PROJECT REPORT: CUSTOMER MANAGEMENT

Submitted by,

Shilpa Balakrishnan

ID: 845394

Batch code: CHN19AJ029

Contents

ABSTRACT

The objective of this project is to develop an application for a customer manager where he is able to create new customers and update/delete/retrieve the customer details. It also has a search module where the manager can search the customer names. This is achieved by making a web application utilizing Spring MVC framework. Spring framework helps in the easy development of JavaEE applications. By following the Model-View-Controller (MVC) designing pattern it implements all the basic features of core spring framework required to complete this application.

This application makes use of a database in SQL to manage the customer details. It has Login page which only allows the authorized users to access the system. The user can view the current customers and their details as a table and also add new customers or update/delete/fetch existing customer details. The application also has a search bar where user can search customer list using the first name of customers.

INTRODUCTION

Customer management is no simple task. Juggling multiple accounts and its details requires a great deal of organization and prioritization. You may have multiple clients that ask for last-minute tasks or any other requirements relating to the business that you provide to those customers. In any sectors where a large number of customers are involved, getting and keeping customer information and documents organized is a struggle. Without a proper process in place, one generally ends up with piles, emails, notes and scraps of information all over the place. This application provides the solution for this problem.

Customer Management System is a web application which can be easily used for storing your customer details and viewing it as a list. For that it makes use of a database for storing customer details, HTML and CSS makes it more attractive and user friendly, and other components in the background helps to do the CRUD functions such as fetching, deleting, updating or even adding new customer details. The background components are implemented on basis Spring framework. It provides ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of this application (input logic, business logic, and UI logic), while providing a loose coupling between these 3 elements such as Model, View and Controller.

REQUIREMENT SPECIFICATIONS

The background of this project includes some major software's which provides different functionalities for the web application. The whole project is setup in an integrated development environment (IDE) called Eclipse IDE. JDK and Tomcat server is required to prepare the development environment to work with Spring Framework.

1. Maven 3.0 or above

   Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central place. This build tool will help tomcat server and eclipse to ease their work. Maven takes care of Dependency Management, Building a deployable unit, to run things you want - like unit tests, Multiple Modules, Running application in Tomcat, Generating sample projects.

2. Eclipse IDE / STS 2018-2019

   Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins. Here, by making use of required plug-ins we create the application for customer management. Similarly, Spring Tool Suite is an IDE to develop spring applications. It is an Eclipse-based development environment. It provides a ready-to-use environment to implement, run, deploy, and debug the application. It validates our application and provides quick fixes for the applications.

3. JAVA 1.8+

   The Java Development Kit is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms released by Oracle Corporation. The JDK includes a private JVM and a few other resources to finish the development of a Java Application. In eclipse, JDK is necessary to Compile the code and convert java code into byte code while JRE is need for executing the byte code. JDK include the JRE plus command- line development tools such as compiler and debuggers that are necessary for developing applications.

4. MySQL database 5.0 or above

   MySQL is an open-source relational database management system (RDBMS). It is based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

5. Tomcat Server 7.0 or above

   Apache Tomcat is used to deploy your Java Servlets and JSPs. So in your Java project you can build your WAR (short for Web ARchive) file, and just drop it in the deploy directory in Tomcat. So basically Apache is an HTTP Server, serving HTTP. Tomcat is a Servlet and JSP Server serving Java technologies. After installing select server runtime environment as the installed version tomcat and run the code.

