IIHT

Time To Complete: 10 to 12 hr

Library Management System

Contents

[1 Problem Statement 2](#_Toc76387085)

[2 Proposed Library Application Wireframe 2](#_Toc76387086)

[3 Tool Chain 3](#_Toc76387087)

[4 Business Requirements: 4](#_Toc76387088)

# Problem Statement

A corporate organization has planned to setup a portal for managing library external projects. Project/library managers can create/update/delete books. This simple application will be used to track the projects the project manager is handling.

Software Checklist

• Spring Tool Suite 4 (Version: 4.10.0.RELEASE) or

• Eclipse Maven plugin (m2e Maven Integration for Eclipse, version: 1.17.2.20210219-1922, Feature ID: org.eclipse.m2e.feature)

• JDK 1.8 (Oracle or Opened)

# Tool Chain

|  |  |  |
| --- | --- | --- |
| Competency | Skill | Skill Detail |
| Engineering Mindset | Code Quality |  |
|  |  |  |
| Programming Languages | Application Language | Java |
|  | Compute & Integration | Spring |
|  |  |  |
|  | Governance & Tooling | JUnit |
|  |  | Mockito |
|  |  | Jasmine |

# Business Requirements:

## 

|  |  |
| --- | --- |
| Screen Name | Console input screen |
| Problem Statement | 1. User needs to enter in to the application. 2. User needs to provide valid credentials, otherwise the system should show an error message: 3. After entering the details , the user should be able to do the particular operations 4. The console show display the menu   i)add a subject  ii)add a book  iii)delete a subject  iv)delete a book  v)search a subject  vi)search a book  vii)exit |
| Design Consideration | 1. Use Spring security for the login feature (provided) 2. Create multiple user profiles with the roles USER and ADMIN and use in-memory authentication (provided) 3. Allow application access for USER and ADMIN roles only (provided) 4. Spring JPA with hibernate for database operation 5. Spring AOP for exception handling |

* You can either write test cases and execute or directly test it in UI(preferred).

Use the below Maven command to execute the tests:

* Spring clean test

Please update the below fields to include traceability between the requirements and scope and blue print. Also please share the test cases for each requirement.

|  |  |  |  |
| --- | --- | --- | --- |
| **Blue Print Topic with complexity** | **Requirement** | **File name** | **Test case** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Implementation/Functional Requirements

## Governance and Tooling:

* + - 1. Follow the practise of Creating Testable Component
      2. Configure your backend application to implement UnitTesting using Junit and Mockito.
      3. Test suites must contain exception situation testing.
      4. Generate the Code Coverage report of the same.
      5. Generate the Code Quality Report of the backend application.

## Code Quality/Optimizations

1. Associates should have written clean code that is readable.
2. Associates need to follow SOLID programming principles.
3. Generate the code quality report of the same.