System Requirements Specification Index

For

Notes-App

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Table of Contents

NOTES APPLICATION

System Requirements Specification

1 Project Abstract

Note App is Spring boot application with MySQL, where it allows any unregistered users (visitors) to manage the notes like create, view, modify and delete.

Visitors can perform the follow actions:

1. Allows to add a note
2. Allows to delete an existing note
3. Allows to update an existing note
4. Allows to search any note based on the id.
5. Allows to display all the notes

2 Assumptions, Dependencies, Risks / Constraints

- While fetching the note by ID, if note id does not exist then the operation should throw an exception.
- While deleting the note by ID, if note id does not exist then the operation should throw an exception.
- While updating the status of note, if note id does not exist then the operation should throw an exception.
- 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.
- Must not go and touch the test resources, as they will be used for Auto-Evaluation
- All RestEndpoint methods and Exception Handlers must return data wrapped in **ResponseEntity**

3 Rest Endpoints

Rest End-points to be exposed in the controller along with method details for the same to be created

3.1 NoteController

<table>
<tr><th>URL Exposed</th><th>Purpose</th></tr>
<tr><td>/noteservice/all<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>-</td></tr><tr><td>Return</td><td>List<Note></td></tr></table></td><td>Fetches all the notes</td></tr>
<tr><td>/noteservice/add<table><tr><td>Http Method</td><td>POST</td></tr><tr><td>Parameter 1</td><td>Note</td></tr><tr><td>Return</td><td>Note</td></tr></table></td><td>Add a new note</td></tr>
<tr><td>/noteservice/delete/{id}<table><tr><td>Http Method</td><td>DELETE</td></tr><tr><td>Parameter 1</td><td>Integer (id)</td></tr><tr><td>Return</td><td>Note</td></tr></table></td><td>Delete note with given note id</td></tr>
<tr><td>/noteservice/get/{id}<table><tr><td>Http Method</td><td>GET</td></tr><tr><td>Parameter 1</td><td>Integer (id)</td></tr><tr><td>Return</td><td>Note</td></tr></table></td><td>Fetches the note with the given id</td></tr>
<tr><td>/noteservice/update<table><tr><td>Http Method</td><td>PUT</td></tr><tr><td>Parameter 1</td><td>Note</td></tr><tr><td>Return</td><td>Note</td></tr></table></td><td>Updates existing note</td></tr>
</table>

4 Template Code Structure

4.1 Package: com.yaksha.assessments.notesservice

Resources

NotesserviceApplication (Class)	This is the SpringBoot starter class of the application.	Already Implemented

4.2 Package: com.yaksha.assessments.notesservice.model

Resources

Class/Interface	Description	Status
Note (class)	o Annotate this class with proper annotation to declare it as an entity class with **Id** as primary key. o Map this class with **note** table. o Generate the **Id** using the **IDENTITY** strategy	Partially implemented.

4.3 Package: com.yaksha.assessments.notesservice.repository

Resources

Class/Interface	Description	Status
NoteRepository (interface)	1. Repository interface exposing CRUD functionality for **Note** Entity. 2. You can go ahead and add any custom methods as per requirements	Partially implemented

4.4 Package: com.yaksha.assessments.notesservice.service

Resources

Class/Interface	Description	Status
NoteService (interface)	Interface to expose method signatures for note related functionality. Do not modify, add or delete any method	Already implemented.

NoteServiceImpl (class)	• Implements **NoteService**. Contains template method implementation. • Need to provide implementation for note related functionalities • Do not modify, add or delete any method signature	To be implemented.

4.5 Package: com.yaksha.assessments.notesservice.controller

Resources

Class/Interface	Description	Status
NoteController (Class)	• Controller class to expose all rest-endpoints for note related activities. • May also contain local exception handler methods	To be implemented

5 Execution Steps to Follow

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. To build your project and run test cases use command:
   mvn clean package
4. To launch your application, move into the target folder (cd target). Run the following command to run the application:
   java -jar noteservice-0.0.1-SNAPSHOT.jar
5. This editor Auto Saves the code
6. 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.
7. 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.
8. 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.
9. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.
   Note: The application will not run in the local browser
10. 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.**