Web Routing

Technical Manual

Slippery Rock University

Fady Aziz [FAA1002@SRU.EDU](mailto:FAA1002@SRU.EDU)

IAN BLACK [IMB1007@SRU.EDU](mailto:IMB1007@SRU.EDU)

LOGAN KIRKWOOD [LLK1005@SRU.EDU](mailto:LLK1005@SRU.EDU)

Contents

[1. Overview 3](#_Toc101295875)

[2. Recommended System Requirements 3](#_Toc101295876)

[3. WebRouting Classes 3](#_Toc101295877)

[4. UML Diagrams 6](#_Toc101295878)

[4.1 Class Diagram 6](#_Toc101295879)

[4.2 Sequence Diagram 8](#_Toc101295880)

[4.3 Activity Diagrams 12](#_Toc101295881)

[4.4 State Diagrams 17](#_Toc101295882)

[5. Figures 18](#_Toc101295883)

# 1. Overview

This document describes in detail the inter-workings of the Web Routing Application. Detailed in this document are the classes used, system requirements, and UML diagrams.

# 2. Recommended System Requirements

**CPU:** Intel® Core i5

**Memory:** 8GB

**OS:** Microsoft® Windows® 10

**Free Space:** 40gb

**Software:**

* Eclipse IDE EE – 2021-12
* MySQL Server 8.0.28
* MySQL Workbench 8.0.28
* Google Chrome v100.0.4896.75

