

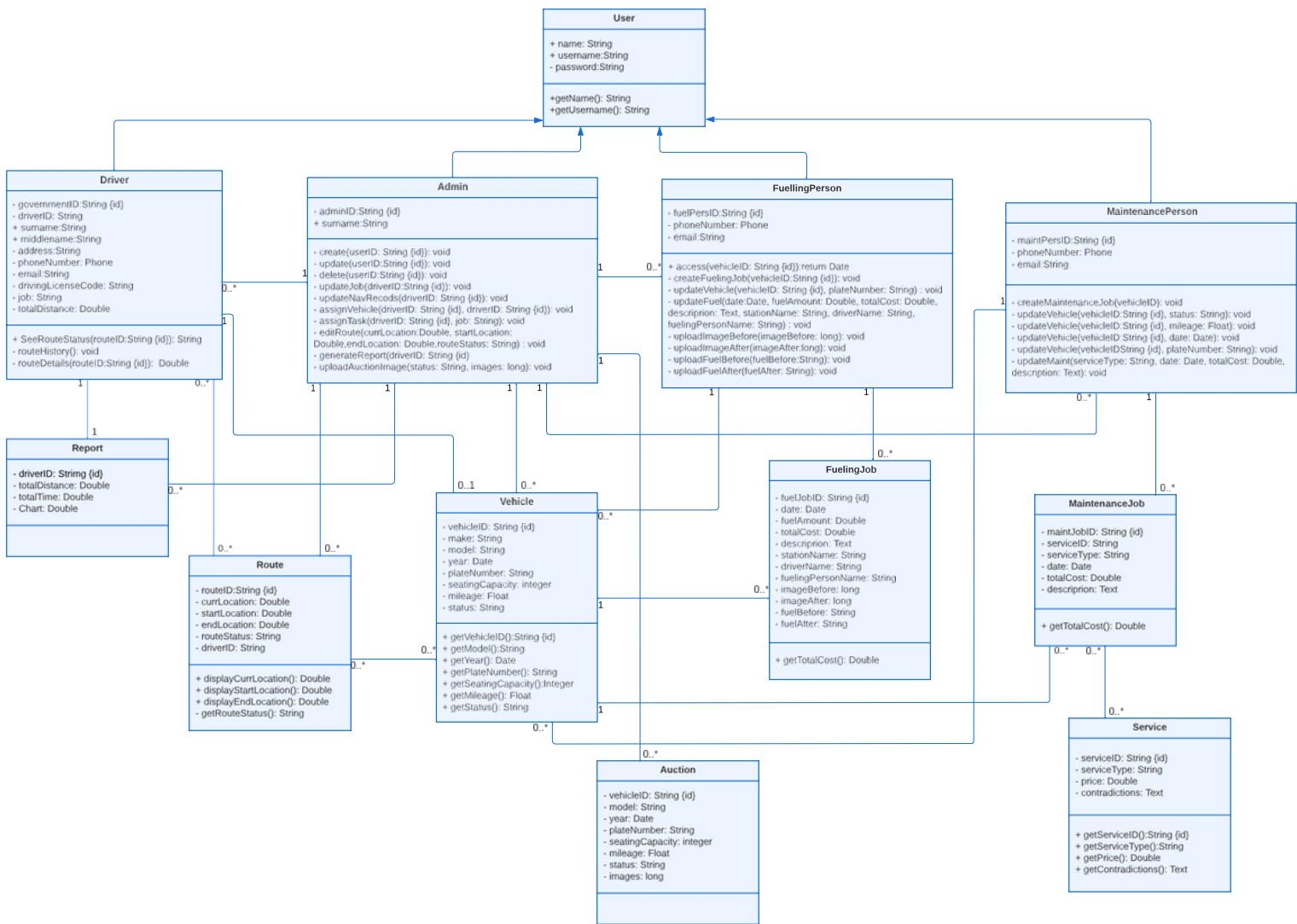
SOFTWARE REQUIREMENTS SPECIFICATION PROJECT MILESTONE

1. Aziz Bekzhanov
2. Alan Igilikov
3. Dias Gaziz
4. Karine Aripova
5. Khorlan Assylbek
6. Yestay Aikyn

Section: 6

Group: 1

Figure 1: Class Diagram

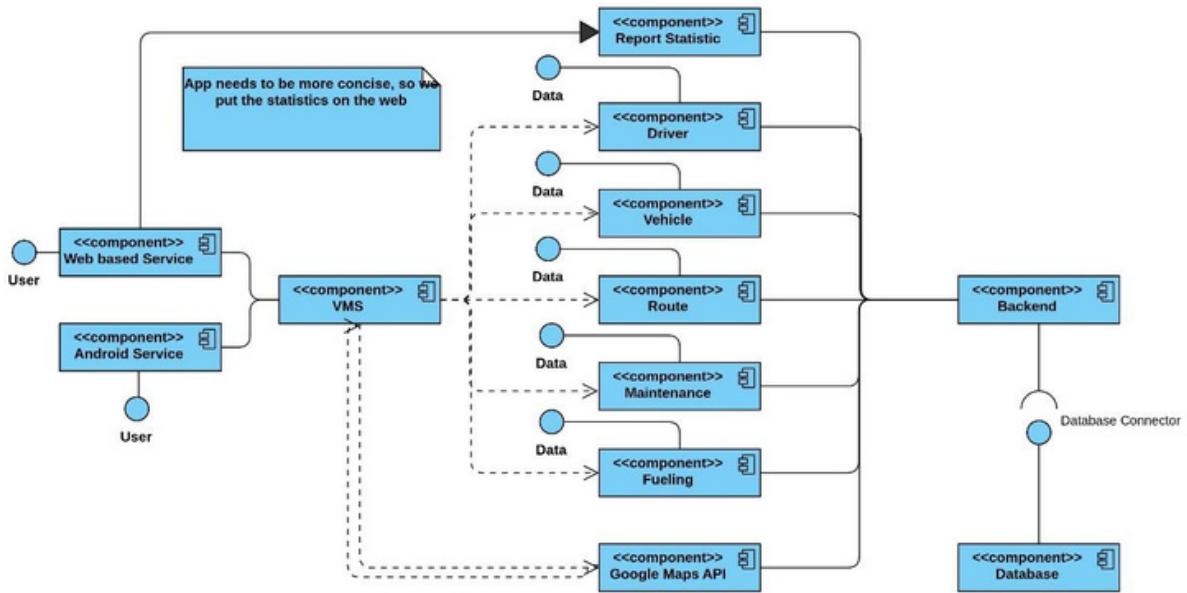


Description of Class Diagram:

There are 12 classes in the VMS class Diagram. They are User, Driver, Admin, FuellingPerson, MaintenancePerson, Report, Route, Vehicle, FuelingJob, MintenanceJob, Auction, and Service. The relationships between them can be seen in the diagram:

Figure 2:

