1. K&D Music Store is a musical instruments shop located in Negombo, Sri Lanka and currently they have 15 employees working under different job roles. You are hired as a freelance software engineer to develop their employee management system and they have provided you below mention system requirements to implement.
    - The application has to be a console-based application.
    - The system has two users, admin and employee.
    - Admin should be able to login to the system. (10 marks)
    - Admin should be able to enter new employees with their salaries to the system, and employee ids should be auto generated from the system. (ie:- you can define relevant employee attributes in the system) (10 marks)
    - Admin should be able to update employee details in the system. (10 marks)
    - Admin should be able to delete employee details in the system. (10 marks)
    - Admin should be able to view all and search employee details in the system. (10 marks)
    - Admin should be able to sort (ascending order) the employee details according to the salaries of the employees. (10 marks)
    - Employee should be able to login to the system with employee id and password. (10 marks)
    - Employee should be able to view their details with salary from the system. (10 marks)

When creating the system, it's not required to store all the details permanently in the system and use collections to store details temporary in the system. As the developer, you need to provide,

- Complete console-based java application with all the functionalities. (Provide necessary methods for each functionality). (Total 80 marks)
- Provide a documentation with system requirement specification, designing (use case diagram, class diagram and sequence diagrams) and important codings. (Total 20 marks)

System Requirements Specification (SRS) of K&D Music Store EMS

Purpose of the project

This is an employee management system that I intend to create for K&D Music Store, a music instrument store in Negombo, Sri Lanka. They decided to create a system to manage all of their employees' processes, such as adding, deleting, updating, searching for, and viewing employee details and salary details. Employees can also use this system to view their personal information. The primary goal of implementing this system is to improve the efficiency of the employee management process.

Functional and non-functional requirements

Functional Requirements **(Admin)**

- Admin should be able to log in to the system
- Admin should be able to add employees to the system
- Admin should be able to delete employees from the system
- Admin should be able to view and search employee details from system
- Admin should be able to update employee details to the system

Functional Requirements **(Employee)**

- Employee should be able to log in to the system
- Employee should be able to view employee details from the system

