PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Object Oriented Analysis and Design using Java (UE20CS352)

Lab Assignment-9&10: Design Patterns

Submitted by: Mahika Gupta
SRN: PES1UG20CS243

Problem Statement:

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.

Represent the design (using appropriate design patterns) in a UML Class Diagram and implement the same.

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.

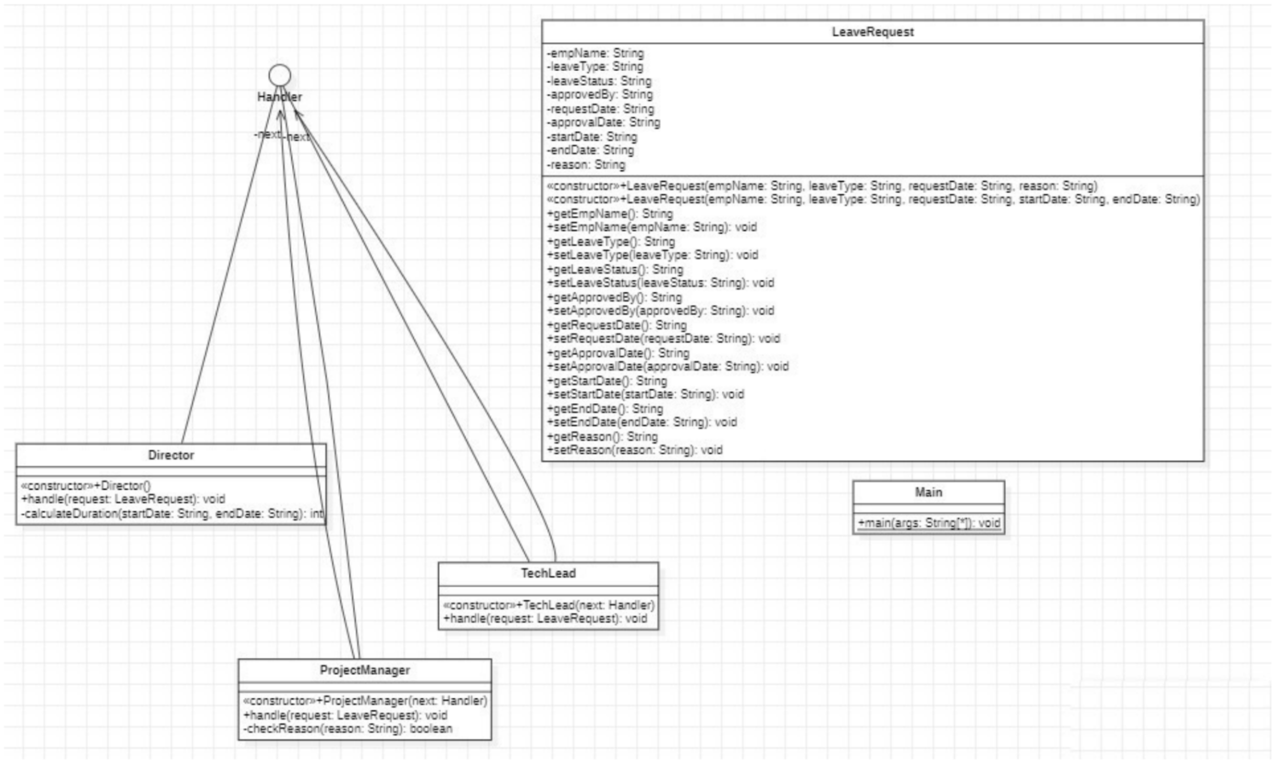
Design Patterns Considered:

1. Factory method: As different types of leaves are to be considered, as well as different types of employees (Employee requesting for leave, Director, tech lead, etc). This pattern is recommended when multiple objects can handle a request and the handler doesn't have to be a specific object. Also, the handler is determined at runtime. Please note that a request not handled at all by any handler is a valid use case.
2. Chain of Responsibility: As to get the approval for the leave, the request is passed on to different levels of authority.

Design Patterns Used:

1. Factory Method pattern is used to create different types of leaves and employees.
2. Chain of Responsibility pattern is used to process leave requests through different levels of approval.

