. Hence, this project work is one of the best version of the Employee record taking system.

Introduction

Earlier systems were manual where there was no way of properly storing information. Employee records were stored manually which lead to errors. There was no proper way of tracking employee records. It was very difficult and required a lot of paperwork which makes the application time consuming and not secured. There was no administrator which could handle the records. So there was the need to develop a system which could manage all these things and reduce the paperwork.

Employee information system is easy to use application which is created to manage the employee data. It is created to record the details of the employees. This reduces the dependency on the manual system which could create errors. This system can easily help in tracking employee records. There is a search feature which allows getting records of a specific employee. There is an administrator which can add, edit, delete and save records in a database. There are two views for this application first is the administrator and the second is employee view. The employee view enables employees to view their details.

Literature Review

In the past few times the employee record system is developed by manual deployment and there is no system generated approach through which this can be automated, but through this approach we can solve this matter and automate the whole system using the code base that we have developed and deployed successfully.

Before implementing this kind of project and code base the record system is worked in the excel sheet or in the form of register sheet, but in that case anyone can view and edit that as this is very open to all of the employees. Hence chance of malfunction is very much logical in this case.

So, to prevent this kind of steps, we have to develop something which will be 100% accurate and 100% efficient. Hence we have developed this project.

Problem Statement

The problem statement for this project is, to make a secure and automated system which will keep the Employee Records of an organization or institution secure and safe.

Also it can do the following jobs automatically with the help of the Admin of the company only,

- Adding new employee data
- Updating previous employee data
- Deleting employee data
- Viewing all the records
- Search a particular employee record based on the employee ID.

These are the problem statements that we are solving using the code base which is implemented in JAVA.

Solution

To overcome the above mentioned problem statements, we need to create 6 procedures. They are,

- Creating a class in the Java program which will add new employee records, as per the admin choice.
- Creating a class in the Java program which will update the previously entered record of the employee with the help of the admin through the system by the employee ID.
- Creating a class in the Java program which will delete the desired record of the database with the permission of the admin.
- Creating a class in the Java program which will search a record with respect to the employee ID of the employee.
- Creating a class which will display the whole data base whenever the admin is wanting to view.
- Finally, to switch off the program we need to end the program and hence we have to implement the exit class or exit operation.

These classes will be combined together to form the program which will help the admin to work on with the database of the employee records.

Methods and Algorithm

To access this project the basic medium will be Visual Studio Code and JDK 15 (any other IDEs will also work on this project). From the JDK we have imported some basic libraries which are useful for this project work to be easier and successful.

Those libraries are,

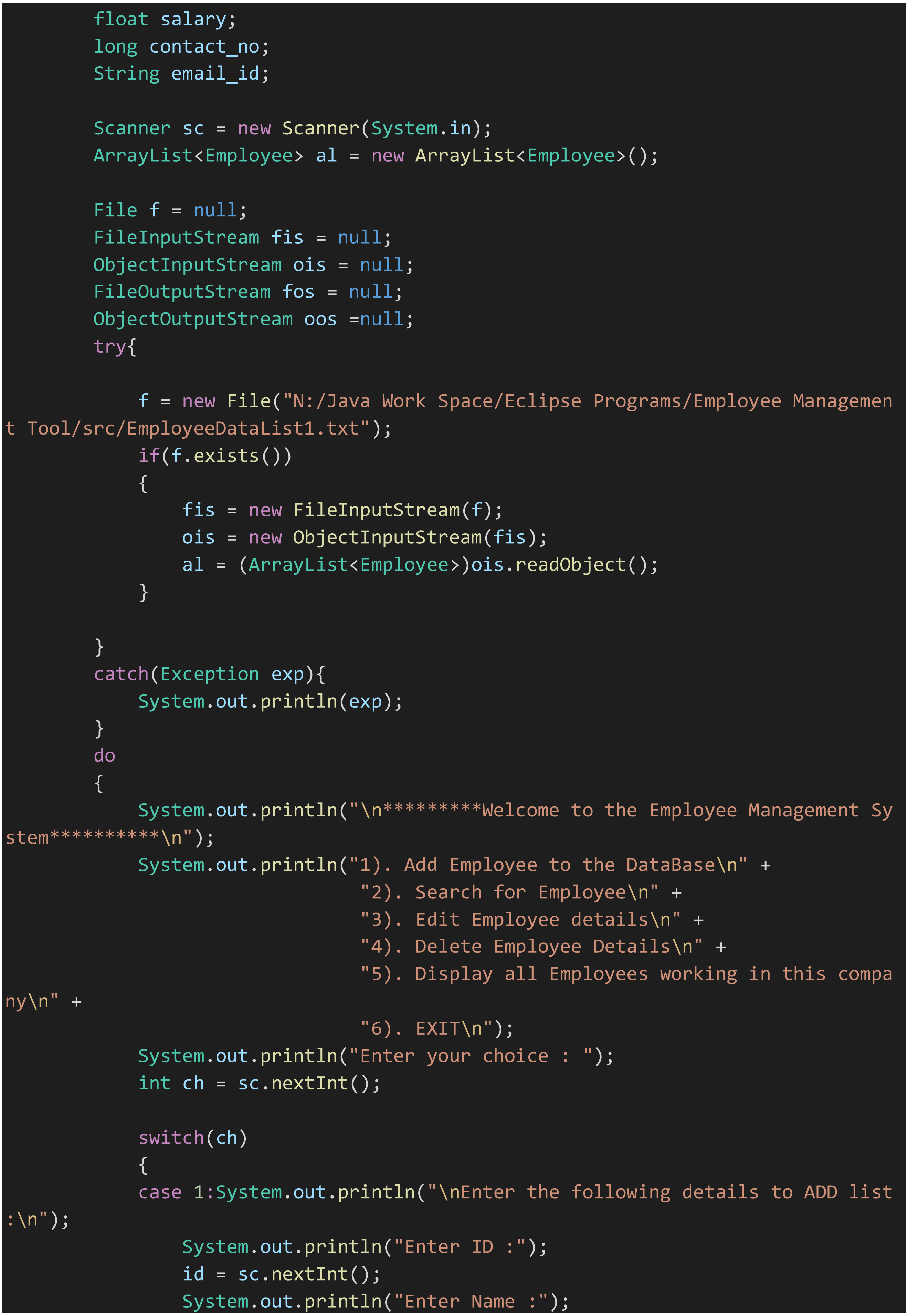
```java
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Scanner;
```