Non-Functional Requirements

- Security
- Usability
- Compatibility
- Maintainability

UML Diagrams

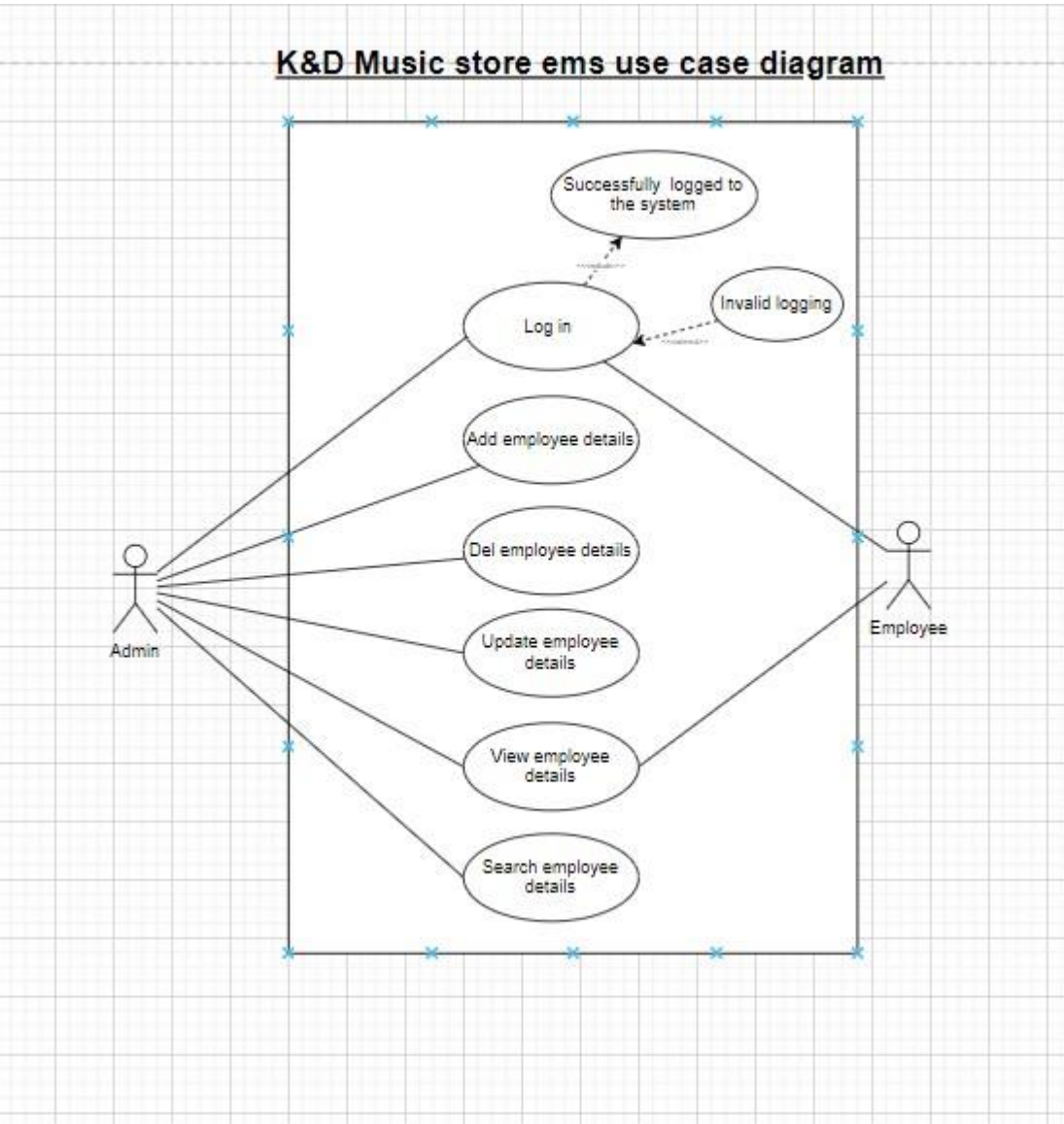


Figure 1 Use case diagram of K&D music store EMS system

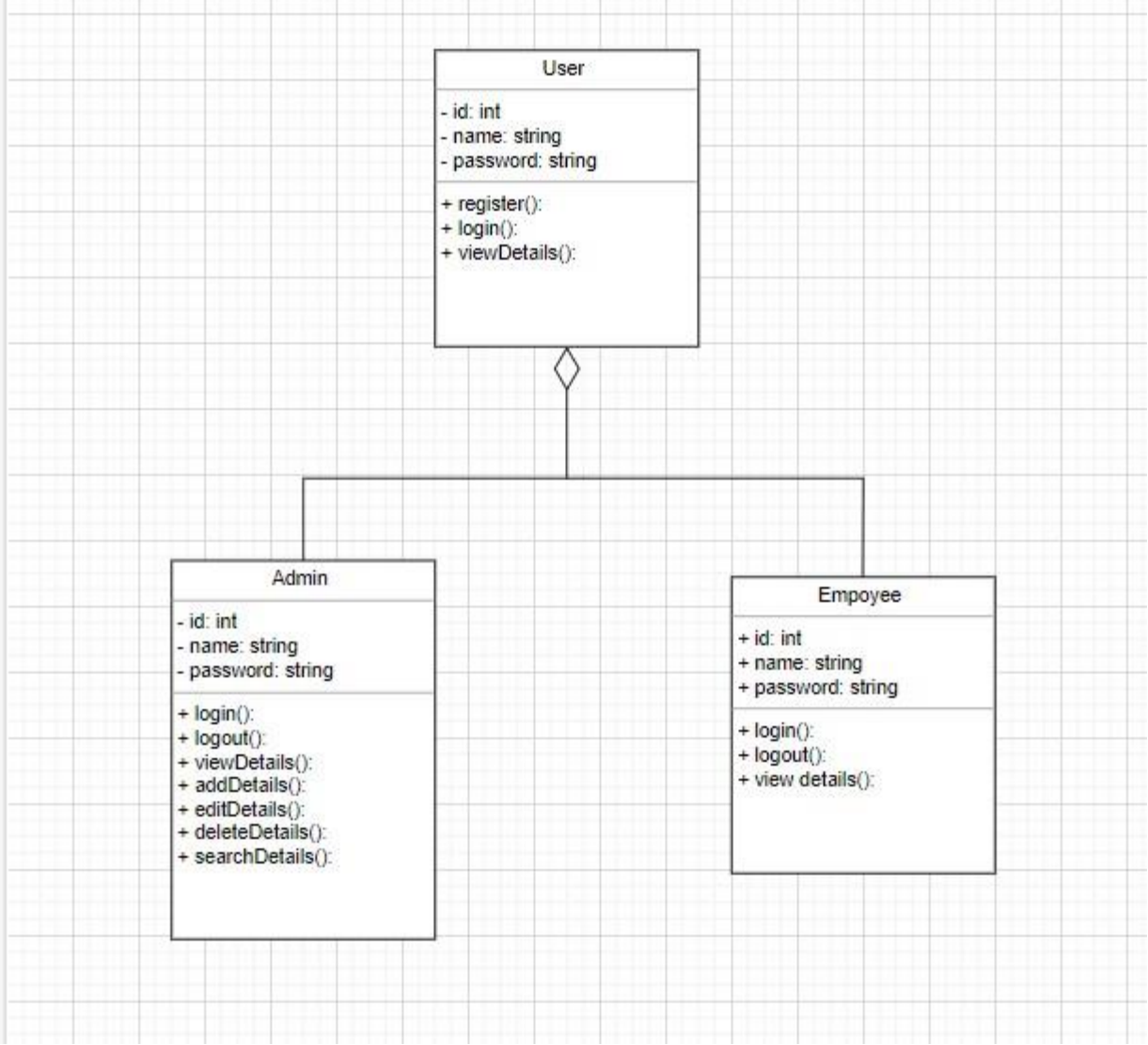


Figure 2 Class diagram of K&D Music Store EMS

