# Object Oriented Analysis and Design using Java (UE20CS352)

## Lab Assignment - 9 & 10

**Name : Tushar J** **Section : H**

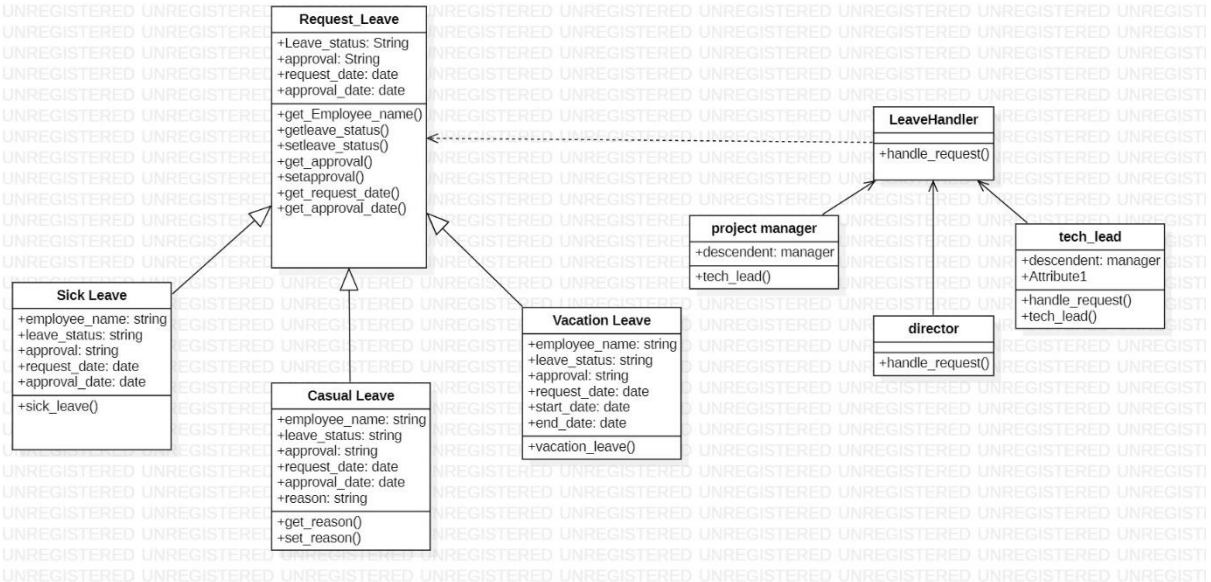
**Roll No : 10** **SRN : PES1UG20CS472**

**Date : 21-04-2023**

**Problem:-**

A Company's Leave Management System has the following features. An Employee (client) can apply for Casual Leave (CL), Sick Leave (SL) and Vacation Leave (VL). The roles in the hierarchy who are responsible for approving or rejecting the leave using the process specified are Director, Project Manager and Tech Lead. The Leave request contains the following details: empName, leaveStatus, approvedBy, requestDate and approvalDate. A CL and SL are for only one day. A VL will have a startDate and endDate. A CL will also need a reason to be specified. The Leave created by the client is assigned a "New" status. If the leave is SL, then it will be processed by Tech Lead, if it is CL, it will be processed by the Project Manager, and if it is VL, will be processed by the Director. The Leave when created is sent to Tech Lead for processing, if it is not SL, the Tech Lead will just pass the request to the next higher level. Similarly, Project Manager will process a CL request or forward the VL request to the next higher level. Once the request is processed, a message should be displayed on the console showing request details and approval details. Note: Design the application in such a way that extensibility is easy. It should be easy to add new types of Employee and new types of Leave.

**UML Class Model :-**




**Output:-**

```
PS C:\Users\Tushar Jumla\Documents\Zoom\OOAD\lab9&10\New folder>  & 'C:\Program Files\Java\jdk-19\bin\java.exe' '--en
able-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Tushar Jumla\AppData\Roaming\Code\User\worksp
aceStorage\2215386587b1556aa771ec6b7aaf3f8f\redhat.java\jdt_ws\New folder_db1198dd\bin' 'Main'
Enter name of the employee:
Tushar J
Enter leave type 1: Sick Leave, 2: Casual Leave, 3: Vacation Leave..
1
leave request got approved by Tech Lead for Tushar J
```

```
PS C:\Users\Tushar Jumla\Documents\Zoom\OOAD\lab9&10\New folder>  & 'C:\Program Files\Java\jdk-19\bin\java.exe' '--en
able-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Tushar Jumla\AppData\Roaming\Code\User\worksp
aceStorage\2215386587b1556aa771ec6b7aaf3f8f\redhat.java\jdt_ws\New folder_db1198dd\bin' 'Main'
Enter name of the employee:
Tushar J
Enter leave type 1: Sick Leave, 2: Casual Leave, 3: Vacation Leave..
2
Enter reason:
I have to take delivery of my new car.
sending leave request to Program Manager from Tech Lead
```

```
sending leave request to Director from Program Manager           & 'C:\Program Files\Java\jdk-19\bin\java.exe' '--enable-preview' '-XX:+Sh
owCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Tushar Jumla\AppData\Roaming\Code\User\workspaceStorage\2215386587b1556aa771ec6b7aaf3f8f\
redhat.java\jdt_ws\New folder_db1198dd\bin' 'Main'
Enter name of the employee:
Tushar J
Enter leave type 1: Sick Leave, 2: Casual Leave, 3: Vacation Leave..
3
Enter starting date:
2023-05-01
Enter ending date:
2023-05-06
sending leave request to Program Manager from Tech Lead
sending leave request to Director from Program Manager
leave request got approved by Director for Tushar J
PS C:\Users\Tushar Jumla\Documents\Zoom\OOAD\lab9&10\New folder>
```

