

WEB APPLICATION FOR CUSTOMER RELATIONSHIP MANAGER USING JAVA

Project report submitted in partial fulfillment of the requirement for the
degree of Bachelor of Technology

in

Computer Science and Engineering

By

Nitish Kumar (151344)& Vansh Mahajan(151210)

Under the supervision of

Mr. Gaurav Bhansali (Assistant Manager)

&

Mr. Sainath Dekate (Lead Consultant)

to



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Wagnaghat, Solan-
173234, Himachal Pradesh**

CERTIFICATE

Candidate's Declaration

We hereby declare that the work presented in this report entitled “Web Application for Customer Relationship Manager Using Java” in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering, Jaypee University of Information Technology Wagnaghat is an authentic record of my own work carried out over a period from Feb 2019 to till now under the supervision of Mr. Gaurav Bhansali (Assistant Manager) & Mr. Sainath Dekate(Lead Consultant).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Nitish Kumar (151344)

Vansh Mahajan (151210)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Gaurav Bhansali

Assistant Manager

LiveSpread Cora AI

Dated :

ACKNOWLEDGEMENT

Learning through the project under the guidance of our esteemed mentor Mr. Gaurav Bhansali (Assistant Manager) and Mr. Sainath Dekate (Lead Consultant & Java Lead), whose expertise knowledge in the domain of Java, Advance Java and others latest technologies based on Java, not only cleared all our ambiguities but also generated a high level of interest and gusto in the subject. We are truly grateful for his guidance and support throughout the project. We would also like to thank our Head of the Department, Brig(Retd.) SP Ghrera for undying faith in the department of computer science and allowing us to join internship program.

The prospect of working in a group with high level of accountability fostered a spirit of teamwork and created a feeling of oneness which thus, expanded our ken, motivated us to perform to the level best of our ability and create a report of the highest quality.

To do the best quality work, with utmost sincerity and precision has been our constant endeavor.

Date:

Nitish Kumar (151344)

Vansh Mahajan (151210)

TABLE OF CONTENT

S. No.	Title	Page No
1	Introduction	1
1.1	Problem Statement	1
1.2	Objective	1
1.3	Methodology	1-2
1.4	Use Case	3
1.5	Implementation steps	3-4
1.6	Crud Information	4
2.	Literature Survey	5
2.1	Super User	7
2.2	Admin User	7
2.2	End User/Analyst	7
3	System Development	8
3.1	Tools	8
3.1.1	Spring Framework	8
3.1.2	Eclipse Oxygen IDE	9
3.1.3	Java	9
3.1.4	MySql Workbench	9
3.2	Libraries	10
3.2.1	Spring Core	10
3.2.2	Annotations	10
3.2.3	Hibernate	10
3.2.4	JPA	11
3.2.5	Maven	12
4	Data Analysis & Algorithm	13
4.1	Data Assembly	13
4.2	Data Preprocessing	13

4.2.1	Missing values	13
4.2.2	Irrelevant Data	14
4.3	Algorithm or Workflow	15
4.3.1	Extraction	15
4.3.2	Mapping	15
4.3.3	Normalization	15
4.4	Code Snippet	16
4.4.1	Web.xml File	16
4.4.2	Servlet File	16
4.4.3	Pom.xml File	17
4.4.4	Domain Class	17
4.4.5	Controller Class	18
4.4.6	Service Interface	18
4.4.7	Service Implementation	19
4.4.8	DAO Interface	19
4.4.9	DAO Implementation	20
4.4.10	Login.jsp File	20
4.4.11	Database Table	21
4.5	UI Snippet	22
4.5.1	Landing Page	22
4.5.2	Read Operation	22
4.5.3	Add Operation	23
4.5.4	Update Operation	23
4.5.5	Delete Operation	24
5	Result & Performance Analysis	25
6	Conclusion	26
7	References	27

LIST OF ABBREVIATIONS

S. No.	ABBREVIATION	FULLFORM
1.	DAO	Data Access Object
2.	JDBC	Java Database Connectivity
3.	JSTL	JSP Standard Tag Library
4.	JSP	Java Server Pages
5.	MVC	Model View Controller
6.	ORM	Object Relational Mapping

LIST OF TABLES

S. No.	Table Name	Page
1.	CRUD Information Table	4

LIST OF FIGURES

FIG. NO.	DESCRIPTION	PAGE
1.	Overview of The Methodology	2
2.	Use Case Diagram	3
3.	Spring Modules Structure	8
4.	JPA Architecture	11
5.	LS Workflow	14

ABSTRACT

We have created “CUSTOMER RELATIONSHIP MANAGER” using spring MVC framework in this we have made a controller that maps all client demands with recommended page. As a matter of first importance, at whatever point client runs the venture, web.xml tells the appreciated page subtleties after the servlet mapping is utilized to scan for the servlet which is spring-servlet for our situation. This record is utilized for characterizing the base bundles and characterizing the database bean besides and DAO class is mapped for jdbc-template. The envelope containing the jsp record is mapped inside this file. The controller on getting the solicitation chooses which page ought to be shown. This is finished utilizing the interior view resolver of the spring system. We have made a POJO class in which all the getter setter techniques for every one of the factors that are utilized in the program. We have DAO class where we have set the layout and all the different usage of sql inquiry is finished.

Chapter – 1

INTRODUCTION

The motivation behind this undertaking is to portray business prerequisite of an application totally, precisely and unambiguously in innovation autonomous way. The sum total of what endeavors have been made in utilizing generally business wording and business language while portraying the prerequisite in this document by utilizing insignificant and normally comprehended specialized phrasing. Main idea behind this all terminology is extract, mapped and normalize the customers data as per requirement.

1.1 PROBLEM STATEMENT

This is the project has its motivation come from the basic crud operation in Java web application also the requirement of our customers. There is basic need of adding, updating, deleting and reading customer data by sending the request to the server and then server responds the request that is coming to the server. So, all basic crud operation is performed in this project and this is basic application requirement in most of the applications.

1.2 OBJECTIVE

The primary objective of this project is:

Extract and normalize the data as per the requirements of the user and develop applications as per the guidance of the user. In this case there is small module in which we are developing a one functionality of the application in which we can create, update, delete and read customer data so that we can communicate with the existing data and adding new data in our web application.

1.3 METHODOLOGY

All the methodology is based on the Spring MVC. MVC is the basic methodology that is used in the most of the web application. MVC stands for the Model View Controller and with help of it we develop our web application. When user make any request to the server then first it goes to the controller: there is two types of controller one front controller and one main controller. Then controller contact with the service layer for the data accessing

from the database. There is one more layer between the database and the service layer that is the data access object (DAO) layer. So, whatever is our business logic that can be performed in this layer. So also, model concept is there which is used to bind the data with the object and it can bind the data that is come from the jsp page or any other web page. After accessing data from the database, it is returning to the controller and controller passes it to the view where we can represent data and give response to the user.