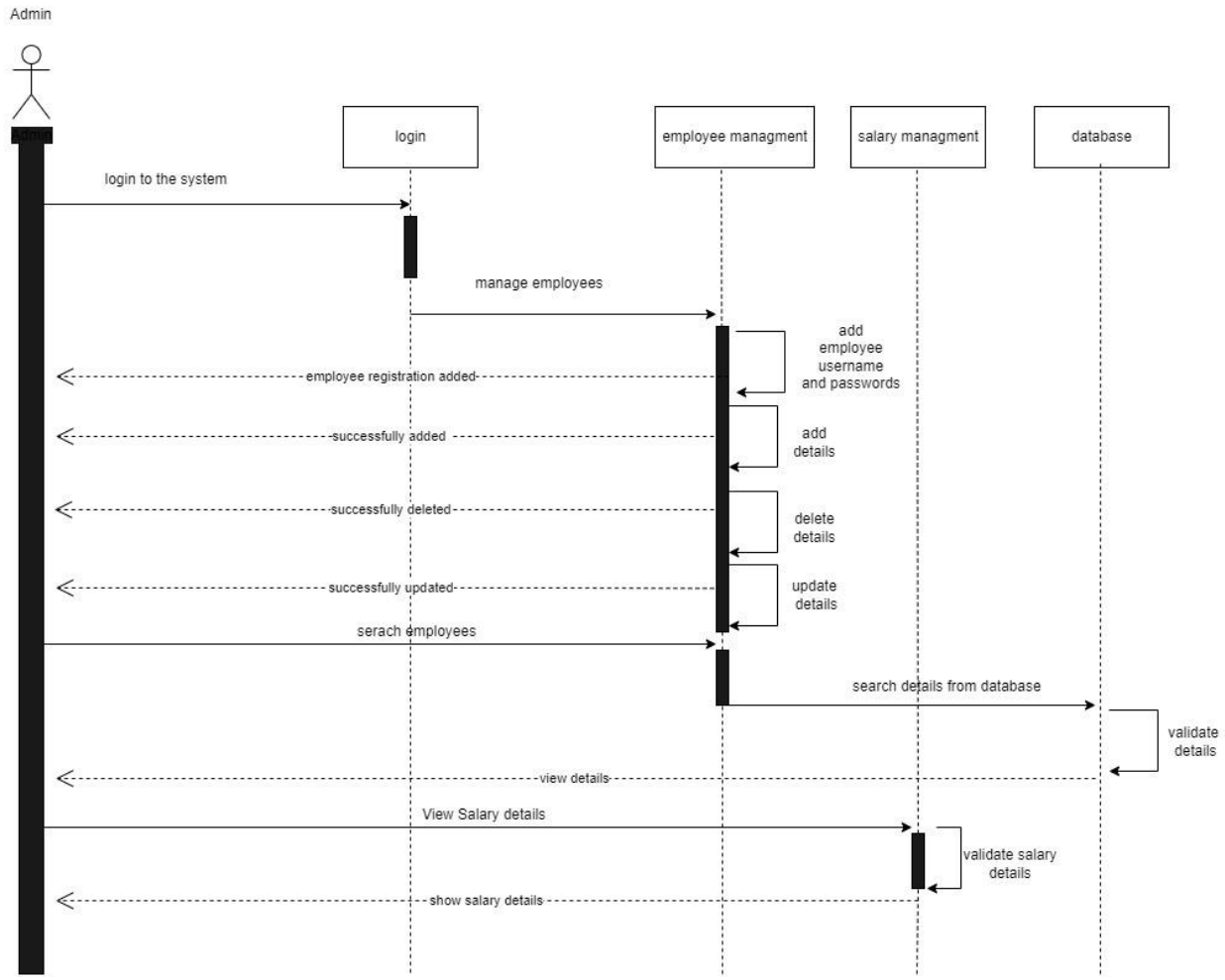


Figure 3 Sequence diagram of K&D music store EMS Admin

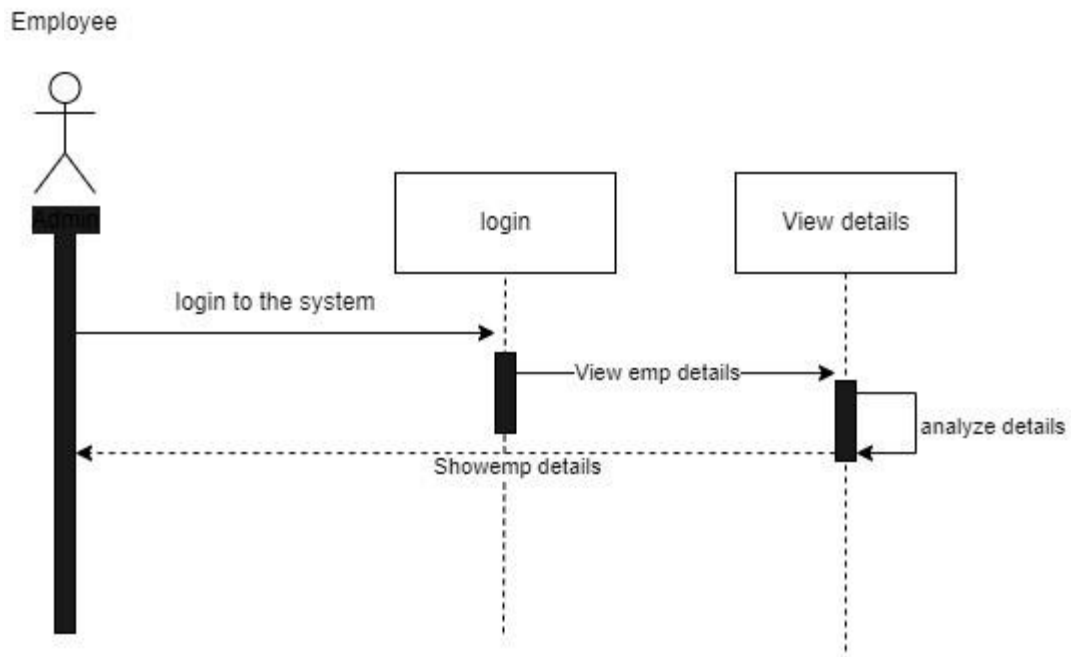
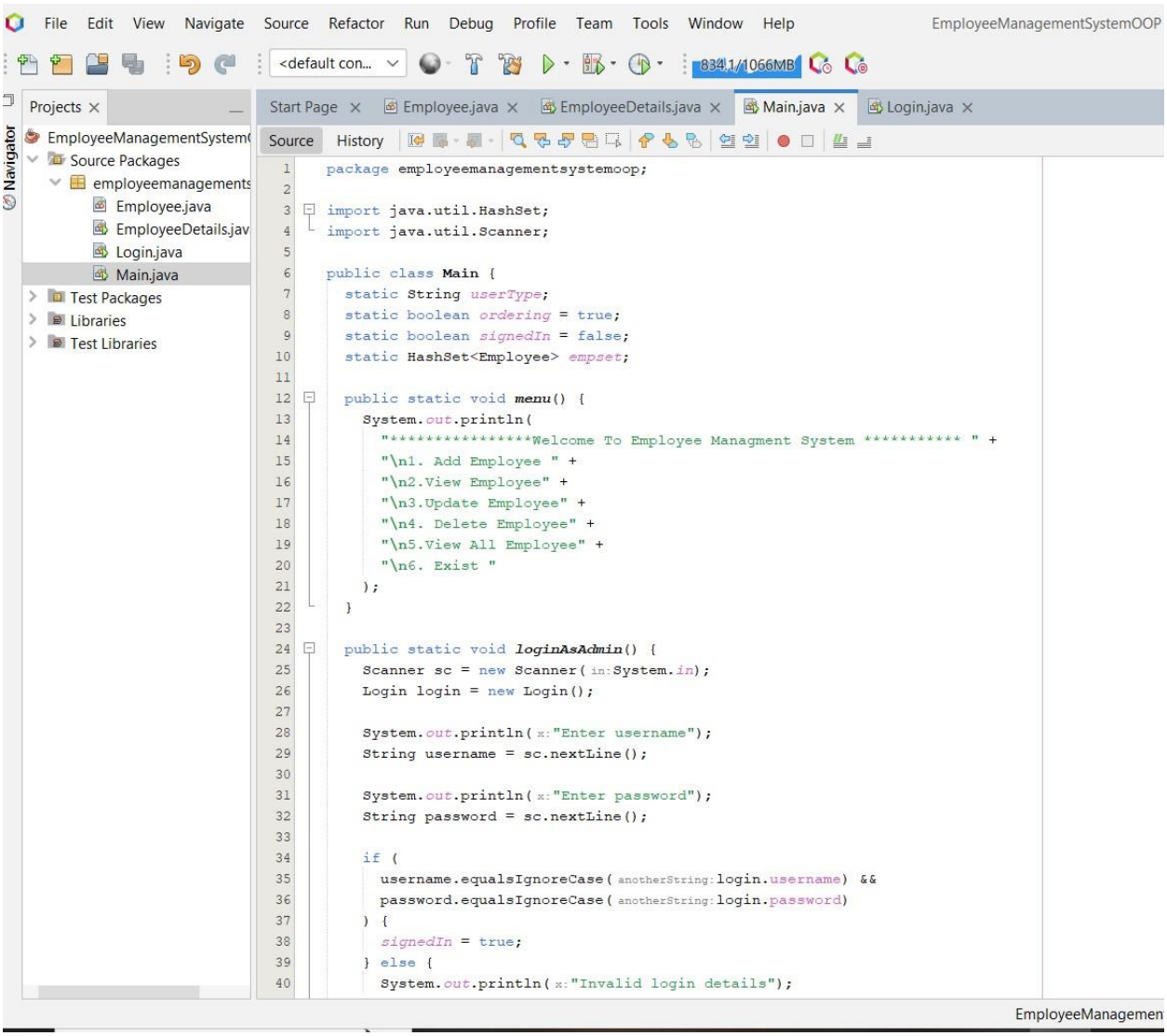
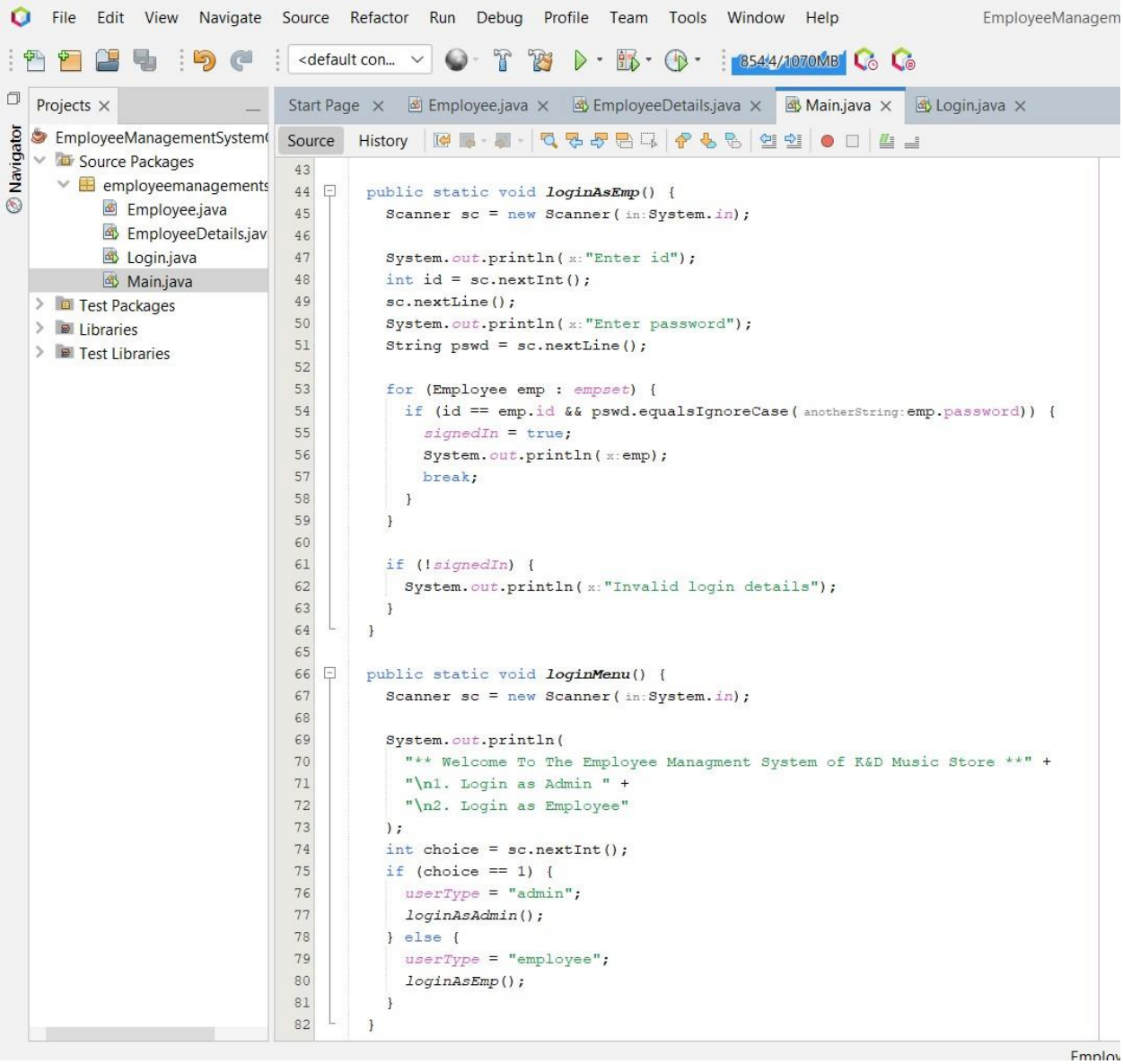


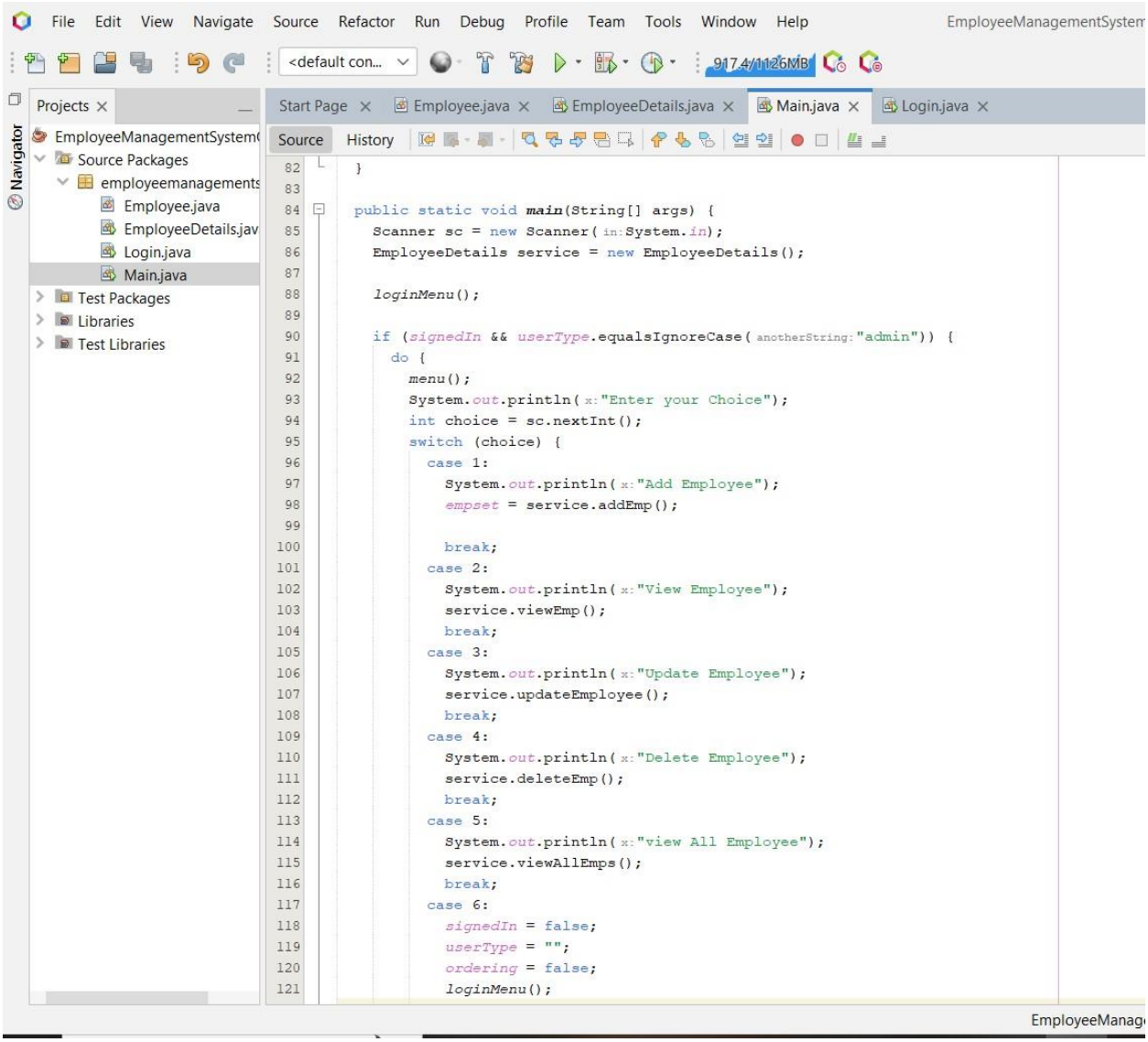
Figure 4 Sequence diagram of Employee K&D Music store EMS