Request_Leave.java :-

```java
import java.time.LocalDate;

public abstract class Request_Leave
{
   private String name_of_employee;
   private String leave_status;
   private String approval;
   private LocalDate date_of_request;
   private LocalDate date_of_approval;
   public Request_Leave(String name_of_employee, LocalDate date_of_request)
   {
      this.name_of_employee = name_of_employee;
      this.leave_status = "New";
      this.approval = null;
      this.date_of_request = date_of_request;
   }
   public String getname_of_employee()
   {
      return name_of_employee;
   }
   public String getleave_status()
   {
      return leave_status;
   }
   public void setleave_status(String leave_status)
   {
      this.leave_status = leave_status;
```

```
    }
    public String getapproval()
    {
        return approval;
    }
    public void setapproval(String approval)
    {
        this.approval = approval;
    }
    public LocalDate getdate_of_request()
    {
        return date_of_request;
    }
    public LocalDate getdate_of_approval()
    {
        return date_of_approval;
    }
    public void setdate_of_approval(LocalDate date_of_approval)
    {
        this.date_of_approval = date_of_approval;
    }
}

class sick_leave extends Request_Leave
{
    public sick_leave(String name_of_employee, LocalDate date_of_request)
    {
        super(name_of_employee, date_of_request);
    }
}

class casual_leave extends Request_Leave
{
    private String reason_description;
    public casual_leave(String name_of_employee, LocalDate date_of_request, String reason_description)
    {
        super(name_of_employee, date_of_request);
        this.reason_description = reason_description;
    }
    public String getreason_description()
    {
        return reason_description;
    }
    public void setreason_description(String reason_description)
    {
        this.reason_description = reason_description;
    }
}
```

```
class vacation_leave extends Request_Leave
{
   private LocalDate starting_date;
   private LocalDate ending_date;
   public vacation_leave(String name_of_employee, LocalDate date_of_request, LocalDate starting_date,
LocalDate ending_date)
   {
      super(name_of_employee, date_of_request);
      this.starting_date = starting_date;
      this.ending_date = ending_date;
   }
   public LocalDate getstarting_date()
   {
      return starting_date;
   }
   public LocalDate getending_date()
   {
      return ending_date;
   }
}

interface leave_manager
{
   void handleRequest(Request_Leave leave);
}

class tech_lead implements leave_manager
{
   private leave_manager descendant;
   public tech_lead(leave_manager leavemanager)
   {
      this.descendant = leavemanager;
   }
   @Override
   public void handleRequest(Request_Leave leave)
   {
      if(leave instanceof sick_leave && leave.getleave_status().equals("New"))
      {
         leave.setleave_status("approved");
         leave.setapproval("Tech Lead");
         leave.setdate_of_approval(LocalDate.now());
         System.out.println("leave request got approved by Tech Lead for " +
leave.getname_of_employee());
      }
      else
      {
         System.out.println("sending leave request to Program Manager from Tech Lead");
```

```
            this.descendant.handleRequest(leave);
        }
    }
}

class project_manager implements leave_manager
{
    private leave_manager descendant;
    public project_manager(leave_manager leavemanager)
    {
        this.descendant = leavemanager;
    }
    @Override
    public void handleRequest(Request_Leave leave)
    {
        if(leave instanceof casual_leave && leave.getleave_status().equals("New"))
        {
            leave.setleave_status("approved");
            leave.setapproval("Project Manager");
            leave.setdate_of_approval(LocalDate.now());
            System.out.println("leave request got approved by Project Manager for " +
leave.getname_of_employee());
        }
        else
        {
            System.out.println("sending leave request to Director from Program Manager");
            this.descendant.handleRequest(leave);
        }
    }
}

class director implements leave_manager
{
    public void handleRequest(Request_Leave leave)
    {
        if(leave instanceof vacation_leave  && !leave.getleave_status().equals("approved"))
        {
            leave.setleave_status("approved");
            leave.setapproval("Director");
            leave.setdate_of_approval(LocalDate.now());
            System.out.println("leave request got approved by Director for "+leave.getname_of_employee());
        }
    }
}
```

Main.java :-

```
import java.time.LocalDate;
import java.util.Scanner;

public class Main
{
   public static void main(String[] args)
   {
      Scanner in = new Scanner(System.in);
      System.out.println("Enter name of the employee:");
      String name = in.nextLine();
      System.out.println("Enter leave type 1: Sick Leave, 2: Casual Leave, 3: Vacation Leave.. ");
      int leave_type = in.nextInt();
      LocalDate start_date = null;
      LocalDate end_date = null;
      String reason = null;
      switch(leave_type)
      {
         case 1: break;
         case 2: in.nextLine();
                 System.out.println("Enter reason:");
                 reason = in.nextLine();
                 break;
         case 3: System.out.println("Enter starting date:");
                 start_date = LocalDate.parse(in.next());
                 System.out.println("Enter ending date:");
                 end_date = LocalDate.parse(in.next());
                 break;
         default: System.out.println("Invalid leave type entered");
                  return;
      }
      Request_Leave leave;
      switch(leave_type)
      {
         case 1: leave = new sick_leave(name, LocalDate.now());
                 break;
         case 2: leave = new casual_leave(name, LocalDate.now(), reason);
                 break;
         case 3: leave = new vacation_leave(name, LocalDate.now(), start_date, end_date);
                 break;
         default: return;
      }
      director director = new director();
      project_manager projectmanager = new project_manager(director);
      tech_lead techlead = new tech_lead(projectmanager);
      techlead.handleRequest(leave);
      in.close();
   }
}
```