# 3. WebRouting Classes

1. **WebroutingApplication.java:** Class to run the application.
2. **WebSecurityConfig.java:** Has instance of the UserValidator class. It Handles all authentication for different roles and authorities such as Admin, Manager, Assistant Manager, and operations Manager. Also, it encodes the specified raw password.
3. **CarriersController.java**: Has instance of the CarriersRepository.
4. **ContactsController.java**: Handles the Thymeleaf controls for pages dealing with Contacts. Constructor for ContactsController, Instantiates the contactsRepository
5. **ContractorsController.java:** Handles the Thymeleaf controls for pages dealing with Contractors. Constructor for ContractorsController, instantiates the contractorsRepository, and instantiates the contactsRepository.
6. **DriverController.java:** Handles the Thymeleaf controls for pages dealing with Drivers. Constructor for DriverController, Instantiates the driverRepository, Instantiates the contactRepository, and Instantiates the contractorsRepositor.
7. **LocationController.java:** Handles the Thymeleaf controls for pages dealing with Locations. Constructor for locationsController, instantiates the locationsRepository, and instantiates the contractorsRepository.
8. **LoginController.java:** Handles the Thymeleaf controls for pages dealing with the login function. Redirects user to the registration page and adds a new instance of a User to the userForm model, also adds all of the roles to the role model
9. **MaintenanceOrderController.java:** Handles the Thymeleaf controls for pages dealing with maintenance order. Constructor for MaintenanceOrderController. instantiates the maintenanceOrderRepository, instantiates the techniciansRepository, instantiates the driverRepository, and instantiates the vehiclesRepository.
10. **RolesController.java:** Handles the Thymeleaf controls for pages dealing with Roles. Constructor for roles, and instantiates the roleRepository
11. **RoutesController.java:** Handles the Thymeleaf controls for pages dealing with Routes. Constructor for RoutesController, instantiates the routesRepository, instantiates the vehiclesRepository, and instantiates the locationsRepository
12. **ShipmentsController.java:** the Thymeleaf controls for pages dealing with Shipments. Constructor for ShipmentsController, and instantiates the shipmentsRepository.
13. **TechniciansController.java:** Handles the Thymeleaf controls for pages dealing with Technicians. Constructor for TechniciansController, instantiates the techniciansRepository, and instantiates the contactsRepository
14. **UserController.java:** Handles the Thymeleaf controls for pages dealing with Users. Constructor for UserController. instantiates the userRepository, and instantiates the roleRepository
15. **VehiclesController.java:** Handles the Thymeleaf controls for pages dealing with Vehicles. Constructor for VehiclesController. Instantiates the vehiclesRepository, instantiates the locationsRepository, instantiates the vehicleTypesRepository, and instantiates the contractorsRepository
16. **VehicleTypesController.java**: Handles the Thymeleaf controls for pages dealing with VehicleTypes. Constructor for VehicleTypesController. Instantiates the vehicleTypesRepository.
17. **BidsController**.java: Handles the Thymeleaf controls for pages dealing with Bids. Constructor for BidsController. Instantiates the bidsRepository.
18. **Carriers.java**: Contains the Carriers database such as Carrier Name, SCAC Code, LTL, FTL, Pallets, Weight, Contact ID, Shipments, Bids, Locations, Vehicles, Drivers.
19. **Contacts.java:** Contains all the Contacts database such as First name, Last name, Middle initial, Email address, Street address1, Street address2, City, State, Zip, PrimaryPhone, and WorkPhone.
20. **Driver.java:** Contains all the Driver database such as ID, license class, License expiration, License number, Contact ID, Contractor ID.
21. **Location.java:** Contains all the Location database. such as ID, City, Latitude, Location type, Longitude, State, Street address1, Street address2, Zip, Contractor ID.
22. **MaintenanceOrderr.java:** Contains all the MaintenanceOrder database Such as ID, Cost, Details, Maintenance type, Scheduled date, Service type key, Status key, Technician ID, and Vehicle ID.
23. **Roles.java:** Contains all the Roles database such as ID, and Roles “it could be Admin, Masterlist, Shipper, or Carrier”
24. **Routes.java:** Contains all the Routes database such as ID, End date, Start Date, Total cubic weight, Total days, Total miles, Total time, Total weight, Vehicle ID, Start Location ID, End location ID.
25. **Shipments.java:** Contains all the Shipments database such as ID, ship date, Client name, Weight, Vehicle id, Carrier name, Client mode, Commodity class, Commodity paid weight, Commodity pieces, Consignee city, Consignee latitude, Consignee longitude, Consignee state, Consignee zip, Freight bill number, Full freight terms, Paid amount, SCAC, Shipper city, Shipper latitude, Shipper longitude, Shipper state, Shipper zip
26. **Technicians.java:** Contains all the Technicians database such as ID, Skill grade, Orders, and contact ID.
27. **User.java:** Contains all the User database such as ID, Email, Encrypted password, User name, Role ID, Carrier ID, Shipments.
28. **Vehicles.java:** Contains all the Vehicles database such ID, Manufactured year, Plate number, VIN number, Contractor ID, Location ID, Vehicle Type ID.
29. **VehicleTypes.java:** Contains all the VehicleTypes database Such as Capacity, Description, Empty weight, Height, Length, Make, Maximum cubic weight, Maximum range, Maximum weight, Minimum cubic weight, Minimum weight, Model Restrictions, Sub type, Type.
30. **Bids.java**: Contains all the Bids database such as Bid ID, price, date, time, shipment ID, Carrier ID.
31. **MyAccessDeniedHandler.java:** Logs when a user accesses a page without permission. Also, redirects to the 403 access denied page (403.html)
32. **CarriersRepository:** Sets Up the Carriers Repository.
33. **ContactsRepository.java:** Sets up the Contacts repository.
34. **ContractorsRepository.java:** Sets up the Contractors repository.
35. **DriverRepository.java:** Sets up the Driver repository.
36. **LocationRepository.java:** Sets up the Location repository.
37. **LoginRepository.java:** Sets up the Login repository.
38. **MaintenanceOrderrRepository.java:** Sets up the MaintenanceOrder repository.
39. **RolesRepository.java:** Sets up the Roles repository.
40. **RoutesRepository.java:** Sets up the Routes repository.
41. **ShipmentsRepository.java:** Sets up the Shipments repository.
42. **TechniciansRepository.java:** Sets up the Technicians repository.
43. **UserRepository.java:** Sets up the User repository.
44. **VehiclesRepository.java:** Sets up the Vehicles repository.
45. **VehiclesTypesRepository.java:** Sets up the VehicleTypes repository.
46. **BidsRepository.**java: Sets up the Bids repository.
47. **SecurityService.java:** Interface class for the isAuthenticated() and autoLogin() methods
48. **SecurityServiceImpl.java:** Implements the SecurityService interface. also, used as a service for Spring Security.
49. **UserDetailsServiceImpl.java:** Implements the UserDetailsService interface. also, used as a service for Spring Security.
50. **UserService.java:** Interface class for the save() and findByUsername() methods.
51. **UserServiceImpl.java:** Implements the UserService Interface. also, used as a service for Spring Security
52. **UserValidator.java:** Checks the username and the password to see if the length is valid.

