Software Design Proposal for a Driverless Car

A vehicle that uses sensors, cameras, and artificial intelligence to navigate without a driver is known as a driverless car (Luckovich, 2019). Driverless cars can be portable offices and bedrooms that let users work or rest while traveling (WSJ 2017). This proposal will be focused on the software design for a driverless-car operating system.

The following terms were used to conduct background research on how driverless cars work.

- **Driverless cars/Robot cars:**

The major and existing features of autonomous cars were researched to get a better understanding of how they work.

- **Vehicle to Everything (V2X):**

The latest trends in V2P, V2I, and V2V were researched to get a better understanding of the technology.

- **Sensors (RADAR, LIDAR, GPS):**

Various sensors that are often found in driverless cars were also researched. These sensors communicate with the control unit to enable the vehicle to perform specific tasks such as detecting an obstacle and changing lanes or directions.

- **Traffic sign recognition:**

There are sensors found in autonomous cars that enable them to detect traffic signs. This is one of the most effective ways to detect traffic signs.

Operations

Due to the requirements for testing driverless cars as Zhou et al. (2019) discussed, the following operations were selected for this software:

- **Detecting an Obstacle**

LIDAR is one of the sensors that enable autonomous vehicles to detect obstacles in a self-driving car (Reddy, 2019). Self-driving cars will have to be able to detect obstacles such as objects on the road, pedestrians, traffic signs, and other vehicles. LIDAR is employed due to the high efficiency and accuracy expected from the vehicles (Aghdam et al. 2016).

- **Detecting Traffic Signs/Warnings**

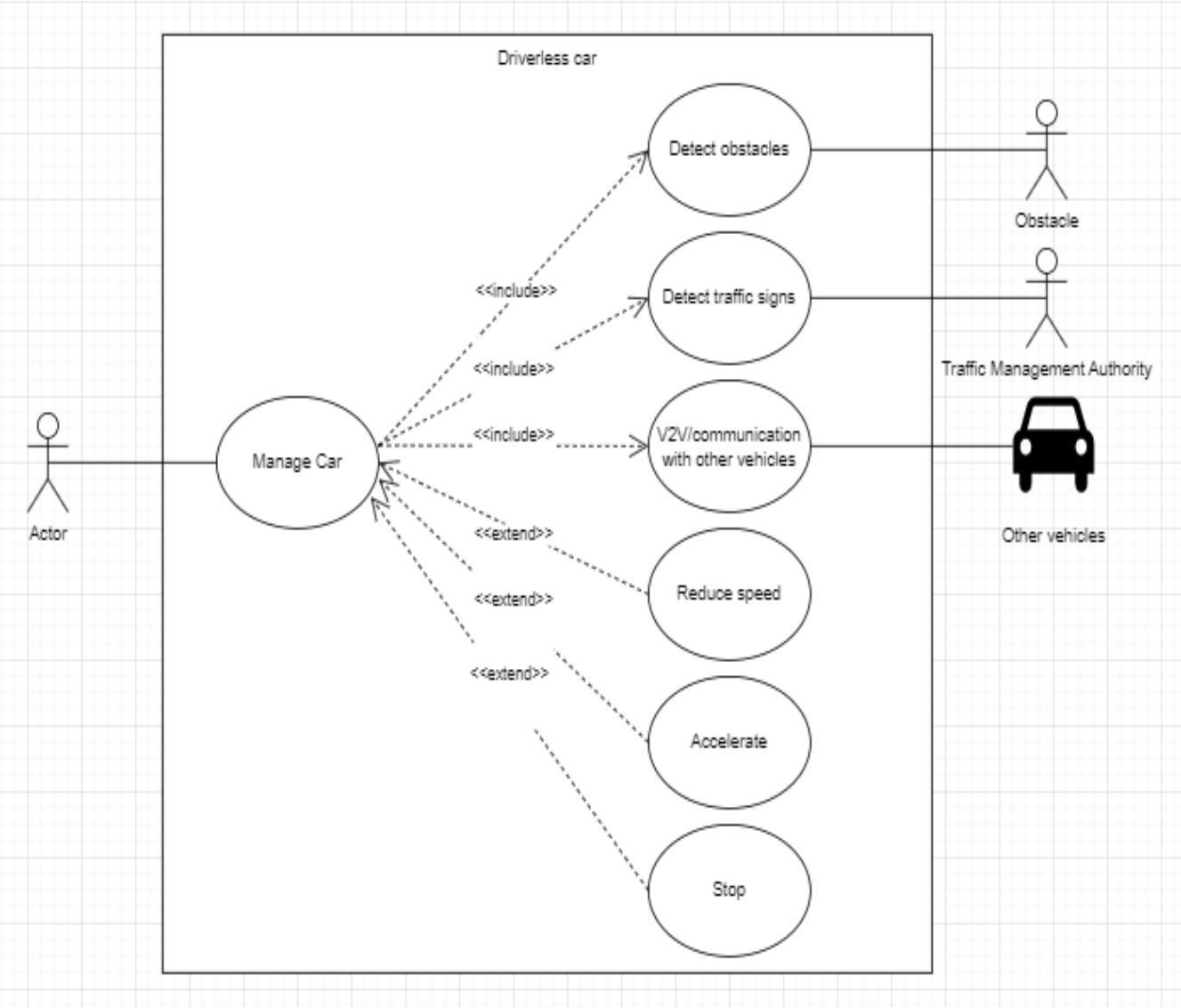
According to Liu et al. (2021), it is possible to detect and classify traffic signs using deep neural networks, convolutional neural networks (CNN), and cameras. To achieve this in this software, the Traffic Sign Recognition System (TSRS) will communicate with the Traffic Management Authority (TMA).