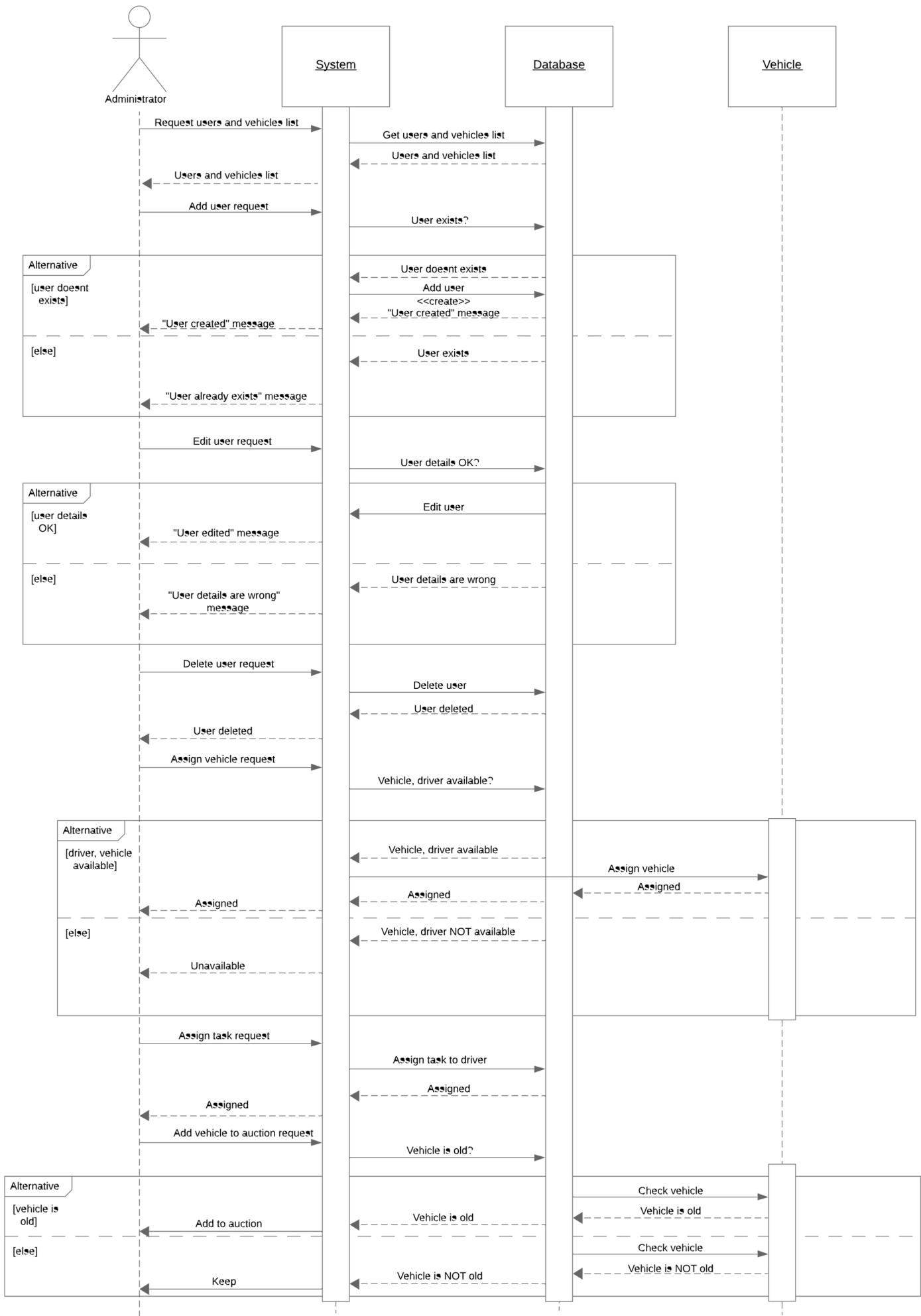
Vehicle Management System (VMS) Component Diagram