Coding parts

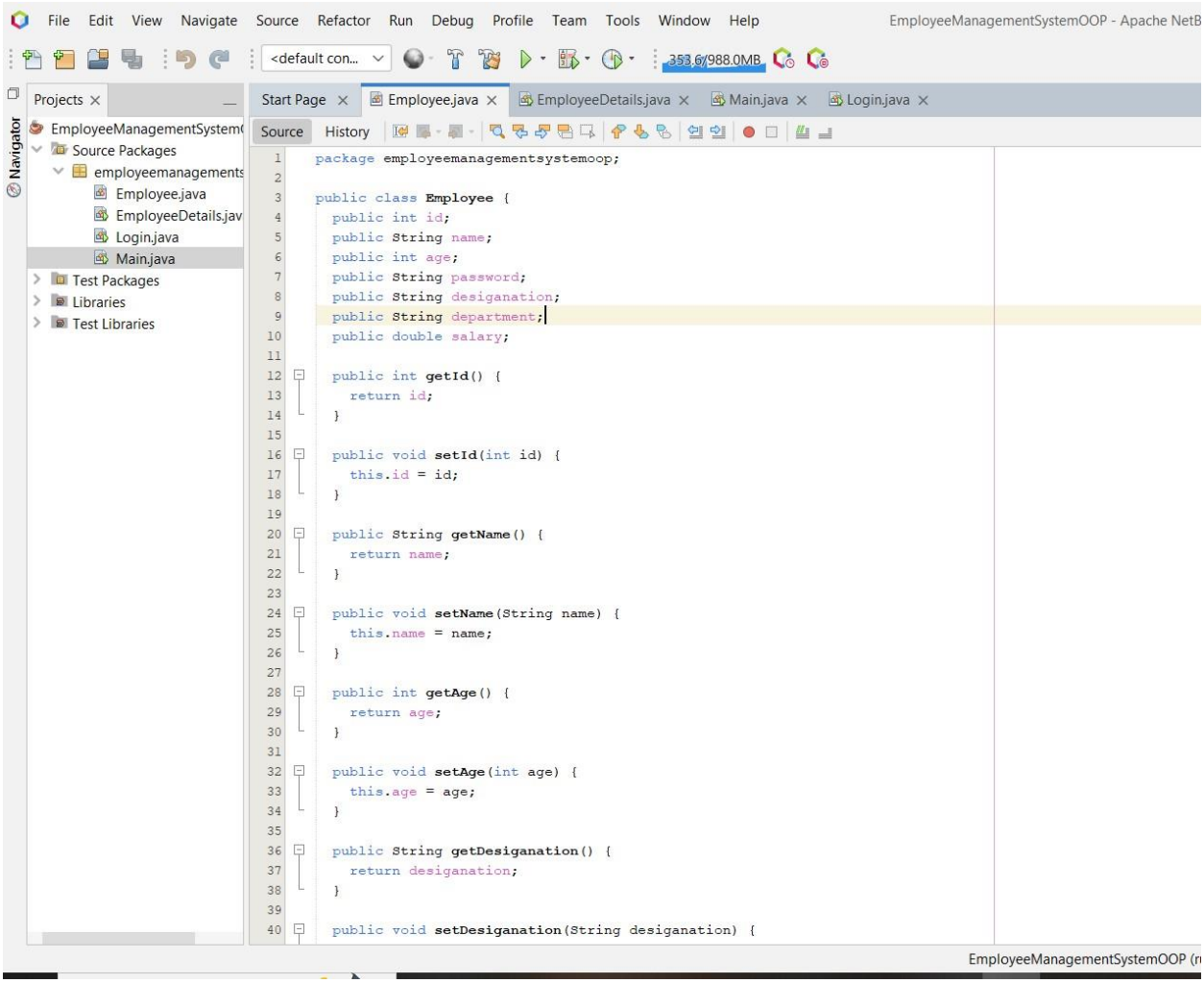
Main file

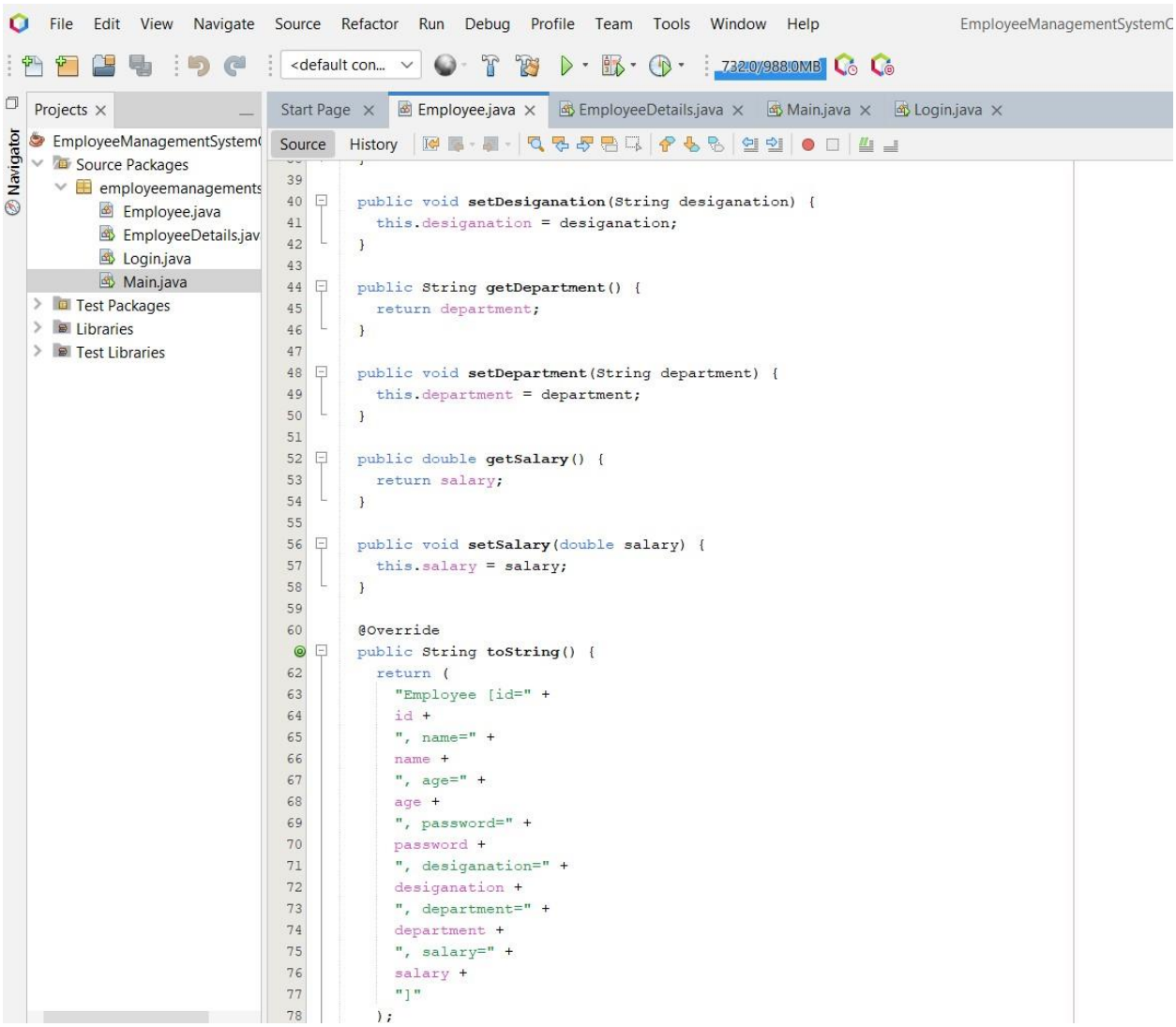

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
EmployeeManagementSystemOOP
Projects
Start Page
Employee.java
EmployeeDetails.java
Main.java
Login.java
Source
History
EmployeeManagementSystem
Source Packages
employeemanagements
Employee.java
EmployeeDetails.jav
Login.java
Main.java
Test Packages
Libraries
Test Libraries
Navigator
1 package employeemanagementsystemoop;
3 import java.util.HashSet;
4 import java.util.Scanner;
6 public class Main {
7 static String userType;
8 static boolean ordering = true;
9 static boolean signedIn = false;
10 static HashSet<Employee> empset;
12 public static void menu() {
13 System.out.println(
14 "***************Welcome To Employee Managment System ********** " +
15 "\n1. Add Employee " +
16 "\n2.View Employee" +
17 "\n3.Update Employee" +
18 "\n4. Delete Employee" +
19 "\n5.View All Employee" +
20 "\n6. Exist "
21);
22 }
24 public static void loginAsAdmin() {
25 Scanner sc = new Scanner(in:System.in);
26 Login login = new Login();
28 System.out.println(x:"Enter username");
29 String username = sc.nextLine();
31 System.out.println(x:"Enter password");
32 String password = sc.nextLine();
34 if (
35 username.equalsIgnoreCase(anotherString:login.username) &&
36 password.equalsIgnoreCase(anotherString:login.password)
37) {
38 signedIn = true;
39 } else {
40 System.out.println(x:"Invalid login details");
EmployeeManagemen