# 4. UML Diagrams

## 4.1 Class Diagram

Timeline

Description automatically generated

Figure 1 – Class Diagram

## 4.2 Sequence Diagram

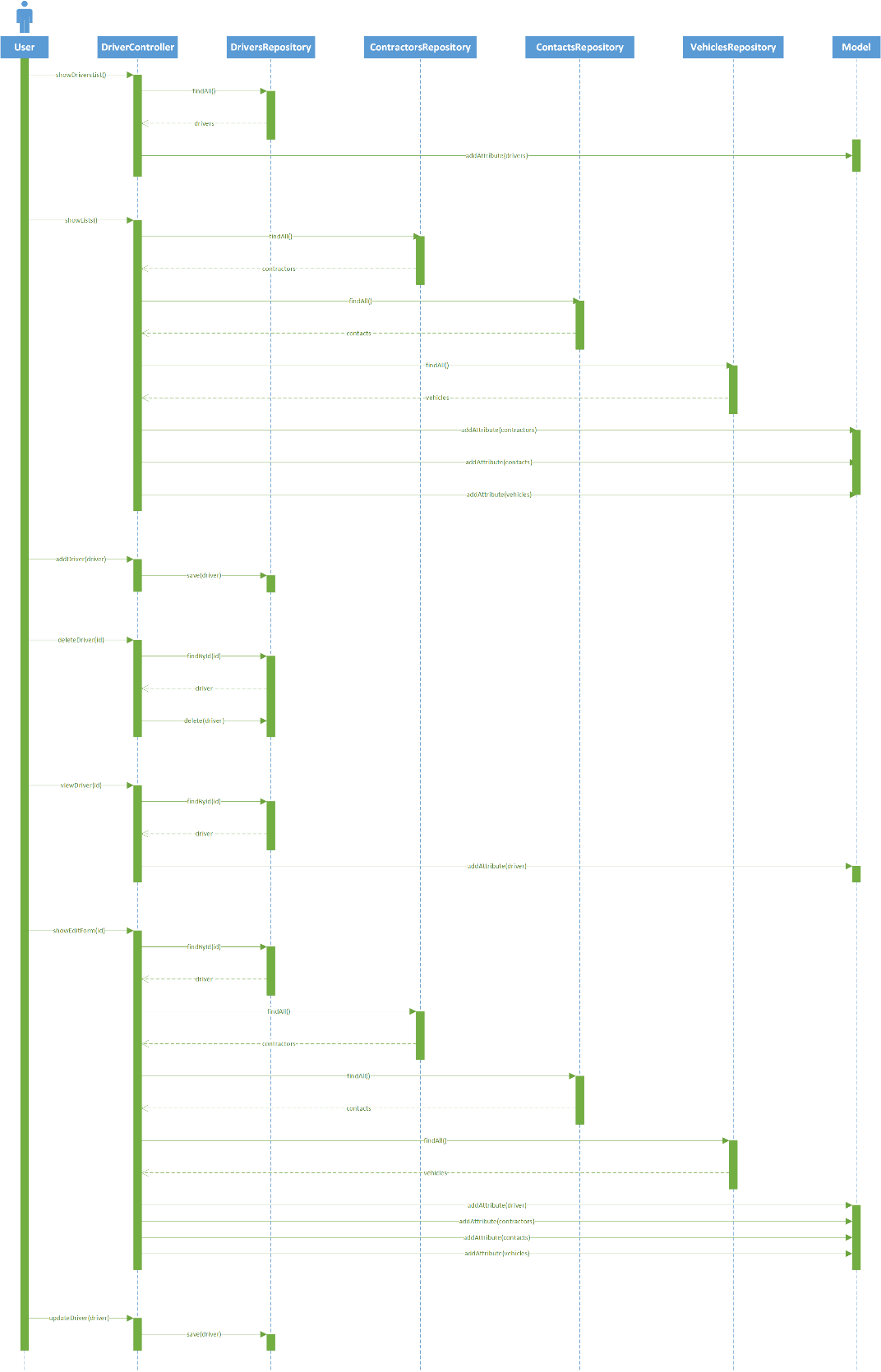


Figure 2 – Drivers Sequence Diagram

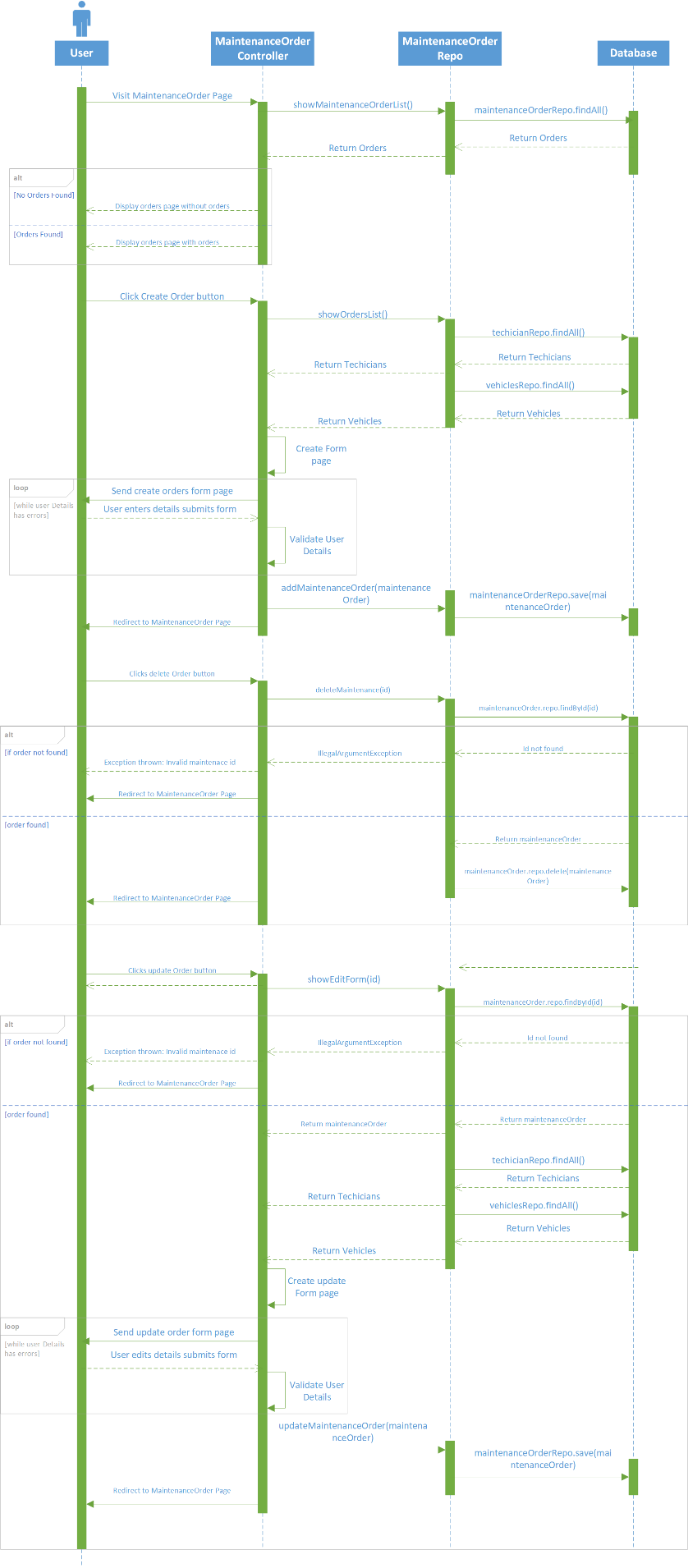


Figure 3 – Maintenance Order Sequence Diagram

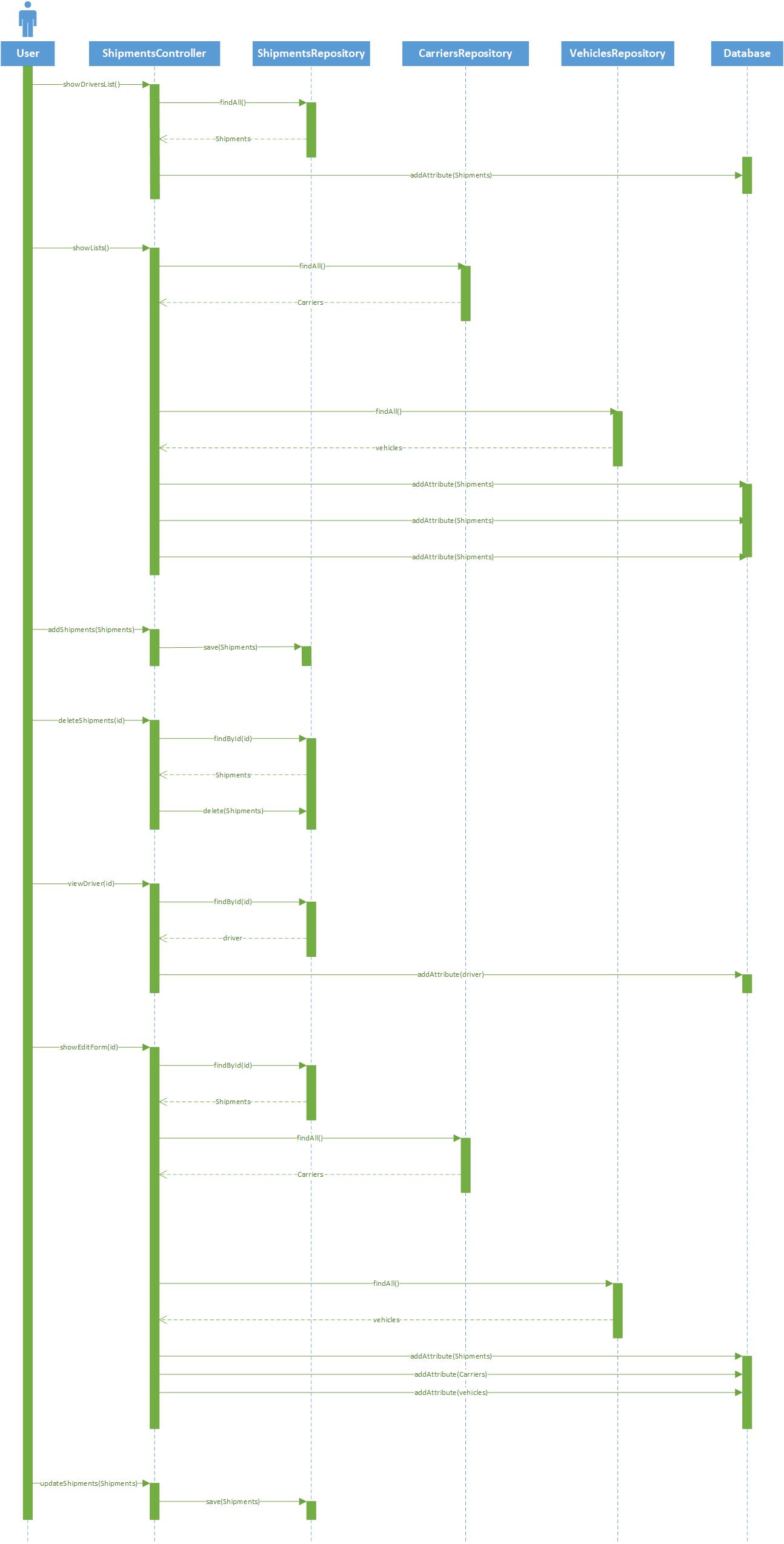


Figure 4 – Shipments Sequence Diagram