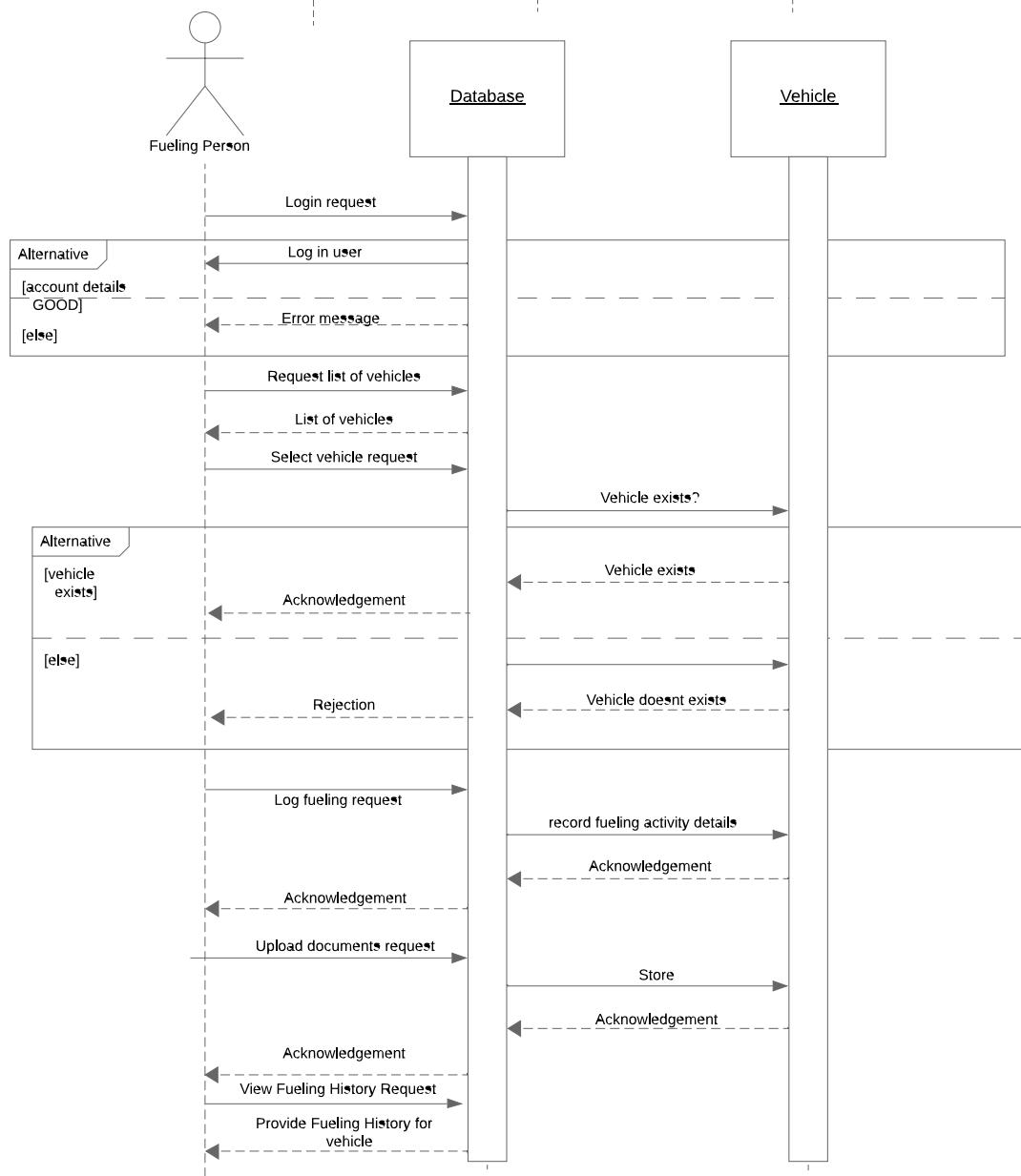
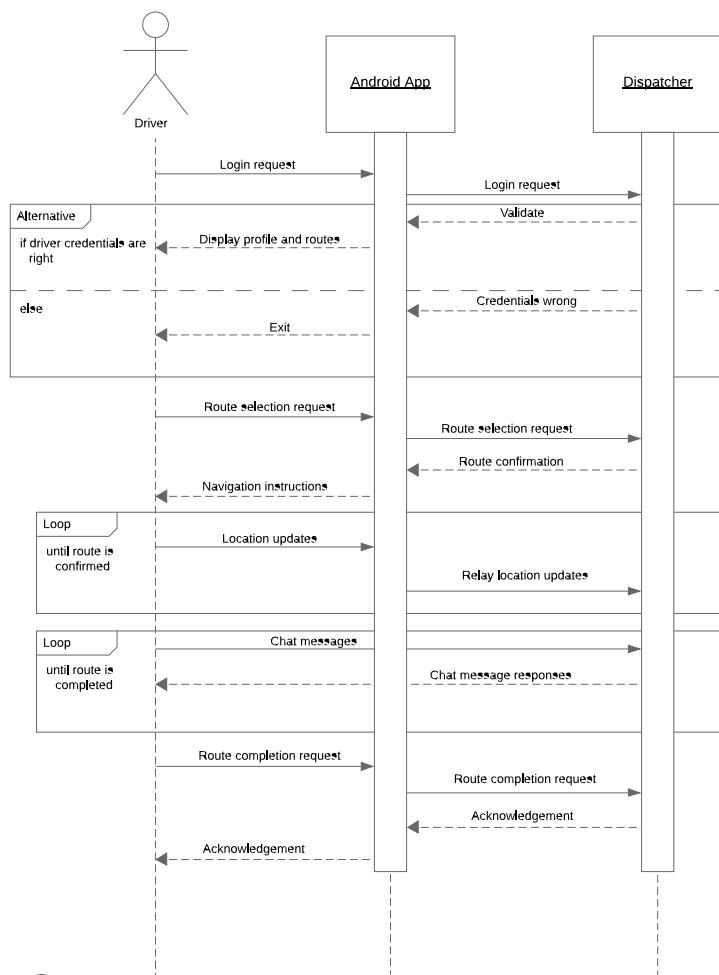