UML Class Model:


Handler
-next
-next
LeaveRequest
-empName: String
-leaveType: String
-leaveStatus: String
-approvedBy: String
-requestDate: String
-approvalDate: String
-startDate: String
-endDate: String
-reason: String
«constructor»+LeaveRequest(empName: String, leaveType: String, requestDate: String, reason: String)
«constructor»+LeaveRequest(empName: String, leaveType: String, requestDate: String, startDate: String, endDate: String)
+getEmpName(): String
+setEmpName(empName: String): void
+getLeaveType(): String
+setLeaveType(leaveType: String): void
+getLeaveStatus(): String
+setLeaveStatus(leaveStatus: String): void
+getApprovedBy(): String
+setApprovedBy(approvedBy: String): void
+getRequestDate(): String
+setRequestDate(requestDate: String): void
+getApprovalDate(): String
+setApprovalDate(approvalDate: String): void
+getStartDate(): String
+setStartDate(startDate: String): void
+getEndDate(): String
+setEndDate(endDate: String): void
+getReason(): String
+setReason(reason: String): void
Director
«constructor»+Director()
+handle(request: LeaveRequest): void
-calculateDuration(startDate: String, endDate: String): int
Main
+main(args: String[*]): void
TechLead
«constructor»+TechLead(next: Handler)
+handle(request: LeaveRequest): void
ProjectManager
«constructor»+ProjectManager(next: Handler)
+handle(request: LeaveRequest): void
-checkReason(reason: String): boolean

CODE:

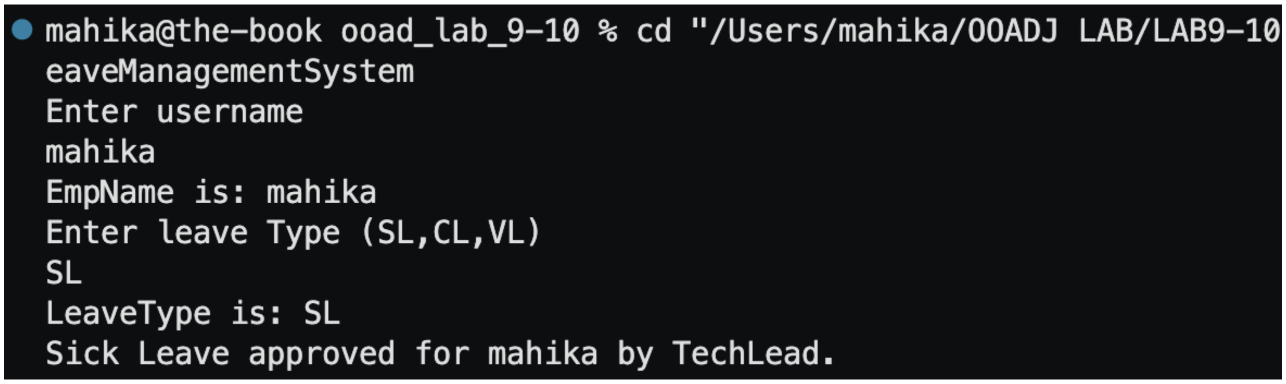
```
import java.time.LocalDate;
import java.util.Scanner;
import java.util.Date;

interface LeaveHandler {
void setNextHandler(LeaveHandler nextHandler);
void processLeave(LeaveRequest leaveRequest);
}

class LeaveRequest {
String empName;
String leaveType;
String leaveStatus;
String approvedBy;
LocalDate requestDate;
LocalDate approvalDate;
String reason;
LocalDate startDate;
LocalDate endDate;
public LeaveRequest(String empName, String leaveType, LocalDate requestDate, String
reason, LocalDate startDate, LocalDate endDate) {
this.empName = empName;
this.leaveType = leaveType;
this.requestDate = requestDate;
this.reason = reason;
this.startDate = startDate;
this.endDate = endDate;
this.leaveStatus = "New";
}
}

class TechLead implements LeaveHandler {

LeaveHandler nextHandler;
public void setNextHandler(LeaveHandler nextHandler) {
```

```
this.nextHandler = nextHandler;
}

public void processLeave(LeaveRequest leaveRequest) {
if (leaveRequest.leaveType.equalsIgnoreCase("SL")) {
leaveRequest.leaveStatus = "Approved";
leaveRequest.approvedBy = "Tech Lead";
leaveRequest.approvalDate = LocalDate.now();
System.out.println("Sick Leave approved for " + leaveRequest.empName+" by TechLead.");
}
else {
if (nextHandler != null) {
nextHandler.processLeave(leaveRequest);
}
}
}
}

class ProjectManager implements LeaveHandler {
LeaveHandler nextHandler;
public void setNextHandler(LeaveHandler nextHandler) {
this.nextHandler = nextHandler;
}

public void processLeave(LeaveRequest leaveRequest) {
if (leaveRequest.leaveType.equalsIgnoreCase("CL")) {
leaveRequest.leaveStatus = "Approved";
leaveRequest.approvedBy = "Project Manager";
leaveRequest.approvalDate = LocalDate.now();
System.out.println("Casual Leave approved for "+leaveRequest.empName+" by Project
Manager.");
}
else {
if (nextHandler != null) {
nextHandler.processLeave(leaveRequest);
}
}
}
}
```

