# System Requirements Specification

## Index

For

# My Time Away

Version 1.0

# TABLE OF CONTENTS

# MY TIME AWAY

## System Requirements Specification

**You need to consume APIs exposed by Backend application in Angular to make application work as FULLSTACK**

## BACKEND-SPRING BOOT RESTFUL APPLICATION

## 1 PROJECT ABSTRACT

**My Time Away** is a FullStack Application with a backend implemented using Spring Boot with a MySQL database and a frontend developed using Angular. It enables users to manage various aspects and streamline the process of managing employee leaves within an organization. The system facilitates efficient management of employee leave requests and approvals, ensuring proper allocation of resources and smooth workflow management.

**Following is the requirement specifications**:

| | My Time Away |
|---|---|
| | |
| Modules | |
| 1 | MyTimeAway |
| | |
| Event Module Functionalities | |
| | |
| 1 | Create an Leave |
| 2 | Update the existing Leave details |
| 3 | Get the Leave by Id |
| 4 | Get all Leaves |
| 5 | Delete a Leave |
| 6 | Search for Leave by Id, Name,and Total Days |
| 7 | Cancel a Leave by Id |
| 8 | Approve a Leave by Id |
| 9 | Reject Leave request by Id |

# 2 Assumptions, Dependencies, Risks / Constraints

## 2.1 Leave Constraints

- When fetching a Leave by ID, if the leave ID does not exist, the operation should throw a custom exception.
- When updating an Leave, if the leave ID does not exist, the operation should throw a custom exception.
- When removing an Leave, if the leave ID does not exist, the operation should throw a custom exception.
- When canceling a Leave, the operation should throw a custom exception.
- When approving a Leave, the operation should throw a custom exception.
- When rejecting a Leave, the operation should throw a custom exception.
- When changing a Leave status, if the leave ID does not exist, the operation should throw a custom exception.

# Common Constraints

- For all rest endpoints receiving @RequestBody, validation check must be done and must throw custom exception if data is invalid
- All the business validations must be implemented in dto classes only.
- All the database operations must be implemented on entity object only
- Do not change, add, remove any existing methods in service layer
- In Repository interfaces, custom methods can be added as per requirements.
- All RestEndpoint methods and Exception Handlers must return data wrapped in **ResponseEntity**

# 3 Business Validations

- Name of the employee is not null.
- Phone number of the employee is not null.
- Email of the employee is not null.
- Email of the employee's manager is not null.
- From_date is not null.
- To_date is not null.
- Total_days is not null.
- Reason for taking leave is not null.
- IsProcessed is optional
- Status is optional

# 4 REST ENDPOINTS

Rest End-points to be exposed in the controller along with method details for the same to be created. Please note, that these all are required to be implemented.

## 4.1 EMPLOYEELEAVE CONTROLLER

| URL Exposed | Purpose |
|---|---|
| 1. /api/leaves<br>Http Method: GET<br>Parameter: -<br>Return: List<Leaves> | Fetches all the leaves |
| 2. /api/leaves<br>Http Method: POST<br>Parameter 1: Leave<br>Return: Leave | Add a new leave |
| 3. /api/leaves/{id}<br>Http Method: DELETE<br>Parameter 1: Long (id)<br>Return: - | Delete leaves with given leave id |
| 4. /api/leaves/{id}<br>Http Method: GET<br>Parameter 1: Long (id)<br>Return: Leave | Fetches the leave with the given id |
| 5. /api/leaves/{id}<br>Http Method: PUT<br>Parameter 1: Long (id)<br>Parameter 2: Leave<br>Return: Leave | Updates existing leave |
| 6. /api/leaves/search?employeeId={id}&employeeName={name}&totalDays={totalDays}<br>Http Method: GET<br>Parameter 1: String (id)<br>Parameter 2: String (name)<br>Parameter 3: Int (totaldays)<br>Return: List<Leaves> | Fetches the leave with the given id, name, total days |

| | | |
|---|---|---|
| 7. /api/leaves/{id}/cancel<br>Http Method: PUT<br>Parameter 1: Long (id)<br>Return: - | | Cancel the leave request |
| 8. /api/leaves/{id}/approve<br>Http Method: PUT<br>Parameter 1: Long (id)<br>Return: - | | Approve the leave request |

| | | |
|---|---|---|
| 9. /api/leaves/{id}/reject<br>Http Method: PUT<br>Parameter 1: Long (id)<br>Return: - | | Reject the leave request |

| | | |
|---|---|---|
| 10. /api/leaves/send-email/{email}<br>Http Method: GET<br>Parameter 1: String (email)<br>Return: String (confirmation or error message) | | Send the notification about leave on shared email |

