Course Project Instructions

Leave Application Processing System (LAPS)

The Application

You have been tasked by NUS-ISS to develop a '**Leave Application Processing System' (LAPS).** Employees of NUS-ISS will be able to access this system from NUS-ISS intranet web site. The system will accommodate three types of employee roles, namely: Administrators, Managers and Employees.

- **Employees** are enabled to apply/cancel/update their leave.
- **Managers** are responsible for leave approval/rejection process. Managers can also print consolidated leave reports.
- **Administrators** are responsible for creating, managing users and respective roles. They are also responsible for managing the approval hierarchy.

There are *three categories* of leave that an employee is entitled to claim. They are annual leave, medical leave, and compensation leave. An employee has to take full day leave for all entitlement except compensation leave. For compensation leave, the granularity is half a day. Every four hours of overtime work makes an employee eligible for half-a day compensation.

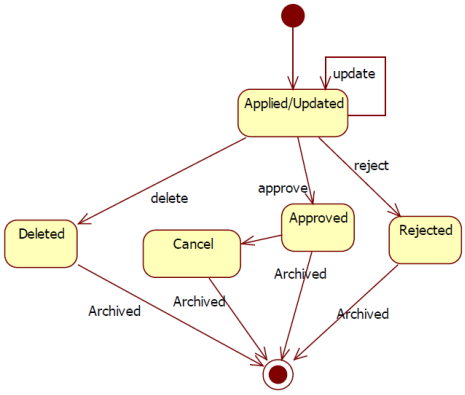
The team must design, develop, test and release a web-based application, using a standard RDBMS such as MySQL. The team should use a model-view-controller architecture using Spring MVC Framework. A reasonable business layer is expected to validate leave type, claim dates, eligibility and approval processes. The persistent data layer can be implemented using Spring Data JPA ORM framework.

The features of the system are listed in two groups. Mandatory features must be implemented. A good implementation of all Mandatory features alone will earn the team a pass. To maximize the marks attained, you should implement some of the Optional features described below. Optional features are not listed in any particular order, and you should prioritize them as you see fit.

It is important to note that implementing features (optional as well as mandatory) badly can actually make you lose marks rather than gain them. It is important you give the appropriate amount of thought to the design of each feature you implement. The functionality of your system should be potentially usable in a real-life site. **Usability and robustness should be high in your list of priorities.**

Use Case Model

Leave Application Status – State Transition Diagram



Mandatory Features

Supported Leave Types

Your system must support three types of leave: Annual, Medical and Compensation.

Login

The application will have one entry point (URL) for all Employees, Managers and Admins. All access to the application will be password-protected, so that user identity can be established. User ids and passwords will be stored in your application database.

For this assignment, you may assume that proper authorization will be added later stage. Therefore, you only need to perform authentication from the main access page, and need not secure access to operations, nor verify authorization for every Web Page Access. This means that it is acceptable that a user is able to bypass the system security by manipulating URLs.

You can implement the authorization process and claim it as one of your optional features.

Leave Application Submission

When an employee (who has already logged in) wishes to submit a leave application, he/she is presented with a leave application form. The Employee fills-in the leave details and submits the form. The system performs validation on the leave details before processing. Details that the form captures minimally include leave period, category of leave, additional reasons, work dissemination (if any) and contact details (if on an overseas trip). The status of application is assigned '***Applied***'.

The Employee could decide to update or delete the leave application before the leave is approved or rejected. In such cases the status is changed to '***Updated'/'Deleted'***. The Employee may also cancel an 'Approved' leave, in which case the status is changed to '***Cancelled'.***

The conditions to be checked by application before applying leave includes but not limited to the following:

- Leave period, reason and leave type are mandatory details.
- Dates *From* and *To* in leave period should be in chronologically increasing order.
- Annual leave computation
    - If the leave period is <= 14 calendar days, weekends / public holidays are excluded.
    - Otherwise, weekends / public holidays are included.
    - Each employee is entitled certain number of annual leave based on their designation. For example, administrative employees receive 14 days annual leave, professional employees receive 18 days annual leave in each calendar year.
    - *From* and *To* dates must be working days.
- Medical Leave is limited to 60 days in a calendar year.

View Personal Leave History and View Leave Application

An Employee can view all previous leave applications raised by him/her by clicking 'View Personal Leave History'. This page lists leave records pertaining to only the current year. All leave records are shown in tabular form with hyperlinks for detailed view. Upon clicking the individual leave

hyperlink, employee can view the complete leave details. He/she can then further update, cancel or delete the application.

View Leave Application for Approval and Subordinate Leave History

Managers can view leave applications raised by his/her subordinates by clicking 'View Leave Application for Approval'. This page shows applications grouped by subordinate name. All leave records are shown in tabular form with hyperlinks for detailed view. Upon clicking the individual leave hyperlink, the manager can view complete leave details.

He/she can then click **Approve** or **Reject** accordingly along with a Comment. If the leave request is rejected, the manager adds a mandatory comment explaining the decision. To help the manager decide on approval, leave records of his/her subordinates during the leave period are listed. The leave records are assigned '**Approved'** and '**Rejected'** status respectively.