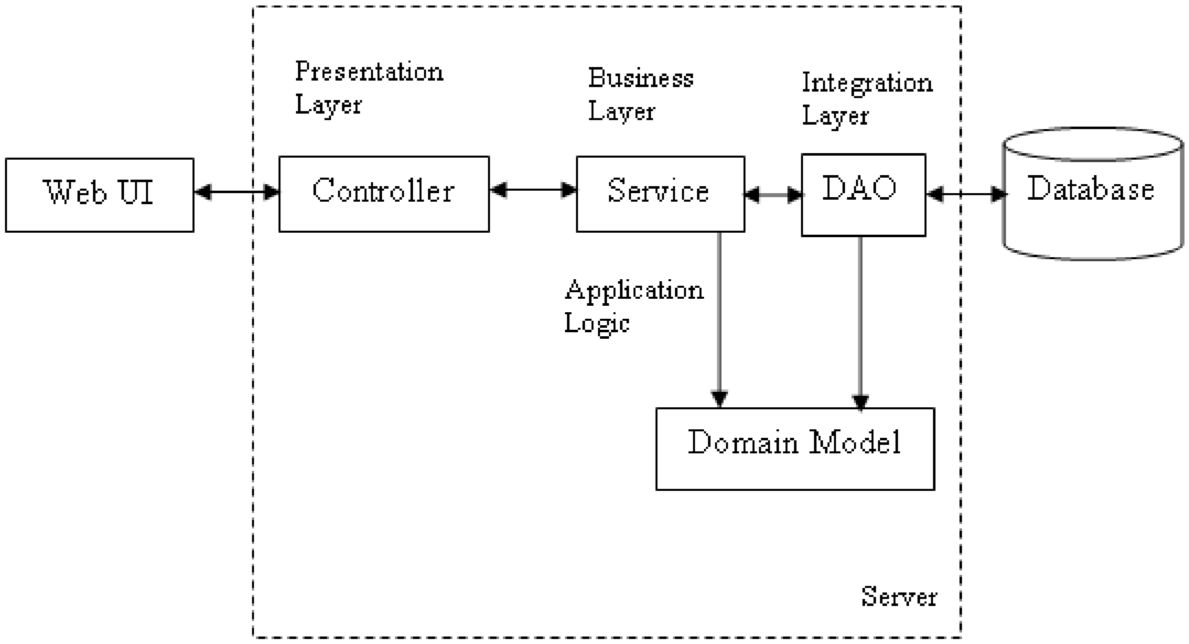
6. JUnit

   JUnit is an open source Unit Testing Framework for JAVA. It is useful for Java Developers to write and run repeatable tests. As the name implies, it is used for Unit Testing of a small chunk of code. Developers who are following test-driven methodology must write and execute unit test first before any code.

ARCHITECTURE DESIGN

Outline of the Project:

The web application is based on the Spring MVC framework. A Spring MVC provides an elegant solution to use MVC in spring framework by the help of “DispatcherServlet” xml file. The required JAR files/Maven dependencies are added to “pom.xml”. Next configuration is done in another xml file where view components such as jsp, css, javascript etc are specified by defining bean. Also define configurations for the controller classes. When a web request is sent to a Spring MVC application, dispatcher servlet first receives the request. Then it organizes requests according to the different components configured in spring’s web application context or annotations present in the controller itself. Models usually consist of POJO objects that are processed by the service layer and persisted by the persistence layer. Views are usually JSP templates written with Java Standard Tag Library (JSTL).

