WEB ROUTING

Technical Manual

SLIPPERY ROCK UNIVERSITY

Joshua Gearhart jjg1018@sru.edu

Nicholas Bushée [nab1017@sru.edu](mailto:nab1017@sru.edu)

Fady Aziz [faa1002@sru.edu](mailto:faa1002@sru.edu)

Ian Black [imb1007@sru.edu](mailto:imb1007@sru.edu)

Logan Kirkwood llk1005@sru.edu

Table of Contents

[1. Overview 4](#_Toc120741837)

[2. System requirements 4](#_Toc120741838)

[3. Hardware Requirements 4](#_Toc120741839)

[4. webrouting application classes 5](#_Toc120741840)

[4.1 SPRING BOOT APPLICATION 5](#_Toc120741841)

[4.1.1 Spring boot application class information 5](#_Toc120741842)

[4.2 Controller Class 5](#_Toc120741843)

[4.2.1 controller class information 5](#_Toc120741844)

[**4**.3 Domain Entity Classes 6](#_Toc120741845)

[4.3.1 Domain Entity Class Information 6](#_Toc120741846)

[4.4 Repository Classes 7](#_Toc120741847)

[4.4.1 Repository Class Information 7](#_Toc120741848)

[4.5 Security Classes 7](#_Toc120741849)

[4.5.1 Security Class Information 8](#_Toc120741850)

[4.6 Emailing Classes 8](#_Toc120741851)

[4.6.1 Emailing Class Information 8](#_Toc120741852)

[4.7 Recaptcha Classes 8](#_Toc120741853)

[4.7.1 Recaptcha Class Information 8](#_Toc120741854)

[4.8 Service Class 9](#_Toc120741855)

[4.8.1 Service Class Information 9](#_Toc120741856)

[4.9 Configuration Classes 9](#_Toc120741857)

[4.9.1 Configuration Class Information 9](#_Toc120741858)

[5. Testing 9](#_Toc120741859)

[5.1 Junit Testing 9](#_Toc120741860)

[6. UML Diagrams 9](#_Toc120741861)

[6.1 Class Diagrams 9](#_Toc120741862)

[6.1.1 9](#_Toc120741863)

[6.2 Sequence Diagrams 10](#_Toc120741864)

[6.2.1 10](#_Toc120741865)

[6.3 Activity Diagrams 15](#_Toc120741866)

[6.3.1 15](#_Toc120741867)

[6.4 State Diagrams 19](#_Toc120741868)

[6.4.1 19](#_Toc120741869)

[7. Figures 21](#_Toc120741870)

# Overview

# System requirements

**1.Operating System**

* 1. ***Required:*** Windows
     1. ***Version:*** 10
     2. ***Link:*** https://www.microsoft.com/en-us/software-download/windows10

**2.Java IDE**

* 1. ***Required:*** Eclipse IDE
     1. ***Version:*** 2022-06
     2. ***Link:*** https://www.eclipse.org/downloads/packages/release/2022-06/r/eclipse-ide-java-developers

**3.Database**

* 1. ***Required:*** MySQL Server & Workbench
     1. ***Version*** 8.0.30
     2. ***Link:*** https://downloads.mysql.com/archives/installer/

**4.Internet Browser**

* 1. Any updated Internet Browser
     1. ***Recommended:*** Updated Google Chrome
     2. ***Link:*** https://www.google.com/chrome/dr/download/?brand=CHBD&geo=US&gclid=Cj0KCQiAsoycBhC6ARIsAPPbeLsLUy2IR\_YPNp57SJUvP5Eobf3w9DZjd2dPmuhr4jSZOqSqYLZwUsUaAoCMEALw\_wcB&gclsrc=aw.ds

# *3.* HARDWARE REQUIREMENTS

**CPU:**

***Multi Core 64bit x86 architecture CPU***

**RAM:**

***12 GB or higher***

**Display:**

***1024x768***

**Storage:**

***250 GBs***

***Note:*** These metrics are for low-end optimal performance

# 4. webrouting application CLASSES

This section contains functional information on each class organized by what the class handles.

## 4.1 SPRING BOOT APPLICATION

This section refers to the “main” of the Spring Boot Application.

See section 4.1.1 for more information.

### 4.1.1 Spring boot application class information

***Class can be located at: edu.sru.thangiah.webrouting***

1. ***WebroutingApplication:*** Main class, used to start the application***(@SpringBootApplication*** annotation), ***@EnableAsync*** annotation configures the **Spring Boot Application** to allow running You will run this class as a **Java Application** to run the whole project ( See Note for more details).