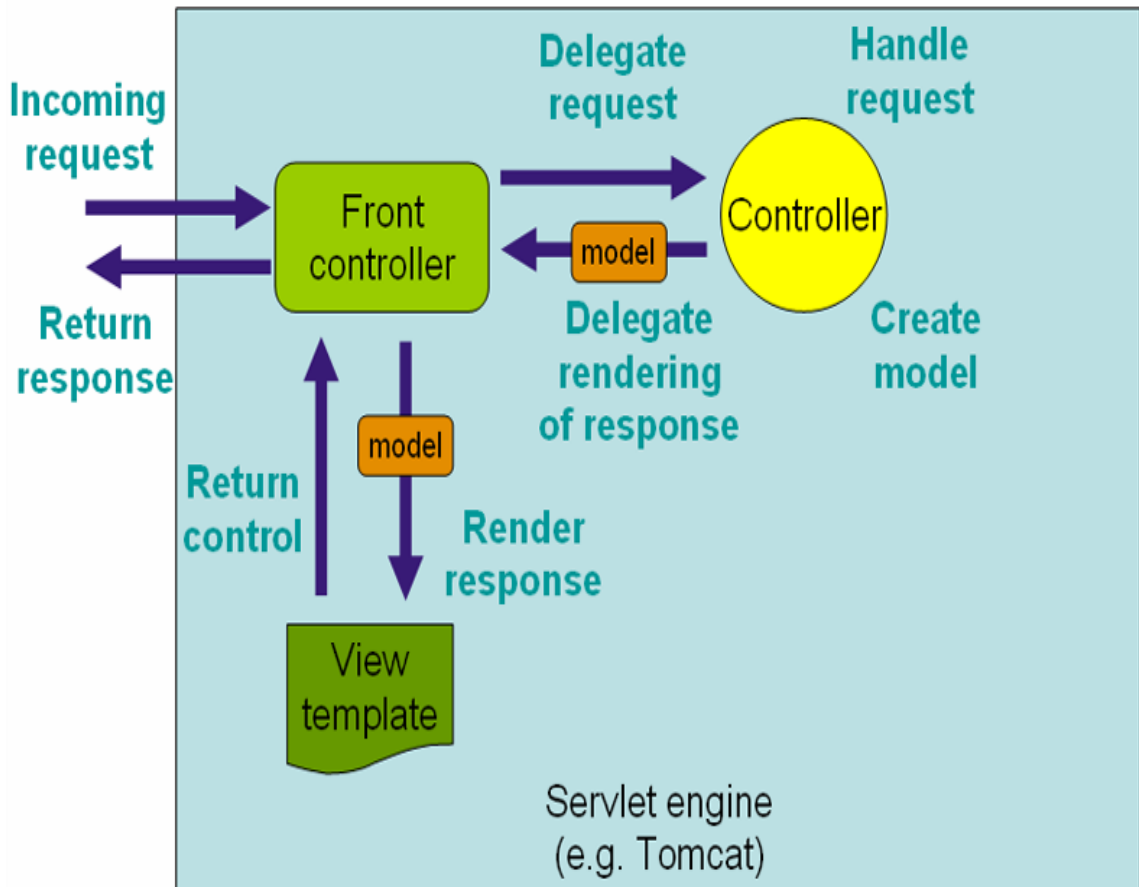


Figure 1: Overview of the methodology

Above diagram shows all the methodologies of this project how the processes and request are processed. so this is the basic methodology of this project.

1.4 USECASE DIAGRAM

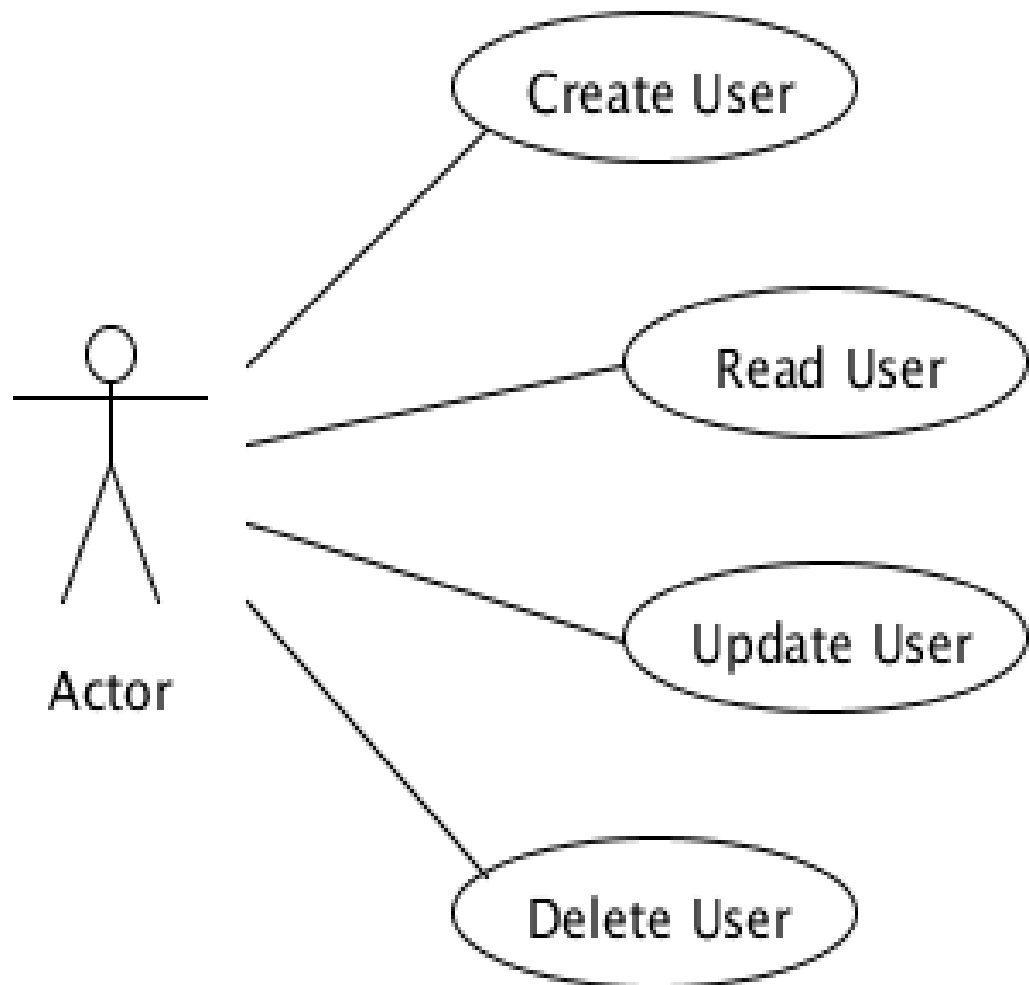


Figure 2: Use Case Diagram

1.5 IMPLEMENTATION STEPS

Java Servlets and Java Server Pages (JSP)

JSP Standard Tag Library (JSTL)

Java Database Connectivity (JDBC)

MySQL database

Apache Tomcat Server

The following tools can be used for the development:

1. Eclipse IDE for Java EE Developers
2. Apache Tomcat version 8.5
3. MySQL Connector for Java

1.6 CRUD INFORMATION

CREATE	Create operation means we can add data to the existing system.
READ	Read operation means we can read the existing data in the system.
UPDATE	Update means we can made changes in the existing data in the system.
DELETE	Delete means we can delete the information existing in the system.

TABLE 1. CRUD OPERATION

CHAPTER - 2

LITERATURE SURVEY

This section discusses the theory used in application design and implementation. It introduces Customer relationship manager and LiveSpread application system's background with several characteristics of existing solutions such as Spring Framework, Hibernate, JPA and Springsecurity, Apache Maven, SQL Database and Apache Tomcat as a server.

Customer relationship manager System is one part of the application in which we perform the basic crud operation to adding, fetching and deleting the records of the various customer in our application. System which is an application that users can have capabilities to manage all ora section of content, data or information of an article, website and so forth. Furthermore, users can manage those contents without technical knowledge of project or application. In this project, CRM is a web application on which business clients can collaboratively modify content and structure. Further, CRM is a content management system, which is similar to database operation but differs in a way that it mainly focuses on interactive UI. CRM and LS help and provides different features for clients to manage the digital content of their records. Evidently, it manages versions because clients can freely come back to any previous ones while they are continuously growing in their business-related activities. Equally important, it increases interactions among customers and users in commending, rating and other features. In addition, it stores, secures and assigns rights to content. In this case, there is a filter for internal and public information or clients created by the product so that internal clients can only see internal information and information which is related to their task. Moreover, LS has a flexible data filter, which stores many different relevant data contents. When clients want to search for something, they can exploit as many subjects as they wish.

In programming improvement process, a lot of prerequisites fill in as a rule to help designers to concentrate on center parts and highlights of the application and avert them from the allurements of making a solid program. This segment makes reference to necessities of the application as far as two primary classifications in particular utilitarian prerequisite and nonfunctional requirement. Functional prerequisites portray the determinations of the application or, at the end of the day, what the application can do. Right off the bat, the application needs to deal with the article executing four

essential elements of constant stockpiling, for example, create, read, update and delete (CRUD). Besides, the application can verify clients into various gatherings for example, administrator gathering, supervisor gathering, interior gathering and outer gathering. The administrator bunch has full benefits to deal with the articles. The trough bunch has comparable benefits to the administrator bunch aside from that a chief can just delicately erase articles. The inner gathering just enables its clients to peruse all open and private articles while the outer gathering limits its clients to open articles. Thirdly, the application gives a variant control to oversee changes or articles. At long last, the application gives an Application Programming Interfaces (APIs) that enable different applications to recover the archive assets. Non-useful necessities allude to the measure that can be utilized to assess the task of the application or, at the end of the day, the properties of the application. Right off the bat, the application is an independent miniaturized scale administration that can be incorporated with another smaller scale service. Secondly, the application can deal with in any event one hundred thousand solicitations for every second. Thirdly, the application can scale into numerous occurrences. At long last, the application utilizes regular server farm Cassandra with other existing miniaturized scale administrations with its very own key space.

