# System Requirements Specification Index

For

# Customer Relationship Management Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

## Contents

# 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 **(must show option to add customer, search by customer first name or last name and show list of customers with number of pages for pagination).**
2. Should list all customers **(should return data in pages and must have buttons as Asc and Desc in each column to sort list ascending or descending respectively).**
3. Should be able to add new customers.
4. It can have basic fields like S.No, First Name, Last Name, Email, Action and Customer Verified.
5. Should be able to edit, delete and verify any customer.
6. Should be able to search any customer on a first name or last name basis. **(should be implemented using custom query).**

# 2 Template Code Structure:

## 2.1 Customer Controller

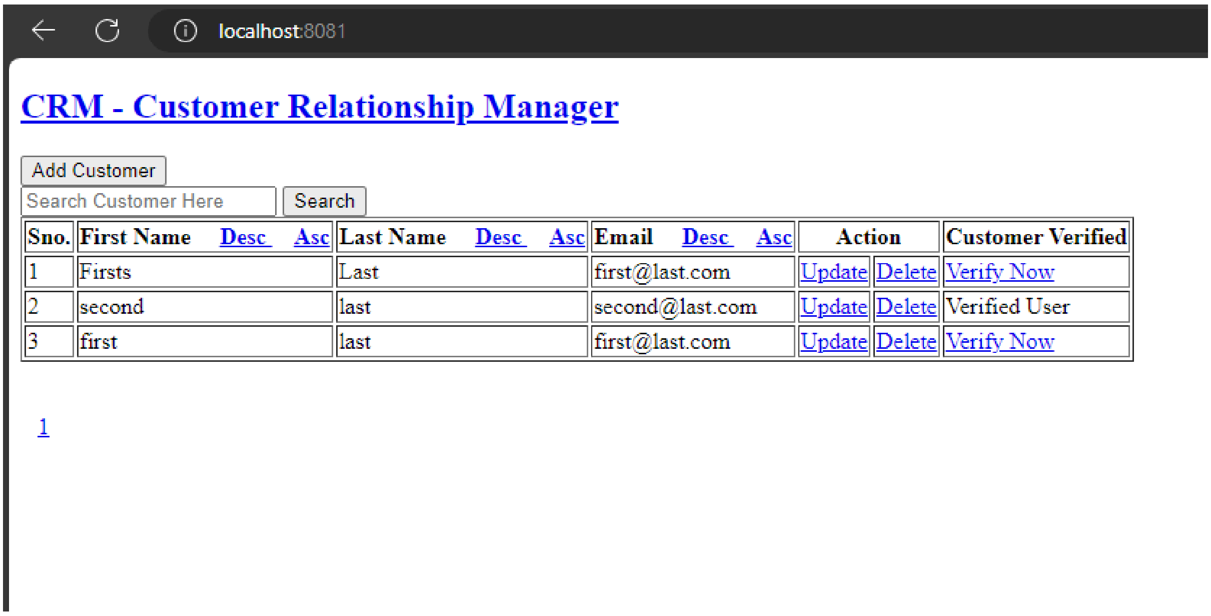
| Method Exposed | Purpose |
|---|---|
| **listCustomers()** | Should return page “list-customers” with required data. |
| **showFormAndAdd()** | Should return the page “add-customer-form” |
| **saveCustomer()** | Should save the customer and return “/customer/list” |
| **showFormAndUpdate()** | Should return the “update-customer-form” page. |
| **deleteCustomer()** | Should delete customer and return “/customer/list”. |
| **updateVerified()** | Verify the customer and return ‘customers/list“ 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 | |