- **Vehicle to Everything (V2X)**

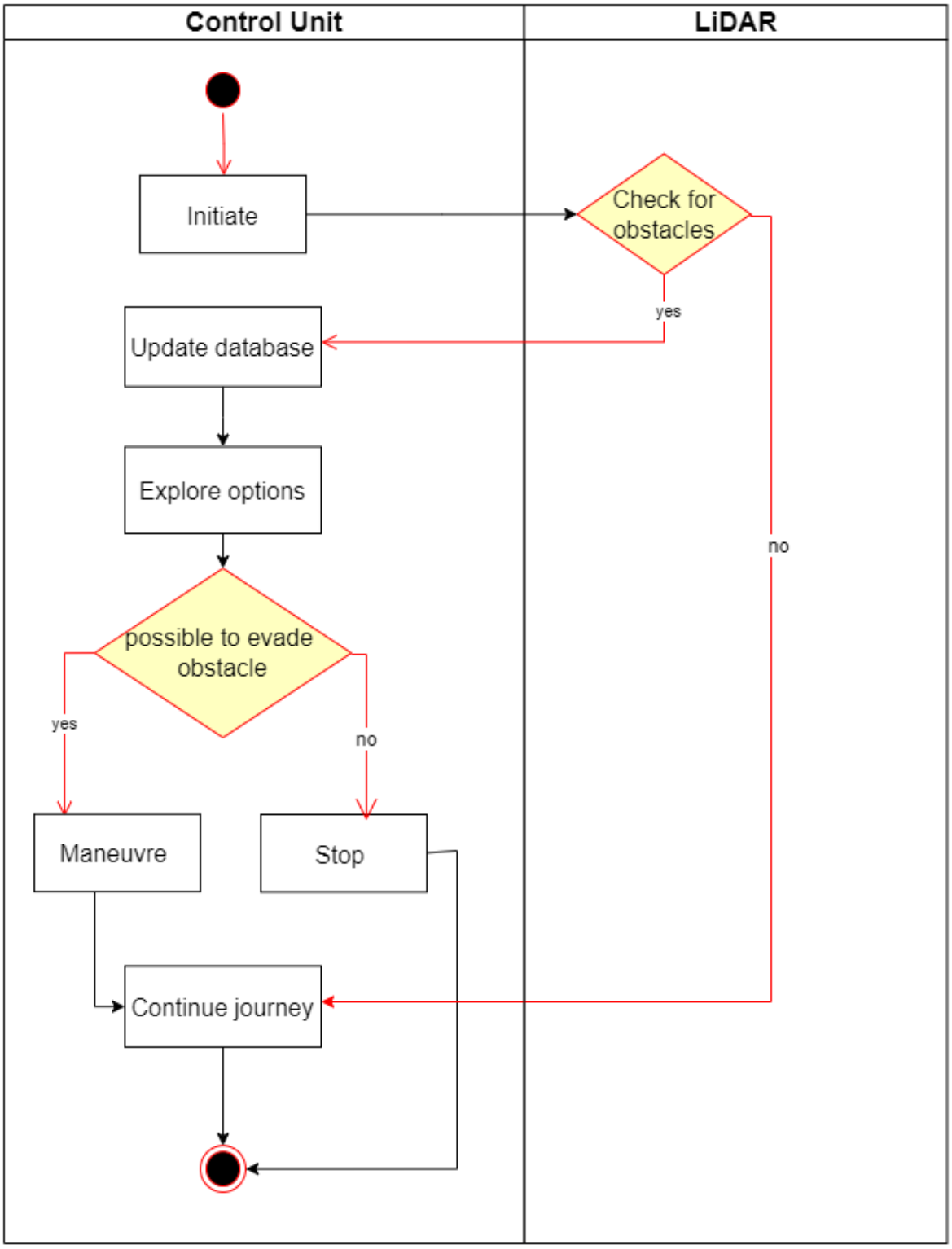
For information on the traffic condition, weather updates, and alternative routes in the event of congestion, communication between automobiles and infrastructure may be useful (Dahlman et al. 2021). This software will support V2V communication, and the information gathered from other vehicles will be stored in a list and sent to the control unit.

