
System Requirements Specification Index

For

Blood Donors Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Contents

1	Business-Requirement:	3
1.1	Problem Statement:	3
1.1.1	Blood Donors Application	3
2.	Template Code Structure	
1.1	Donor 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 provide a platform for all blood donors. Where users can add, edit and delete their profiles.

1.1.1 Blood Donor Application:

The Blood Donor Application allows you to:

1. Access home page
2. Should be able to add a new donor.
3. It can have basic fields like first name, last name, email, mobile number, state, city and blood group.
4. Should be able to get the list of donors.
5. Should be able to edit and delete any donor.
6. Should bbe able to search any donor.

2. TEMPLATE CODE STRUCTURE:

2.1 DONOR CONTROLLER

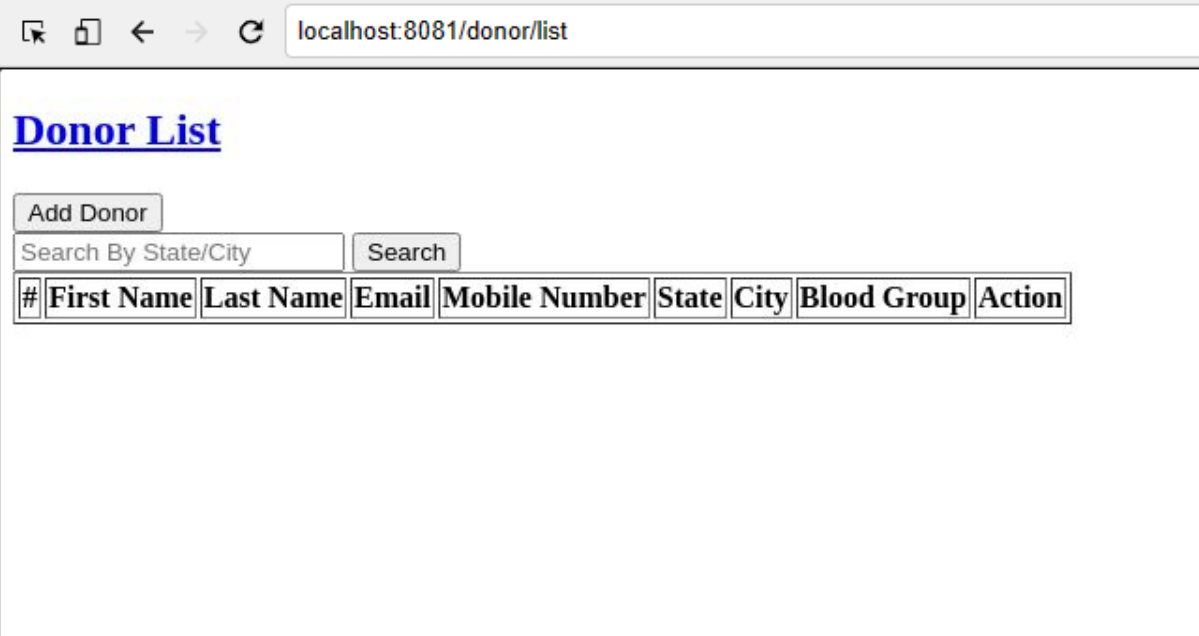
Method Exposed	Purpose
listDonor()	Should return page "list-donors" with required data.
showFormForAdd()	Should return page "add-donor-form" for adding a donor.
saveDonor()	Should save a donor and return "donor/list" with required data.
showFormForUpdate()	Should show donor details in page "update-donor-form" to edit an donor.
deleteDonor()	Should delete a donor and return "donor/list" with required data.
searchDonors()	Should search a donor and return "list-donors" with required data.

3. RESOURCES AVAILABLE:

Description	View Pages Name	Remarks
Common UI		
Home Page	list-donors	Contains a homepage which shows a list of all donors along with options to add, edit , delete and search a donor.
All donors	list-donors	
Add a donor	add-donor-form	
Update a donor	update-donor-form	
Search a donor	list-donors	

3 SUGGESTED WIREFRAMES:

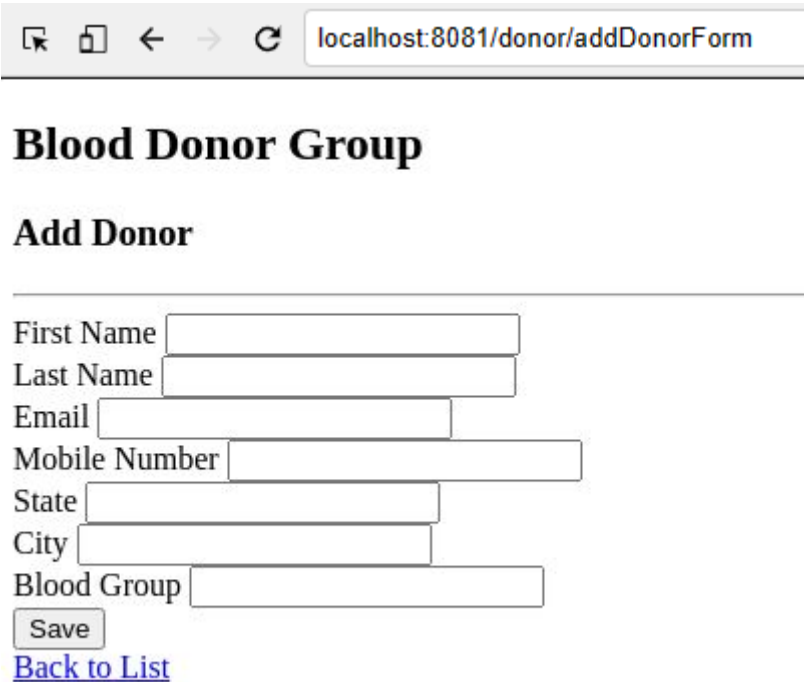
1. Homepage – Visitor Landing Page



A browser window showing the URL `localhost:8081/donor/list`. The page has a title **Donor List**. Below the title is an **Add Donor** button. Underneath is a search bar with the placeholder text "Search By State/City" and a **Search** button. Below the search bar is a table with the following headers: #, First Name, Last Name, Email, Mobile Number, State, City, Blood Group, and Action.

#	First Name	Last Name	Email	Mobile Number	State	City	Blood Group	Action
---	------------	-----------	-------	---------------	-------	------	-------------	--------

2. Create a Donor



A browser window showing the URL `localhost:8081/donor/addDonorForm`. The page has a title **Blood Donor Group** and a subtitle **Add Donor**. Below the subtitle is a form with the following fields: First Name, Last Name, Email, Mobile Number, State, City, and Blood Group. Each field has a corresponding input box. Below the form is a **Save** button and a [Back to List](#) link.

First Name

Last Name

Email

Mobile Number

State

City

Blood Group

Save

[Back to List](#)

Blood Donor Group

Add Donor

First Name

Last Name

Email

Mobile Number

State

City

Blood Group

[Back to List](#)

Donor List

#	First Name	Last Name	Email	Mobile Number	State	City	Blood Group	Action
1	first	last	first@last.com	9999999992	ABC	XYZ	A-	Update Delete

BUSINESS VALIDATIONS

1. Id must be of type id.
2. First name value not blank, min 2 and max 40 characters.
3. Last name value not blank, min 2 and max 40 characters.
4. Email value not blank and of type email.
5. Mobile number not blank.
6. State not blank.
7. City not blank.
8. Blood group not blank and min 1 and max 4 characters.

4 CONSIDERATIONS

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

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 use command:
mvn clean package -Dmaven.test.skip
4. To launch your application:
java -jar <your application war file name>
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.