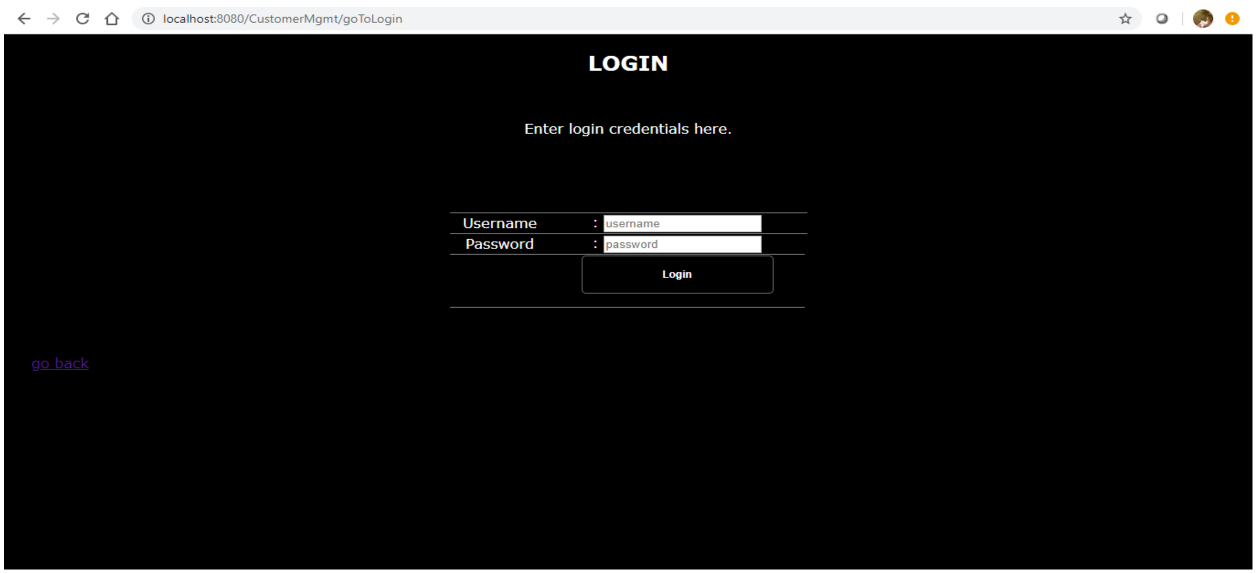


The overall functioning of the web application can be divided into 3 parts - Web UI, Middleware Layer and Persistence Layer as shown in figure. The Middleware

Layer provides Spring MVC components including the domain model, which is the POJO class corresponding to the database created in MySQL, Service layer which is also known as the business layer, Repository layer also known as integration layer and finally the controller also known as the presentation layer that coordinates with the Web UI layer according to the mapping given.

Snapshots:

1] Login Page:



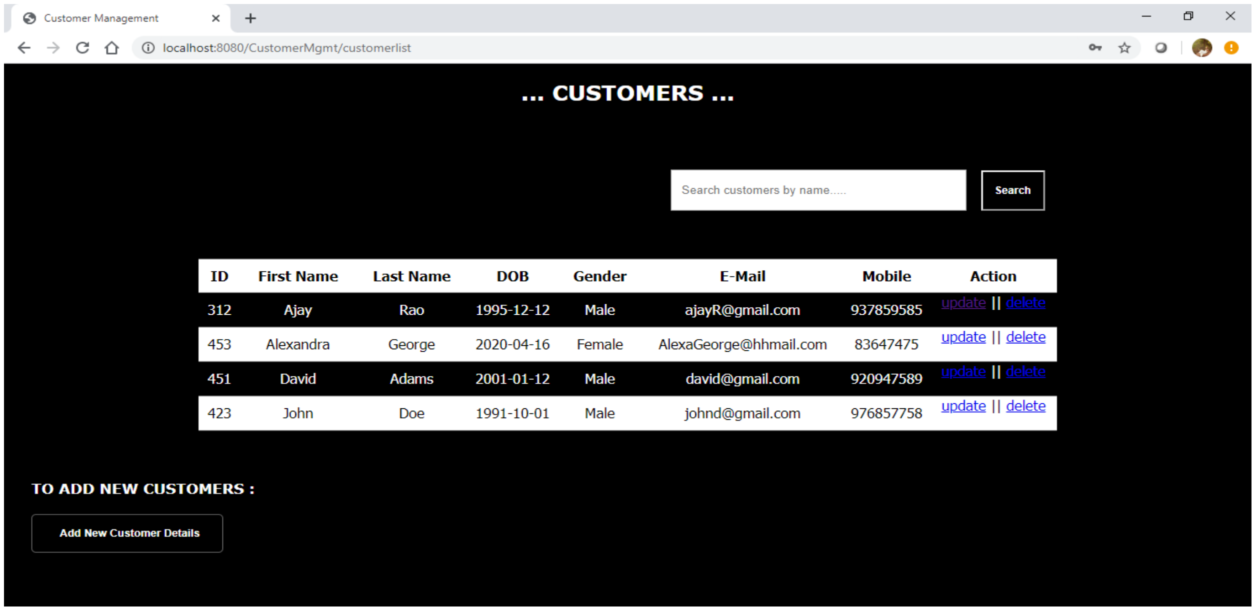
localhost:8080/CustomerMgmt/goToLogin

LOGIN

Enter login credentials here.

