Untitled-2

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

public class PayrollSystem {
    private static final String FILE_NAME = "employees.txt";
    private static Map<String, Employee> employees = new HashMap<>();

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

        while (true) {
            System.out.println("\nEmployee Payroll System");
            System.out.println("1. Add Employee");
            System.out.println("2. Calculate Salary");
            System.out.println("3. View Employee");
            System.out.println("4. Exit");
            System.out.print("Enter command: ");
            command = scanner.nextLine();

            switch (command) {
                case "1":
                    addEmployee(scanner);
                    break;
                case "2":
                    calculateSalary(scanner);
                    break;
                case "3":
                    viewEmployee(scanner);
                    break;
                case "4":
                    saveEmployees();
                    System.out.println("Exiting...");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid command. Please try again.");
            }
        }
    }

    private static void addEmployee(Scanner scanner) {
        System.out.print("Enter employee ID: ");
        String id = scanner.nextLine();
        if (employees.containsKey(id)) {
            System.out.println("Employee ID already exists.");
            return;
        }

        System.out.print("Enter employee name: ");
        String name = scanner.nextLine();
        System.out.print("Enter employee hourly wage: ");
        double hourlyWage = Double.parseDouble(scanner.nextLine());
        System.out.print("Enter hours worked: ");
        double hoursWorked = Double.parseDouble(scanner.nextLine());

        Employee employee = new Employee(id, name, hourlyWage, hoursWorked);
        employees.put(id, employee);
        System.out.println("Employee added successfully.");
    }

    private static void calculateSalary(Scanner scanner) {
        System.out.print("Enter employee ID to calculate salary: ");
        String id = scanner.nextLine();
        Employee employee = employees.get(id);

        if (employee != null) {
            double salary = employee.calculateSalary();
            System.out.println("Salary for " + employee.getName() + ": VSCODE_PRINT_CONTENTquot; + salary);
        } else {
            System.out.println("Employee not found.");
        }
    }

    private static void viewEmployee(Scanner scanner) {
        System.out.print("Enter employee ID to view details: ");
        String id = scanner.nextLine();
        Employee employee = employees.get(id);

        if (employee != null) {
            System.out.println("Employee ID: " + employee.getId());
            System.out.println("Name: " + employee.getName());
            System.out.println("Hourly Wage: VSCODE_PRINT_CONTENTquot; + employee.getHourlyWage());
            System.out.println("Hours Worked: " + employee.getHoursWorked());
        } else {
            System.out.println("Employee not found.");
        }
    }

    private static void loadEmployees() {
        try (BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length == 4) {
                    String id = parts[0];
                    String name = parts[1];
                    double hourlyWage = Double.parseDouble(parts[2]);
                    double hoursWorked = Double.parseDouble(parts[3]);

                    Employee employee = new Employee(id, name, hourlyWage, hoursWorked);
                    employees.put(id, employee);
                }
            }
        } catch (IOException e) {
            System.out.println("No existing employee file found. Starting fresh.");
        }
    }

    private static void saveEmployees() {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE_NAME))) {
            for (Employee employee : employees.values()) {
                writer.write(employee.getId() + "," + employee.getName() + ","
                        + employee.getHourlyWage() + "," + employee.getHoursWorked());
                writer.newLine();
            }
        } catch (IOException e) {
            System.out.println("Error saving employee data.");
        }
    }

    private static class Employee {
        private String id;
        private String name;
        private double hourlyWage;
        private double hoursWorked;

        public Employee(String id, String name, double hourlyWage, double hoursWorked) {
            this.id = id;
            this.name = name;
            this.hourlyWage = hourlyWage;
            this.hoursWorked = hoursWorked;
        }

        public String getId() {
            return id;
        }

        public String getName() {
            return name;
        }

        public double getHourlyWage() {
            return hourlyWage;
        }

        public double getHoursWorked() {
            return hoursWorked;
        }

        public double calculateSalary() {
            return hourlyWage * hoursWorked;
        }
    }
}

```