UML Models

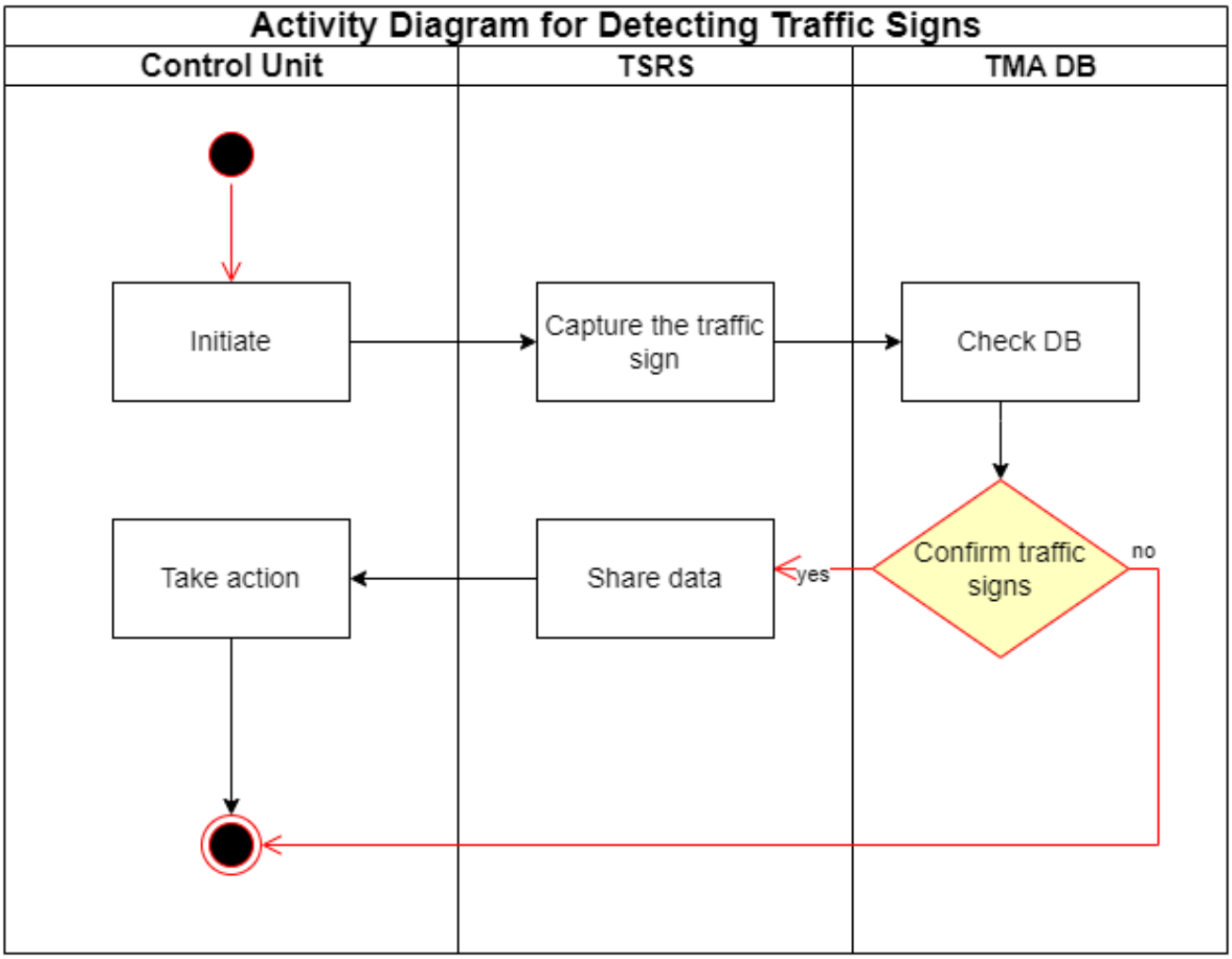
Use Case Diagram

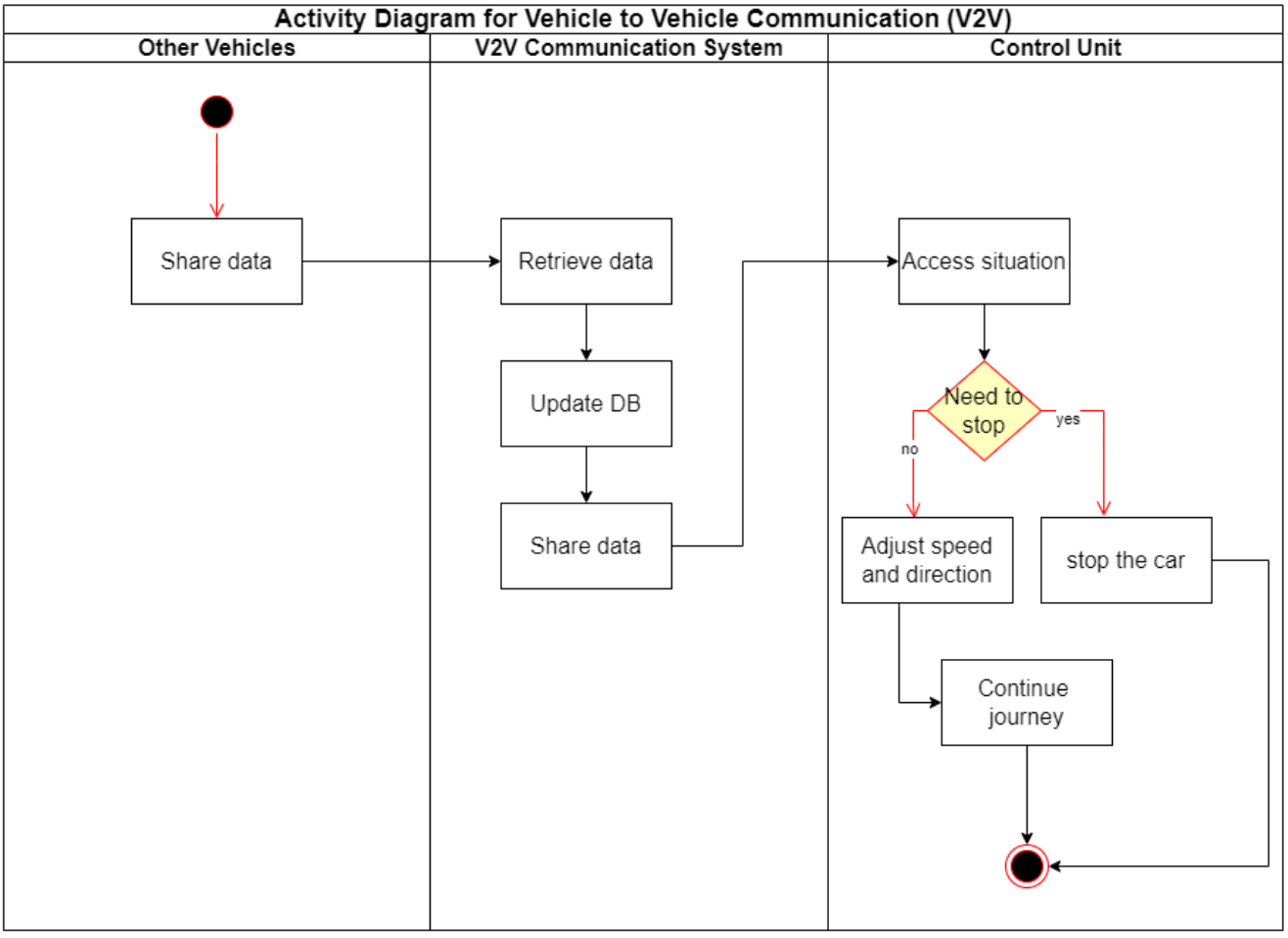