These libraries will ensure the working principle of the project as well as the code base!

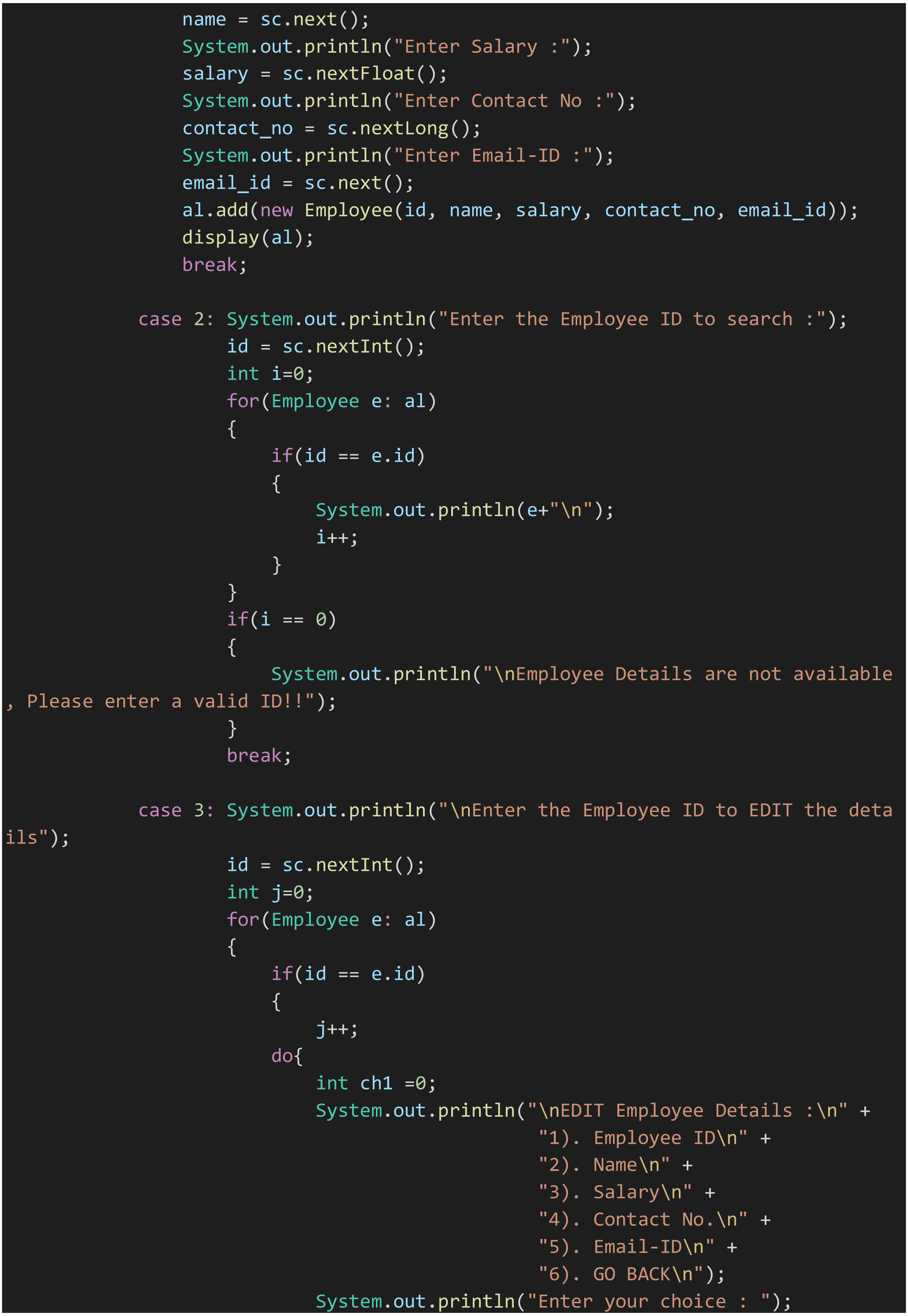
Now let's get to the codebase,

```java
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Scanner;
// Code contributed by, Abhishek Sharma, 2021
@SuppressWarnings("serial")
```

