Project Submission Report

Title of the Project:

Human Resource Management System

2. Brief on the project-

The HRMS is designed to facilitate efficient management of employee information, attendance records, and leave management for organizations. It should provide various functionalities for HR administrators to perform employee related tasks effectively

3. Deliverables of the project-

Functionality and User Interaction:

The HRMS project provides a menu-driven console interface that allows HR personnel to interact with the system. Upon running the application, the user is presented with a menu containing options for each functionality. The user can enter the corresponding number to access the desired feature.

Adding Employees: HR personnel can add new employees to the system by providing their name, employee ID, designation, department, and contact information.

Viewing Employee Details: This option displays a list of all employees stored in the system along with their details.

Updating Employee Details: HR personnel can update employee information, including their name, designation, department, and contact information, by providing the employee's ID.

Deleting Employees: Users can delete an employee from the system by providing their employee ID. Marking Attendance: HR personnel can mark the attendance of employees by specifying their employee ID and attendance status (Present, Absent, or Leave).
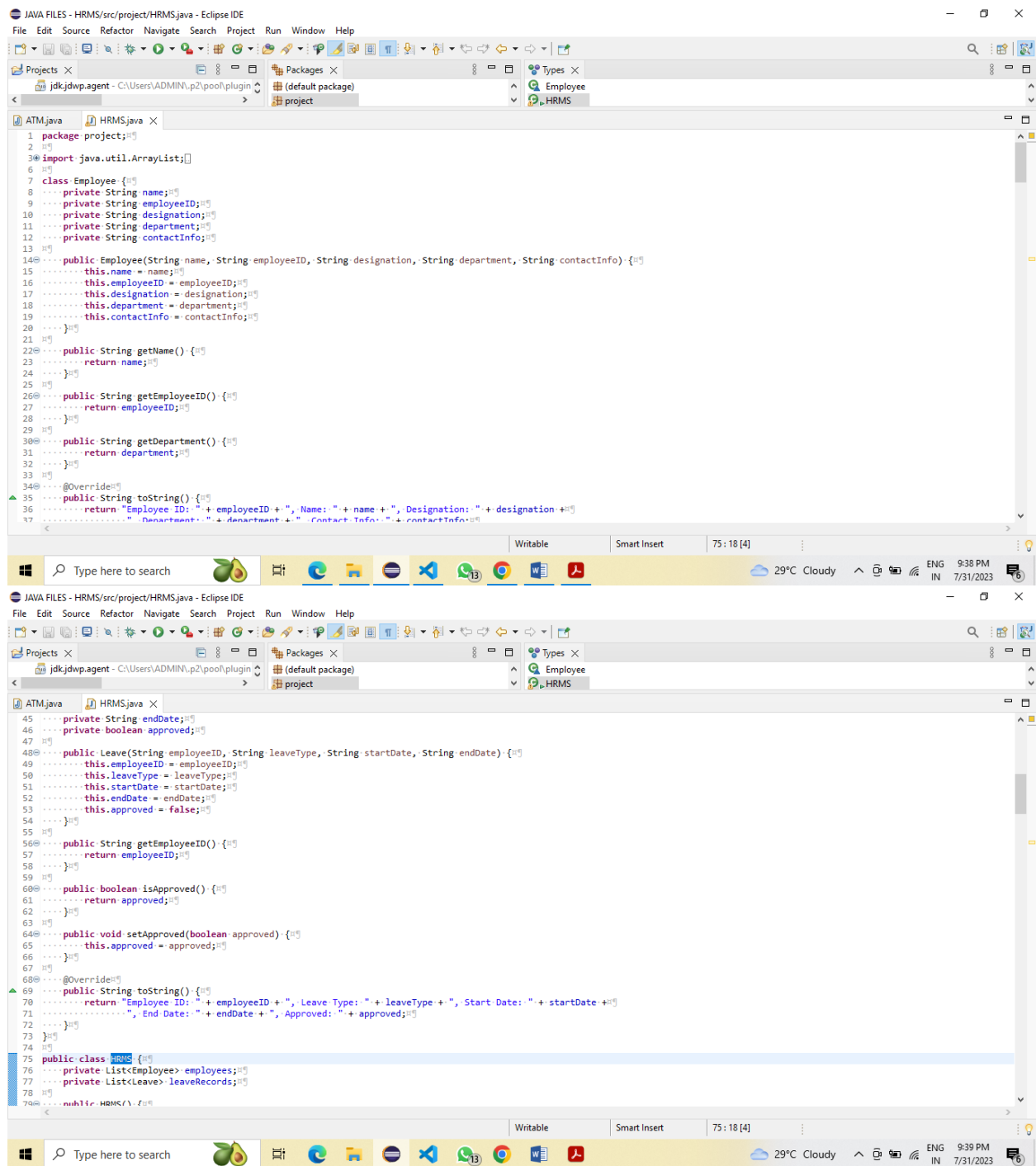
Adding Leave Requests: Employees can submit leave requests by providing their employee ID, leave type, start date, and end date.