Activity Diagram

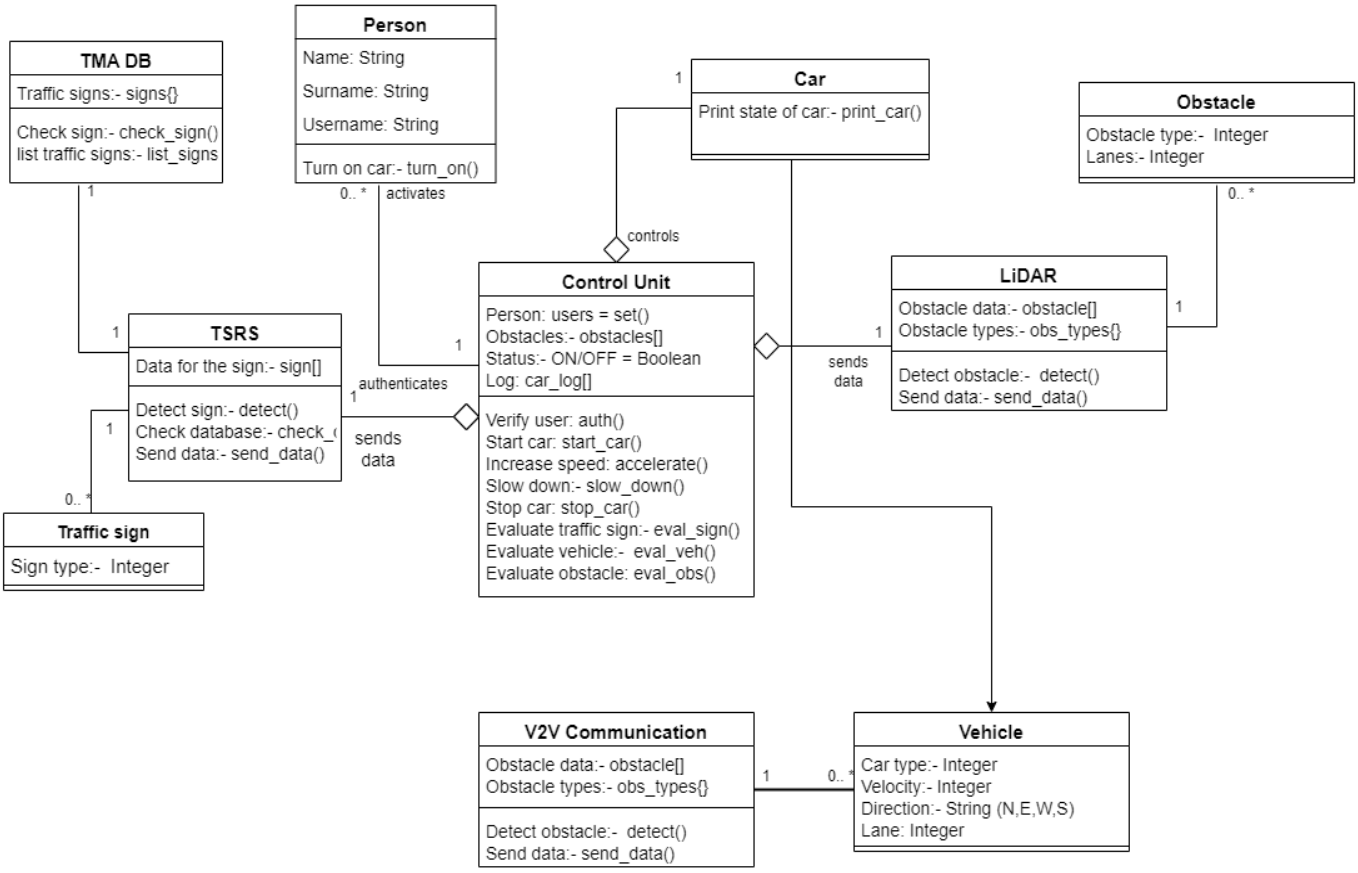


Activity diagram for Detect obstacles Use case using LiDAR