Graphical user interface, chart

Description automatically generated

Figure 5 – Accept Bid Sequence Diagram

Graphical user interface

Description automatically generated

Figure 6 – Create Bid Sequence Diagram

## 4.3 Activity Diagrams

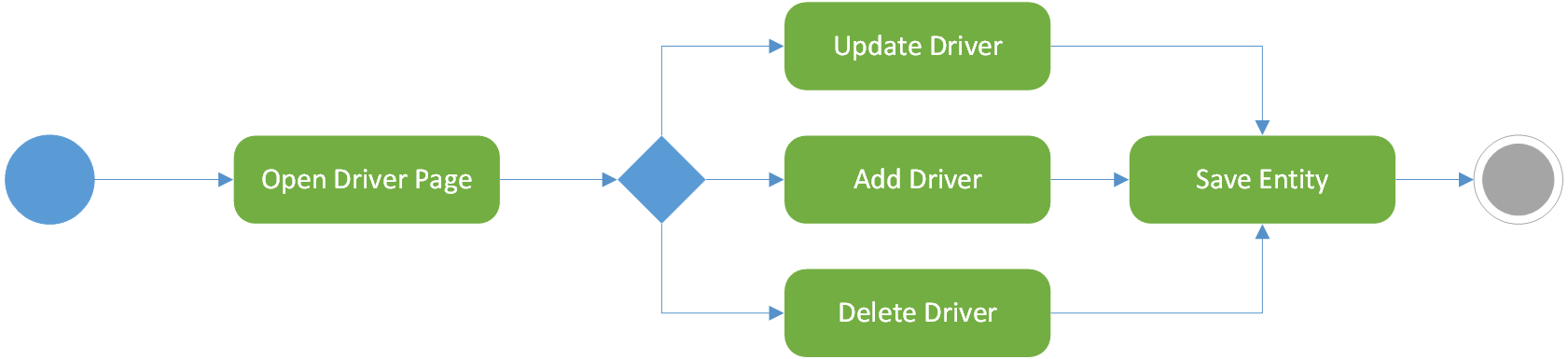


Figure 7 – Drivers Activity Diagram

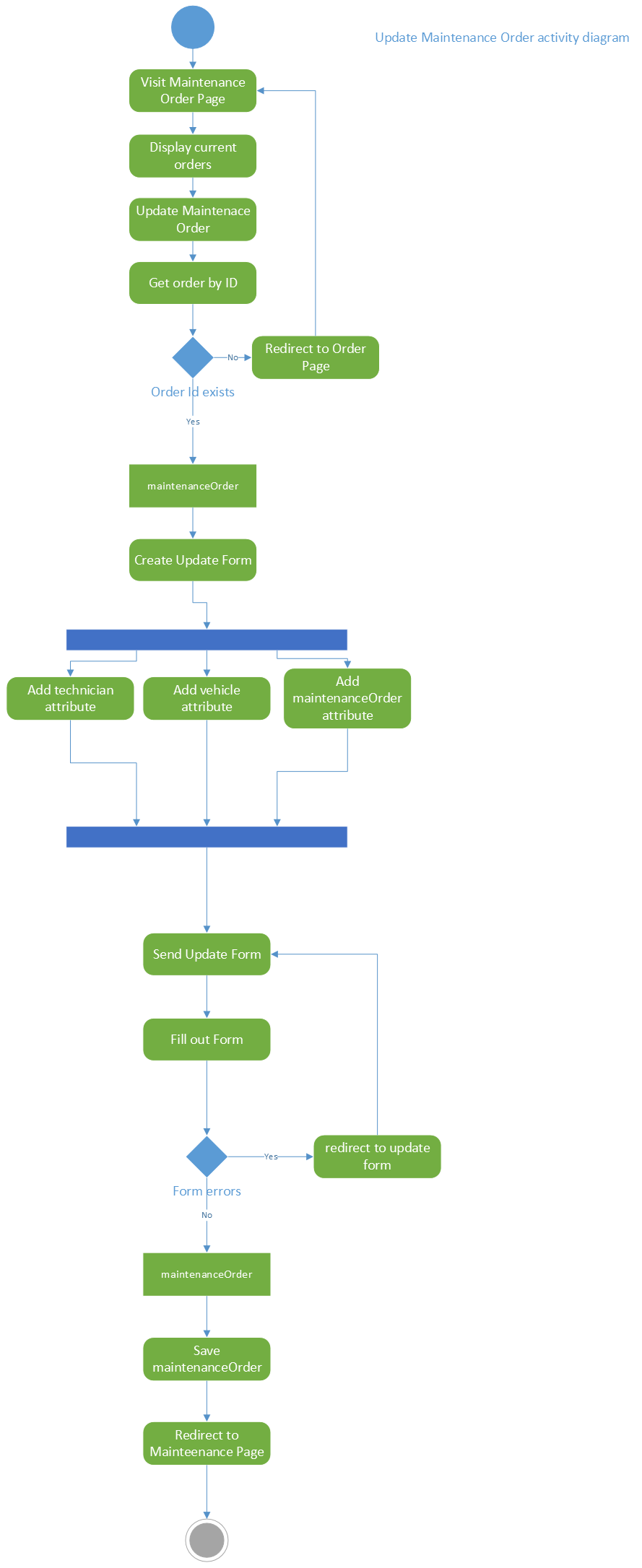