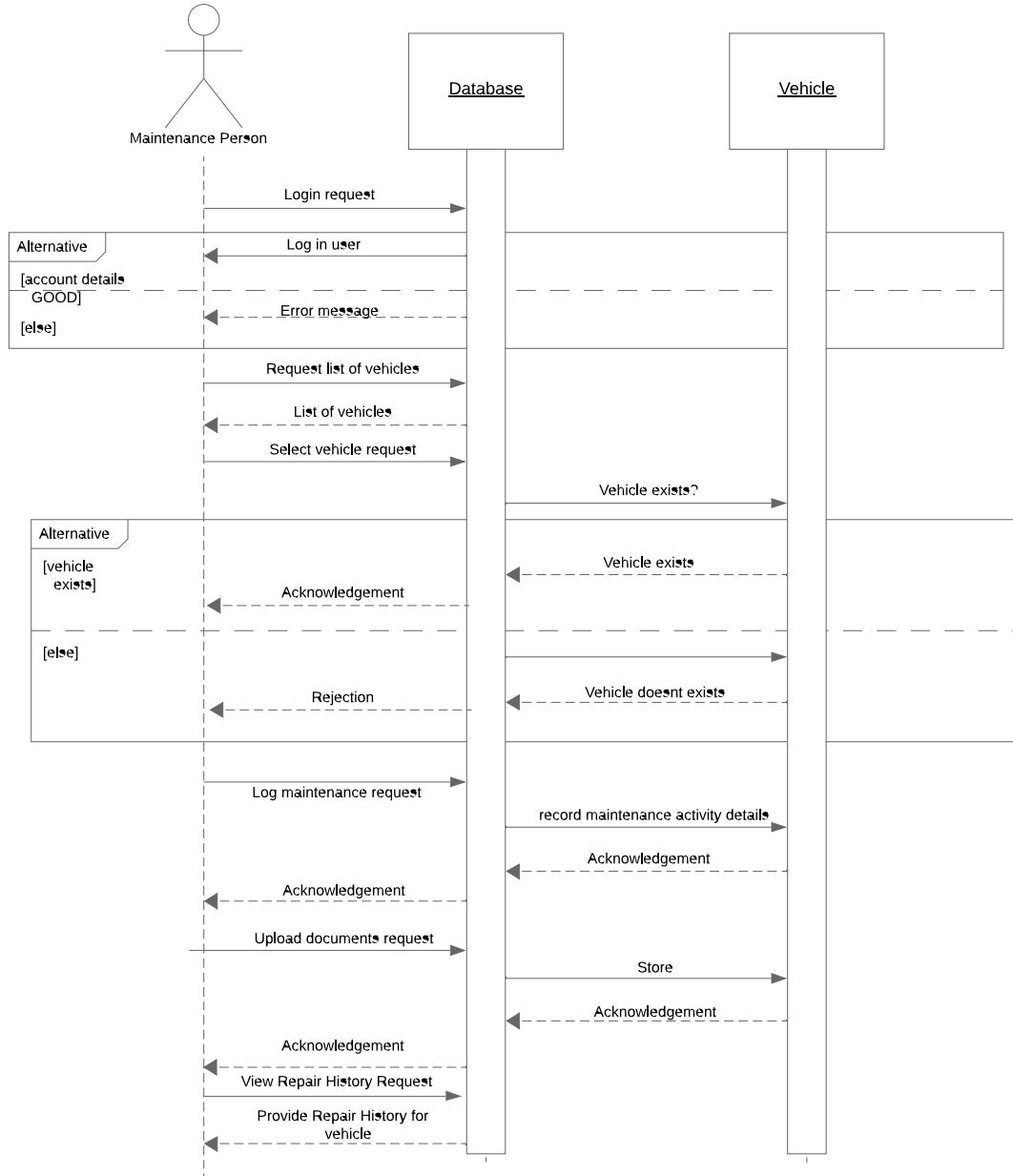
Description of Component diagram:

The diagram has 10 components (Android Service, Web based Service, Backend, Database, Google Maps API, Report Statistic, VMS, Vehicle, Driver and Route) and 5 connectors (Maintenance, Fueling, Data, App and User). The diagram shows how the components interact with each other to provide a vehicle management system (VMS). The Android Service component allows the driver to access the VMS app on their mobile device. The Web based Service component allows the admin to access the VMS website on their browser. The Backend component handles the logic and communication between the other components. The Database component stores the data related to the vehicles, drivers, routes, maintenance and fueling. The Google Maps API component provides the map and navigation features for the VMS app and website. The Report Statistic component generates reports based on the data from the Database component. The VMS component represents the overall system that provides the vehicle management services. The Vehicle component represents a physical vehicle that is managed by the VMS. The Driver component represents a person who drives a vehicle assigned by the admin. The Route component represents a path that a driver follows for a trip. The Maintenance connector represents the process of maintaining a vehicle by a maintenance person. The Fueling connector represents the process of fueling a vehicle by a fueling person. The Data connector represents the flow of data between the components. The App connector represents the interaction between the driver and the VMS app. The User connector represents the interaction between the admin and the VMS website.

Figure 3: Sequence Diagrams



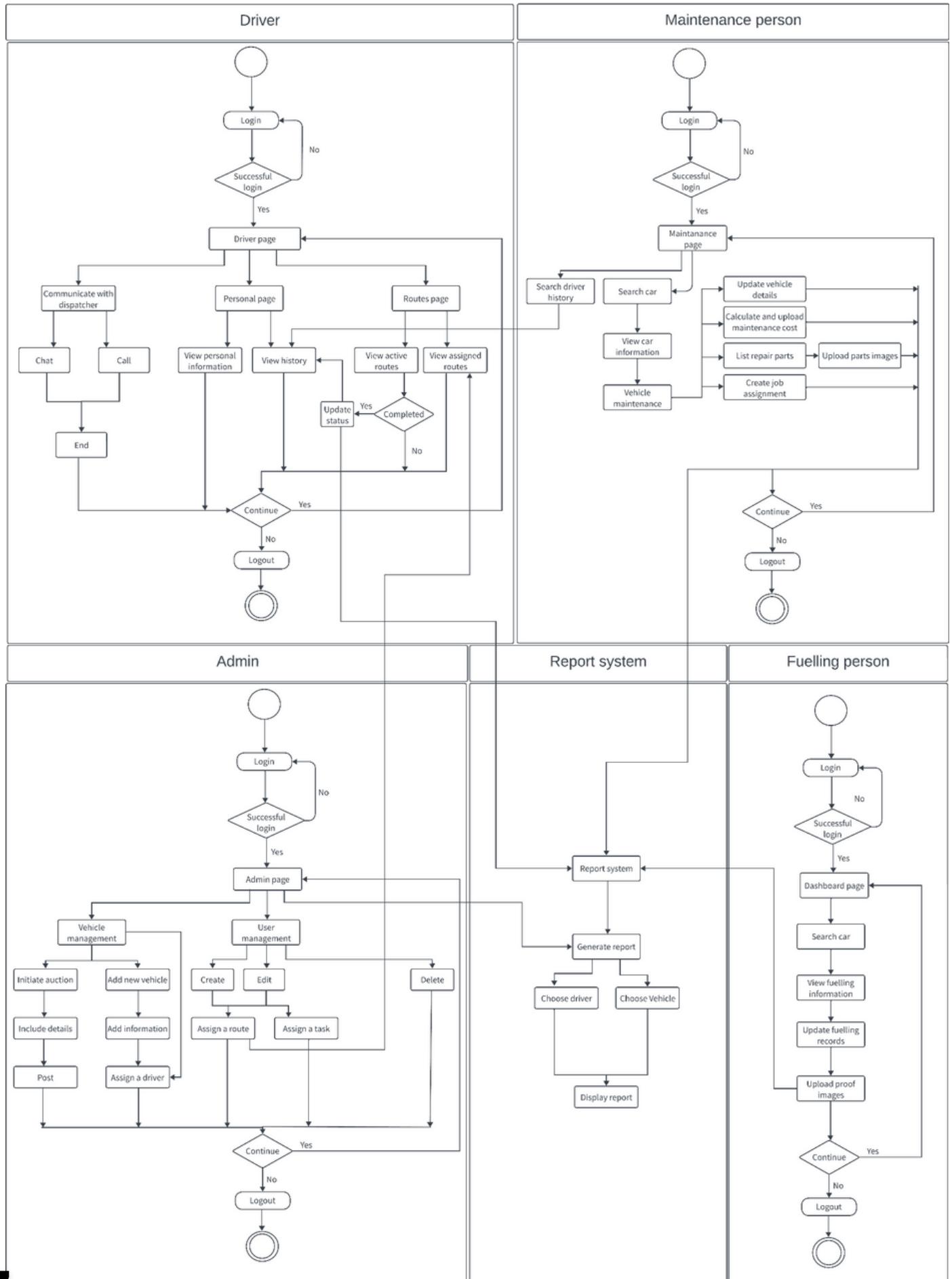