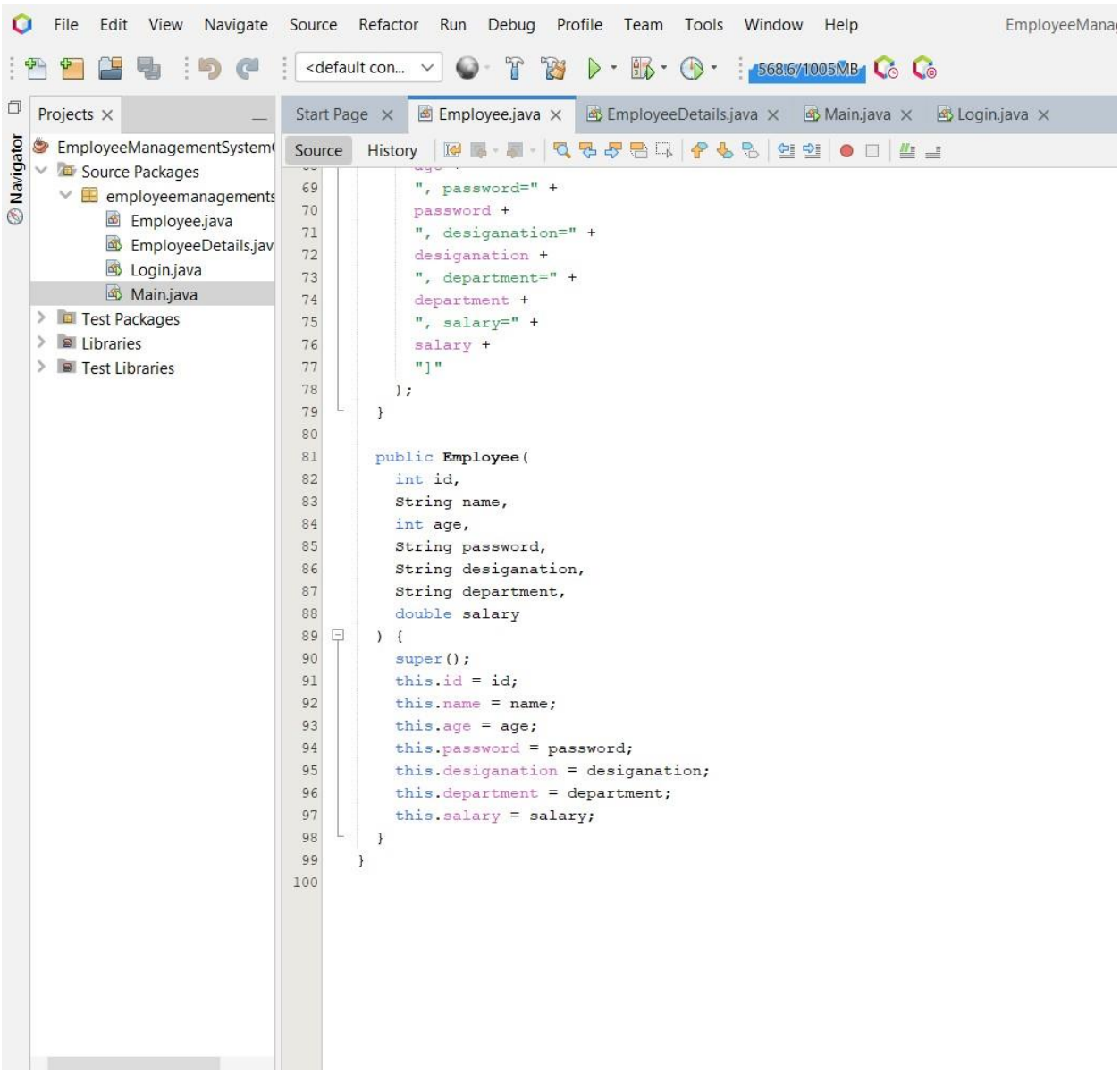

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
EmployeeManagem
<default con...
854/1070MB
Projects
Navigator
EmployeeManagementSystem
Source Packages
employeemanagements
Employee.java
EmployeeDetails.jav
Login.java
Main.java
Test Packages
Libraries
Test Libraries
Start Page
Employee.java
EmployeeDetails.java
Main.java
Login.java
Source
History
43
44 public static void loginAsEmp() {
45 Scanner sc = new Scanner(in:System.in);
46
47 System.out.println(x:"Enter id");
48 int id = sc.nextInt();
49 sc.nextLine();
50 System.out.println(x:"Enter password");
51 String pswd = sc.nextLine();
52
53 for (Employee emp : empset) {
54 if (id == emp.id && pswd.equalsIgnoreCase(anotherString:emp.password)) {
55 signedIn = true;
56 System.out.println(x:emp);
57 break;
58 }
59 }
60
61 if (!signedIn) {
62 System.out.println(x:"Invalid login details");
63 }
64 }
65
66 public static void loginMenu() {
67 Scanner sc = new Scanner(in:System.in);
68
69 System.out.println(
70 "** Welcome To The Employee Managment System of K&D Music Store **" +
71 "\n1. Login as Admin " +
72 "\n2. Login as Employee"
73);
74 int choice = sc.nextInt();
75 if (choice == 1) {
76 userType = "admin";
77 loginAsAdmin();
78 } else {
79 userType = "employee";
80 loginAsEmp();
81 }
82 }