Username	:	username
Password	:	password

2] Customer List:



Customer Management

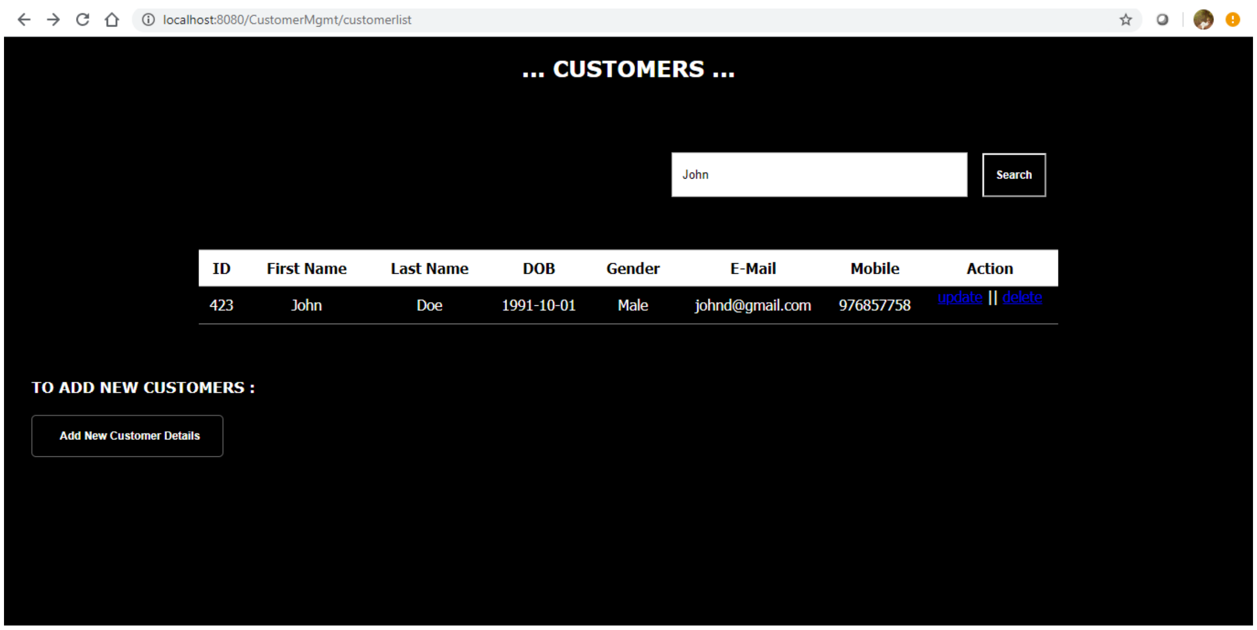
localhost:8080/CustomerMgmt/customerlist

... CUSTOMERS ...

ID	First Name	Last Name	DOB	Gender	E-Mail	Mobile	Action
312	Ajay	Rao	1995-12-12	Male	ajayR@gmail.com	937859585	update || delete
453	Alexandra	George	2020-04-16	Female	AlexaGeorge@hhmail.com	83647475	update || delete
451	David	Adams	2001-01-12	Male	david@gmail.com	920947589	update || delete
423	John	Doe	1991-10-01	Male	johnd@gmail.com	976857758	update || delete

TO ADD NEW CUSTOMERS :