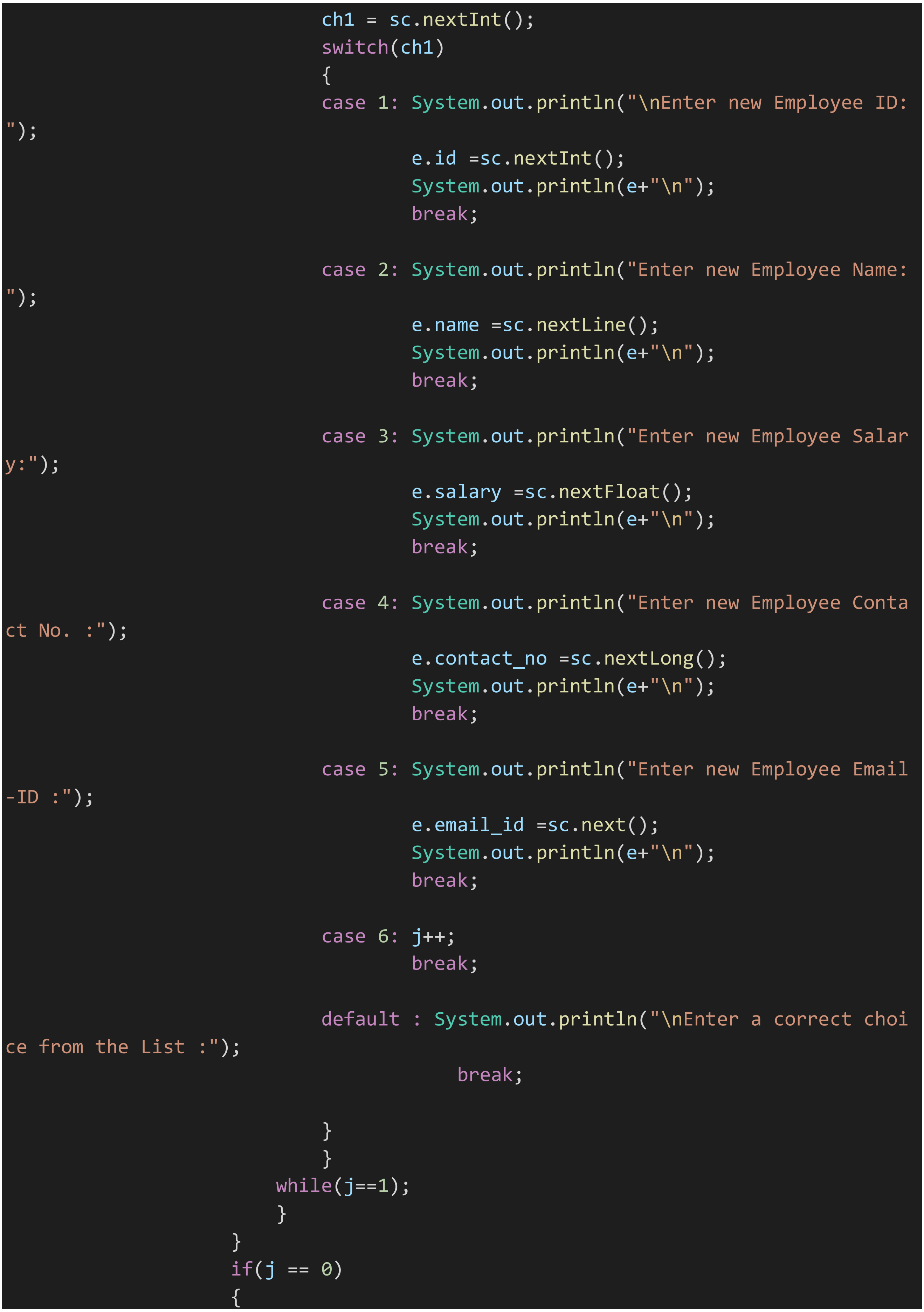
```java
class Employee implements Serializable{

    int id;
    String name;
    float salary;
    long contact_no;
    String email_id;

    public Employee(int id, String name, float salary, long contact_no, String email_id)
    {
        this.id = id;
        this.name = name;
        this.salary = salary;
        this.contact_no = contact_no;
        this.email_id = email_id;
    }

    public String toString()
    {
        return "\nEmployee Details :" + "\nID: " + this.id + "\nName: " + this.name + "\nSalary: " +
                this.salary + "\nContact No: " + this.contact_no + "\nEmail-id: " + this.email_id;
    }

}

public class EmployeeManagement
{
    static void display(ArrayList<Employee> al)
    {
        System.out.println("\n------------------------ Employee List ------------------------\n");
        System.out.println(String.format("%-10s%-15s%-10s%-20s%-10s", "ID","Name","salary","contact-no","Email-Id"));
        for(Employee e : al)
        {
            System.out.println(String.format("%-5s%-20s%-10s%-15s%-10s",e.id,e.name,e.salary,e.contact_no,e.email_id));
        }
    }

    @SuppressWarnings("unchecked")
    public static void main(String[] args)
    {
        int id;
        String name;
```