# 4 SUGGESTED WIREFRAMES:

## 1. Homepage – Visitor Landing Page



localhost:8081

CRM - Customer Relationship Manager

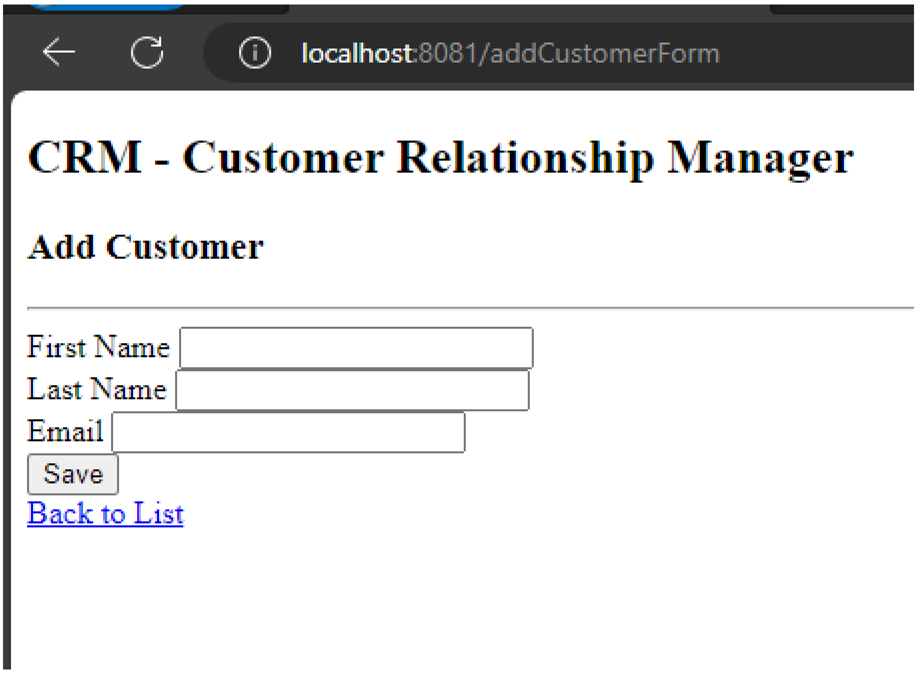
Add Customer

Search Customer Here Search

| Sno. | First Name Desc Asc | Last Name Desc Asc | Email Desc Asc | Action | | Customer Verified |
|---|---|---|---|---|---|---|
| 1 | Firsts | Last | first@last.com | Update | Delete | Verify Now |
| 2 | second | last | second@last.com | Update | Delete | Verified User |
| 3 | first | last | first@last.com | Update | Delete | Verify Now |