***Note:*** Install Spring tools in eclipse to bypass netstat command.

## 4.2 cONTROLLER CLASSES

See Sub-section 4.2.1 for information on each Controller class of the Webrouting application controller classes are web handlers. All controller classes are annotated with **@Controller.** Controller classes documented for programmer readability and understanding.

**Note**: Attributes are used to view information to users

**Ex**: Model objects, etc.

### 4.2.1 controller class information

***Classes can be located at: edu.sru.thangiah.webrouting.controller***

**Deprecated**: *Constructor methods* are only used for testing, **@Autowired** Annotation handles this in Application

1. ***CarriersController***: The Carriers Controller class handles all methods of the carrier pages this can be viewed in html source at ***carriers.html***, or the user interface ***localhost:8080/carriers*** when the application is running.
2. **ContactsController**: Handles the controls for pages dealing with Contacts. Viewed in html source at ***contacts.html*** or via the user interface ***localhost:8080/carriers*** when the application is running.
3. **DriverController:** Web handler for pages dealing with Drivers. Viewed in html source at ***drivers.html*** or via the user interface ***localhost:8080/drivers*** when the application is running
4. **LocationController:** Web handler for pages dealing with Locations. Viewed in html source at

***Locations.html*** or via the user interface ***localhost:8080/locations*** when the application is running

1. **LoginController:** Web handler for pages dealing with Logging in and Registration. Viewed in html source at ***registrationhome.html, registrationshipper.html, registrationcarrier.html and index.html*** or via the user interface ***localhost:8080/login or by following the create account registration***  when the application is running
2. **MaintenanceOrderController:** Web handler for the Thymeleaf controls for pages dealing with maintenance orders. Viewed in html source at ***maintenanceorders.html,*** or via the user interface ***localhost:8080/maintenanceorders*** when the application is running
3. **RolesController:** Web controller for handling web aspects to the ***roles.html*** page and can be viewed at ***localhost:8080/roles***
4. **RoutesController:** Handles the Web controls for pages dealing with Routes, and functions dealing with routes Viewed in html source at ***routes.html*** or can be accessed via the user interface ***localhost:8080/routes***
5. **ShipmentsController:** Handles web controls for pages dealing with Shipments in all aspects. Html source can be located at ***shipments.html*** or via the user interface at ***localhost:8080/shipments***
6. **TechniciansController:** Works with web controls for pages working on technicians. Html source can be located at ***technicians.html*** or by accessing the user interface at ***localhost:8080/technicians.***
7. **UserController:** Handles the web controls for pages dealing with Users. Notice Mapping Annotations above methods for html and user interface.
8. **VehiclesController:** Class for handling all web controls dealing with vehicles.
9. **VehicleTypesController**: Handles the web controls for pages dealing with Vehicle Types.
10. **BidsController**: Web controller class dealing with all bid’s methods.
11. **VerifyEmailController**: Web controller class dealing with verifying a user’s email address and workflow of that process once registered. Follow mapping annotations for html and user interface.
12. **ForgottenPasswordController**: Web handling class working with a user forgetting their password. Navigate mapping annotations for html and user interface.

## **4**.3 Domain Entity Classes

This section highlights the Domain Classes, Each domain class is an entity located in the webrouting schema. Domain classes are annotated with ***@Entity*** to apply this as well as explicitly named with ***@Table*** annotation to define table names.

### 4.3.1 Domain Entity Class Information

***Classes can be located at: edu.sru.thangiah.webrouting.domain***

View section 4.3.1.1 for detailed information on Domain Class annotations.

1. **Bids**: Defines Bids entity and contains methods associated with updating and retrieving information on bids.
2. **Carriers**: Defines Carriers entity and contains methods associated with updating and retrieving information on carriers.
3. **Contacts**: Defines Contacts entity and includes methods associated with updating and retrieving information on carriers.
4. **Driver**: Defines Driver entity and implements methods for updating and retrieving information on drivers.
5. **Locations**: Creates the locations entity and implements methods for retrieval and updating of locations
6. **MaintenanceOrders**: Defines Maintenance Orders entity and creates methods for updating and retrieving maintenance orders.
7. **Role**: Sets up Role entity and methods allowing access to the Roles entity.
8. **Shipments**: Creates the Shipments Entity and methods allowing access to the Shipments Entity as well as updating and creating these entities.
9. **Technicians**: Generates Technicians entity and methods associated with assigning values and getting them.
10. **User**: Creates the User entity. Has ***@Transient*** Annotation for values not persisting into the database. Allows for user to be updated and user information retrieval.
11. **Vehicles**: Defines Vehicles entity and implements methods for updating and retrieving information on vehicles.
12. **VehicleTypes**: Creates the Vehicle Types entity and implements methods for updating and retrieving information on vehicle types.