There are some powerful frameworks which will discuss below, and it is widely used: -

Spring Framework: This is a Java innovation most recent system which is lightweight, and it offers help to its past structure, for example, Struts, Hibernate, Tapestry, EJB, JSF and so on. The Spring system has a few modules, for example, IOC, AOP, DAO, Context, ORM, WEB MVC and so on.

Rest Framework: This is additionally another all the more dominant system which has tackled numerous issues which happened building up java association with database. It is an open source, lightweight, ORM (Object Relational Mapping) instrument.

Spring Boot:

The Spring gives fundamental and advance ideas of the Spring Framework. This device utilized show over design so there is no need XML arrangement like Spring system. It is for the most part utilized independent spring-based application and to make tranquil webservices.

Also, there are three types of user in the main application of LiveSpread and they have different functionalities and roles in the application. Every user has access according to their role so we cannot say that a particular user has all the access it is totally depends upon the roles which type of user they are.

2.1 SUPER USER

The super user can view/search cases which are uploaded by main group user or child users.

2.2 ADMIN USER

The admin user can view/search cases which are uploaded by himself/herself and child group users.

2.3 END USER /ANALYST

The End user can view/search cases which are uploaded by parent group users or child group users.

CHAPTER – 3

SYSTEM DEVELOPMENT

3.1 TOOLS

3.1.1 SPRING FRAMEWORK

Spring Framework is a Java stage that gives far reaching foundation backing to creating Java applications. Spring handles the foundation so you can concentrate on your application.

The Spring Framework consists of features organized into about 20 modules. These modules are grouped into Core Container, Data Access/Integration, Web, AOP (Aspect Oriented Programming), Instrumentation, and Test, as we can see in diagram: -

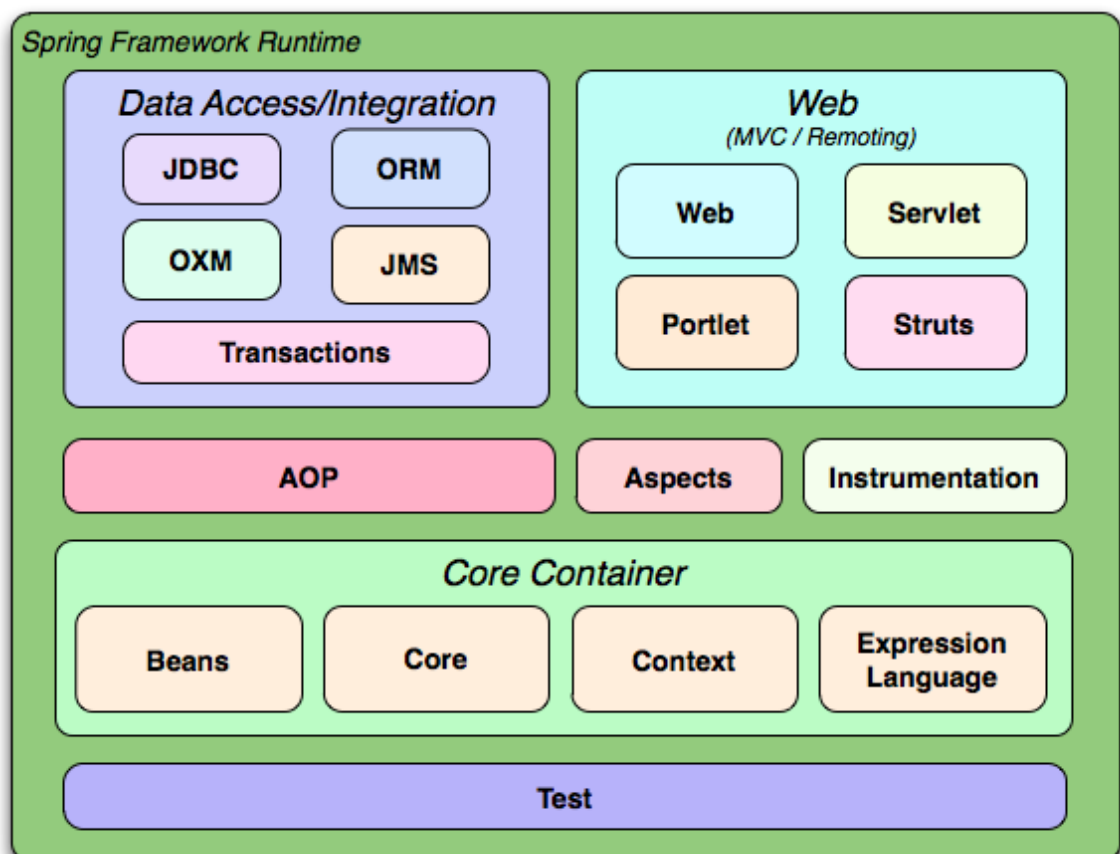


Figure 3: Spring Modules Structure

3.1.2 ECLIPSE OXYGEN IDE

Eclipse is a Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF and in different definition terms it can be defined as best software for doing java coding for business when we do coding in Java and other programming dialects like C, C++, PHP, and Ruby and so forth. Advancement platform given by Eclipse includes Java improvement devices (JDT) for Java, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

3.1.3 JAVA

Java is a language that is used in our company for business use. We perform business logic in our application using java and we find it easy with java that's why we use java in our business. Web application environment is the one of the challenging tasks when we develop application for our client but when we use java and spring mvc module of the Spring then it becomes easy and we can do this easily and all the stuff related to our business is related to the programming language which is java in our case. It is language that is used in most of the company's business for developing enterprise application. Advantage of using this is that we can write code once and use that code across the different platforms.

3.1.4 MySQL WORKBENCH

Concept of database is very critical when we create a web application. Without database no application can be called perfect application. We are performing crud operation in our application and for crud use of database is necessary. It is the backbone of every application. There are many relational databases available in the market and we are using Mysql 8.0.15 version. It has updated functionalizes as compare to the previous version and we can say that this plays an important role in our application because when we performing crud operation and using Hibernate for that in backend Mysql performs critical role.

3.2 LIBRARIES

3.2.1 SPRING CORE

Spring core means parent of all jar files and all the basic functionalities of spring framework. There are many modules similar to this that are used in Python and other constructs. It does most of the work behind the scenes for us.

Many updates come but base working is same as compare to the previous versions and some additional details are added.

It is very difficult that go to internet then find particular website and then download and add that file into our project build. Spring core is parent of all jar file so we use this in maven dependency and it will do all work for us.

3.2.2 ANNOTATIONS

Java Annotation means lot of work is happened behind the scenes and you need not to worry lot of code and it will be handled by your framework behind the scenes and it is the biggest advantage of annotations.

Xml file work is reducing while we use Annotations so lot of hard code is not necessary in case of Annotations.

3.2.3 HIBERNATE

There are many models present in the market related to the ORM. But in our company, we are using hibernate for development. When we take connection from the database then it plays big role. it implements the functions of the persistent framework and we can map our table to the database by adding @entity annotations. So, annotations make our role easy and we need not to worry about lot of backend performance. But as a developer it is necessary to understand backend functionalities so that we understand everything in our code. Also, performance issue is also resolved if we use this framework in our application because it gives the concept of cache for performance.

3.2.4 JPA

As we familiar with the concept of interface and classes in the java ,here also JPA and hibernate is same. JPA is an interface and it has many mapping modules that provide its implementation and we can say that hibernate is one of them. JPA is many updates available time to time so we can say that it is up to date persistent API so we can use this in our company because it developers of hibernate also suggest to use JPA instead of hibernate because due to up to date functions.