# 5 TEMPLATE CODE STRUCTURE

## 5.1 PACKAGE: COM.MYTIMEAWAY

**Resources**

| | | |
|---|---|---|
| **MyTimeAwayApplication (Class)** | This is the Spring Boot starter class of the application. | Already Implemented |

| | | |
|---|---|---|
| **MailConfig (Class)** | This is the Spring Boot bean to configure the details for sending email. | Partially Implemented |

## 5.2 PACKAGE: COM.MYTIMEAWAY.REPOSITORY

**Resources**

| Class/Interface | Description | Status |
|---|---|---|

| | | |
|---|---|---|
| **EmployeeLeaveRepository (interface)** | • Repository interface exposing CRUD functionality for EmployeeLeave Entity.<br>• You can go ahead and add any custom methods as per requirements. | Partially implemented. |

## 5.3 PACKAGE: COM.MYTIMEAWAY.SERVICE

**Resources**

| Class/Interface | Description | Status |
|---|---|---|
| **EmployeeLeaveService (interface)** | • Interface to expose method signatures for employee leave related functionality.<br>• Do not modify, add or delete any method. | Already implemented. |

## 5.4 PACKAGE: COM.MYTIMEAWAY.SERVICE.IMPL

**Resources**

| Class/Interface | Description | Status |
|---|---|---|
| **EmployeeLeaveServiceImpl (class)** | • Implements EmployeeLeaveService.<br>• Contains template method implementation.<br>• Need to provide implementation for employee leave related functionalities. | To be implemented. |

| | | |
|---|---|---|
| | • Do not modify, add or delete any method signature | |

## 5.5 PACKAGE: COM.MYTIMEAWAY.CONTROLLER

**Resources**

| Class/Interface | Description | Status |
|---|---|---|
| **EmployeeLeaveController (Class)** | • Controller class to expose all rest-endpoints for employee leave related activities.<br>• May also contain local exception handler methods | To be implemented |

## 5.6 PACKAGE: COM.MYTIMEAWAY.DTO

**Resources**

| Class/Interface | Description | Status |
|---|---|---|
| **EmployeeLeaveDTO (Class)** | Use appropriate annotations from the Java Bean Validation API for validating attributes of this class. | Partially implemented. |

## 5.7 PACKAGE: COM.MYTIMEAWAY.ENTITY

**Resources**

| Class/Interface | Description | Status |
|---|---|---|

| | | |
|---|---|---|
| **EmployeeLeave (Class)** | • This class is partially implemented.<br>• Annotate this class with proper annotation to declare it as an entity class with **employeeId** as primary key.<br>• Map this class with an **employeeleave table**.<br>• Generate the **employeeId** using the IDENTITY strategy | Partially implemented. |

## 5.8 PACKAGE: COM.MYTIMEAWAY.EXCEPTION

**Resources**

| Class/Interface | Description | Status |
|---|---|---|
| **ApplicationNotFoundException (Class)** | • Custom Exception to be thrown when trying to fetch or delete the leave info which does not exist.<br>• Need to create Exception Handler for same wherever needed (local or global) | Already implemented. |
| **CustomExceptionHandler (Class)** | • RestControllerAdvice Class for defining global exception handlers.<br>• Contains Exception Handler for **InvalidDataException** class.<br>• Use this as a reference for creating exception handler for other custom | Already implemented. |