#### 4.3.1.1 Annotation Information

Domainannotations are persisted through domain classes. **@Id** annotation signifies a primary key, **@GenericGenerator** and **@GeneratedValue** are coupled to automatically create user Id’s and incremented for each Id. Other annotations include **@Column** definition annotation to set values like name, nullability, variable type and length as well as default values. Other annotations included in domain classes are **@JoinColumn** annotations and **@OneToMany** as well as **@ManyToOne** annotations.

## 4.4 Repository Classes

Section 4.4 highlights the Repository Classes

### 4.4.1 Repository Class Information

***Classes can be located at: edu.sru.thangiah.webrouting.repository***

All Repository classes are linked to domain classes by domain data their perspective domain data type (See Section 4.3) repository classes extend CRUD Repositories (With exception to the User Repository which is extending the JpaRepository) allows for Spring Framework to interface Entity classes with Database and repository.

**Note:** UserRepository is the only class with defined methods for finding information regarding a User entity. **@Query** annotations are used to define a query upon method call.

## 4.5 seCURITY CLASSES

This section highlights classes focused on security, commonly implementing Spring Framework classes.

### 4.5.1 Security Class Information

**Deprecated**: *Constructor methods* are only used for testing, **@Autowired** Annotation handles this in Application

1. **WebSecurityConfig:** extends Spring’s **WebSecurityConfigurerAdapter**, used to explicitly define authority between pages based on roles and permissions. Also defines login page as well as logging out, continuing to define exception handling and access denied handling.
   1. **Class can be located at:**  ***edu.sru.thangiah.webrouting.config***
2. **UserValidator:** Implements Spring’s Validator, used in aspects of the program to validate and approve user input or deny using conditional statements which provide messages to be created to send to a user.
   1. **Class can be located at:**  ***edu.sru.thangiah.webrouting.web***
3. **SecurityServiceImpl:** Class used for checking authentication within the context as well as permitting autologin, annotated with **@Service**
   1. **Class can be located at: edu.sru.thangiah.webrouting.services**
4. **UserDetailsServiceImpl:** Class used for loading a user by their username and granting authorities with class annotated with **@Service** annotation, as well as a **@Transactional** annotation.
   1. **Class can be located at: edu.sru.thangiah.webrouting.services**
5. **MyAccessDeniedHandler:** Implements Springs **AccessDeniedHandler** to implement an access denied page to the user.
   1. **Class can be located at: edu.sru.thangiah.webrouting.error**

## 4.6 Emailing Classes

This section highlights Emailing Classes and talks about the classes dealing with sending emails to users within the Webrouting Application.

### 4.6.1 Emailing Class Information

**Deprecated**: *Constructor methods* are only used for testing, **@Autowired** Annotation handles this in Application

**Classes can be located at: edu.sru.thangiah.webrouting.mailsending**

1. **EmailingImpl:** Implements the Emailing class and is annotated as a service, contains methods relevant to sending any form of emails to users.
2. **MailSending:** Class used for a helper to the **EmailingImpl** for parsing applications servlet path.

## 4.7 Recaptcha Classes

Section 4.7 overviews classes relating to the implementation of the Google ReCAPTCHA v2.

### 4.7.1 Recaptcha Class Information

**Deprecated**: *Constructor methods* are only used for testing, **@Autowired** Annotation handles this in Application

**Classes can be located at: edu.sru.thangiah.webrouting.captcha**

1. **Recaptcha:** Class used to receive Captcha responses, and post the object to the API for Google ReCAPTCHA
2. **RecieveCaptcha:** Helper class used for googles response values are annotated with the **@JsonProperty** annotation due to how captcha responds.

## 4.8 Service Class

This section will talk about service classes within the **Webrouting Application**.

### 4.8.1 Service Class Information

**Class can be located at: edu.sru.thangiah.webrouting.services**

1. **UserServiceImpl:** Class used throughout the program to implement repository methods.

## 4.9 Configuration Classes

Section 4.9 highlights classes that are used to configure different aspects of the application.

