
System Requirements Specification Index

For

Customer Relationship Management Application

Version 1.0

Contents

| | | |
|-------|---------------------------|---|
| 1 | Business-Requirement: | 3 |
| 1.1 | Problem Statement: | 3 |
| 1.1.1 | CRM Application | 3 |
| 2. | Template Code Structure | |
| 1.1 | Customer 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 the company to manage customers very easily.

1.1.1 CRM Application:

The CRM Application allows you to:

1. Access the home page.
2. Should list all customers.
3. Can add new customers.
4. It can have basic fields like First Name, Last Name and Email.
5. Should be able to edit and delete any customer.
6. Search for any customer.

2. TEMPLATE CODE STRUCTURE:

2.1 CUSTOMER CONTROLLER

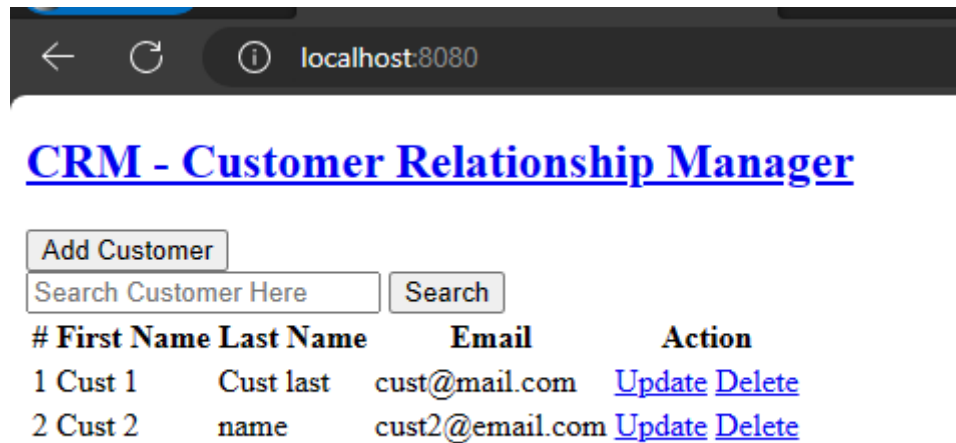
| Method Exposed | Purpose |
|---------------------|---|
| listCustomers() | Should return page "list_customers" with required data. |
| showFormAndAdd() | Should return the page "add_customer_form" |
| saveCustomer() | Should save the customer and return "/customer/list" |
| showFormAndUpdate() | Should return the "update_customer_form" page. |
| deleteCustomer() | Should delete customer and return "/customer/list". |
| searchCustomers() | Should search customers and return 'list_customers' page. |

3. RESOURCES AVAILABLE:

| Description | View Pages Name | Remarks |
|-----------------|----------------------|--|
| Common UI | | |
| Home Page | list_customers | Contains a homepage which shows a list of all customers along with options to add, edit and search a customer. |
| Add Customer | add_customer_form | |
| Update Customer | update_customer_form | |

3 SUGGESTED WIREFRAMES:

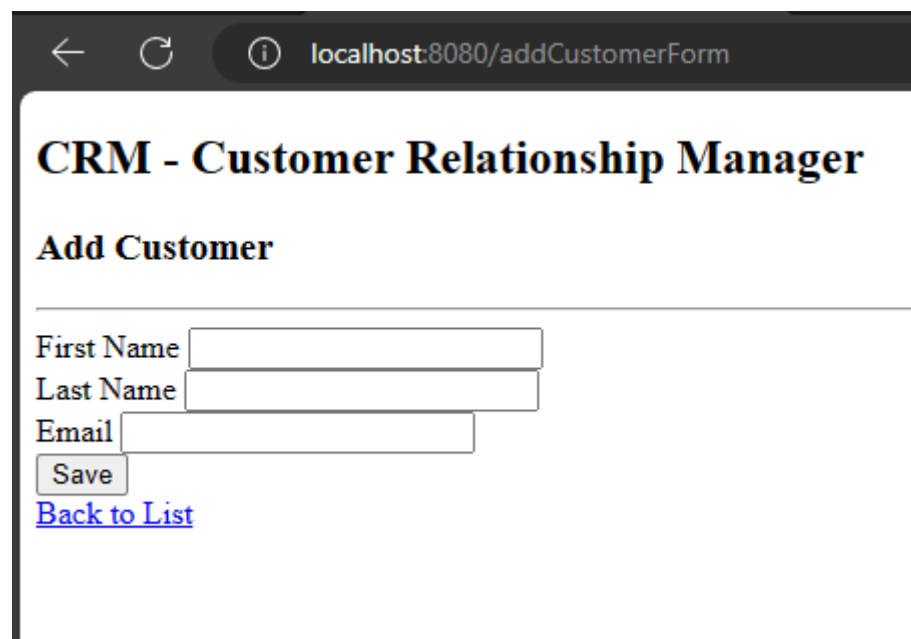
1. Homepage – Visitor Landing Page



A screenshot of a web browser showing the homepage of a CRM system. The browser's address bar displays 'localhost:8080'. The page title is 'CRM - Customer Relationship Manager'. Below the title, there is a button labeled 'Add Customer' and a search bar with the placeholder text 'Search Customer Here' and a 'Search' button. A table lists two customers with columns for '#', 'First Name', 'Last Name', 'Email', and 'Action'. The first customer is 'Cust 1' with email 'cust@mail.com', and the second is 'Cust 2' with email 'cust2@email.com'. Both have 'Update' and 'Delete' links in the action column.

