# System Requirements Specification Index

For

## **Matrimony Application**

Version 1.0



## Contents

1	Business-Requirement	3
	1.1 Problem Statement	3
	1.1.1 Matrimony Application	3
2	Template Code Structure	4
2	2.1 Matrimony Controller	4
2	2.2 Resources Available	4
3	Suggested Wireframes	5
4	Business Validations	9
5	Considerations	9
6	Execution Steps to Follow	9

#### 1 BUSINESS-REQUIREMENT:

#### 1.1 PROBLEM STATEMENT:

The Matrimony Application offers a seamless platform for individuals seeking meaningful relationships.

You are tasked with creating a robust platform where users can effortlessly navigate to the homepage to register new matrimonial profiles, inputting essential details such as full name, gender, address, religion, occupation, and hobbies. The application provides an organized view of all matrimony entries, which can be sorted by various fields in either ascending or descending order to facilitate user preferences. The portal enables easy editing and deletion of any matrimony profiles, enhancing user control and data accuracy. Users can perform searches specifically by name to find relevant matrimony profiles. The system also allows users to update the status of each entry, indicating whether a suitable match has been found, thereby optimizing the search experience and providing timely updates on matchmaking progress.

#### 1.1.1 Matrimony Application:

#### The Matrimony Application allows you to:

- 1. Access home page
- 2. Should be able to add a new matrimony.
- 3. It should have basic fields like full name, gender, address, religion, occupation and hobbies.
- 4. Should be able to get the matrimony list along with options to sort in ascending and descending order in each field.
- 5. Should be able to edit and delete any matrimony.
- 6. Should be able to search for a matrimony by their name.
- 7. Should be able to update the status of a matrimony whether a match is found or not.

## 2. TEMPLATE CODE STRUCTURE:

## 2.1 MATRIMONY CONTROLLER

Method Exposed	Purpose
listMatrimonys()	Should return page "list-matrimonys" with required
	data.
showFormForAdd()	Should return page "add-matrimony-form" for adding
	a matrimony.
saveMatrimony()	Should save a matrimony and return "matrimony/list"
	with required data.
showFormForUpdate()	Should show matrimony details in page
	"update-matrimony-form" to edit a matrimony.
deleteMatrimony()	Should delete a matrimony and return
	"matrimony/list" with required data.
searchMatrimonys()	Should search a matrimony and return
	"list-matrimonys" with required data.
updateStatus()	Should show the status of a matrimony whether a
	match is found or not.

#### 2.2 Resources Available:

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

#### 3 SUGGESTED WIREFRAMES:

## **1. Homepage –** Visitor Landing Page



## 2. Create a Matrimony



## Matrimony

#### **Add Matrimony**

Full Name:	
Gender: Male 🕶	
Address: Religion:	
Occupation:	
Hobbies: Save Back to List	

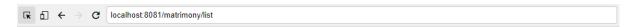
R	Ð	<b>←</b>	G	localhost:8081/addMatrimonyForm

## Matrimony

#### **Add Matrimony**

Full Nam	e: John Doe
Gender:	Male 🗸
	Whitefield, Bangalore
Address:	2
Religion:	Hindu
Occupation	on: Software Developer
	Swimming, Singing
Hobbies:	//
Save	
Back to L	<u>.ist</u>

## 3. Matrimony List

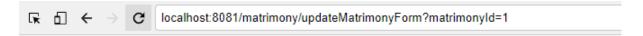


#### **Matrimony List**



1

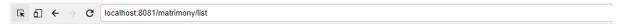
## 4. Update Matrimony



#### Matrimony

#### **Update Matrimony**



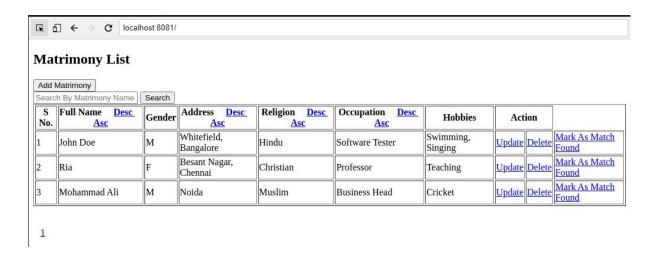


#### **Matrimony List**



1

#### 5. Delete a Matrimony



## 6. Search Matrimony



#### **Matrimony List**

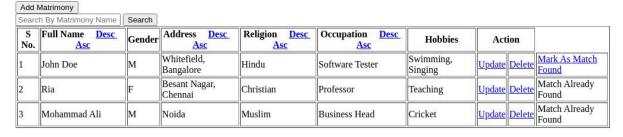


1

## 7. Matrimony Status



#### **Matrimony List**



1

#### 4 Business Validations

- 1. Id must be of type id.
- 2. Full name value should not be blank, min 2 and max 40 characters.
- 3. Gender is not null.
- 4. Address should not be blank, min 2 and max 100 characters.
- 5. Religion should not be blank.
- 6. Occupation should not be blank.

#### **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.
- 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

**NOTE:** After typing any of the above commands you might encounter any warnings.

- >> Please note that these warnings are expected and can be disregarded. Proceed to the next step.
- 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.