3] Search Result:



localhost:8080/CustomerMgmt/customerlist

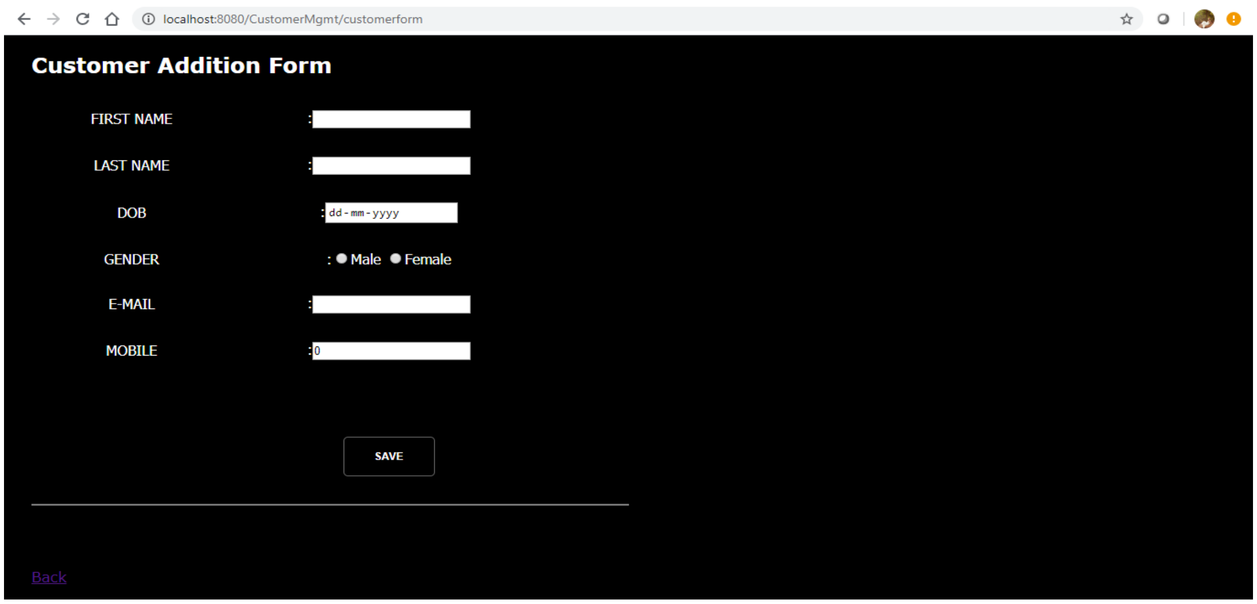
... CUSTOMERS ...

John

ID	First Name	Last Name	DOB	Gender	E-Mail	Mobile	Action
423	John	Doe	1991-10-01	Male	johnd@gmail.com	976857758	update || delete

TO ADD NEW CUSTOMERS :

4] Add Customer:



localhost:8080/CustomerMgmt/customerform

Customer Addition Form

FIRST NAME	:
LAST NAME	:
DOB	: dd-mm-yyyy
GENDER	: ● Male ● Female
E-MAIL	:
MOBILE	: 0

CONCLUSION AND FUTURE SCOPE

Customer management is a method of managing the relationship between a company and its customer base, regardless of the industry. As part of this project a solution to this tedious task is proposed. A web application that can provide assistance for a customer manager that eases his job. It simplifies the task by keeping record of each and every customer. The same can be used for any future needs like updating/deleting the existing customer details and also the use can add new customers when needed. Also, it has a search module to search among the list of customers. It has login page which makes sure that only authorized users can access the system.

As a future scope, it will be more beneficial to add a calendar so that the manager can keep track of requirements and deadlines associated with each customer. Also alerts and reminders can be very useful for the manager to organize day-to-day works. Considering the large amount of customers in any sector, it’s hard to give customers our undivided attention. So adding more fields relating to the customer details on information regarding the business between them and also a review note based on each and every meeting conducted with them.

REFERENCES

1. https://docs.spring.io/spring/docs/current/spring-framework-reference/
2. https://docs.spring.io/spring/docs/current/spring-framework-reference/core.html#spring-core
3. https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html
4. https://hibernate.org/orm/documentation/5.0/
5. https://maven.apache.org/guides/getting-started/index.html