End to End Project Assignment Solutions

Assignment 1 : Data Access Layer Assignment

```
     create table projectdb.Customer(
  id int PRIMARY KEY AUTO_INCREMENT,
  name varchar(50),
  email varchar(50)
);
```

```
package com.nickodemus.customer;
import com.nickodemus.customer.entities.Customer;
import com.nickodemus.customer.repositories.CustomerRepository;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;
@RunWith(SpringRunner.class)
@SpringBootTest
```

```
      public class CustomerdataApplicationTests {
 @Autowired
 CustomerRepository customerRepository;
 @Test
 public void createCustomerTest() {
 Customer customer = new Customer();
      customer.setName("John");
 customer.setEmail("john@arbuckle.com");
 customerRepository.save(customer);
}
@Test
public void findCustomerTest() {
Customer customer = customerRepository.findOne(1L);
System.out.println(customer);
}
@Test
public void updateCustomerTest() {
Customer customer = customerRepository.findOne(1L);
```

```
  customer.setName("Joan");
  customerRepository.save(customer);
  System.out.println(customer);
  }
  @Test
  public void deleteCustomerTest() {
  customerRepository.delete(1L);
  }
}
        package com.nickodemus.customer.repositories;
        import com.nickodemus.customer.entities.Customer;
        import org.springframework.data.repository.CrudRepository;
        public interface CustomerRepository extends CrudRepository<Customer,Long>{
}
        package com.nickodemus.customer.entities;
        import lombok.Data;
        import lombok.ToString;
```

```
    import javax.persistence.*;
    @Data
    @Entity
    @ToString
    public class Customer {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String name;
    private String email;
}
    spring.datasource.url=jdbc:mysql://localhost:3306/projectdb
    spring.datasource.username=root
    spring.datasource.password=
    spring.jpa.show_sql=true
```

Assignment 2 :   Create a Vendor Module

```
    @Controller
    public class VendorController {
    @Autowired
    private VendorService service;
    @RequestMapping({"/", "index"})
    public String index(ModelMap modelMap) {
    modelMap.addAttribute("vendors", service.getAll());
    return "index";
}
    @RequestMapping("/create")
    public String create() {
    return "create";
}
    @RequestMapping("/edit")
    public String edit(@RequestParam("id") int id, ModelMap modelMap) {
    modelMap.addAttribute("vendor", service.getById(id));
    return "edit";
}
    @RequestMapping("/vendor-create")
```

```
        public String vendorCreate(@ModelAttribute("vendor") Vendor vendor,
ModelMap modelMap) {

        service.save(vendor);

        modelMap.addAttribute("msg", "Vendor saved with id: " +
vendor.getId());

        return "create";

}

        @RequestMapping("/vendor-update")

        public String vendorUpdate(@ModelAttribute("vendor") Vendor vendor,
ModelMap modelMap) {

        service.save(vendor);

        modelMap.addAttribute("vendors", service.getAll());

        return "index";

}

        @RequestMapping("/vendor-delete")

        public String vendorDelete(@RequestParam("id") int id, ModelMap
modelMap) {

        service.delete(id);

        modelMap.addAttribute("vendors", service.getAll());

        return "index";
}
}
        @Entity
        public class Vendor {
        @Id
        private int id;
        private String code;
        private String name;
        private String type;
        private String email;
        private String phone;
        private String address;
        public int getId() {
        return id;
}
        public void setId(int id) {
        this.id = id;
}
```

```
        public String getCode() {

        return code;

}

        public void setCode(String code) {

        this.code = code;

}

        public String getName() {

        return name;

}

        public void setName(String name) {

        this.name = name;

}

        public String getType() {

        return type;

}

        public void setType(String type) {

        this.type = type;

}
```

```
    public String getEmail() {
    return email;
}
    public void setEmail(String email) {
    this.email = email;
}
    public String getPhone() {
    return phone;
}
    public void setPhone(String phone) {
    this.phone = phone;
}
    public String getAddress() {
    return address;
}
    public void setAddress(String address) {
    this.address = address;
}
```

```
        @Override

        public String toString() {

        return "Vendor [id=" + id + ", code=" + code + ", name=" + name + ",
type=" + type + ", email=" + email
        + ", phone=" + phone + ", address=" + address + "]";

}

}

        public interface VendorRepository extends JpaRepository<Vendor,
Integer> {

}

        public interface VendorService {

        Vendor save(Vendor vendor);

        Vendor update(Vendor vendor);

        void delete(Vendor vendor);

        void delete(int id);

        Vendor getById(int id);

        List<Vendor> getAll();

}

        @Service

        public class VendorServiceImpl implements VendorService {
}

    @Autowired
    private VendorRepository repository;
    @Override
    public Vendor save(Vendor vendor) {
    return repository.save(vendor);
}
    @Override
    public Vendor update(Vendor vendor) {
    return repository.save(vendor);
}
    @Override
    public void delete(Vendor vendor) {
    repository.delete(vendor);
}
    @Override
    public void delete(int id) {
    repository.delete(id);
}
```

```
        @Override

        public Vendor getById(int id) {

        return repository.findOne(id);

}

        @Override

        public List<Vendor> getAll() {

        return repository.findAll();

}

}

        <%@ page language="java" contentType="text/html;
charset=ISO-8859-1"

        pageEncoding="ISO-8859-1"%>

        <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

        <%@ page isELIgnored="false" %>

        <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

        <html>

        <head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Vendor - Create</title>

</head>

<body>

<h2>Vendor - Create</h2>

<form action="vendor-create" method="post">

<pre>

ID: <input type="text" name="id" /><br/>

Code: <input type="text" name="code" /><br/>

Name: <input type="text" name="name" /><br/>

Type: <select name="type">

<option>Default</option>

<option>Custom</option>

</select><br/>

Email: <input type="text" name="email" /><br/>

Phone: <input type="text" name="phone" /><br/>

Address: <textarea rows="4" name="address"></textarea><br/>

<input type="submit" value="save"/>
```

```
</pre>

</form>

${msg}

<a href="index">View All</a>

</body>

</html>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<%@ page isELIgnored="false" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Vendor - Create</title>

</head>
```

```
        <body>

        <h2>Vendor - Edit</h2>

        <form action="vendor-update" method="post">

        <pre>

        ID: <input type="text" name="id" value="${vendor.id}" /><br/>

        Code: <input type="text" name="code" value="${vendor.code}"/><br/>

        Name: <input type="text" name="name" value="${vendor.name}"/><br/>

        Type: <select name="type">

        <option ${vendor.type=='Default'?'selected':''}>Default</option>

        <option ${vendor.type=='Custom'?'selected':''}>Custom</option>

        </select><br/>

        Email: <input type="text" name="email" value="${vendor.email}"/><br/>

        Phone: <input type="text" name="phone" value="${vendor.phone}" /><br/>

        Address: <textarea rows="4" name="address">${vendor.address}</textarea><br/>

        <input type="submit" value="save"/>

        </pre>

        </form>
        <a href="index">View All</a>

        </body>

        </html>

        <%@ page language="java" contentType="text/html;
charset=ISO-8859-1"

        pageEncoding="ISO-8859-1"%>

        <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

        <%@ page isELIgnored="false" %>

        <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

        <html>

        <head>

        <meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1">

        <title>Vendor - Index</title>

        </head>

        <body>

        <h2>Vendor - Index</h2>

        <table>
```

```
<tr>
<th>id</th>
<th>code</th>
<th>name</th>
<th>type</th>
<th></th>
</tr>
<c:forEach items="${vendors}" var="vendor">
<tr>
<td>${vendor.id}</td>
<td>${vendor.code}</td>
<td>${vendor.name}</td>
<td>${vendor.type}</td>
<td><a href="vendor-delete?id=${vendor.id}">delete</a> | <a href="edit?id=${vendor.id}">edit</a></td>
</tr>
</c:forEach>
</table>
<a href="create">create</a>
```

```
}

    </body>

    </html>
```

