

```
import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.List;
import java.util.Scanner;

class Employee {
    private int id;
    private String name;
    private String department;
    private double salary;

    public Employee(int id, String name, String department, double salary) {
        this.id = id;
        this.name = name;
        this.department = department;
        this.salary = salary;
    }

    public void display() {
        System.out.printf("ID: %d | Name: %s | Department: %s | Salary: %.2f%n", id, name,
            department, salary);
    }
}

public class EmployeeManagement {
    private static Scanner scanner = new Scanner(System.in);
    private static List<Employee> employees = new ArrayList<>();

    public static void main(String[] args) {
        while (true) {
            showMenu();
        }
    }
}
```

```

int choice = getUserChoice();

switch (choice) {
    case 1:
        addEmployee();
        break;
    case 2:
        displayEmployees();
        break;
    case 3:
        System.out.println("Exiting... Have a great day!");
        scanner.close();
        return;
    default:
        System.out.println("Invalid option! Please enter a valid choice.");
}
}
}

```

```

private static void showMenu() {
    System.out.println("\nEmployee Management System");
    System.out.println("1. Add Employee");
    System.out.println("2. Display Employees");
    System.out.println("3. Exit");
    System.out.print("Select an option: ");
}

```

```

private static int getUserChoice() {
    int choice = -1;
    while (choice < 1 || choice > 3) {
        try {

```

```

        choice = scanner.nextInt();

        scanner.nextLine(); // consume the newline character
    } catch (InputMismatchException e) {
        scanner.nextLine(); // clear invalid input

        System.out.println("Invalid input. Please enter a number between 1 and 3.");
    }
}

return choice;
}

```

```

private static void addEmployee() {
    int id = getValidInt("Employee ID: ");
    String name = getValidString("Employee Name: ");
    String department = getValidString("Employee Department: ");
    double salary = getValidSalary("Employee Salary: ");

    employees.add(new Employee(id, name, department, salary));
    System.out.println("Employee added successfully.");
}

```

```

private static String getValidString(String prompt) {
    String input = "";
    while (input.trim().isEmpty()) {
        System.out.print(prompt);
        input = scanner.nextLine().trim();
        if (input.isEmpty()) {
            System.out.println("This field cannot be empty. Please enter again.");
        }
    }
    return input;
}

```

```
private static int getValidInt(String prompt) {  
    int value = -1;  
    while (value <= 0) {  
        System.out.print(prompt);  
        try {  
            value = scanner.nextInt();  
            scanner.nextLine(); // consume the newline character  
            if (value <= 0) {  
                System.out.println("ID must be a positive number. Please enter again.");  
            }  
        } catch (InputMismatchException e) {  
            scanner.nextLine(); // clear invalid input  
            System.out.println("Invalid input. Please enter a valid integer.");  
        }  
    }  
    return value;  
}
```

```
private static double getValidSalary(String prompt) {  
    double salary = -1;  
    while (salary <= 0) {  
        System.out.print(prompt);  
        try {  
            salary = scanner.nextDouble();  
            scanner.nextLine(); // consume the newline character  
            if (salary <= 0) {  
                System.out.println("Salary must be a positive number. Please enter again.");  
            }  
        } catch (InputMismatchException e) {  
            scanner.nextLine(); // clear invalid input  
        }  
    }  
}
```

```
        System.out.println("Invalid input. Please enter a valid number for salary.");
    }
}
return salary;
}

private static void displayEmployees() {
    if (employees.isEmpty()) {
        System.out.println("No employees found.");
    } else {
        System.out.println("\nEmployee List:");
        for (Employee employee : employees) {
            employee.display();
        }
    }
}
}
```