1

## 2. Add a Customer



localhost:8081/addCustomerForm

CRM - Customer Relationship Manager

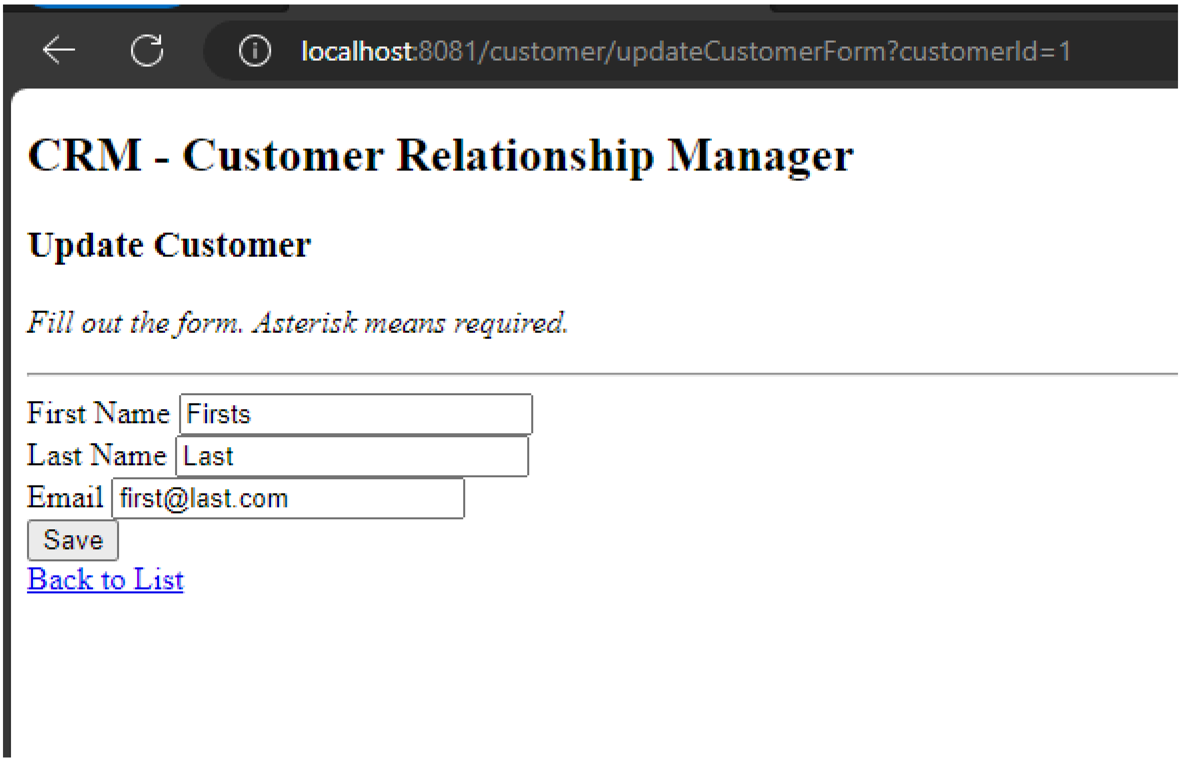
Add Customer

First Name
Last Name
Email

Save

Back to List

## 3. Edit a Customer



localhost:8081/customer/updateCustomerForm?customerId=1

# CRM - Customer Relationship Manager

### Update Customer

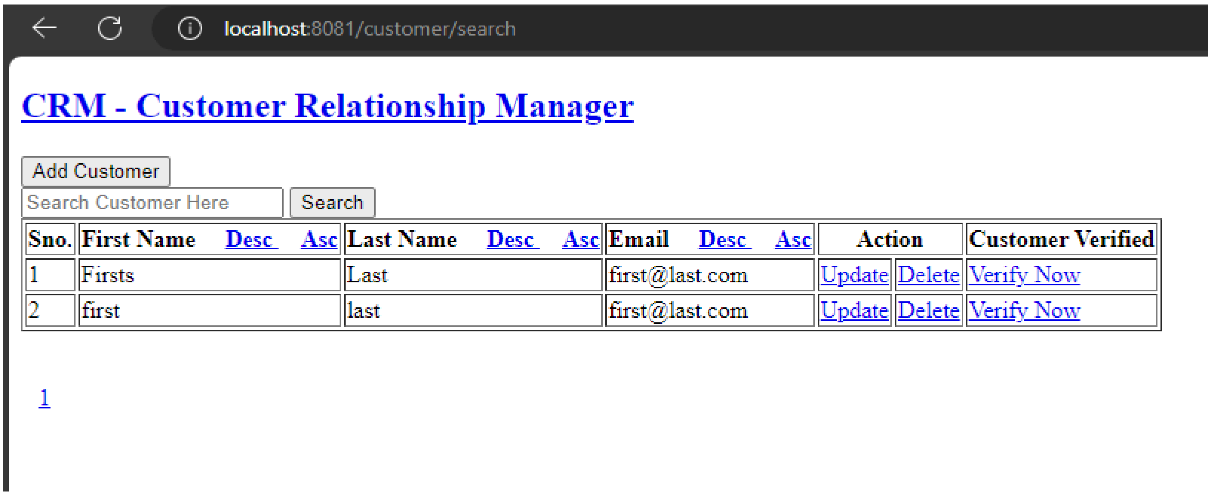
*Fill out the form. Asterisk means required.*

First Name: Firsts

Last Name: Last

Email: first@last.com

## 4. Search a Customer



localhost:8081/customer/search

# CRM - Customer Relationship Manager

| Sno. | First Name Desc Asc | Last Name Desc Asc | Email Desc Asc | Action | | Customer Verified |
|---|---|---|---|---|---|---|
| 1 | Firsts | Last | first@last.com | Update | Delete | Verify Now |
| 2 | first | last | first@last.com | Update | Delete | Verify Now |

1

## 5 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. Verify value should not be null.

## 6 Considerations

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

## 7 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.**