Description of Sequence diagrams:

Overall, four sequence diagrams depict distinct interactions and flows. The Driver Interaction Sequence Diagram elucidates how a driver engages with the Android App and Dispatcher, from login to route completion. The Admin Interaction Sequence Diagram outlines administrative tasks undertaken by the admin through interactions with the system and the database. Additionally, the Maintenance Person Interaction Sequence Diagram demonstrates how maintenance personnel log and retrieve maintenance activities for specific vehicles, involving interactions with the Database and Vehicle objects. Similarly, the Fueling Person Interaction Sequence Diagram illustrates how fueling personnel log and access fueling activities for specific vehicles, encompassing interactions with the Database and Vehicle objects.

Figure 4: Activity Diagram



Description of Activity diagram:

The diagram has 4 users (driver, admin, maintenance person and fueling person) and a report generation system. To begin with, admin has access to vehicle and user management as shown in diagram. The admin will assign routes to drivers; thus, an arrow leads to driver's view assigned routes option. In addition, admin can also request for report generation.

This report will include information regarding driver or a vehicle depending on what the admin has chosen. The report system itself saves information from activities of all other users. For example, it tracks a data about driver whenever they update status regarding their assigned routes. Lastly, the maintenance person has an option to view driver's history, hence an arrow leads to driver's view history option.

Figure 5: Use Case Diagram