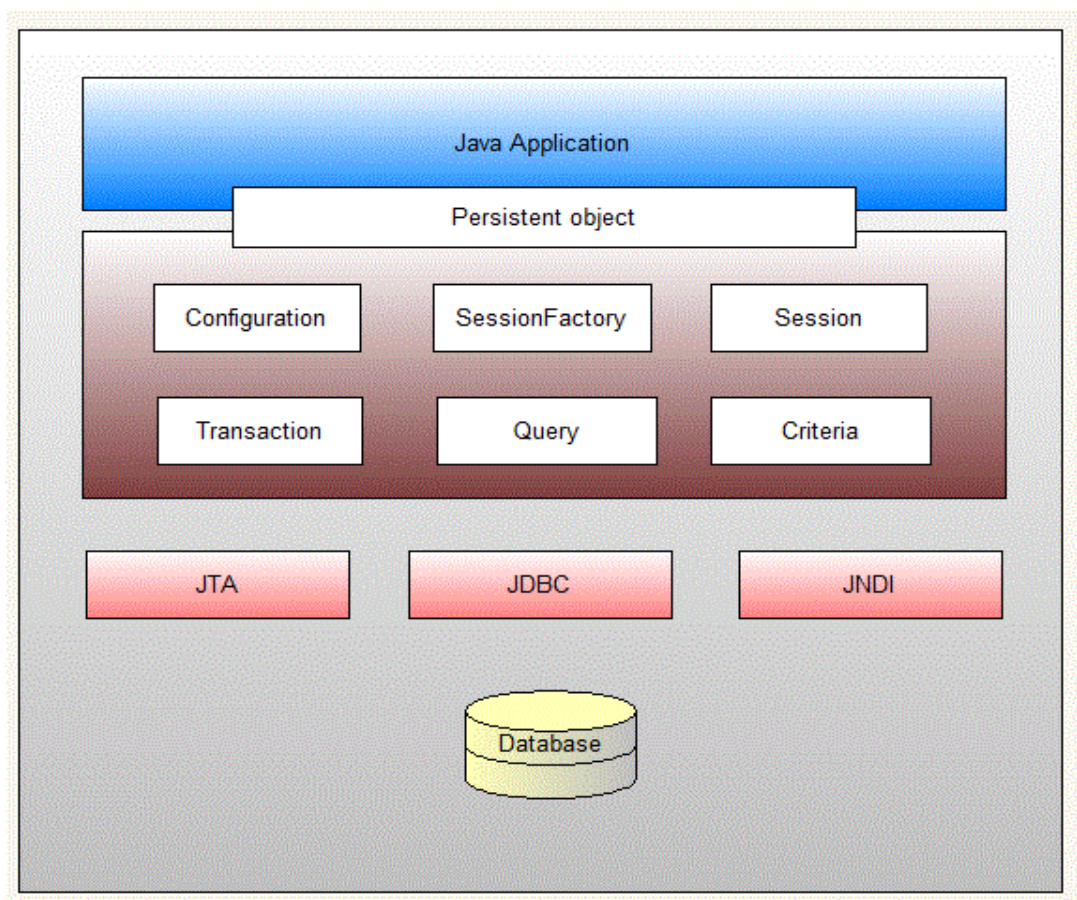


Figure 4: JPA Architecture

3.2.5 MAVEN

Jar files issue is one of the big issues while we develop our web application. Developers also worry how to solve this problem and then they come with the one of the good technologies and we can say that Maven is that technology.

Idea behind using Maven is remove the issue of jar files and making the developer task easy so that they can develop their code easily. And management of maven is given below: -

- 1.Maven make task very easy.
- 2.It can provide us one system or one integrated system.
- 3.Performcae also improved.
- 4.Best code review can be developed.
- 5.It find jar from the net behind the scenes.

CHAPTER - 4

DATA ANALYSIS&ALGORITHM

4.1 DATA ASSEMBLY

Work queue is the landing screen of the application, where the user is able to view the cases and fill their requirements and information that is required. After this our requirement is clear and based on that we can start our process. So, customer or user add their financial data or sheets in our existing Live Spread application landing page. So, there is various documents related to the financial data and out of all the information that are present in the financial sheet we can extract information that is relevant to the requirement based on the rules that are made for extraction. So, in this way collect the data. After this mapping process is start and there is some business logic for that. After mapped there is normalization process in which we represent the result in terms of web application as per user requirement template.

4.2 DATA PREPROCESSING

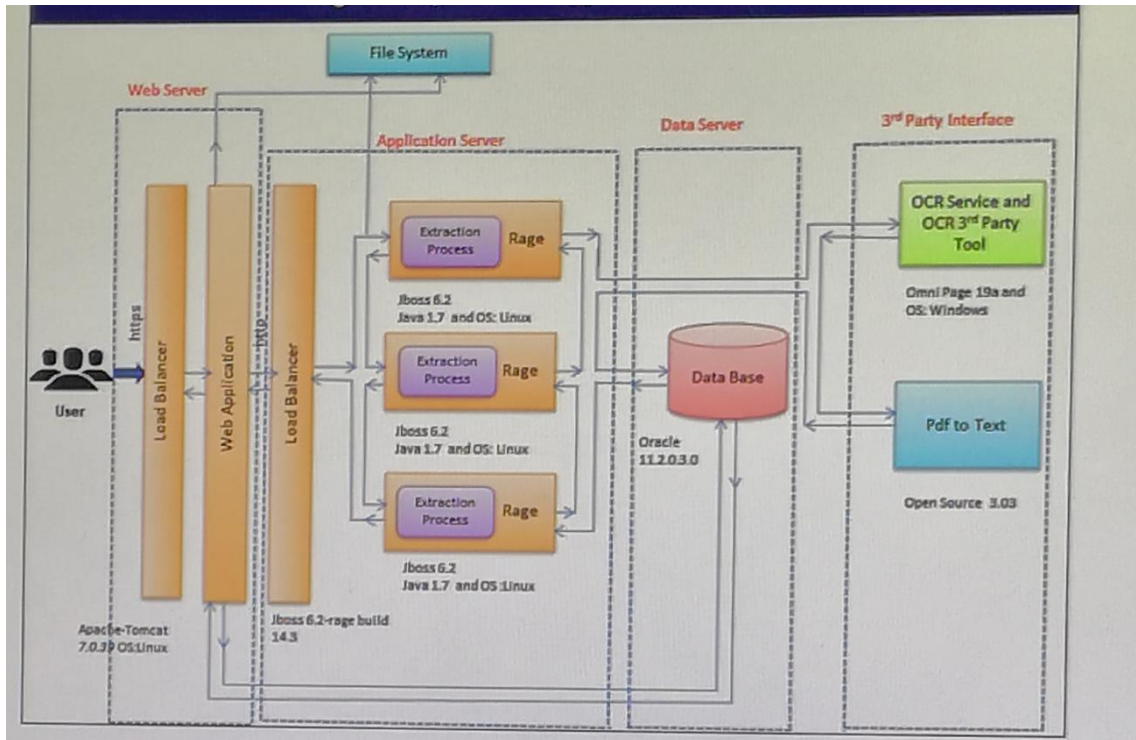
Data preprocessing means finding data that is relevant to our business logic out of the all the data. So, this is done in the extraction part and extraction team uses various data science algorithms to prepare and process data as per our business requirement.

4.2.1Missing Values

Missing value concept is very important. Missing value means data in the financial document which is important data and not mapped to the extraction rules correctly or half of the data is mapped or half of the data is not mapped. This is the rare case but if it comes then we can map that data by manually and add that rule in our extraction rules so that if next time this case is comes then it can detect it automatically. So, there is manual testers for this purpose who can verify the results of the extracted data and confirms whether the data is extracted correctly or not.

4.2.2Irrelevant Data

Irrelevant Data means that is not our useable data which we can find by our extraction rules. If any such data is there then it can go to the supplementary section. So supplementary section simply means data that is not mapped to our extraction rules or not related to the financial statements. So Irrelevant Data is nothing but the data that is not most likely the financial data.