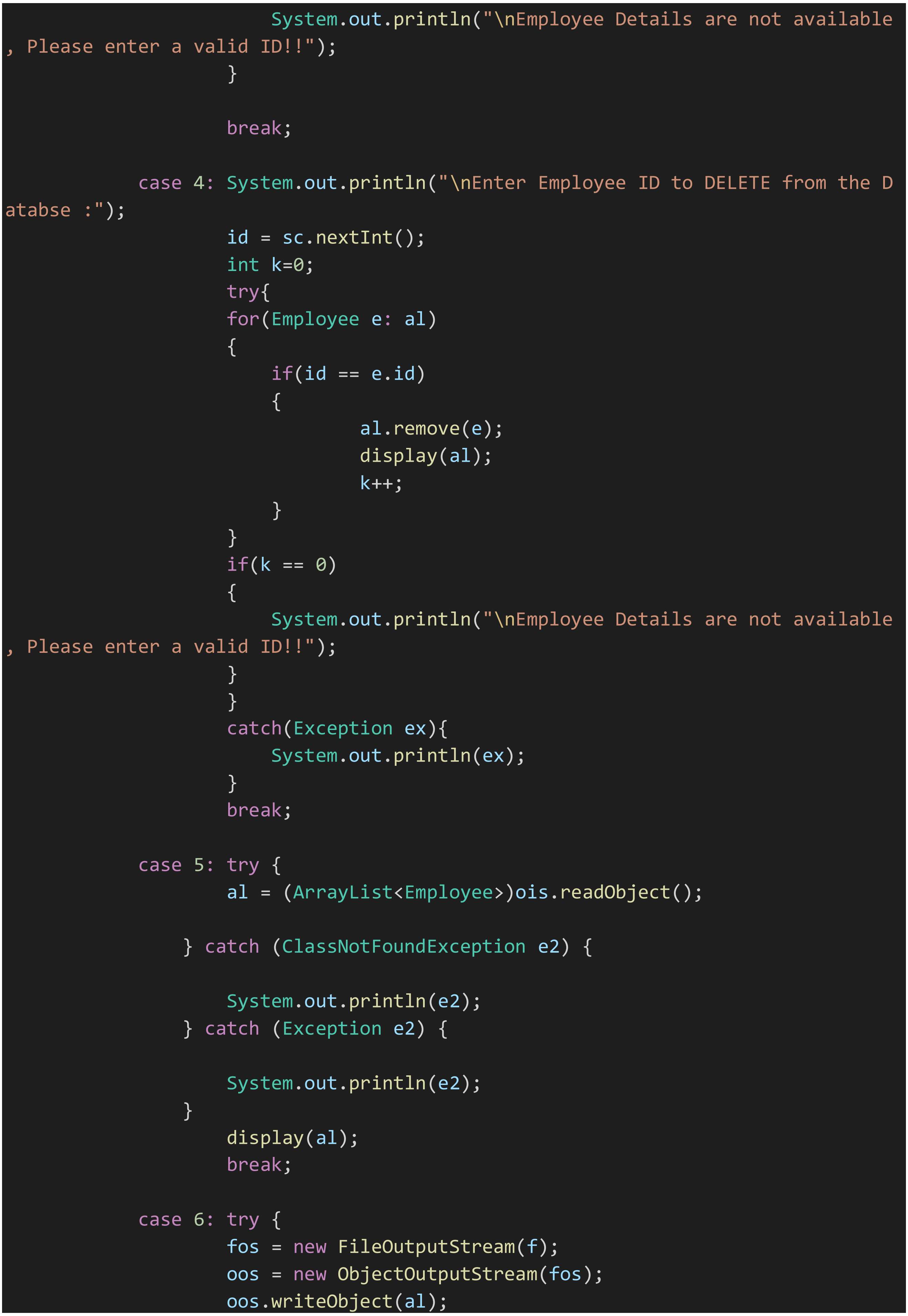
```
        float salary;
        long contact_no;
        String email_id;

        Scanner sc = new Scanner(System.in);
        ArrayList<Employee> al = new ArrayList<Employee>();

        File f = null;
        FileInputStream fis = null;
        ObjectInputStream ois = null;
        FileOutputStream fos = null;
        ObjectOutputStream oos =null;
        try{

            f = new File("N:/Java Work Space/Eclipse Programs/Employee Management Tool/src/EmployeeDataList1.txt");
            if(f.exists())
            {
                fis = new FileInputStream(f);
                ois = new ObjectInputStream(fis);
                al = (ArrayList<Employee>)ois.readObject();
            }

        }
        catch(Exception exp){
            System.out.println(exp);
        }
        do
        {
            System.out.println("\n*********Welcome to the Employee Management System**********\n");
            System.out.println("1). Add Employee to the DataBase\n" +
                                "2). Search for Employee\n" +
                                "3). Edit Employee details\n" +
                                "4). Delete Employee Details\n" +
                                "5). Display all Employees working in this company\n" +

                                "6). EXIT\n");
            System.out.println("Enter your choice : ");
            int ch = sc.nextInt();

            switch(ch)
            {
            case 1:System.out.println("\nEnter the following details to ADD list:\n");
                System.out.println("Enter ID :");
                id = sc.nextInt();
                System.out.println("Enter Name :");
```



```
                name = sc.next();
                System.out.println("Enter Salary :");
                salary = sc.nextFloat();
                System.out.println("Enter Contact No :");
                contact_no = sc.nextLong();
                System.out.println("Enter Email-ID :");
                email_id = sc.next();
                al.add(new Employee(id, name, salary, contact_no, email_id));
                display(al);
                break;

            case 2: System.out.println("Enter the Employee ID to search :");
                    id = sc.nextInt();
                    int i=0;
                    for(Employee e: al)
                    {
                        if(id == e.id)
                        {
                            System.out.println(e+"\n");
                            i++;
                        }
                    }
                    if(i == 0)
                    {
                        System.out.println("\nEmployee Details are not available
, Please enter a valid ID!!");
                    }
                    break;

            case 3: System.out.println("\nEnter the Employee ID to EDIT the deta
ils");
                    id = sc.nextInt();
                    int j=0;
                    for(Employee e: al)
                    {
                        if(id == e.id)
                        {
                            j++;
                        do{
                            int ch1 =0;
                            System.out.println("\nEDIT Employee Details :\n" +
                                                "1). Employee ID\n" +
                                                "2). Name\n" +
                                                "3). Salary\n" +
                                                "4). Contact No.\n" +
                                                "5). Email-ID\n" +
                                                "6). GO BACK\n");
                            System.out.println("Enter your choice : ");
```



```
                            ch1 = sc.nextInt();
                            switch(ch1)
                            {
                            case 1: System.out.println("\nEnter new Employee ID:
");

                                    e.id =sc.nextInt();
                                    System.out.println(e+"\n");
                                    break;

                            case 2: System.out.println("Enter new Employee Name:
");

                                    e.name =sc.nextLine();
                                    System.out.println(e+"\n");
                                    break;

                            case 3: System.out.println("Enter new Employee Salar
y:");

                                    e.salary =sc.nextFloat();
                                    System.out.println(e+"\n");
                                    break;

                            case 4: System.out.println("Enter new Employee Conta
ct No. :");

                                    e.contact_no =sc.nextLong();
                                    System.out.println(e+"\n");
                                    break;

                            case 5: System.out.println("Enter new Employee Email
-ID :");

                                    e.email_id =sc.next();
                                    System.out.println(e+"\n");
                                    break;

                            case 6: j++;
                                    break;

                            default : System.out.println("\nEnter a correct choi
ce from the List :");

                                        break;

                            }
                            }
                        while(j==1);
                        }
                    }
                    if(j == 0)
                    {
```