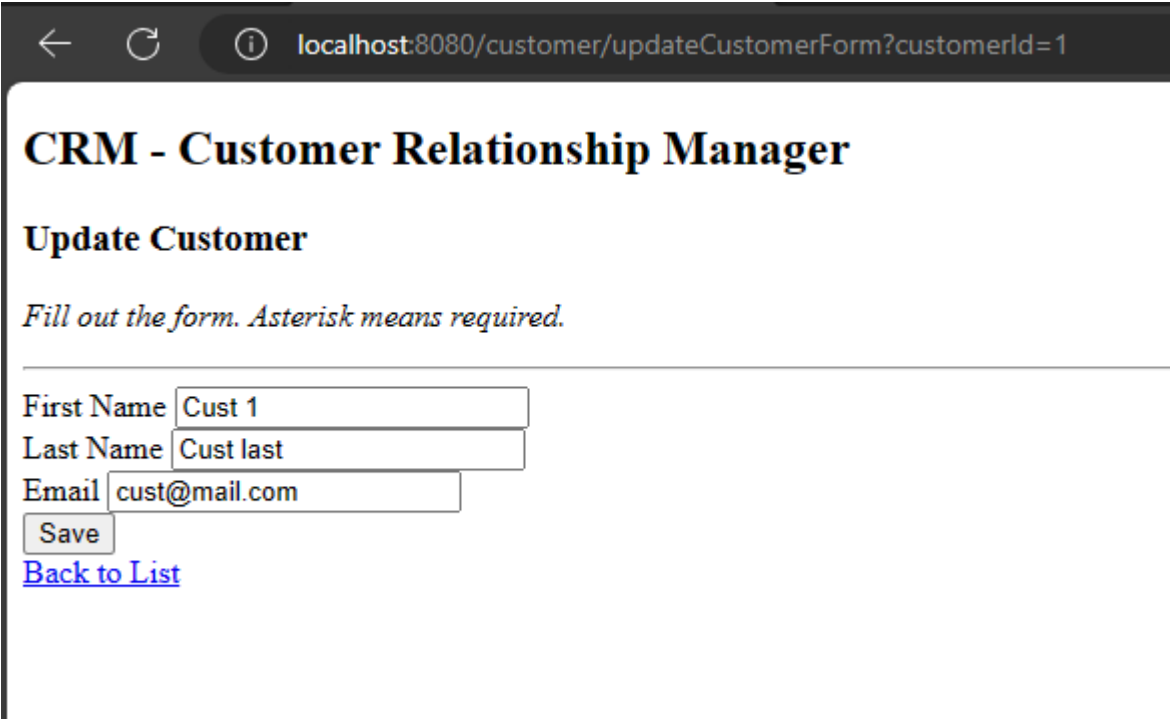
| # | First Name | Last Name | Email | Action |
|---|------------|-----------|-----------------|---|
| 1 | Cust 1 | Cust last | cust@mail.com | Update Delete |
| 2 | Cust 2 | name | cust2@email.com | Update Delete |

2. Add a Customer



A screenshot of a web browser showing the 'Add Customer' form. The browser's address bar displays 'localhost:8080/addCustomerForm'. The page title is 'CRM - Customer Relationship Manager'. Below the title, there is a section titled 'Add Customer'. The form contains three input fields: 'First Name', 'Last Name', and 'Email'. Below the 'Email' field is a 'Save' button and a link labeled 'Back to List'.

3. Edit a Customer



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/customer/updateCustomerForm?customerId=1'. The page title is 'CRM - Customer Relationship Manager'. Below the title is the section 'Update Customer'. A note states 'Fill out the form. Asterisk means required.' The form contains three input fields: 'First Name' with the value 'Cust 1', 'Last Name' with the value 'Cust last', and 'Email' with the value 'cust@mail.com'. Below the form is a 'Save' button and a blue link labeled 'Back to List'.

← ↻ ⓘ localhost:8080/customer/updateCustomerForm?customerId=1

CRM - Customer Relationship Manager

Update Customer

Fill out the form. Asterisk means required.

First Name

Last Name

Email

[Back to List](#)

4 BUSINESS VALIDATIONS

1. Id must be of type id.
2. FirstName value not blank, min 2 and max 40 characters.
3. LastName value not blank, min 2 and max 40 characters.
4. Email value not blank and must be of type email.

5 CONSIDERATIONS

The Code template already contains skeleton methods for service, repository 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-crm-service-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.