Assignment 3 : Integration Layer

```
    @RestController

    @RequestMapping("/vendors")

    public class VendorRestController {

    @Autowired

    private VendorRepository vendorRepository;

    @GetMapping

    public List<Vendor> getVendors() {

    return vendorRepository.findAll();

}

    @PostMapping

    public Vendor createVendor(@RequestBody Vendor vendor) {

    Vendor savedVendor = vendorRepository.save(vendor);

    return savedVendor;

}
```

```
    @PutMapping
    public Vendor updateVendor(@RequestBody Vendor vendor) {
    Vendor updatedVendor = vendorRepository.save(vendor);
    return updatedVendor;
}
    @DeleteMapping("/{id}")
    public void deleteVendor(@PathVariable("id") Integer id) {
    vendorRepository.delete(id);
}
    @GetMapping("/{id}")
    public Vendor getVendor(@PathVariable("id") Integer id) {
    return vendorRepository.findOne(id);
}
}
```

Assignment 4 : Check In Home Page

```
    <!DOCTYPE html>
    <html xmlns:th="http://www.thymeleaf.org">
```

```
        <head>

        <meta charset="UTF-8"/>

        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
/>

        <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></s
cript>

        <title>Check In Reservation Application</title>

        </head>

        <body>

        <div class="container">

   <br/>

   <h2>Welcome to the Check In Reservation Application: </h2>

   <br/>

   <a href="showStartCheckin" class="btn btn-info" role="button"
align="center">Go to Check In</a>

   </div>

        </body>
```

```
</html>
```

Assignment 5 : Externalized Configuration

```
server:

context-path: /flightcheckin

port: 9090

com:

fabian81:

flightcheckin:

url: http://localhost:8080/flightreservation/reservations/
```

```
import org.springframework.beans.factory.annotation.Value;

import org.springframework.stereotype.Component;

import org.springframework.web.client.RestTemplate;

import com.fabiang.flightcheckin.integration.dto.Reservation;

import com.fabiang.flightcheckin.integration.dto.ReservationUpdateRequest;

@Component

public class ReservationRestClientImpl implements ReservationRestClient {

@Value("${com.fabian81.flightcheckin.url}")
```

```
        private String RESERVATION_REST_URL;

        @Override

        public Reservation findReservation(Long id) {

        RestTemplate restTemplate = new RestTemplate();

        Reservation reservation =

        restTemplate.getForObject(RESERVATION_REST_URL+id,
Reservation.class);

        return reservation;

 }

        @Override

        public Reservation updateReservation(ReservationUpdateRequest
request) {

        RestTemplate restTemplate = new RestTemplate();

        Reservation reservation =
restTemplate.postForObject(RESERVATION_REST_URL, request,
Reservation.class);

        return reservation;

}

}
```

Assignment 6 : Transaction Management

```
        @Transactional

        @RequestMapping("/completeCheckIn")

        public String completeCheckIn(@RequestParam("reservationId") Long reservationId,

        @RequestParam("numberOfBags") int numberOfBags) {

        ReservationUpdateRequest reservationUpdateRequest = new ReservationUpdateRequest();

        reservationUpdateRequest.setId(reservationId);

        reservationUpdateRequest.setCheckedIn(true);

        reservationUpdateRequest.setNumberOfBags(numberOfBags);

        reservationRestClient.updateReservation(reservationUpdateRequest);

        return "checkInConfirmation";

 }
```