Viewing Leave Records: This option displays a list of all leave requests along with their details, including the approval status.

Approving Leave Requests: HR personnel can approve leave requests by providing the corresponding leave request ID.

Rejecting Leave Requests: HR personnel can reject leave requests by providing the corresponding leave request ID. Searching Employees: HR personnel can search for employees based on their name, employee ID, or department.

Screenshots of the IDE:

```java
 90            if (employees.isEmpty()) {
 91                System.out.println("No employee records found.");
 92            } else {
 93                for (Employee employee : employees) {
 94                    System.out.println(employee);
 95                }
 96            }
 97        }
 98
 99        public void updateEmployee(String employeeID) {
100            boolean found = false;
101            for (Employee employee : employees) {
102                if (employee.getEmployeeID().equalsIgnoreCase(employeeID)) {
103                    Scanner scanner = new Scanner(System.in);
104                    System.out.print("Enter new name: ");
105                    String newName = scanner.nextLine();
106                    System.out.print("Enter new designation: ");
107                    String newDesignation = scanner.nextLine();
108                    System.out.print("Enter new department: ");
109                    String newDepartment = scanner.nextLine();
110                    System.out.print("Enter new contact information: ");
111                    String newContactInfo = scanner.nextLine();
112
113                    employee = new Employee(newName, employeeID, newDesignation, newDepartment, newContactInfo);
114                    found = true;
115                    System.out.println("Employee details updated successfully.");
116                    break;
117                }
118            }
119            if (!found) {
120                System.out.println("Employee not found.");
121            }
122        }
123
124        public void deleteEmployee(String employeeID) {
```

```java
168            }
169        }
170
171        public void approveLeave(String leaveID) {
172            boolean found = false;
173            for (Leave leave : leaveRecords) {
174                if (leave.getEmployeeID().equalsIgnoreCase(leaveID) && !leave.isApproved()) {
175                    leave.setApproved(true);
176                    found = true;
177                    System.out.println("Leave request approved for Employee ID: " + leaveID);
178                    break;
179                }
180            }
181            if (!found) {
182                System.out.println("Leave request not found or already approved.");
183            }
184        }
185    }
186    public void rejectLeave(String leaveID) {
187        boolean found = false;
188        for (Leave leave : leaveRecords) {
189            if (leave.getEmployeeID().equalsIgnoreCase(leaveID) && !leave.isApproved()) {
190                leaveRecords.remove(leave);
191                found = true;
192                System.out.println("Leave request rejected for Employee ID: " + leaveID);
193                break;
194            }
195        }
196        if (!found) {
197            System.out.println("Leave request not found or already approved.");
198        }
199    }
200
201    public void searchEmployee(String searchQuery) {
202        boolean found = false;
```

```java
285                         hrms.addLeave(leave);
286                         break;
287                     case 7:
288                         hrms.viewLeaveRecords();
289                         break;
290                     case 8:
291                         System.out.print("Enter leave request ID to approve: ");
292                         String approveLeaveID = scanner.nextLine();
293                         hrms.approveLeave(approveLeaveID);
294                         break;
295                     case 9:
296                         System.out.print("Enter leave request ID to reject: ");
297                         String rejectLeaveID = scanner.nextLine();
298                         hrms.rejectLeave(rejectLeaveID);
299                         break;
300                     case 10:
301                         System.out.print("Enter search query (name, employee ID, or department): ");
302                         String searchQuery = scanner.nextLine();
303                         hrms.searchEmployee(searchQuery);
304                         break;
305                     case 0:
306                         System.out.println("Exiting HRMS. Goodbye!");
307                         break;
308                     default:
309                         System.out.println("Invalid choice. Please try again.");
310                         break;
311                 }
312
313             System.out.println();
314             }
315
316         scanner.close();
317     }
318 }
319
```

```java
285                         hrms.addLeave(leave);
```

Console

HRMS (3) [Java Application] C:\Users\ADMIN\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe (Jul 31, 2023, 9:39:22 PM) [pid: 7580]

```
----- HRMS Menu -----
1. Add Employee
2. View Employee Details
3. Update Employee Details
4. Delete Employee
5. Mark Attendance
6. Add Leave Request
7. View Leave Records
8. Approve Leave
9. Reject Leave
10. Search Employee
0. Exit HRMS
Enter your choice:
```

Conclusion: The HRMS project is a basic yet functional application that provides essential HR-related functionalities. It demonstrates core Java concepts such as classes, objects, lists, and user interaction through a console interface. The system allows HR personnel to efficiently manage employee records, attendance, and leave requests. However, for real-world application, additional features like user authentication, database integration, and error handling should be considered. Nonetheless, the project serves as a valuable learning exercise and can be expanded upon to meet specific organizational requirements.

4. Resources

Software: Eclipse IDE.

References: YouTube, Upskill Campus Videos

5. Personal Details-

Name: - Amey Rajendra Tambat

E-mail Id: - ameytambat143@gmail.com

Phone Number: - 9373911618