Figure 8 – Update Maintenance Order Activity Diagram

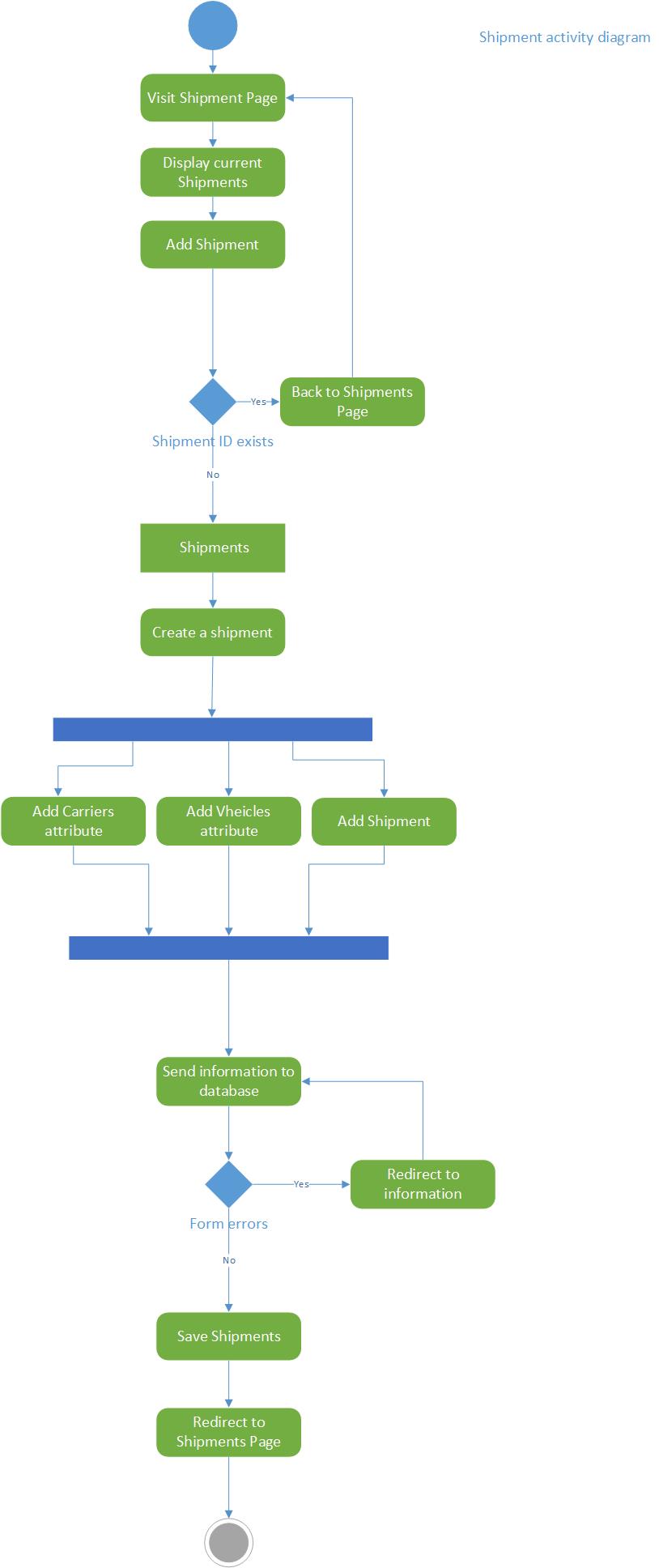


Figure 9 – Shipments Activity Diagram

Graphical user interface, application

Description automatically generated

Figure 10 – Add Bid Activity Diagram

Graphical user interface, application, table, Excel

Description automatically generated

Figure 11 – Accept Bid Activity Diagram