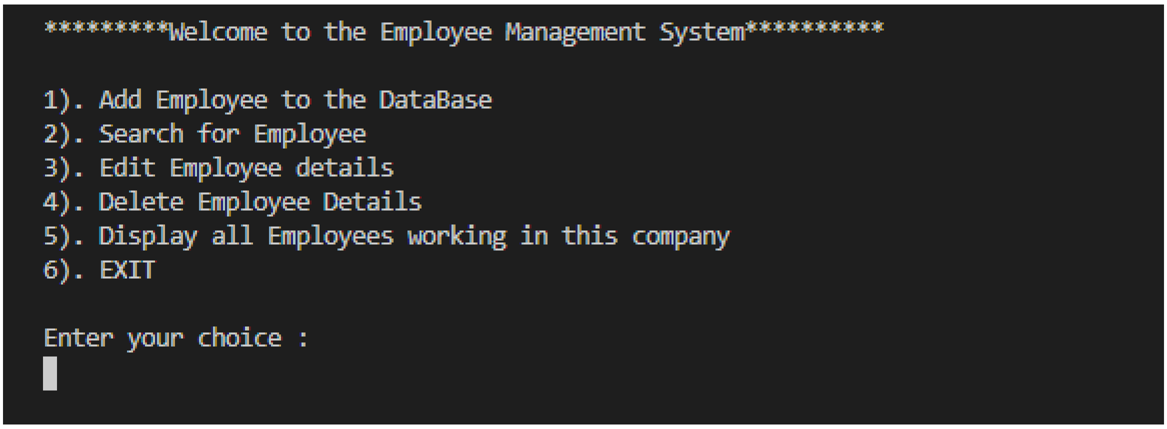
```
                        System.out.println("\nEmployee Details are not available
, Please enter a valid ID!!");
                    }

                    break;

            case 4: System.out.println("\nEnter Employee ID to DELETE from the D
atabse :");
                    id = sc.nextInt();
                    int k=0;
                    try{
                    for(Employee e: al)
                    {
                        if(id == e.id)
                        {
                                al.remove(e);
                                display(al);
                                k++;
                        }
                    }
                    if(k == 0)
                    {
                        System.out.println("\nEmployee Details are not available
, Please enter a valid ID!!");
                    }
                    }
                    catch(Exception ex){
                        System.out.println(ex);
                    }
                    break;

            case 5: try {
                    al = (ArrayList<Employee>)ois.readObject();

                } catch (ClassNotFoundException e2) {

                    System.out.println(e2);
                } catch (Exception e2) {

                    System.out.println(e2);
                }
                    display(al);
                    break;

            case 6: try {
                    fos = new FileOutputStream(f);
                    oos = new ObjectOutputStream(fos);
                    oos.writeObject(al);
                } catch (IOException e1) {
                    e1.printStackTrace();
                }
                catch(Exception e2){
                    e2.printStackTrace();
                }
                finally{
                    try {
                        fis.close();
                        ois.close();
                        fos.close();
                        oos.close();
                    } catch (Exception e1) {
                        e1.printStackTrace();
                    }

                }
                    System.out.println("\nYou have chosen EXIT !! Saving Files a
nd closing the tool.");
                    sc.close();
                    System.exit(0);
                    break;

            default : System.out.println("\nEnter a correct choice from the List
 :");
                        break;

            }
        }
        while(true);
    }

}
```