### 4.9.1 Configuration Class Information

**Deprecated**: *Constructor methods* are only used for testing, **@Autowired** Annotation handles this in Application

I. **PopulateDatabase:** Class used to populate database Implements **Springs Application Runner** to run when the application context starts.

II. **RestConfig:** Class used to create Rest Templates

# 5. Testing

This section will discuss testing within the Webrouting Application

## 5.1 Junit Testing

The Webrouting Application Junit Tests may be found in the projects ***src/test/java*** folder.

All Test classes can be found in their respective packages, test classes implement methods from their non-test versions. All Tests are written with **JUnit 5**. All Tests files can be run at once using by right clicking src/test/java and selecting **Run As JUnit Test**.

# 6. UML Diagrams

This section will highlight UML diagrams conveying code pertinent to the Webrouting Application.

## 6.1 Class Diagrams

This section contains all Class Diagrams of the Webrouting Application

### 6.1.1

Timeline

Description automatically generated

Figure 1 – Class Diagram

## 6.2 Sequence Diagrams

Section 6.2 contains all sequence diagrams relating to the Webrouting Application.

### 6.2.1

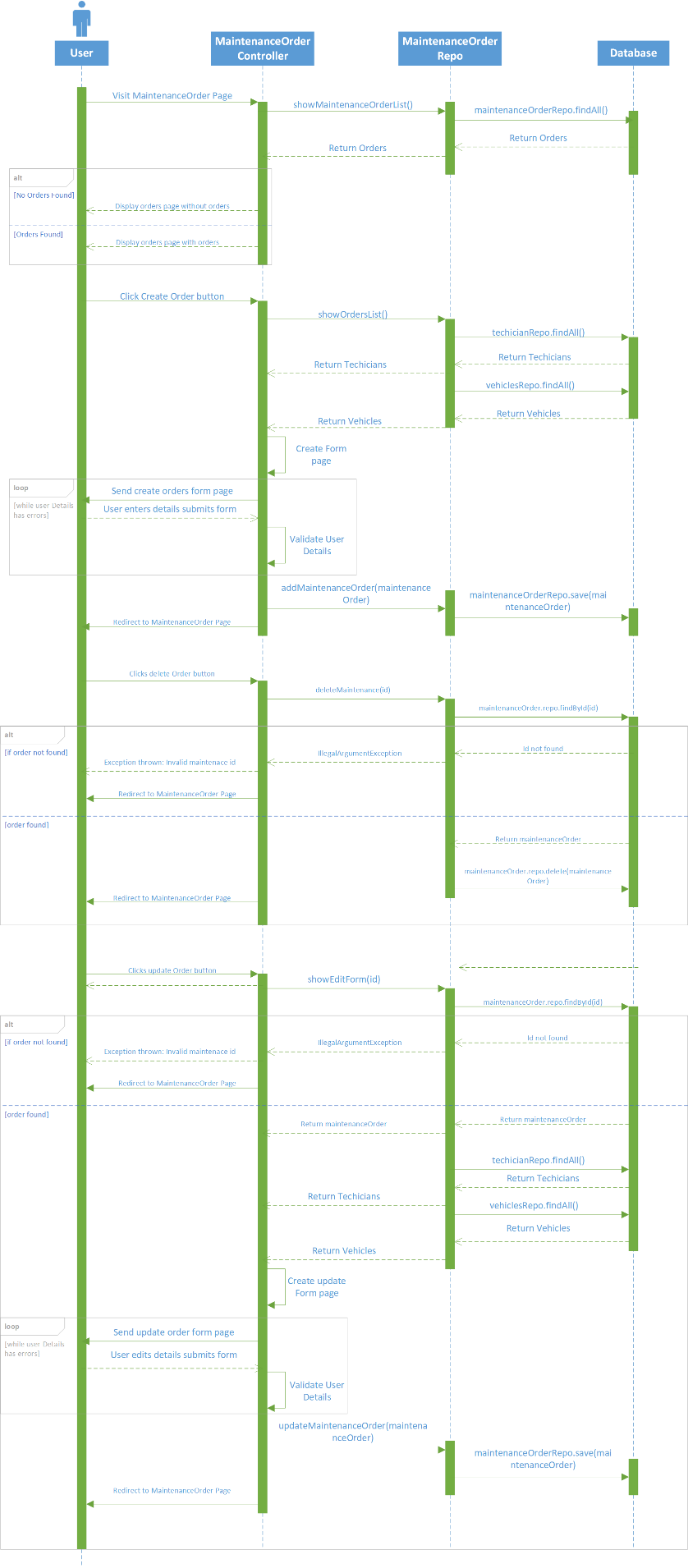


Figure 2 – Maintenance Order Sequence Diagram

Diagram

Description automatically generated

Figure 3 – Shipments Sequence Diagram

Graphical user interface, chart

Description automatically generated

Figure 4 – Accept Bid Sequence Diagram

Graphical user interface

Description automatically generated

Figure 5 – Create Bid Sequence Diagram

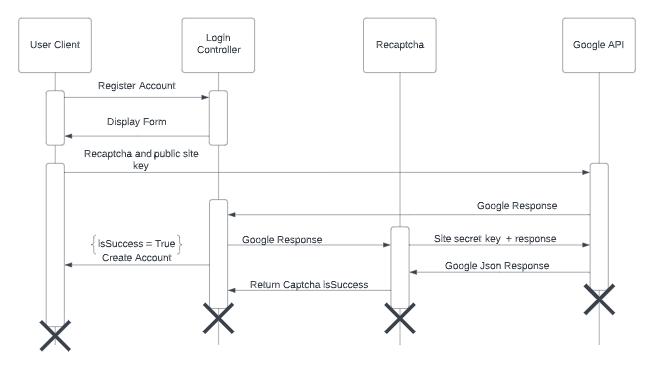


Figure 6 – Captcha State Chart Diagram

## 6.3 Activity Diagrams

Section 6.3 Contains activity diagrams of functionality in the Webrouting Application.