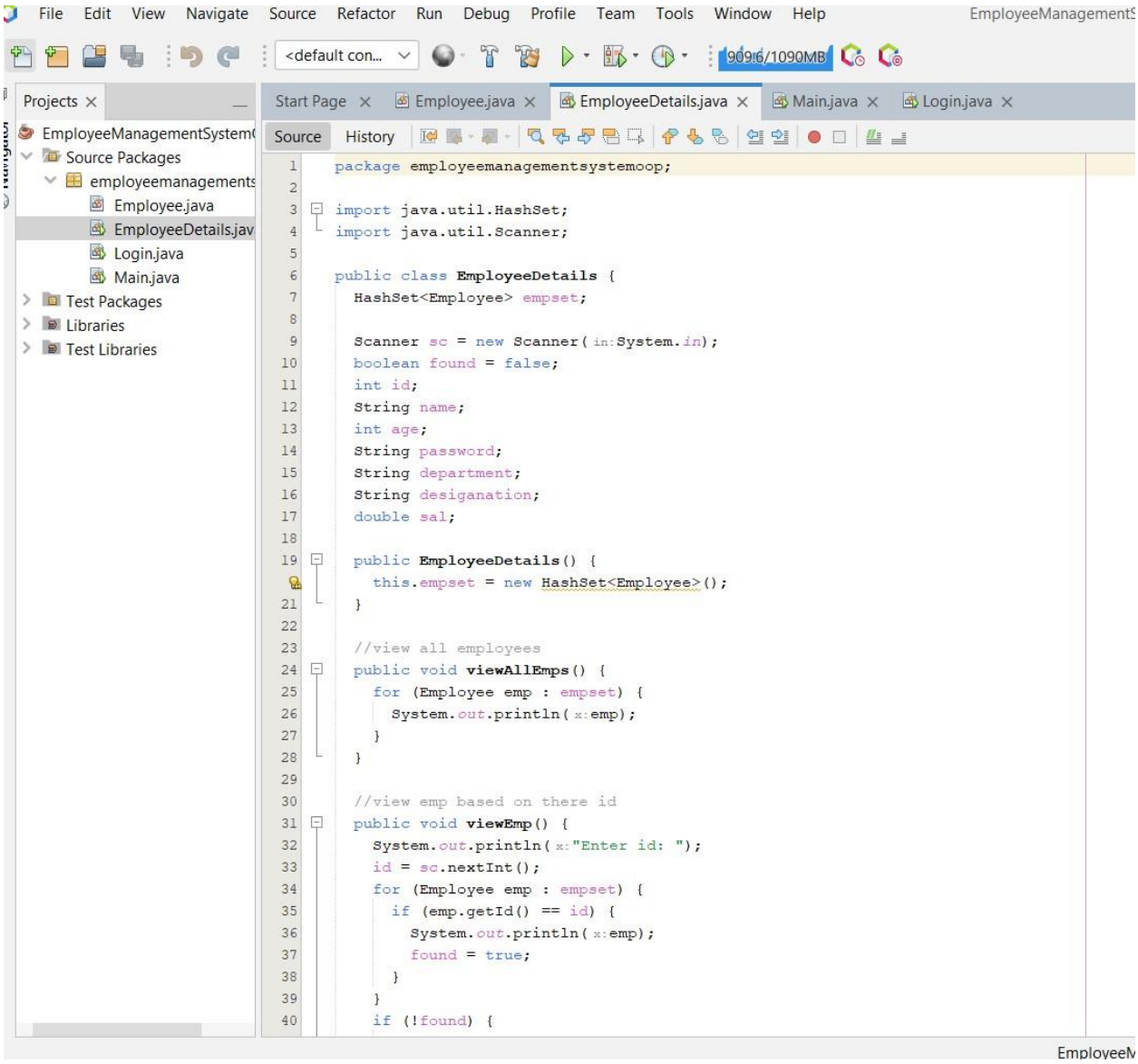
Employee file

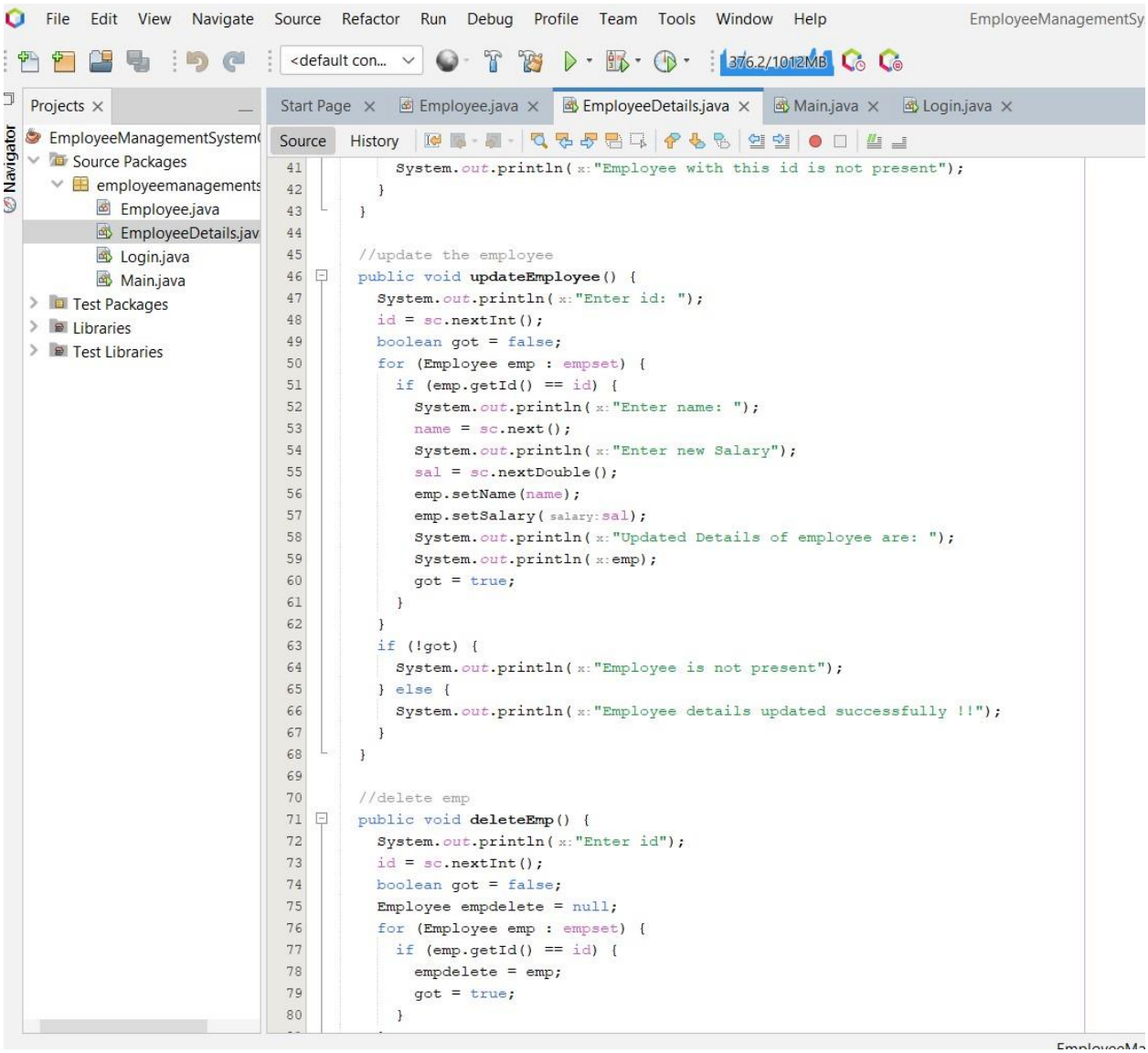

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
EmployeeManagementSystemOOP - Apache NetB
<default con...
353.6/988.0MB
Projects
Navigator
EmployeeManagementSystem
Source Packages
employeemanagements
Employee.java
EmployeeDetails.jav
Login.java
Main.java
Test Packages
Libraries
Test Libraries
Start Page
Employee.java
EmployeeDetails.java
Main.java
Login.java
Source
History
1 package employeemanagementsystemoop;
2
3 public class Employee {
4 public int id;
5 public String name;
6 public int age;
7 public String password;
8 public String desiganation;
9 public String department;
10 public double salary;
11
12 public int getId() {
13 return id;
14 }
15
16 public void setId(int id) {
17 this.id = id;
18 }
19
20 public String getName() {
21 return name;
22 }
23
24 public void setName(String name) {
25 this.name = name;
26 }
27
28 public int getAge() {
29 return age;
30 }
31
32 public void setAge(int age) {
33 this.age = age;
34 }
35
36 public String getDesiganation() {
37 return desiganation;
38 }
39
40 public void setDesiganation(String desiganation) {
EmployeeManagementSystemOOP (r

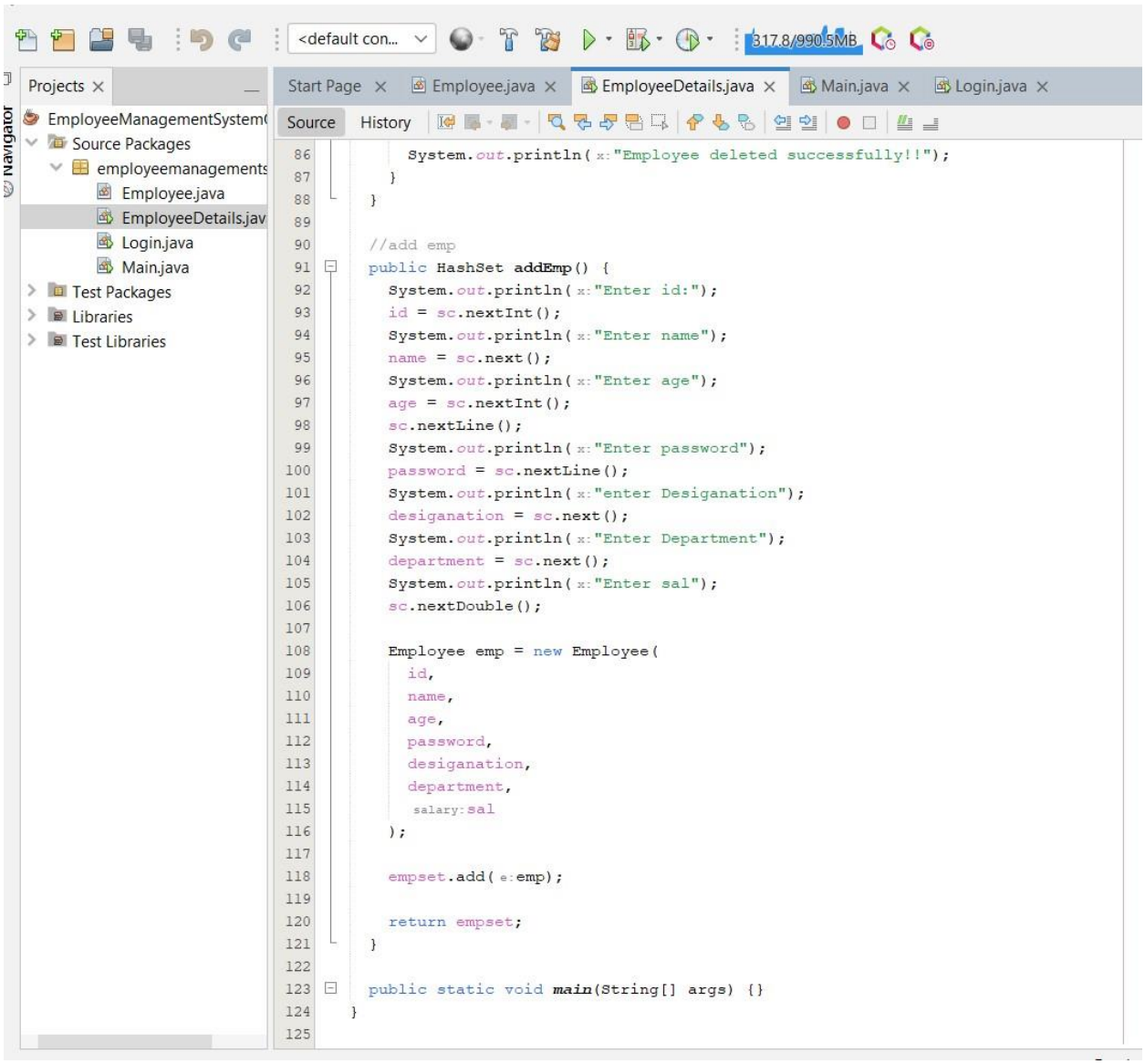

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
EmployeeManagementSystemC
<default con...
732:0/988:0MB
Projects
Navigator
EmployeeManagementSystem(
Source Packages
employeemanagements
Employee.java
EmployeeDetails.jav
Login.java
Main.java
Test Packages
Libraries
Test Libraries
Start Page
Employee.java
EmployeeDetails.java
Main.java
Login.java
Source
History
public void setDesiganation(String desiganation) {
this.desiganation = desiganation;
}
public String getDepartment() {
return department;
}
public void setDepartment(String department) {
this.department = department;
}
public double getSalary() {
return salary;
}
public void setSalary(double salary) {
this.salary = salary;
}
@Override
public String toString() {
return (
"Employee [id=" +
id +
", name=" +
name +
", age=" +
age +
", password=" +
password +
", desiganation=" +
desiganation +
", department=" +
department +
", salary=" +
salary +
"]"
);