4.3 ALGORITHM (METHODS) or WORKFLOW

4.3.1 EXTRACTION

Extraction is the first phase of every application or project. In this phase we extract the relevant information from our customer's financial document that they upload on our main application landing page. In Extraction phase we use the various ML algorithms to extract and finding useful data from the financial document. LiveSpread provides a fast, consistent way to spread the numbers, automating the painstaking part of financial spreading.

4.3.2 MAPPING

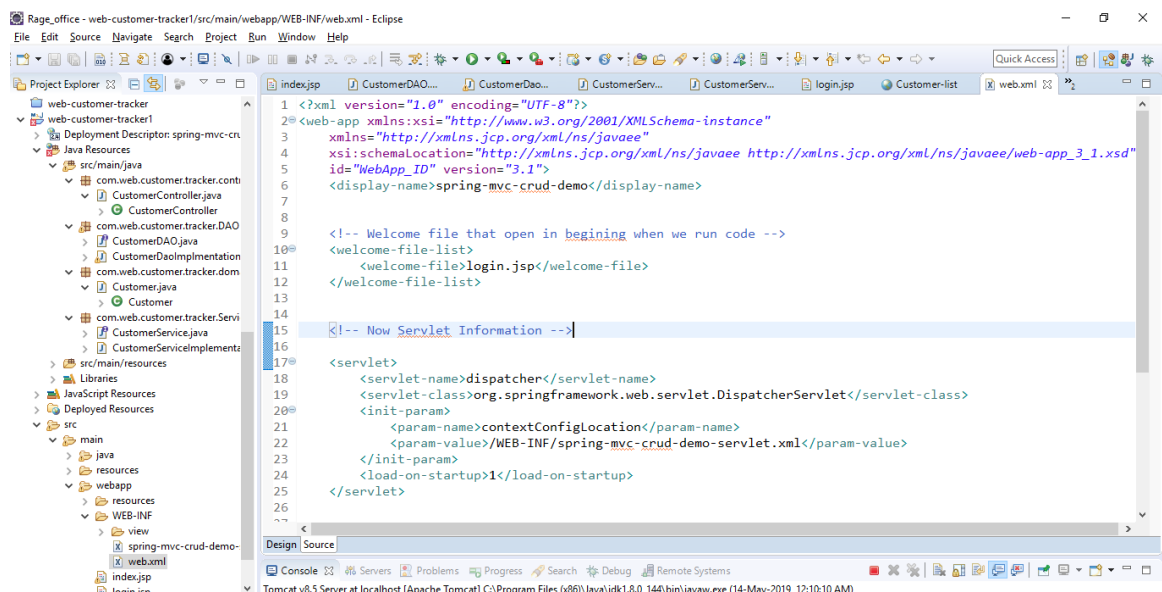
Mapping is the second phase of our application. In this we apply the business logic or the logic according to the customer requirements. So, there are many mapping methods that we are used and it is totally based on the requirements of the user and based on the requirements of user we find whichever is suitable mapping method that we used for this and then we apply that method on the financial data.

4.3.3 NORMALIZATION

Normalization is the last step in our application and this is the approach that we use across all project. So, in this section we develop a user interface for our customers so that they can communicate and handle the application easily. Main motivation behind this section is develop an application or UI which totally user friendly and user can easily understand the application so that there is no problem with the UI. We add different functionalities in our UI and it is as per the requirements of the customer.

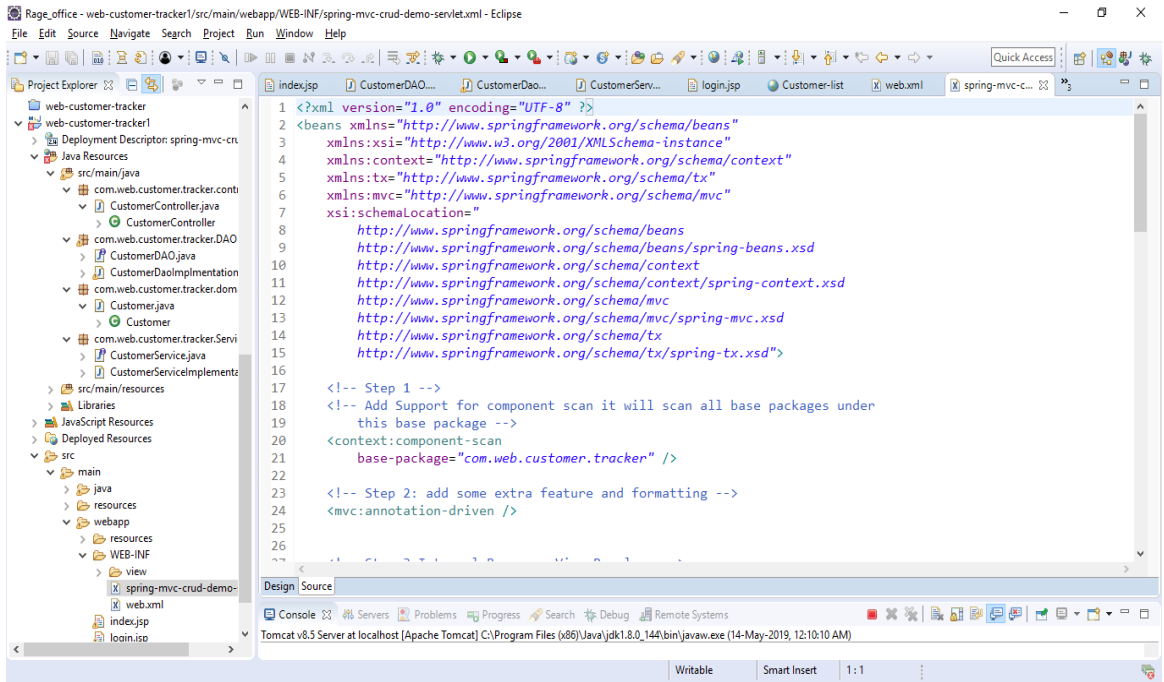
4.4 CODE SNIPPET

4.4.1 WEB.xml

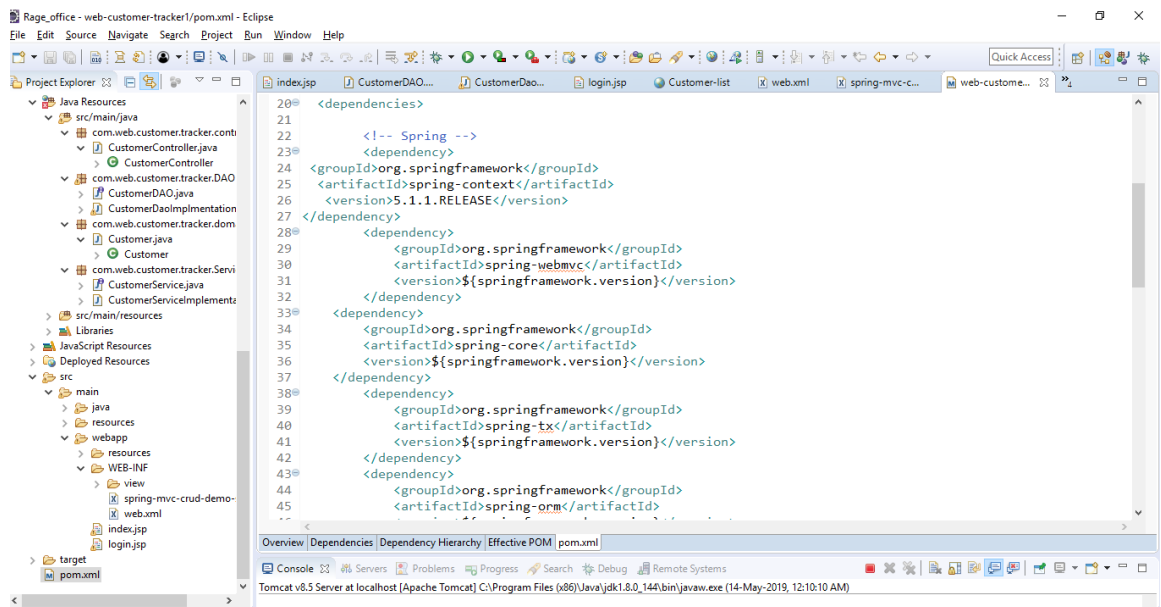


```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3         xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
5         id="WebApp_ID" version="3.1">
6     <display-name>spring-mvc-crud-demo</display-name>
7
8
9     <!-- Welcome file that open in begining when we run code -->
10    <welcome-file-list>
11        <welcome-file>login.jsp</welcome-file>
12    </welcome-file-list>
13
14
15    <!-- Now Servlet Information -->
16
17    <servlet>
18        <servlet-name>dispatcher</servlet-name>
19        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
20        <init-param>
21            <param-name>contextConfigLocation</param-name>
22            <param-value>/WEB-INF/spring-mvc-crud-demo-servlet.xml</param-value>
23        </init-param>
24        <load-on-startup>1</load-on-startup>
25    </servlet>
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

4.4.2 SERVLET FILE



4.4.3 POM.XML FILE



4.4.4 DOMAIN OR POJO CLASS