### 6.3.1

Diagram

Description automatically generated

Figure 7 – Drivers Activity Diagram

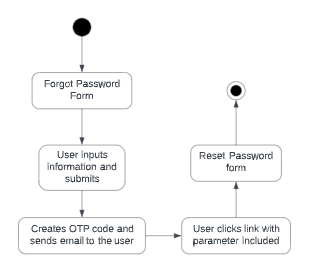


Figure 8 – Email verification diagram

Diagram

Description automatically generated

Figure 9 – Update Maintenance Order Activity Diagram

Diagram

Description automatically generated

Figure 10 - Shipments Activity Diagram

Graphical user interface, application

Description automatically generated

Figure 11 – Add Bid Activity Diagram

Graphical user interface, application, table, Excel

Description automatically generated

Figure 12 – Accept Bid Activity Diagram

## 6.4 State Diagrams

Section 5.4 Will highlight and show state diagrams relevant to the Webrouting Application

### 6.4.1

A picture containing graphical user interface

Description automatically generated

Figure 13 – Drivers State Diagram

Timeline

Description automatically generated

Figure 14 – Maintenance Order State Diagram

Diagram

Description automatically generated

Figure 15 – Shipments State Diagram

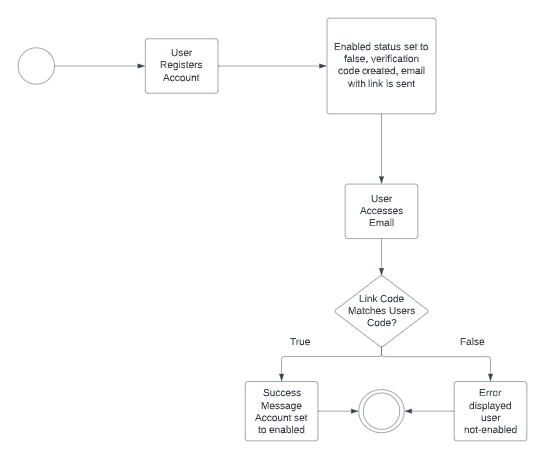


Figure 16 – Email Verification State Chart Diagram

# 7. Figures

Figure 1 – Class Diagram 9

Figure 2 – Maintenance Order Sequence Diagram 10

Figure 3 – Shipments Sequence Diagram 11

Figure 4 – Accept Bid Sequence Diagram 12

Figure 5 – Create Bid Sequence Diagram 13

Figure 6 – Captcha State Chart Diagram 13

Figure 7 – Drivers Activity Diagram 14

Figure 8 – Email verification diagram 14

Figure 9 – Update Maintenance Order Activity Diagram 15

Figure 10 - Shipments Activity Diagram 16

Figure 11– Add Bid Activity Diagram 17

Figure 12 – Accept Bid Activity Diagram 18

Figure 13 – Drivers State Diagram 19

Figure 14 – Maintenance Order State Diagram 19

Figure 15 – Shipments State Diagram 19

Figure 16 – Email Verification State Chart Diagram 20