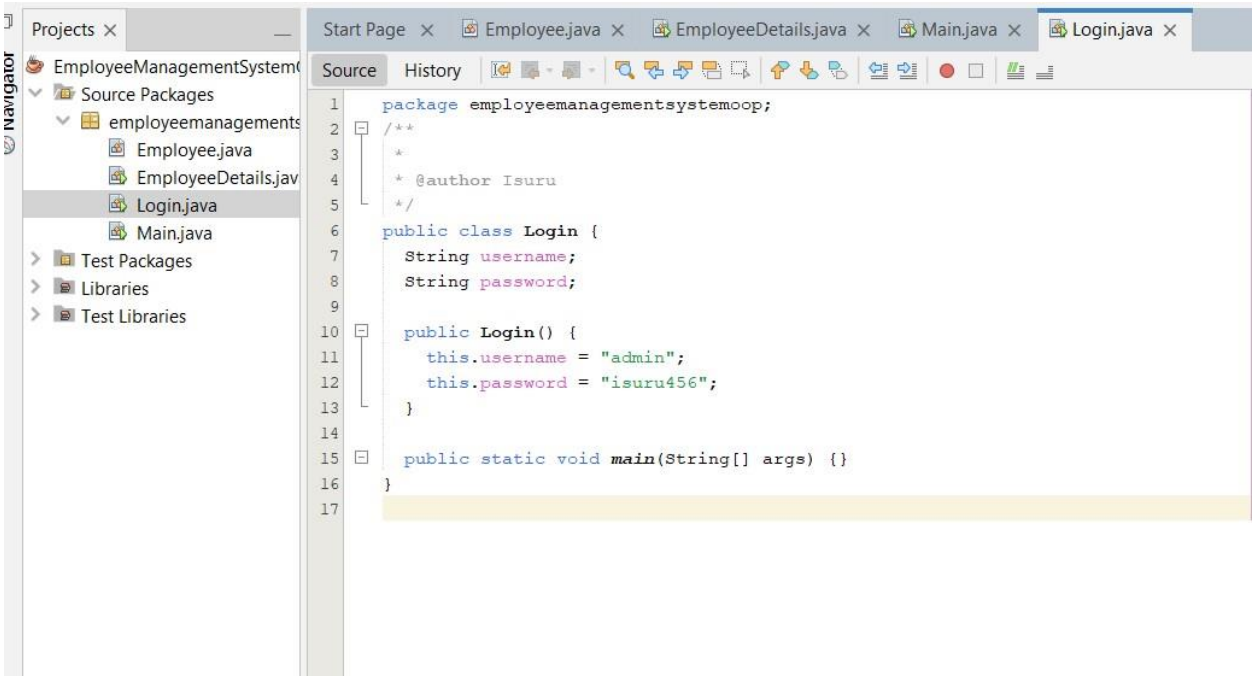
Employee details


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
EmployeeManagementS
<default con...
909:6/1090MB
Projects
EmployeeManagementSystem(
Source Packages
employeemanagements
Employee.java
EmployeeDetails.jav
Login.java
Main.java
Test Packages
Libraries
Test Libraries
Start Page
Employee.java
EmployeeDetails.java
Main.java
Login.java
Source
History
1 package employeemanagementsystemoop;
2
3 import java.util.HashSet;
4 import java.util.Scanner;
5
6 public class EmployeeDetails {
7 HashSet<Employee> empset;
8
9 Scanner sc = new Scanner(in:System.in);
10 boolean found = false;
11 int id;
12 String name;
13 int age;
14 String password;
15 String department;
16 String desiganation;
17 double sal;
18
19 public EmployeeDetails() {
21 this.empset = new HashSet<Employee>();
22 }
23
24 //view all employees
25 public void viewAllEmps() {
26 for (Employee emp : empset) {
27 System.out.println(x:emp);
28 }
29 }
30
31 //view emp based on there id
32 public void viewEmp() {
33 System.out.println(x:"Enter id: ");
34 id = sc.nextInt();
35 for (Employee emp : empset) {
36 if (emp.getId() == id) {
37 System.out.println(x:emp);
38 found = true;
39 }
40 }
if (!found) {
EmployeeM


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
EmployeeManagementSy
<default con...
376.2/1012MB
Projects
Navigator
EmployeeManagementSystem
Source Packages
employeemanagements
Employee.java
EmployeeDetails.jav
Login.java
Main.java
Test Packages
Libraries
Test Libraries
Start Page
Employee.java
EmployeeDetails.java
Main.java
Login.java
Source
History
41 System.out.println(x: "Employee with this id is not present");
42 }
43 }
44
45 //update the employee
46 public void updateEmployee() {
47 System.out.println(x: "Enter id: ");
48 id = sc.nextInt();
49 boolean got = false;
50 for (Employee emp : empset) {
51 if (emp.getId() == id) {
52 System.out.println(x: "Enter name: ");
53 name = sc.next();
54 System.out.println(x: "Enter new Salary");
55 sal = sc.nextDouble();
56 emp.setName(name);
57 emp.setSalary(salary: sal);
58 System.out.println(x: "Updated Details of employee are: ");
59 System.out.println(x: emp);
60 got = true;
61 }
62 }
63 if (!got) {
64 System.out.println(x: "Employee is not present");
65 } else {
66 System.out.println(x: "Employee details updated successfully !!");
67 }
68 }
69
70 //delete emp
71 public void deleteEmp() {
72 System.out.println(x: "Enter id");
73 id = sc.nextInt();
74 boolean got = false;
75 Employee empdelete = null;
76 for (Employee emp : empset) {
77 if (emp.getId() == id) {
78 empdelete = emp;
79 got = true;
80 }


Projects
EmployeeManagementSystem(
Source Packages
employeemanagements
Employee.java
EmployeeDetails.jav
Login.java
Main.java
Test Packages
Libraries
Test Libraries
Start Page
Employee.java
EmployeeDetails.java
Main.java
Login.java
Source
History
317.8/990.5MB
System.out.println(x: "Employee deleted successfully!!");
}
}
//add emp
public HashSet addEmp() {
System.out.println(x: "Enter id:");
id = sc.nextInt();
System.out.println(x: "Enter name");
name = sc.next();
System.out.println(x: "Enter age");
age = sc.nextInt();
sc.nextLine();
System.out.println(x: "Enter password");
password = sc.nextLine();
System.out.println(x: "enter Desiganation");
desiganation = sc.next();
System.out.println(x: "Enter Department");
department = sc.next();
System.out.println(x: "Enter sal");
sc.nextDouble();
Employee emp = new Employee(
id,
name,
age,
password,
desiganation,
department,
salary: sal
);
empset.add(e: emp);
return empset;
}
public static void main(String[] args) {}
}

Log in file


Projects
EmployeeManagementSystem(
Source Packages
employeemanagements
Employee.java
EmployeeDetails.jav
Login.java
Main.java
Test Packages
Libraries
Test Libraries
Navigator
Start Page
Employee.java
EmployeeDetails.java
Main.java
Login.java
Source
History
1 package employeemanagementsystemoop;
2 /**
3 *
4 * @author Isuru
5 */
6 public class Login {
7 String username;
8 String password;
9
10 public Login() {
11 this.username = "admin";
12 this.password = "isuru456";
13 }
14
15 public static void main(String[] args) {}
16 }
17