```

1 package com.web.customer.tracker.domain;
2
3 import javax.persistence.Column;
4
5
6
7
8
9
10 @Entity
11 @Table(name="customer")
12 public class Customer {
13
14     @Id
15     @GeneratedValue(strategy=GenerationType.IDENTITY)
16     @Column(name="id")
17     private int id;
18
19     @Column(name="firstname")
20     private String first_name;
21
22     @Column(name="lastname")
23     private String last_name;
24
25     @Column(name="email")
26     private String email;
27
28     public Customer() {
29
30     }
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

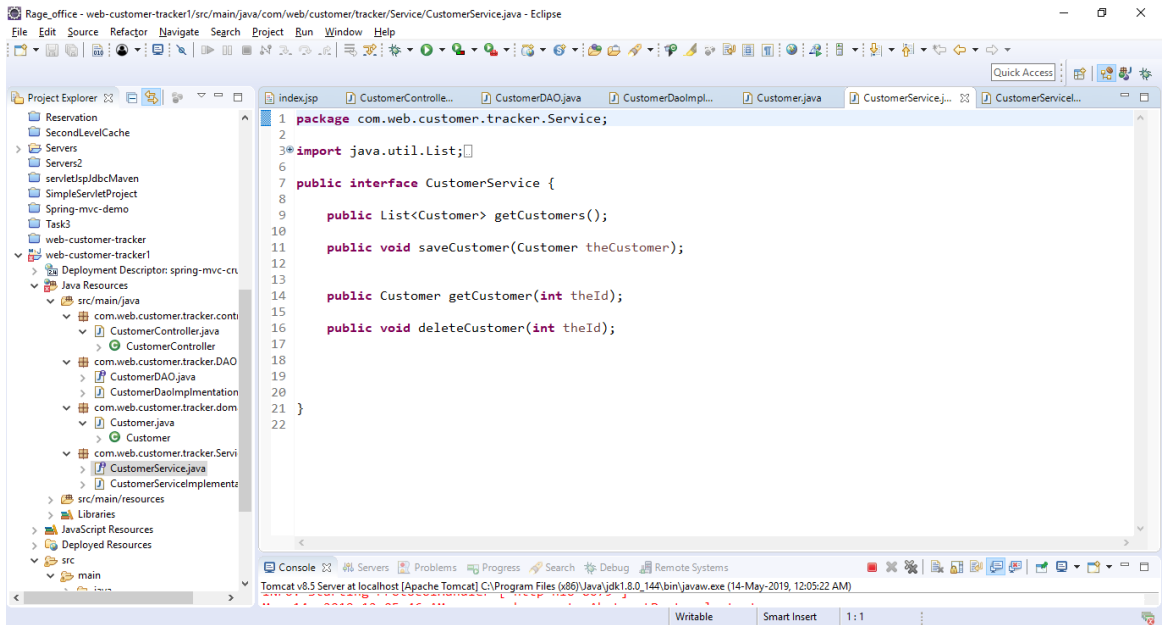
4.4.5 CONTROLLER CLASS

```

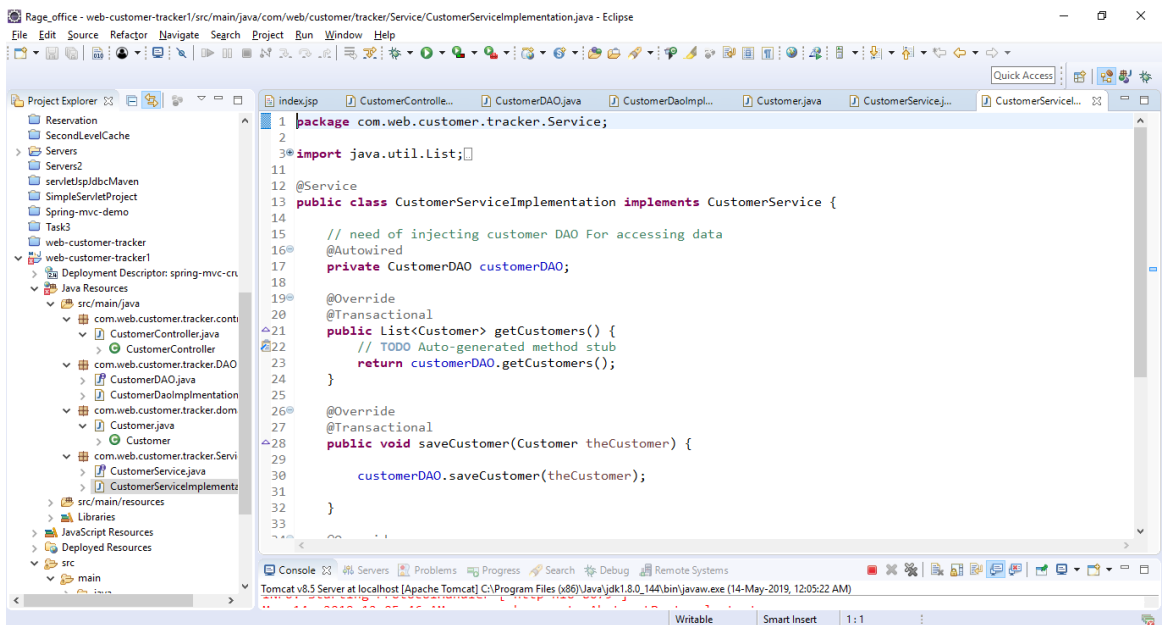
1 package com.web.customer.tracker.controller;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 @Controller
23 @RequestMapping("/customer")
24 public class CustomerController {
25
26
27
28     // for injecting customerService which used customerDAO----in controller class
29     @Autowired
30     private CustomerService customerService;
31     @GetMapping("/list")
32     public String listcustomer(Model model) {
33         // get customer by using interface dao object
34         List<Customer> theCustomer = customerService.getCustomers();
35
36         // now add thecustomer object into spring mvc model
37         model.addAttribute("customers", theCustomer);
38
39         return "customers-list";
40     }
41
42     @RequestMapping("/LoginServlet")
43     protected void doPost(HttpServletRequest request, HttpServletResponse response)
44         throws ServletException, IOException {
