

---

# System Requirements Specification Index

For

## Issue Tracker Application

Version 1.0

IIHT Pvt. Ltd.  
fullstack@iiht.com

## Contents

1	Business-Requirement:	3
1.1	Problem Statement:	3
1.1.1	Issue Track Application	3
2.	Template Code Structure	
1.1	Issue Controller	
3.	Resources AVAILABLE:	4
4	Suggested WIREFRAMES:	5
5	Business Validations	9
6	Considerations	9
7	Execution Steps to Follow	9

# 1 BUSINESS-REQUIREMENT:

---

## 1.1 PROBLEM STATEMENT:

The purpose of this application is to allow a manager to keep track of all open issues.

### 1.1.1 Issue Track Application:

**The Issue Track Application** allows you to:

1. Access home page
2. Should be able to add a new issue
3. It can have basic fields like Ticket Number, Issue, Filling Date & Owner.
4. Should be able to get the list of Issues.
5. Should be able to edit and delete any Issue.

## 2. TEMPLATE CODE STRUCTURE:

### 2.1 ISSUE CONTROLLER

---

Method Exposed	Purpose
listIssues()	Should return page "list_issues" with required data.
showFormForAdd()	Should return page "issue-add" for adding an Issue.
saveIssue()	Should save an Issue and return "issue/list" with required data.
showFormForUpdate()	Should show issue details in page "issue-add" to edit an Issue.
showFormForDelete()	Should delete an issue and return "issue/list" with required data.

## 3. RESOURCES AVAILABLE:

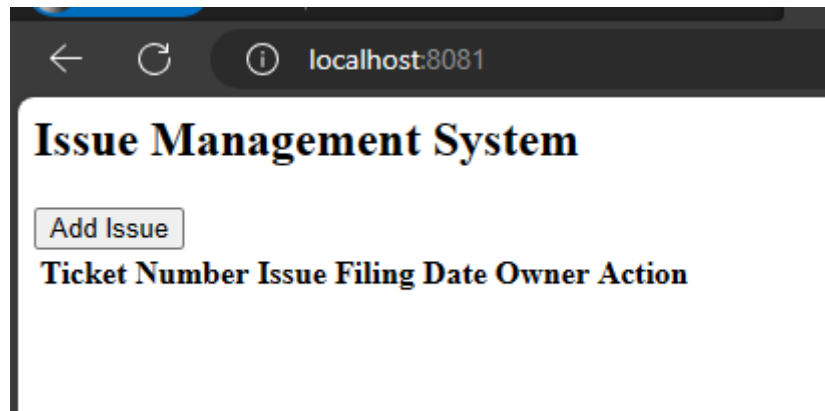
---

Description	View Pages Name	Remarks
Common UI		
Home Page	list-issues	Contains a homepage which shows a list of all issues along with options to add, edit and delete an issue.
All Issues	list-issues	
Add an issue	issue-add	
Update an issue	issue-add	

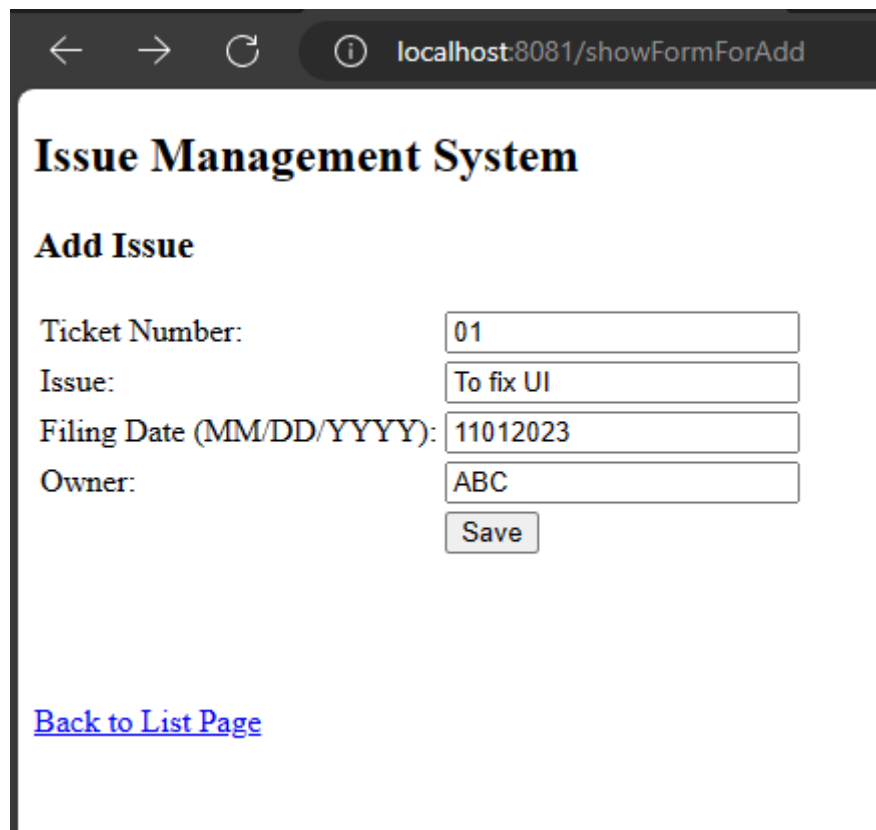
### 3 SUGGESTED WIREFRAMES:

---

#### 1. Homepage – Visitor Landing Page



## 2. Create an Issue



The screenshot shows a web browser window with the address bar displaying 'localhost:8081/showFormForAdd'. The page title is 'Issue Management System'. Below the title is a section header 'Add Issue'. The form contains four input fields: 'Ticket Number' with the value '01', 'Issue' with the value 'To fix UI', 'Filing Date (MM/DD/YYYY)' with the value '11012023', and 'Owner' with the value 'ABC'. A 'Save' button is located below the 'Owner' field. At the bottom left of the form, there is a blue hyperlink labeled 'Back to List Page'.

← → ↻ ⓘ localhost:8081/showFormForAdd

### Issue Management System

#### Add Issue

Ticket Number:

Issue:

Filing Date (MM/DD/YYYY):

Owner:

[Back to List Page](#)

### 3. Edit an Issue

---

## Issue Management System

### Add Issue

Ticket Number:	<input type="text" value="01"/>
Issue:	<input type="text" value="To fix UI"/>
Filing Date (MM/DD/YYYY):	<input type="text" value="11012023"/>
Owner:	<input type="text" value="ABC"/>
	<input type="button" value="Save"/>

[Back to List Page](#)

## 4 BUSINESS VALIDATIONS

---

1. Id must be of type id.
2. Bouncer value not null, min 2 and max 40 characters.
3. Issue value not null, min 2 and max 200 characters.
4. FillingDate value not null.
5. Owner not null and min 2 and max 40 characters.

## 5 CONSIDERATIONS

---

The Code template already contains skeleton methods for service and controller layer. Please write your logic in it.

## 6 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 use command:  
**mvn clean package -Dmaven.test.skip**
4. To launch your application:  
**java -jar springboot-issue-tracker-0.0.1-SNAPSHOT.war**
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.