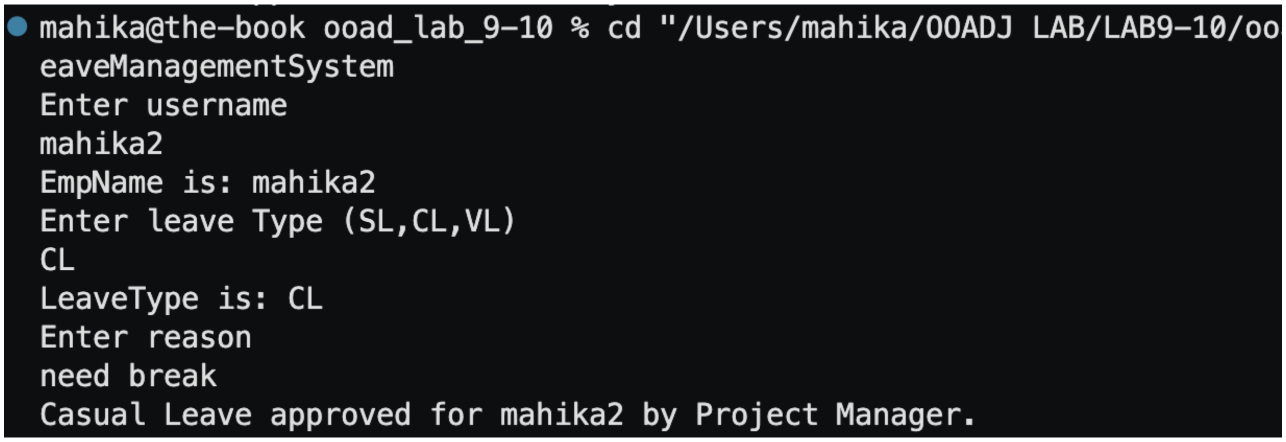
```
class Director implements LeaveHandler {
LeaveHandler nextHandler;
public void setNextHandler(LeaveHandler nextHandler) {
this.nextHandler = nextHandler;
}

public void processLeave(LeaveRequest leaveRequest) {
if (leaveRequest.leaveType.equalsIgnoreCase("VL")) {
leaveRequest.leaveStatus = "Approved";
leaveRequest.approvedBy = "Director";
leaveRequest.approvalDate = LocalDate.now();
System.out.println("Vacation Leave approved for "+leaveRequest.empName+" by Director.");
}
else {
if (nextHandler != null) {
nextHandler.processLeave(leaveRequest);
}
}
}
}

public class LeaveManagementSystem {

public static void main(String[] args) {
TechLead techLead = new TechLead();
ProjectManager projectManager = new ProjectManager();
Director director = new Director();
techLead.setNextHandler(projectManager);
projectManager.setNextHandler(director);
boolean flag=true;
while(flag)
{
Scanner myObj = new Scanner(System.in); // Create a Scanner object
System.out.println("Enter username");
String empName = myObj.nextLine(); // Read user input
System.out.println("EmpName is: " + empName);
System.out.println("Enter leave Type (SL,CL,VL)");
String leaveType = myObj.nextLine();
System.out.println("LeaveType is: " + leaveType);
if(leaveType.equals("SL")||leaveType.equals("CL")||leaveType.equals("VL")){
```

```java
switch(leaveType){
case "SL":{
LeaveRequest sickLeave = new LeaveRequest(empName,leaveType,LocalDate.now(), null, null,
null);
techLead.processLeave(sickLeave);
flag= false;
break;
}
case "CL":{
System.out.println("Enter reason");
String reason = myObj.nextLine();

LeaveRequest casualLeave = new LeaveRequest(empName, leaveType,
LocalDate.now(), reason ,null, null);
techLead.processLeave(casualLeave);
flag= false;
break;
}
case "VL":{

try{
System.out.println("Enter start date for leave in yyyy-mm-dd format");
String startDate = myObj.nextLine();
LocalDate StartDate = LocalDate.parse(startDate);
if(StartDate.compareTo(LocalDate.now()) < 0) {
System.out.println("Invalid start date");
break;
}
System.out.println("Enter end date for leave in yyyy-mm-dd format");
String endDate = myObj.nextLine();
LocalDate EndDate = LocalDate.parse(endDate);

if(EndDate.compareTo(StartDate) < 0 || EndDate.compareTo(LocalDate.now())< 0 ){
System.out.println("Invalid end date");
break;
}
LeaveRequest vacationLeave = new LeaveRequest(empName, leaveType,LocalDate.now(), null,
StartDate, EndDate);
techLead.processLeave(vacationLeave);
flag= false;
break;
}
catch(Exception DateTimeException){
System.out.println("Wrong date format, please enter again");
break;
}
}
}
}
else{
System.out.println("Wrong type of leave");
}
}
}
}
```