| | | |
|---|---|---|
| | exception classes. | |

# 6 CONSIDERATIONS

A. There is no roles in this application
B. You can perform the following possible action

| EmployeeLeave |
|---|

# FRONTEND-ANGULAR SPA

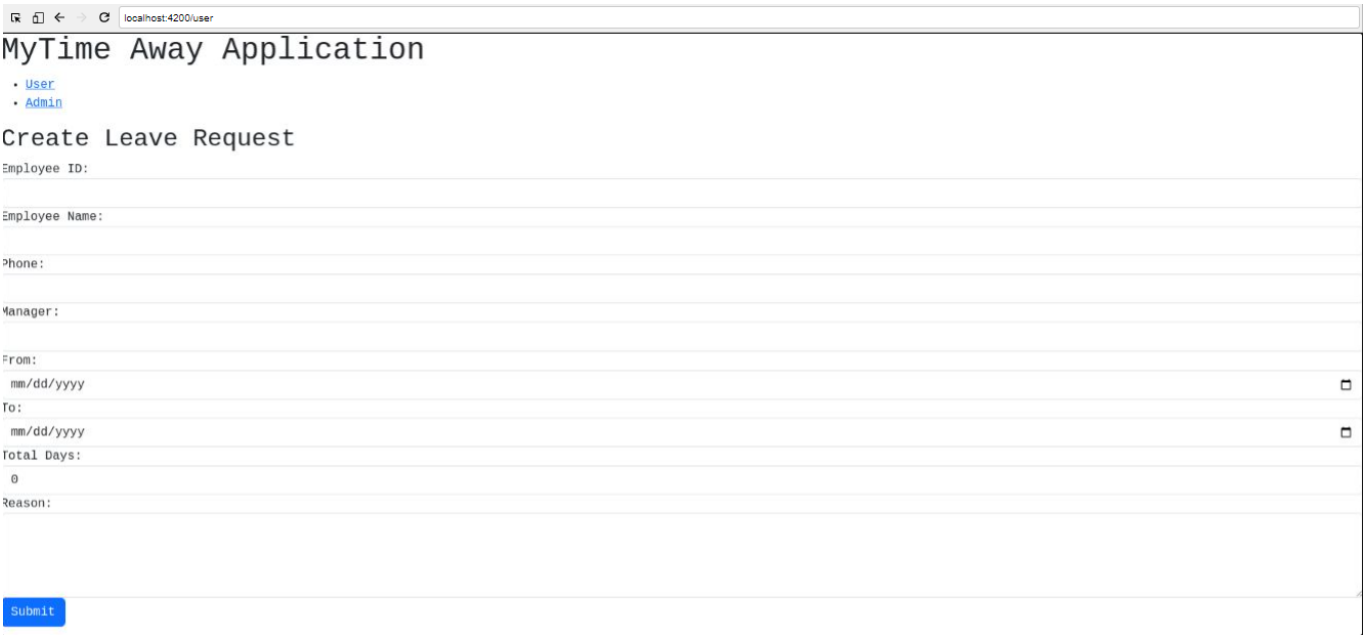
## 1 PROBLEM STATEMENT

My Time Away is SPA (Single Page Application), it enables users to manage various aspects and streamline the process of managing employee leaves within an organization like it allows to add leave, update leave, delete leave, get leave by id, get all leave, search leave by id, name and total days, cancel leave request, approve leave request and reject leave request.

## 2 PROPOSED MY TIME AWAY WIREFRAME

UI needs improvisation and modification as per given use case and to make test cases passed.