## 4.4 State Diagrams

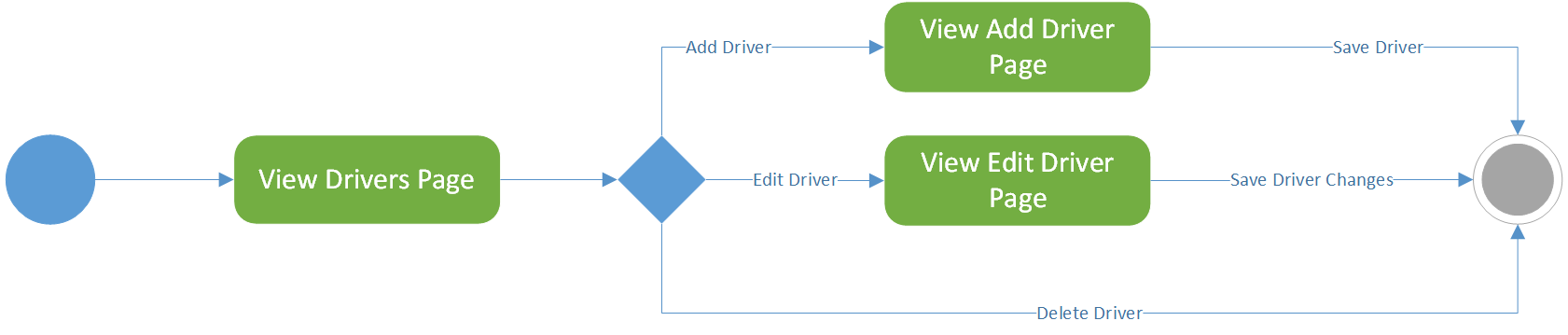


Figure 12 – Drivers State Diagram

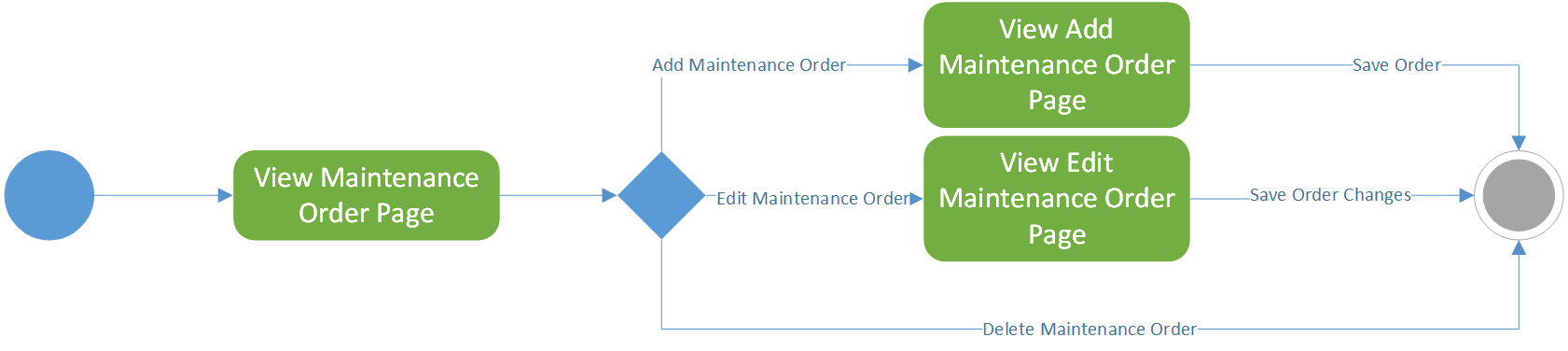


Figure 13 – Maintenance Order State Diagram

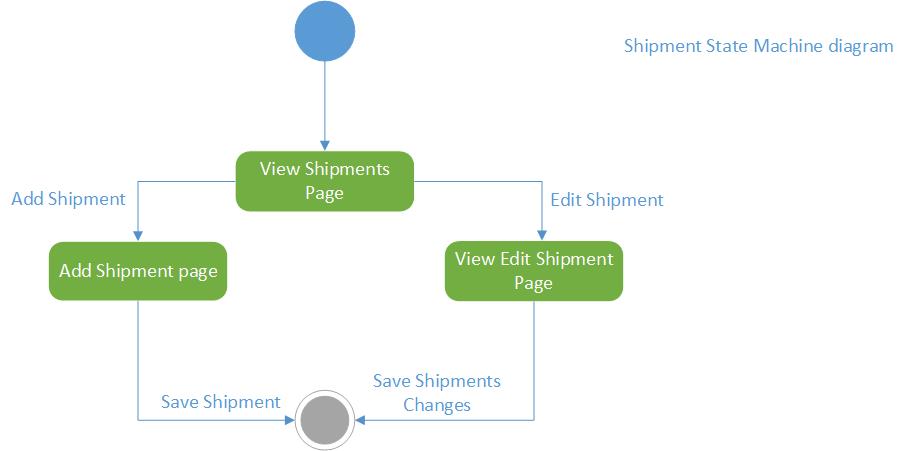


Figure 14 – Shipments State Diagram

# 5. Figures

[Figure 1 – Class Diagram 7](#_Toc101350793)

[Figure 2 – Drivers Sequence Diagram 8](#_Toc101350794)

[Figure 3 – Maintenance Order Sequence Diagram 9](#_Toc101350795)

[Figure 4 – Shipments Sequence Diagram 10](#_Toc101350796)

[Figure 5 – Accept Bid Sequence Diagram 11](#_Toc101350797)

[Figure 6 – Create Bid Sequence Diagram 12](#_Toc101350798)

[Figure 7 – Drivers Activity Diagram 13](#_Toc101350799)

[Figure 8 – Update Maintenance Order Activity Diagram 14](#_Toc101350800)

[Figure 9 – Shipments Activity Diagram 15](#_Toc101350801)

[Figure 10 – Add Bid Activity Diagram 16](#_Toc101350802)

[Figure 11 – Accept Bid Activity Diagram 17](#_Toc101350803)

[Figure 12 – Drivers State Diagram 18](#_Toc101350804)

[Figure 13 – Maintenance Order State Diagram 18](#_Toc101350805)

[Figure 14 – Shipments State Diagram 18](#_Toc101350806)