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

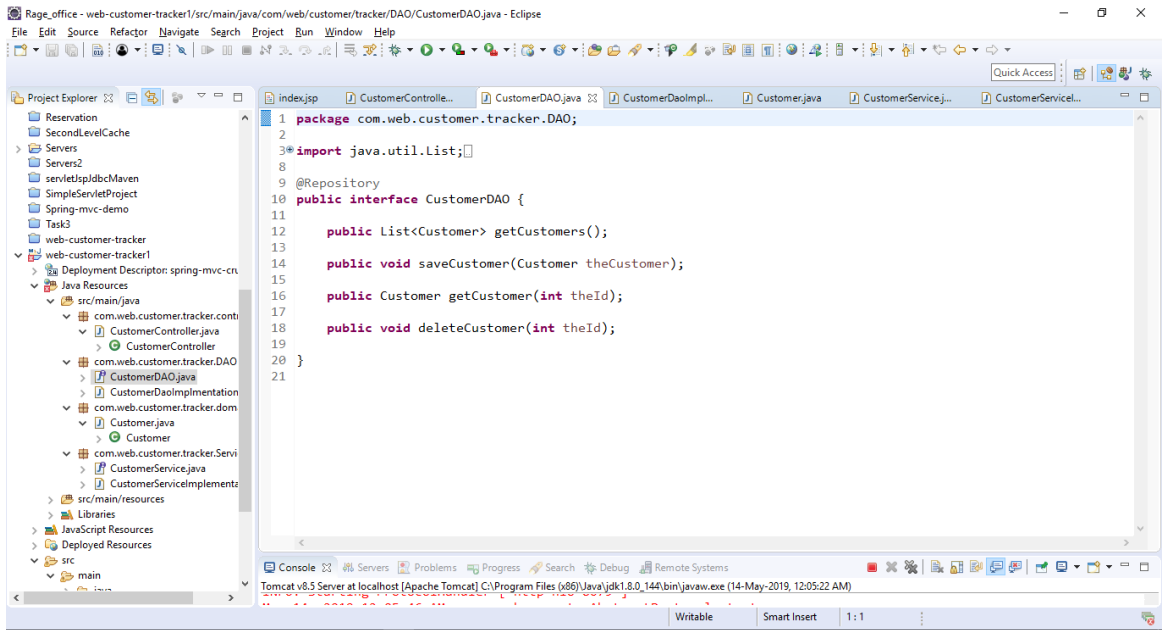
4.4.6 SERVICE INTERFACE



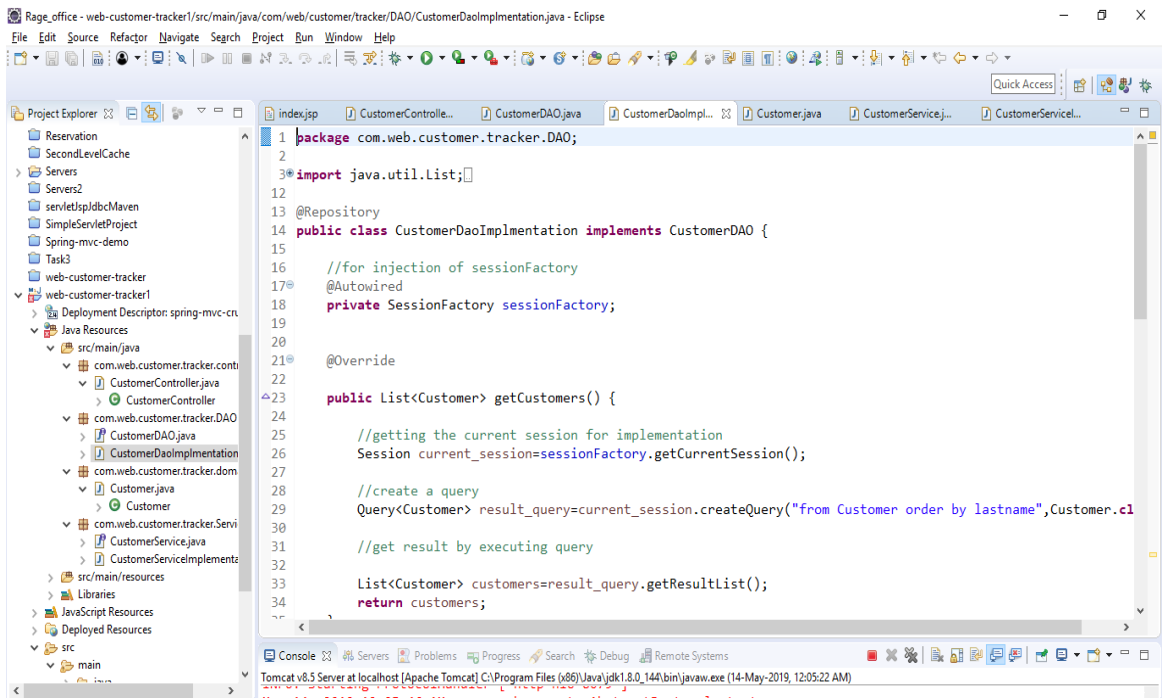
4.4.7 SERVICE IMPLEMENTATION



4.4.8 DAO INTERFACE



4.4.9 DAO IMPLEMENTATION



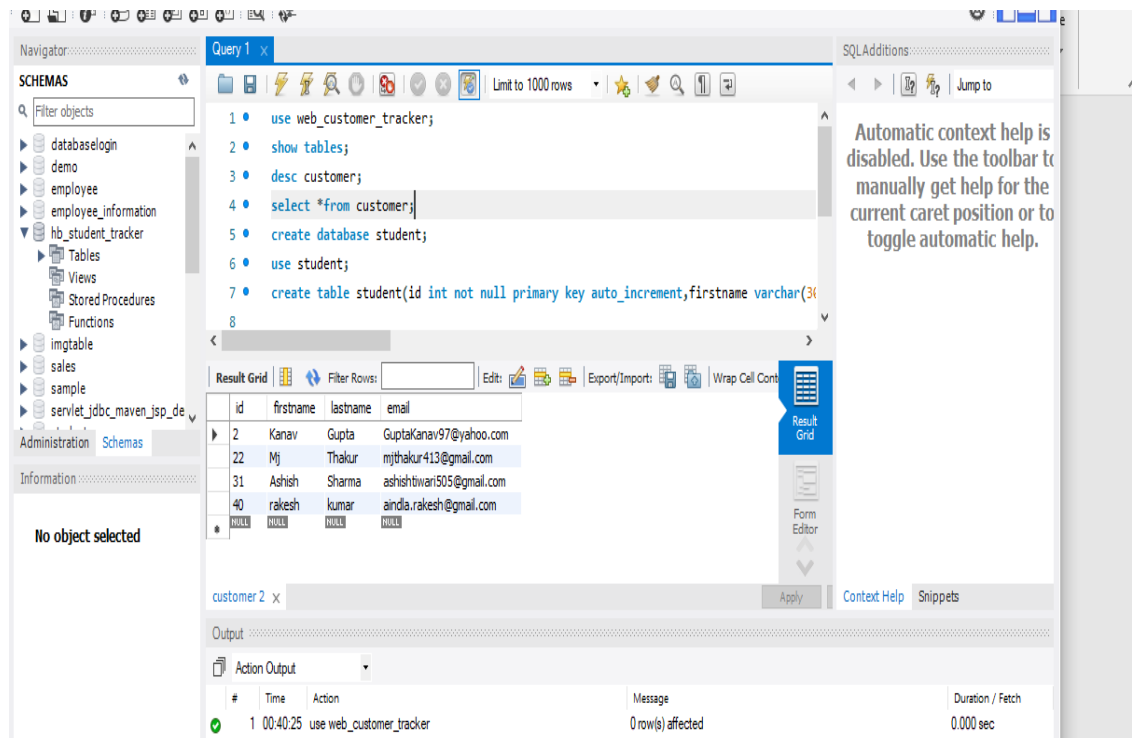
4.4.10 LOGIN.JSP

The screenshot shows the Eclipse IDE with the following components:

- Project Explorer:** Shows the project structure for 'web-customer-tracker'. The 'login.jsp' file is selected under 'src/main/resources'.
- Code Editor:** Displays the content of 'login.jsp'. The code is as follows:

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
7 <title>Login Page</title>
8 <meta charset="utf-8">
9 <meta name="viewport" content="width=device-width, initial-scale=1">
10 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
11 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
12 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
13 </head>
14 <body>
15
16
17 <div class="container">
18 <div class="row">
19 <div class="col-md-4"></div>
20 <div class="col-md-4">
21 <h2>Details form</h2>
22 <form class="form-horizontal" action="customer/LoginServlet" autocomplete="off" >
23 <div class="form-group">
24 <label class="control-label col-md-3" for="email">Email:<br></label>
25 <div class="input-group"> <span class="input-group-addon"><i class="glyphicon glyphicon-user">
26 </i></span>
27
```
- Console:** Shows the Tomcat v8.5 Server at localhost [Apache Tomcat] C:\Program Files (x86)\Java\jdk1.8.0_144\bin\javaw.exe (14-May-2019, 12:05:22 AM) with a timestamp of May 14, 2019, 12:05:46 AM.