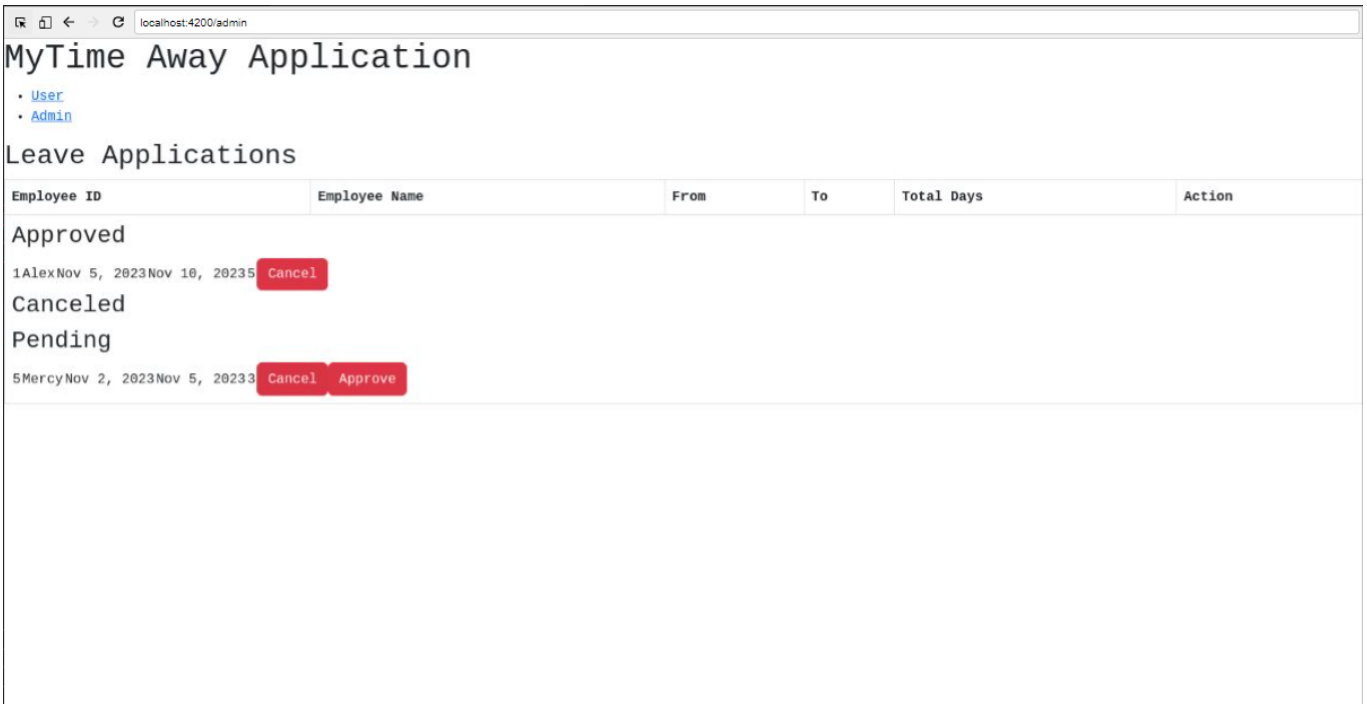
### 2.1 HOME PAGE


localhost:4200/user
MyTime Away Application
User
Admin
Create Leave Request
Employee ID:
Employee Name:
Phone:
Manager:
From:
mm/dd/yyyy
To:
mm/dd/yyyy
Total Days:
0
Reason:
Submit

## 2.2 USER PAGE


localhost:4200/user
MyTime Away Application
User
Admin
Create Leave Request
Employee ID:
524
Employee Name:
John
Phone:
9587412354
Manager:
Navin
From:
10/28/2023
To:
10/30/2023
Total Days:
2
Reason:
Sick Leave
Submit

## 2.3 ADMIN PAGE




# 3 BUSINESS-REQUIREMENT:

As an application developer, develop the Social Networking App (Single Page App) with below guidelines:

| User Story # | User Story Name | User Story |
|---|---|---|
| US_01 | Home Page | As a user I should be able to visit the Home page as the default page. |

| | | |
|---|---|---|
| US_01 | User Page | As a user I should be able to see the homepage and perform all operations:<br><br>Acceptance criteria:<br><br>1. As a user I should be able to furnish the following details at the time of creating a leave application.<br>1.1 Employee ID<br>1.2 Employee Name<br>1.3 Phone Number<br>1.4 Manager Name<br>1.5 From Date<br>1.6 To Date<br>1.7 Total Days<br>1.8 Reason<br>2. The Update button should be disabled by default, and should be enabled when you click on the Edit button.<br>3. Employee ID is not null.<br>4. Name of the employee is not null.<br>5. Phone number of the employee is not null.<br>6. Manager Name of the employee is not null.<br>7. From Date is not null.<br>8. To Date is not null.<br>9. Total Days is not null<br>10. Reason is not null<br>11. All the mentioned fields are mandatory. If any constraint is not satisfied, a validation message must be shown. |

| | | |
|---|---|---|
| US_02 | Admin Page | As a admin I should be able to see the admin and perform all operations:<br><br>Acceptance criteria:<br><br>12. As an admin I should be able to furnish the following details at the time of creating a leave application.<br>13. In Pending field you can see the employee leave request with details you can do two following actions:<br>1.1 Approve<br>1.1 Cancel<br>14. Approve Button can approve the leave requested by the employee.<br>15. Cancel Button can cancel the leave requested by the employee. |

# 4 EXECUTION STEPS TO FOLLOW FOR BACKEND

1. **All actions like build, compile, running application, running test cases will be through**

Command Terminal.

2. To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. cd into your backend project folder
4. To build your project use command:

   mvn clean package -Dmaven.test.skip
5. To launch your application, move into the target folder (cd target). Run the following command to run the application:

   java -jar <your application jar file name>
6. This editor Auto Saves the code.
7. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
8. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
9. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.
10. To test any UI based application the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.
11. Default credentials for MySQL:
    a. Username: root
    b. Password: pass@word1
11. To login to mysql instance: Open new terminal and use following command:
    a. sudo systemctl enable mysql
    b. sudo systemctl start mysql
    c. mysql -u root -p

    The last command will ask for password which is 'pass@word1'
12. Mandatory: Before final submission run the following command:

    mvn test

13. You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

# 1 EXECUTION STEPS TO FOLLOW FOR FRONTEND

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to**

   **Application menu (Three horizontal lines at left top) -> Terminal ->New Terminal.**
3. **This is a web-based application, to run the application on a browser, use the internal browser in the environment.**
4. **You can follow series of command to setup Angular environment once you are in your project-name folder:**
   a. **npm install -> Will install all dependencies -> takes 10 to 15 min**
   b. **npm run start -> To compile and deploy the project in browser. You can press <Ctrl> key while clicking on localhost:4200 to open project in browser -> takes 2 to 3 min**
   c. **npm run test -> to run all test cases. It is mandatory to run this command before submission of workspace -> takes 5 to 6 min**
5. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**