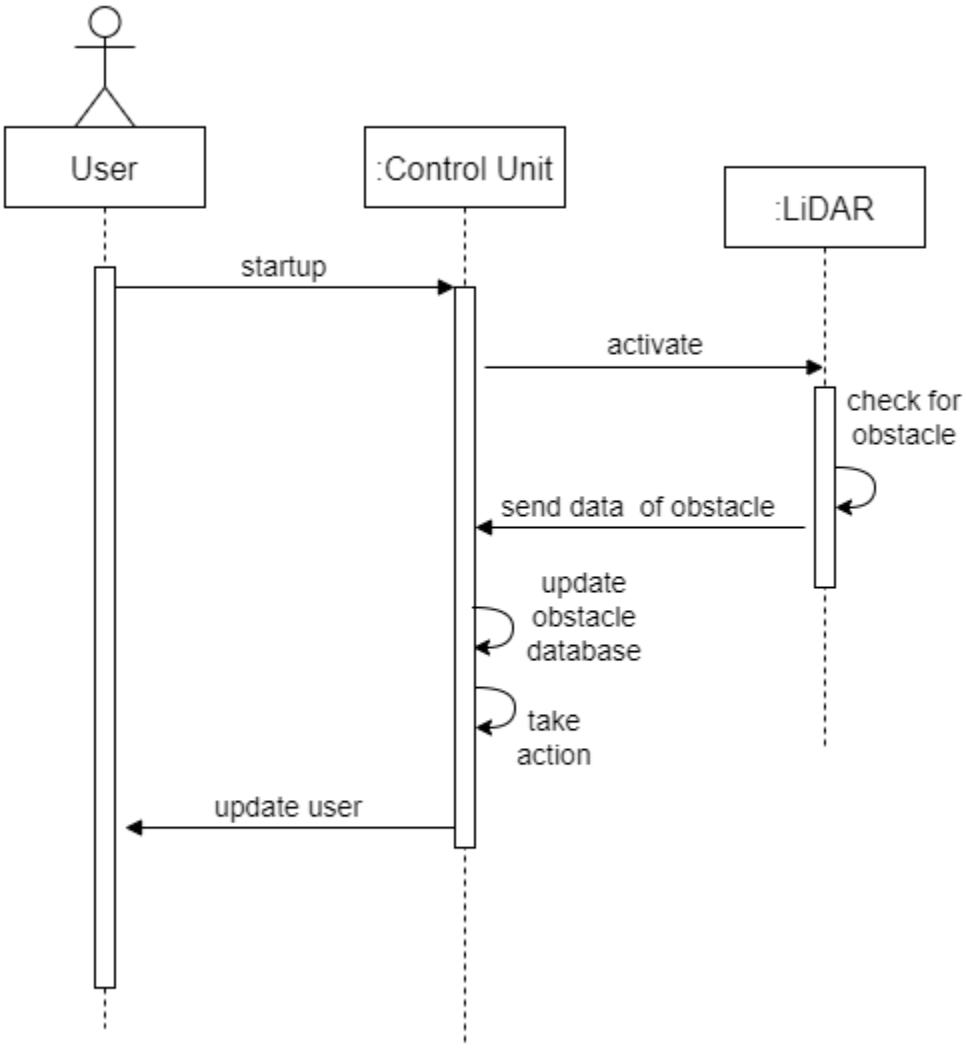

Activity Diagram for Detecting Traffic Signs
Control Unit
TSRS
TMA DB
Initiate
Capture the traffic sign
Check DB
Confirm traffic signs
yes
no
Share data
Take action