Here's the complete codebase of this project. And this is the final outcome of the codebase.

Result and Analysis

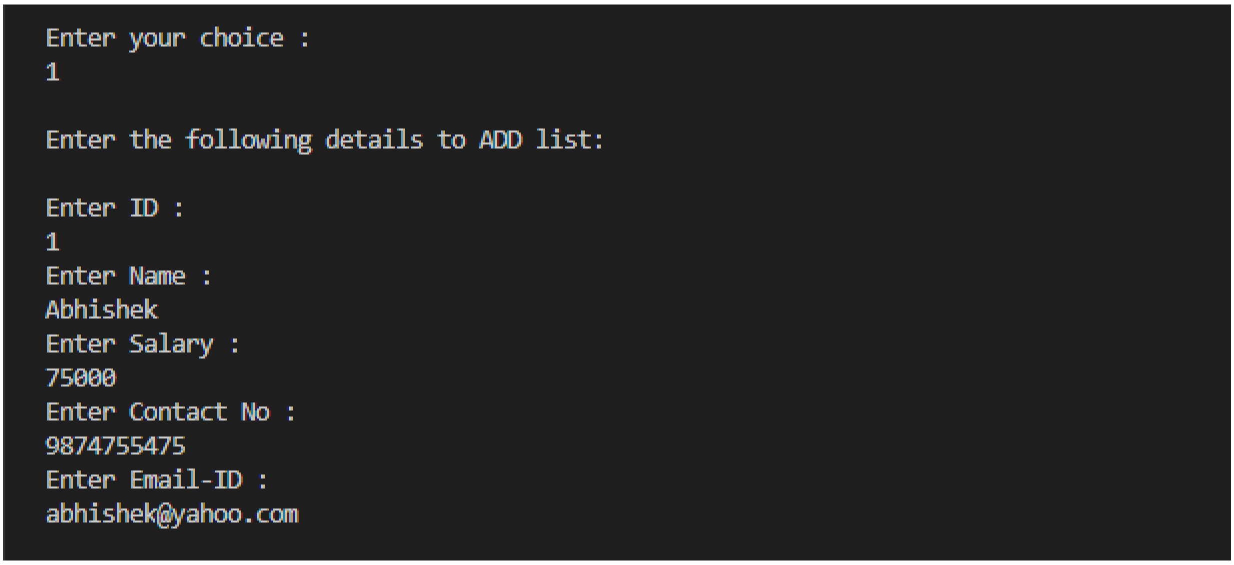
Due to this project work and code base we have deployed the Employee Record System, now let's have look how this will work when it is running.

Firstly it will ask the admin what to do, and the admin will enter the desired number as per the work to be done.



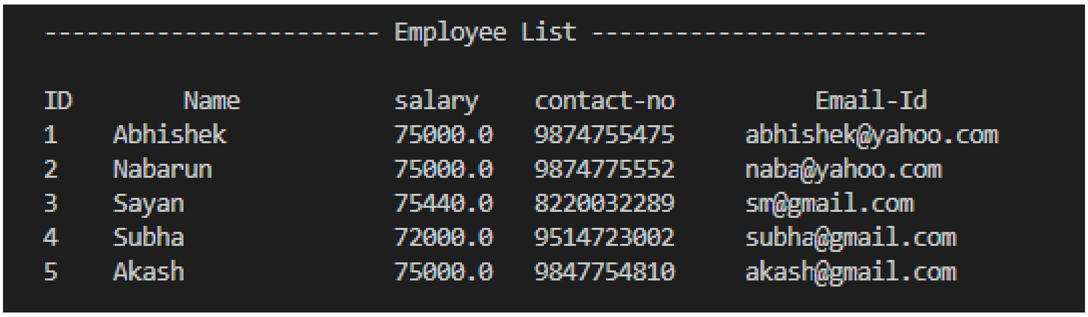
```
*********Welcome to the Employee Management System**********

1). Add Employee to the DataBase
2). Search for Employee
3). Edit Employee details
4). Delete Employee Details
5). Display all Employees working in this company
6). EXIT

Enter your choice :