Outputs

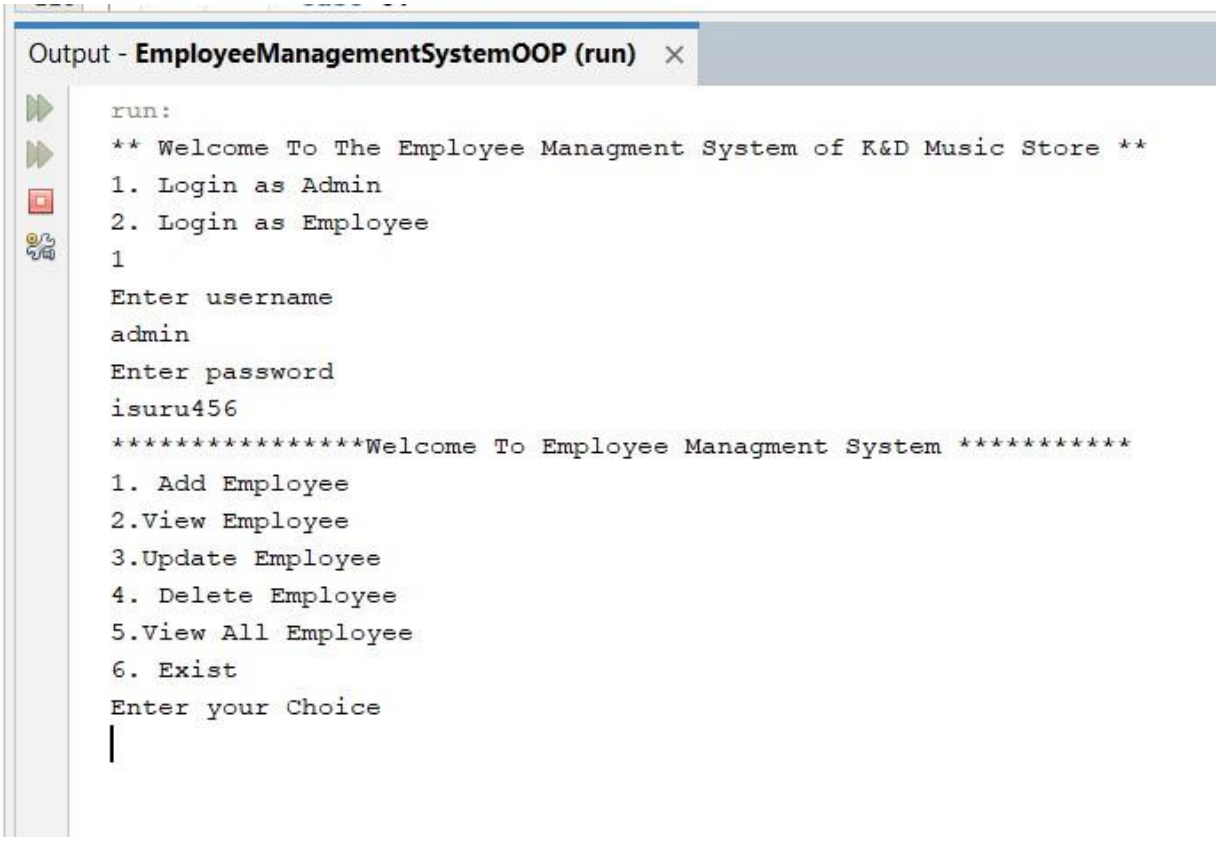


Figure 5 Log in section

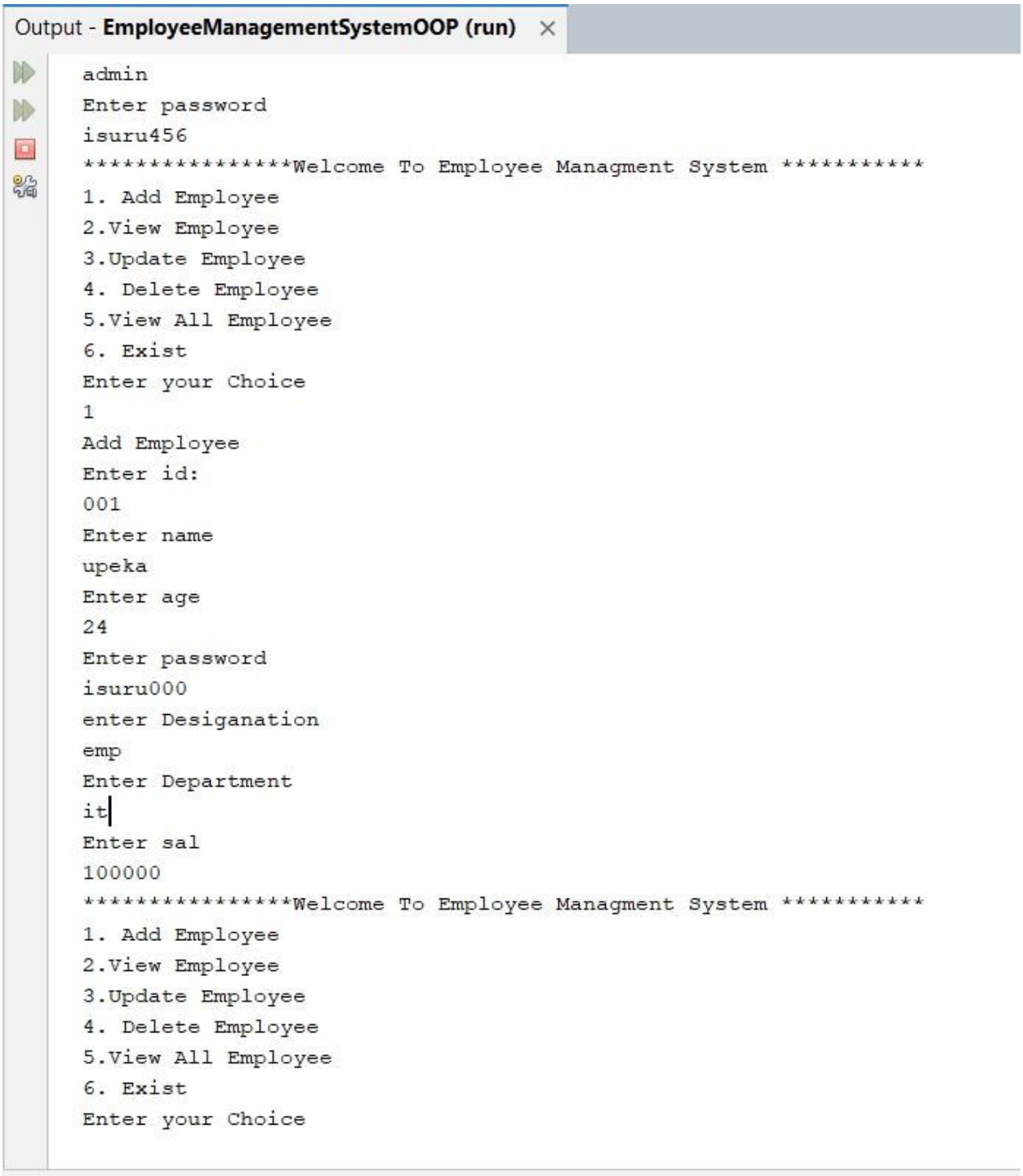


Figure 6 Add employee

```
Enter Sal
100000
*****************Welcome To Employee Managment System ************
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
2
View Employee
Enter id:
001
Employee [id=1, name=upeka, age=24, password=isuru000, desiganation=emp, department=it, salary=0.0]
*****************Welcome To Employee Managment System ************
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
```

EmployeeManagemer

Figure 7 View employee

```
Employee [id=1, name=upeka, age=24, password=isuru000, desiganation=emp, department=it, salary=0.0]
*****************Welcome To Employee Managment System ************
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
3
Update Employee
Enter id:
001
Enter name:
upeka
Enter new Salary
150000
Updated Details of employee are:
Employee [id=1, name=upeka, age=24, password=isuru000, desiganation=emp, department=it, salary=150000.0]
Employee details updated successfully !!
*****************Welcome To Employee Managment System ************
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
```

Figure 8 Update employee details

```
****************Welcome To Employee Managment System ***********
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
5
view All Employee
Employee [id=1, name=upeka, age=24, password=isuru000, desiganation=emp, department=it, salary=150000.0]
****************Welcome To Employee Managment System ***********
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
```

Figure 9 View all employee

```
5
view All Employee
Employee [id=1, name=upeka, age=24, password=isuru000, desiganation=emp, department=it, salary=150000.0]
****************Welcome To Employee Managment System ***********
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
6
** Welcome To The Employee Managment System of K&D Music Store **
1. Login as Admin
2. Login as Employee
2
Enter id
001
Enter password
isuru000
Employee [id=1, name=upeka, age=24, password=isuru000, desiganation=emp, department=it, salary=150000.0]
Thank you for using the application!!
Please enter valid choice
BUILD SUCCESSFUL (total time: 4 minutes 26 seconds)
```

Figure 10 Log in as employee and view details

```
Enter id:
001
Enter name
upeka
Enter age
24
Enter password
isuru000
enter Desiganation
IT
Enter Department
ITDE
Enter sal
1000000
****************Welcome To Employee Managment System ***********
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
4
Delete Employee
Enter id
001
Employee deleted successfully!!
****************Welcome To Employee Managment System ***********
1. Add Employee
2.View Employee
3.Update Employee
4. Delete Employee
5.View All Employee
6. Exist
Enter your Choice
```

Figure 11 Delete employee by admin