Experiment – 2.4

Student Name: Rohan Jaiswal
UID: 21BCS2856
Branch: BE-CSE
Section: KRG_CC_1-B
Semester: 06
Date of Performance: 19-03-2024
Subject Name: Project-Based Learning in Java with Lab
Subject Code: 21CSH-319

1. **Aim:** Create a menu-based Java application with the following options. 1. Add an Employee 2.Display All 3.Exit If option 1 is selected, the application should gather details of the employee like employee name, employee ID, designation, and salary and store it in a file. If option 2 is selected, the application should display all the employee details. If option 3 is selected the application should exit.

2. **Objective:** To learn about the concept of File Handling in Java and to learn about LinkedList and Exception Handling in Java.

3. **Input/Apparatus Used:** Hardware Requirements: - Minimum 384MB RAM, 100 GB hard Disk, processor with 2.1 MHz Software Requirements: - Eclipse, NetBeans, IntelliJ, etc.

4. **Procedure/Algorithm/Pseudocode:**

1. Create a class Employee to store its details like id, name ,salary, age, etc.
2. Create an EmployeeManager class where you ask for your choice using:
   - Add an employee
   - Display All
   - Exit.

5. **Script and Output:**

```
import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;

class Employee {
   private int id;
   private String name;
   private int age;
```

```
    private double salary;
    public Employee(int id, String name, int age, double salary) {
        this.id = id;
        this.name = name;
        this.age = age;
        this.salary = salary;
    }
    public String toString() {
        return id + " " + name + " " + age + " " + salary;
    }
}

public class EmployeeManagementSystem {
    private static final String FILE_NAME = "employees.txt";
    private static ArrayList<Employee> employees = new ArrayList<>();

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        int choice;
        do {
            System.out.println("Main Menu");
            System.out.println("1. Add an Employee");
            System.out.println("2. Display All");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    addEmployee(scanner);
                    break;
                case 2:
                    displayAll();
                    break;
                case 3:
                    System.out.println("Exiting the System");
```

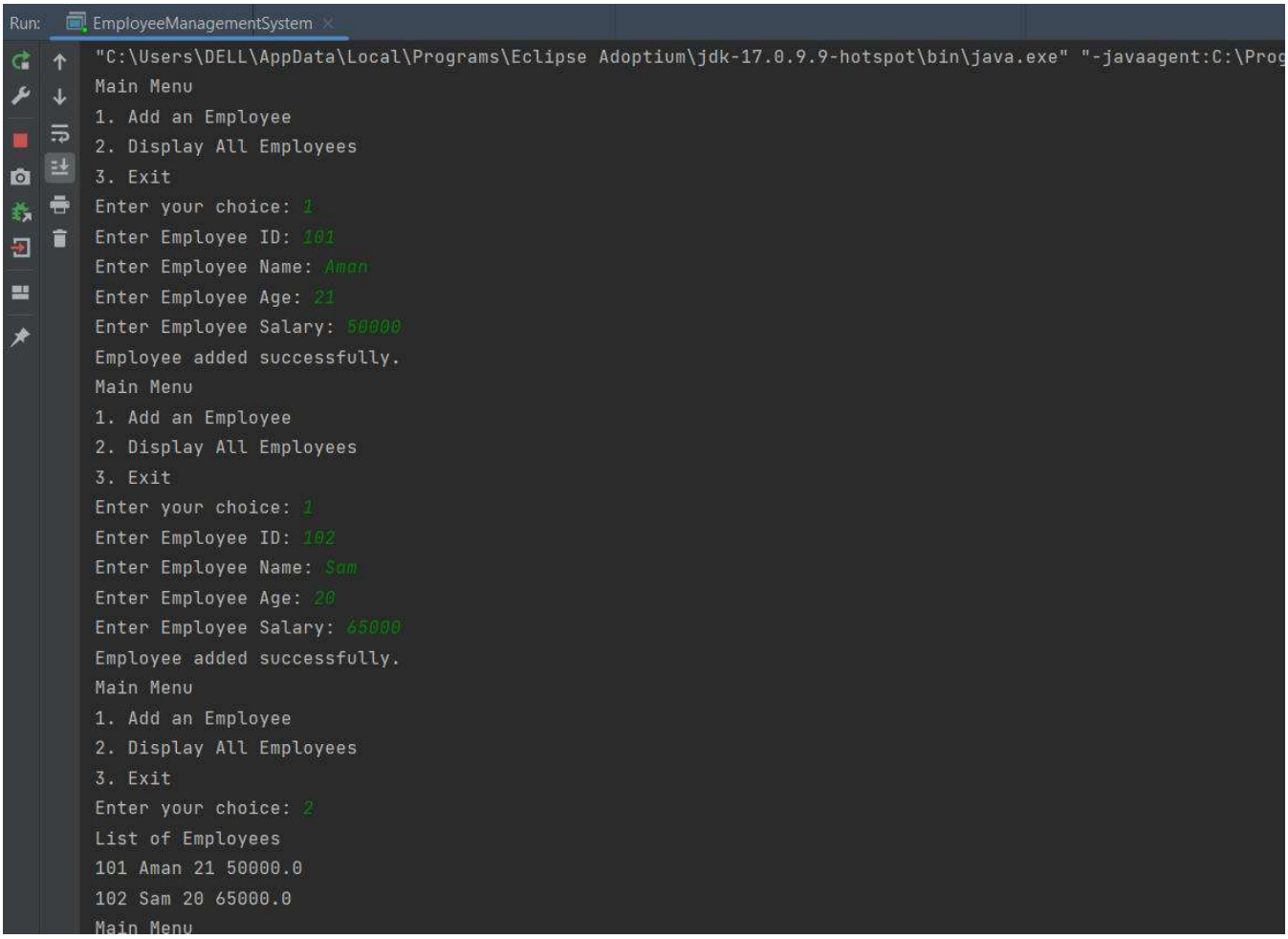
```
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 3);
}

private static void addEmployee(Scanner scanner) {
    System.out.print("Enter Employee ID: ");
    int id = scanner.nextInt();
    scanner.nextLine();
    System.out.print("Enter Employee Name: ");
    String name = scanner.nextLine();
    System.out.print("Enter Employee Age: ");
    int age = scanner.nextInt();
    System.out.print("Enter Employee Salary: ");
    double salary = scanner.nextDouble();

    employees.add(new Employee(id, name, age, salary));

    try (PrintWriter writer = new PrintWriter(new
FileWriter(FILE_NAME, true))) {
        writer.println(id + "," + name + "," + age + "," + salary);
        System.out.println("Employee added successfully.");
    } catch (IOException e) {
        System.err.println("Error occurred while writing to file: " +
e.getMessage());
    }
}

private static void displayAll() {
    System.out.println("List of Employees");
    for (Employee employee : employees) {
        System.out.println(employee);
    }
}
}
```



6. **Learning Outcome:** In this experiment, we learned about
   a. concept of File Handling in Java
   b. LinkedList, Exception Handling in Java.