```

Now adding the records in the database,



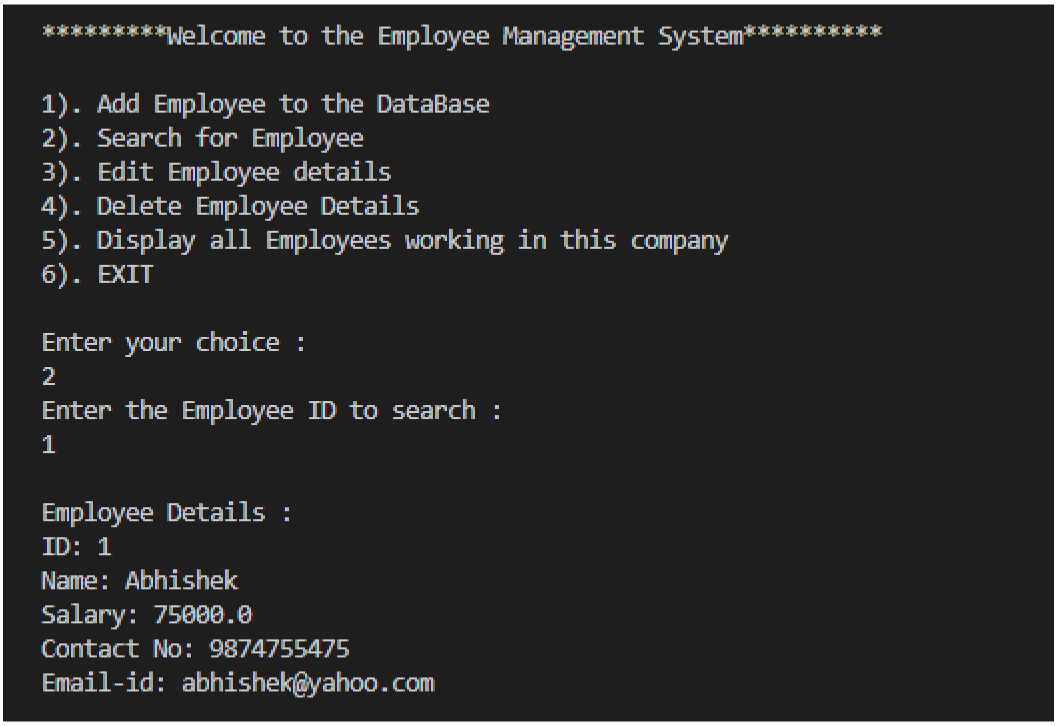
```
Enter your choice :
1

Enter the following details to ADD list:

Enter ID :
1
Enter Name :
Abhishek
Enter Salary :
75000
Enter Contact No :
9874755475
Enter Email-ID :
abhishek@yahoo.com
```

Now after adding the desired data in the record system, let's view all the data in the record system.



```
------------------------ Employee List ------------------------

ID        Name          salary    contact-no        Email-Id
1    Abhishek           75000.0   9874755475     abhishek@yahoo.com
2    Nabarun            75000.0   9874775552     naba@yahoo.com
3    Sayan              75440.0   8220032289     sm@gmail.com
4    Subha              72000.0   9514723002     subha@gmail.com
5    Akash              75000.0   9847754810     akash@gmail.com
```

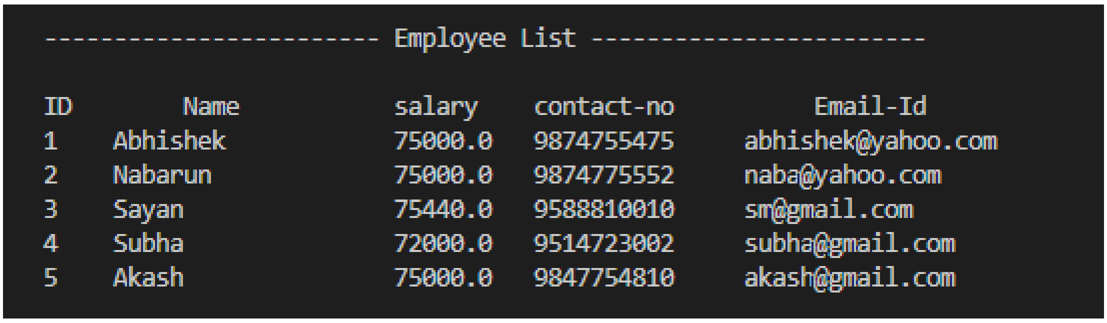
Now search the employee record depending upon the ID of the employees.



```
*********Welcome to the Employee Management System**********

1). Add Employee to the DataBase
2). Search for Employee
3). Edit Employee details
4). Delete Employee Details
5). Display all Employees working in this company
6). EXIT

Enter your choice :
2
Enter the Employee ID to search :
1

Employee Details :
ID: 1
Name: Abhishek
Salary: 75000.0
Contact No: 9874755475
Email-id: abhishek@yahoo.com
```

Here's the employee details of the employee having the ID as 1.