4.4.11 DATABASE TABLE



The screenshot displays a database management interface with the following components:

- Navigator:** Shows a tree view of schemas including 'databaselogin', 'demo', 'employee', 'employee_information', 'hb_student_tracker', 'imgtable', 'sales', 'sample', and 'servlet_jdbc_maven_jsp_de'.
- Query Editor:** Contains the following SQL script:

```
1 use web_customer_tracker;
2 show tables;
3 desc customer;
4 select *from customer;
5 create database student;
6 use student;
7 create table student(id int not null primary key auto_increment,firstname varchar(30)
8
```
- Result Grid:** Displays the output of the 'select *from customer;' query. The table has columns 'id', 'firstname', 'lastname', and 'email'. The data rows are:

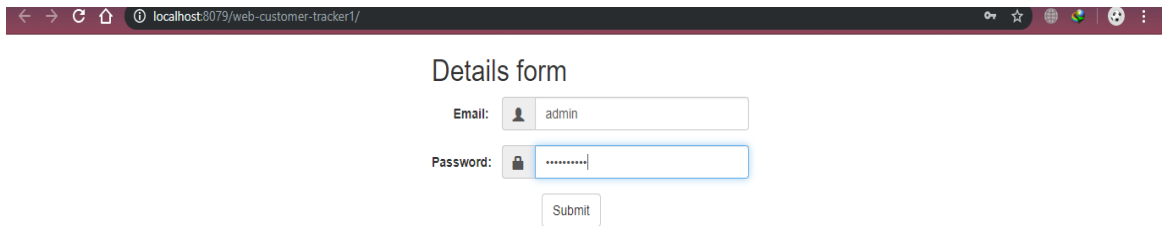
id	firstname	lastname	email
2	Kanav	Gupta	GuptaKanav97@yahoo.com
22	Mj	Thakur	mjthakur413@gmail.com
31	Ashish	Sharma	ashishtiwari505@gmail.com
40	rakesh	kumar	ainda.rakesh@gmail.com
- Action Output:** Shows a log of database actions:

#	Time	Action	Message	Duration / Fetch
1	00:40:25	use web_customer_tracker	0 row(s) affected	0.000 sec

This is the database table information that we create in the relational database and here we use the MySQL database version.

4.5 UI SNIPPET

4.5.1 LANDING PAGE



4.5.2 READ OPERATION

The screenshot shows the Eclipse IDE interface. The Project Explorer on the left displays the project structure for 'web-customer-tracker1'. The main editor window shows a web browser displaying the CRM - Customer Relationship Manager landing page. The page has a green header with the title 'CRM - Customer Relationship Manager' and an 'Add Customer' button. Below the button is a table with the following data:

First Name	Last Name	Email	Action
Kanav	Gupta	GuptaKanav97@yahoo.com	Update Delete
rakesh	kumar	aindla.rakesh@gmail.com	Update Delete
Ashish	Sharma	ashishtiwari505@gmail.com	Update Delete
Mj	Thakur	mjthakur413@gmail.com	Update Delete
Nitish	Tiwari	nitishtiwari97@gmail.com	Update Delete

4.5.3 ADD OPERATION

The screenshot shows the Eclipse IDE with a web browser window displaying the 'Customer Relationship Manager' application. The browser address bar shows the URL: `http://localhost:8079/web-customer-tracker1/customer/showFormForAdd`. The page content includes a green header with the text 'Customer Relationship Manager' and a section titled 'Save Customer'. Below this, there are three input fields: 'First Name' with the value 'Nitish', 'Last Name' with the value 'Kumar', and 'Email' with the value 'kumarnitish@genpact.com'. A 'save' button is located below the form, and a link labeled 'Back to Customer List' is positioned below the button. The Project Explorer on the left side of the IDE shows the project structure for 'web-customer-tracker', including packages like 'com.web.customer.tracker.controller', 'com.web.customer.tracker.DAO', and 'com.web.customer.tracker.domain'. The console at the bottom shows the Tomcat v8.5 Server log.

4.5.4 UPDATE OPERATION

The screenshot shows the Eclipse IDE with a web browser window displaying the 'Customer Relationship Manager' application. The browser address bar shows the URL: `http://localhost:8079/web-customer-tracker1/customer/showFormForUpdate?customerId=54`. The page content includes a green header with the text 'Customer Relationship Manager' and a section titled 'Save Customer'. Below this, there are three input fields: 'First Name' with the value 'Nitish', 'Last Name' with the value 'pandit', and 'Email' with the value 'kumarnitish@genpact.com'. A 'save' button is located below the form, and a link labeled 'Back to Customer List' is positioned below the button. The Project Explorer on the left side of the IDE shows the project structure for 'web-customer-tracker', including packages like 'com.web.customer.tracker.controller', 'com.web.customer.tracker.DAO', and 'com.web.customer.tracker.domain'. The console at the bottom shows the Tomcat v8.5 Server log.

4.5.5 DELETE OPERATION

The screenshot shows the Eclipse IDE interface with a web browser displaying the CRM application. The browser address bar shows `http://localhost:8079/web-customer-tracker/customer/list`. The page title is "CRM - Customer Relationship Manager". There is an "Add Customer" button and a table of customers. A modal dialog box is open over the table, asking "Are you sure you want to delete this customer?" with "OK" and "Cancel" buttons.

First Name	Last Name	Email	Action
Kanav	Gupta	GuptaKanav97@yahoo.com	Update Delete
rakesh			Update Delete
Nitish			Update Delete
Ashish			Update Delete
Mj			Update Delete
Nitish			Update Delete

Chapter – 5

RESULT & PERFORMANCE ANALYSIS

As we know Result and performance analysis is one of the important measures of every web application. So generally, we use the caching concept in web application to improve the performance. Because every time we not hit the database if same query is fired so that's why caching concept is important. So, there are two types of cache first level cache and second level cache. First level cache is associated with the session and second level cache is associated with the session factory.

Performance means how much data is extracted and mapped correctly out of all the relevant data based on rules stored in the LS product library. Application execution will dependably be influenced by asset limitations. Scalability is the capacity to conquered execution restrains by including assets. Regardless of how much equipment we have, at one point, we will see diminishing execution. This implies expanding reaction times or a breaking point in throughput. Will including extra equipment tackle the issue? In the event that truly, at that point we can scale. If not, we have a Scalability issue. At the end of the day, Scalability is a proportion of how including assets (typically equipment) influences execution. A versatile framework is one that enables you to include equipment and get a comparable execution improvement, for example, multiplying what number of servers you need to twofold your throughput. Vertical adaptability, or scaling up, implies adding more capacity to a solitary server, for example, more memory. Flat versatility, or scaling out, implies including more servers.

Chapter – 6

CONCLUSION

Livespread can speed up your financial spreading for commercial underwriting and risk management. It automates the tedious steps involved in spreading financial statements—and does so with fewer errors. All this means you can rapidly analyse data to make faster, smarter credit decisions. LiveSpread uses artificial intelligence (AI) to pull numbers from documents and online systems. Then it puts them into your preferred format for quick, fully auditable insights that you can easily trace back to the source. You get faster cycle times, higher underwriter productivity, improved compliance, and greater operational consistency. With Cora Livespread, you can power through more data. Even when your volume peaks, you're ready to handle the load, so your operating model stays stable. No more worrying about staffing in busy seasons.

In finance, rules and regulations are inevitable. Cora Livespread learns and remembers all the fine print, which means you spend more time analysing data instead of double-checking the inputs. But if you do need to check something, you're a click away from the source.

REFERENCES

1. Craig Walls, Ryan Breidenbach. Spring in Action 2nd Edition. Manning Publishing Co. 2008.
2. Laufer, K. A hike through post-EJB J2EE Web application architecture. Computing in Science & Engineering. Volume 7, Issue 5, Sept.-Oct. 2005 Page(s):80 – 88.
3. Johnson, R. J2EE development frameworks. Computer Volume 38, Issue 1, Jan. 2005 Page(s):107 – 110.
4. The Spring Framework Reference Documentation.
<http://static.springframework.org/spring/docs/2.5.x/reference/index.html>
5. Apache Maven
<http://maven.apache.org/>
6. The Java EE 5 Tutorial. <http://java.sun.com/javaee/5/docs/tutorial/doc/>