Managers can also view complete leave history of his/her subordinates by clicking 'Subordinate Leave History'. This works similar to 'View Personal Leave History'.

Administration

An Admin must be able to create, modify and delete Employee. He/she must be able to assign, modify roles (employee, admin and manager) of each employee. The admin user will maintain the leave types mentioned earlier and calendar of public holidays for the current year. In addition, admins must be able to enter/update Employee's annual leave entitlements for the current year.

REST Controller and REST Repository

The team can choose any use case and implement that via **REST Controllers** and **REST enabled Data exchange** on the server side. Also, implement a client-side technology to consume the same API End Points.

Scope

The minimum number of use cases (or features) per team is ***x*** plus login and logout features, where **x = team size * 2**. For example, the minimum number of features is 10 for a team of 5 members, plus login and logout functions. The selected features must include significant functions for the leave application.

Optional Features (for additional marks)

Compensation Leave Management

Employees are allowed to claim compensation for overtime work, in a granularity of half a day. This claim is subject to manager's approval. Each employee has an associated ledger to keep track of compensation leave and respective claim day. When the employee claims to use his compensation leave, it is again approved/rejected by the managers.

Reporting

Managers must be able to produce various reporting views such as

- Employee on annual/compensation/medical/all leave during a selected period.
- Compensation claims for all/particular employee.
- Managers must be able to export the above reports to comma-delimited (CSV) file format.

Movement Register

This is a menu available for all users and upon clicking the hyperlink, the system displays details of all employees on leave during the current month and the category of leave. Users can also navigate to the previous and next month using a dropdown choice list.

Pagination

If many leave results are returned (say more than 30 records), the system should present them over several pages. Each page will contain navigation facilities so that other pages of search results can be shown (similar to search engine results). The number of results per page should ideally be selectable by the user (e.g. 10, 20, 25).

Email interaction

When the employee applies for leave, system should send a notification email to his/her manager. Also, the employee is informed of results on both accept and reject cases. The email message should contain a direct link to the login page where the user can view the comments.

ReactJS client

Implement one or two use cases with ReactJS app, consuming the REST APIs.

Secure the application using Spring Security

Any other interesting extensions the team can think of creatively would be highly appreciated

Software development tools

Technology	Tool Name	Details
Java SE	JDK	You can download a copy online
Spring Framework	STS	You can download a copy online
ReactJS	Visual Studio Code	You can download a copy online
Database	JPA	Preferably MySQL Database

Design Consideration

The following are the system design consideration.

- The systems are developed using Spring components.
- The system uses Database to store the data. Do design appropriate data scripts and also populate the database with sufficient test data.

Evaluation Criteria

Everybody contributes and no excuses. **You will be assessed both in team and individually.**

The team will be evaluated on the **technical aspects** and **user-friendly aspects** of the application deliverable. Beyond the working solution, some expectations would be:

- Implementation **best practices**; for example, object encapsulation, layering of architecture etc.
- Proper **exception handling** implementations, server validation logic, test cases and utility classes.
- User Interface (**UI**) and User Experience (**UX**).
- **Quality** of the **presentation**, including presentation scripts, timing, clarity, etc.

Deliverables

This assignment is part of the continuous assessment for this course. You will be evaluated for **30 marks** on the whole.

You will work in your team. No individual work will be accepted.

The followings are the deliverables:

- **A video**, maximum 30 mins, including:
  - Demonstrating how the users use your system.
  - Showing the functionalities and important business logics **working properly.**
  - Explanation of **ER Diagram** and **highlighting** of **design choices** if any.
  - **Highlighting** the main / **interesting designs** and **codes** of your teamwork.
- A simple 6 pointer **presentation slides**, mentioned at the **end** of the video, covering the following details is appreciated:
  - Your team number and members (title slide).
  - Explanation of how the Team carried work distribution.

    - Explanation of Architecture/Layer Description/Any other additional dependencies used.
    - Explanation of Class Diagram or ER Diagram.
    - Explanation of Technologies Used.
    - Summary of lessons Learnt.
- Your project **source codes:**
    - Java codes.
    - React.JS codes (if any).
    - A readme.txt, instructing how to run your project **locally**.
- You will complete a **Peer evaluation form** during the exam study period.

Submission

File and Location

You should zip everything into a single file, named ***Java_CA_Team_<xx>.zip*** and upload it into the submission folder in Canvas. If the video is too big, you should reduce its size accordingly.

Schedule

FINAL SUBMISSION: 5pm Friday 26rd December 2023.

Plagiarism Notice*: All students share the responsibility for upholding the academic standards and reputation of the University. Academic honesty is a prerequisite condition in the pursuit and acquisition of knowledge. Academic dishonesty is any misrepresentation with the intent to deceive or failure to acknowledge the source or falsification of information or inaccuracy of statements or cheating at examinations/tests or inappropriate use of resources. Please avoid sharing code, you will be penalized if found guilty.*

COOPERATIVE TEAM IS LIKELY SUCCESSFUL!!!

GOOD LUCK!