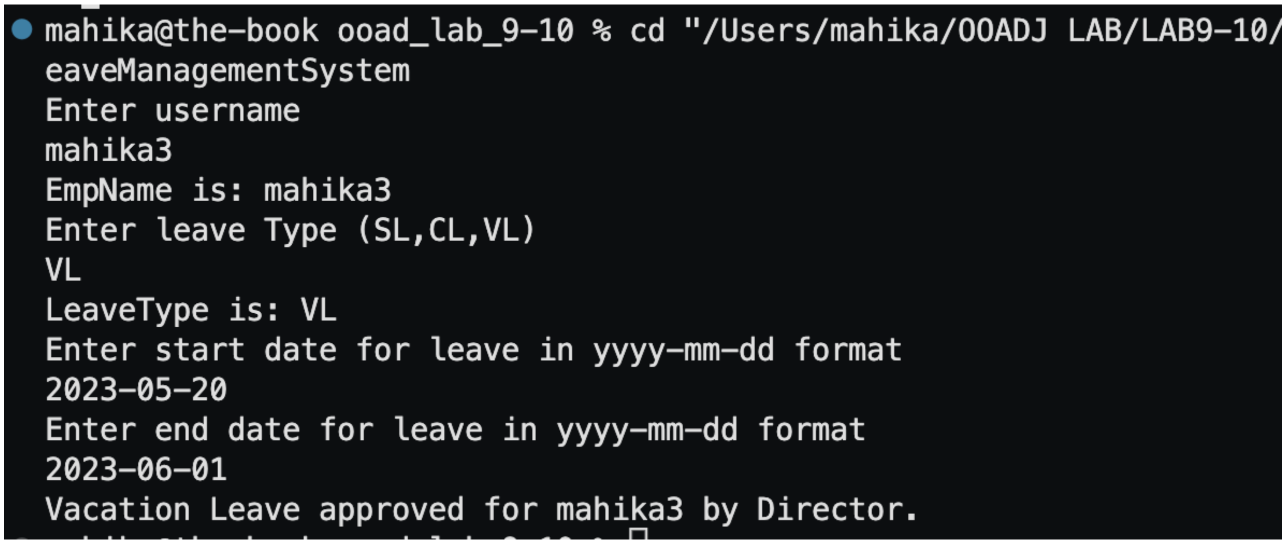
Screenshots:



```
mahika@the-book ood_lab_9-10 % cd "/Users/mahika/OOADJ LAB/LAB9-10
eaveManagementSystem
Enter username
mahika
EmpName is: mahika
Enter leave Type (SL,CL,VL)
SL
LeaveType is: SL
Sick Leave approved for mahika by TechLead.
```



```
mahika@the-book ood_lab_9-10 % cd "/Users/mahika/OOADJ LAB/LAB9-10/oo
eaveManagementSystem
Enter username
mahika2
EmpName is: mahika2
Enter leave Type (SL,CL,VL)
CL
LeaveType is: CL
Enter reason
need break
Casual Leave approved for mahika2 by Project Manager.
```



```
● mahika@the-book ooad_lab_9-10 % cd "/Users/mahika/OOADJ LAB/LAB9-10/
  eaveManagementSystem
  Enter username
  mahika3
  EmpName is: mahika3
  Enter leave Type (SL,CL,VL)
  VL
  LeaveType is: VL
  Enter start date for leave in yyyy-mm-dd format
  2023-05-20
  Enter end date for leave in yyyy-mm-dd format
  2023-06-01
  Vacation Leave approved for mahika3 by Director.
```