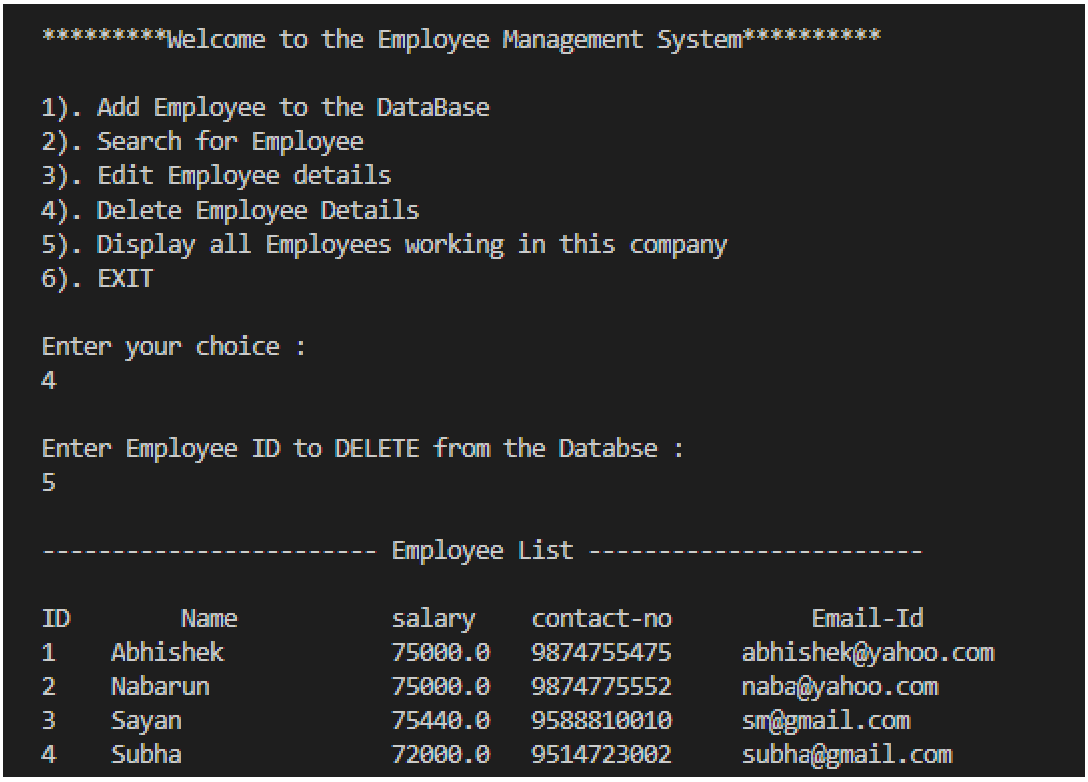
Now edit some employee details. Let's change the Contact no. of the ID 3 and then view that records.



```
------------------------ Employee List ------------------------

ID        Name          salary    contact-no        Email-Id
1    Abhishek           75000.0   9874755475     abhishek@yahoo.com
2    Nabarun            75000.0   9874775552     naba@yahoo.com
3    Sayan              75440.0   9588810010     sm@gmail.com
4    Subha              72000.0   9514723002     subha@gmail.com
5    Akash              75000.0   9847754810     akash@gmail.com
```

This is the updated contact no. of employee ID 3.

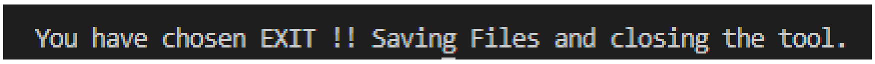
Now delete the record of the employee ID 5.



```
*********Welcome to the Employee Management System**********

1). Add Employee to the DataBase
2). Search for Employee
3). Edit Employee details
4). Delete Employee Details
5). Display all Employees working in this company
6). EXIT

Enter your choice :
4

Enter Employee ID to DELETE from the Databse :
5

------------------------ Employee List ------------------------

ID        Name            salary    contact-no          Email-Id
1    Abhishek           75000.0   9874755475     abhishek@yahoo.com
2    Nabarun            75000.0   9874775552     naba@yahoo.com
3    Sayan              75440.0   9588810010     sm@gmail.com
4    Subha              72000.0   9514723002     subha@gmail.com
```

After deleting the record 5, the record database looks like this.

Now to EXIT from the system just enter the choice as 6 and the database work will be closed.



```
You have chosen EXIT !! Saving Files and closing the tool.
```

This is how the employee record system will work and data are kept by the admin only and also it is secure.

Conclusion and Discussion

By this project work the data keeping procedure for the admin of the company will be more accurate and more efficient. All the procedures have been working correctly. The code base is accurately,

- Adding the data in the database
- Updating the desired column
- Deleting the desired data
- Searching the desired the employee's record
- Perfectly showing the whole database

Hence, we can conclude that the project has been done correctly and implemented properly to do the aforementioned procedures.

Future Scope of Work

This code base is the back bone of this project. Now this project can be modified using the Front end development and My-SQL.

The database can be managed using the Database Management Systems to ensure the data privacy and the data will be compact enough for further use.

The front end of the system can be designed using HTML, PHP, CSS, JavaScript for better look and easy to handle situations.

By this, the project will look more attractive and as well as the data will be secured enough for the future use.

Resources and References

We would like to thank all the sources which help us in the project,

- Visual Studio Code : https://code.visualstudio.com/
- Java Development Kit : https://www.oracle.com/in/java/technologies/javase-downloads.html
- Java Documentation : https://docs.oracle.com/en/java/
- Complete Code Base stored in my GitHub : https://github.com/abhisheks008/Employee-Record-System

----