Activity Diagram for Vehicle to Vehicle Communication (V2V)
Other Vehicles
V2V Communication System
Control Unit
Share data
Retrieve data
Update DB
Share data
Access situation
Need to stop
yes
no
Adjust speed and direction
stop the car
Continue journey

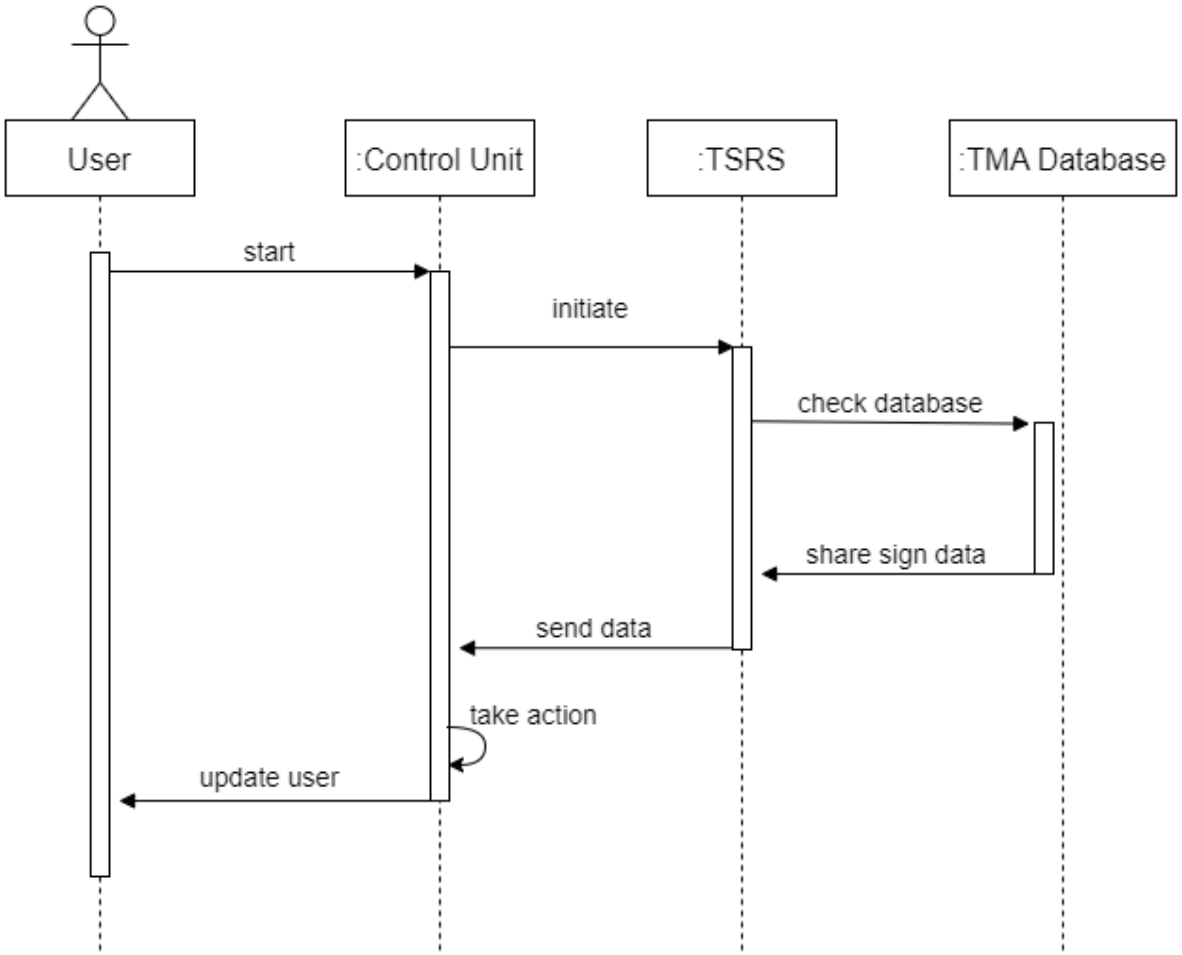
Class Diagram



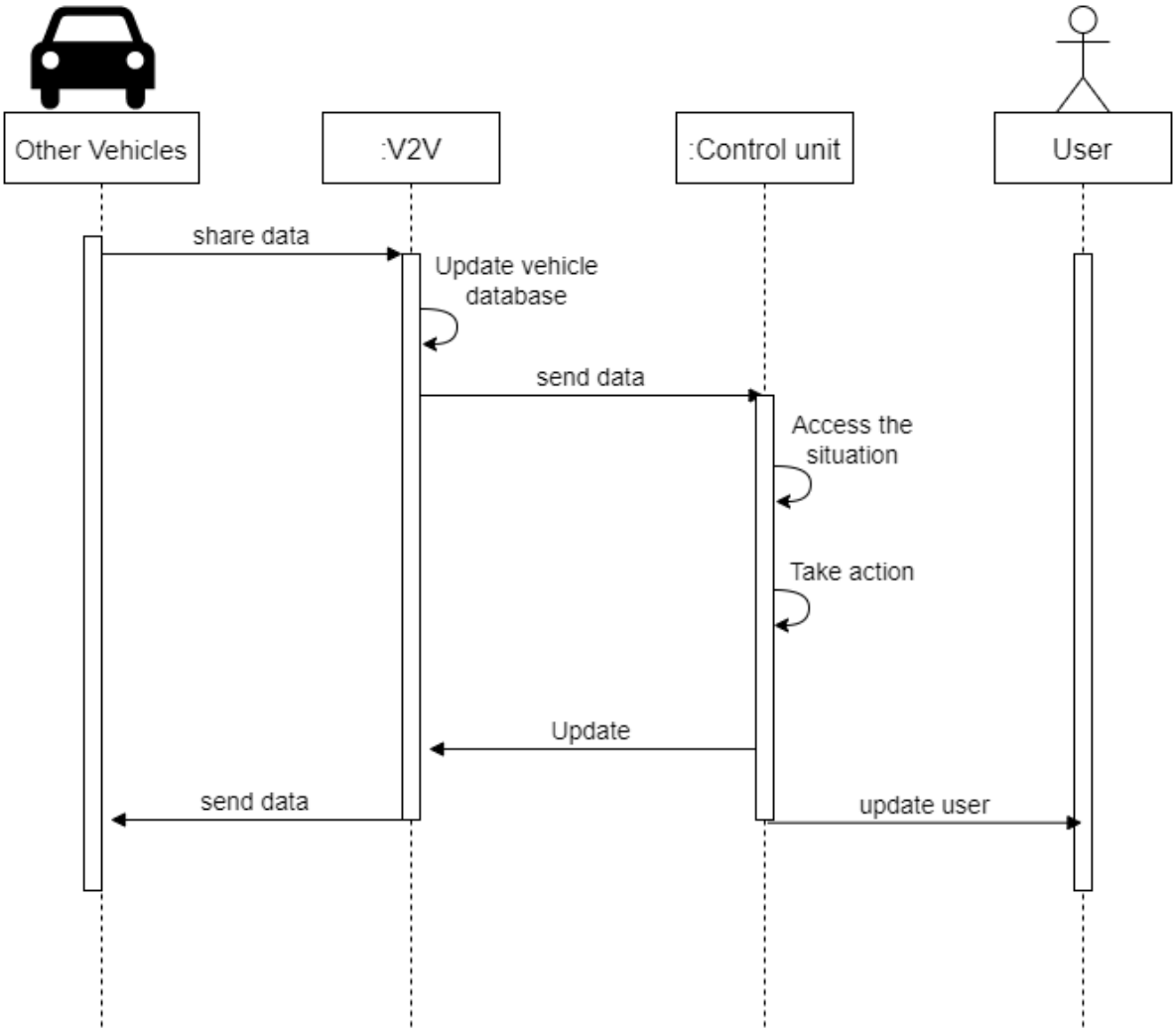
Sequence Diagram



Sequence Diagram for detecting obstacles

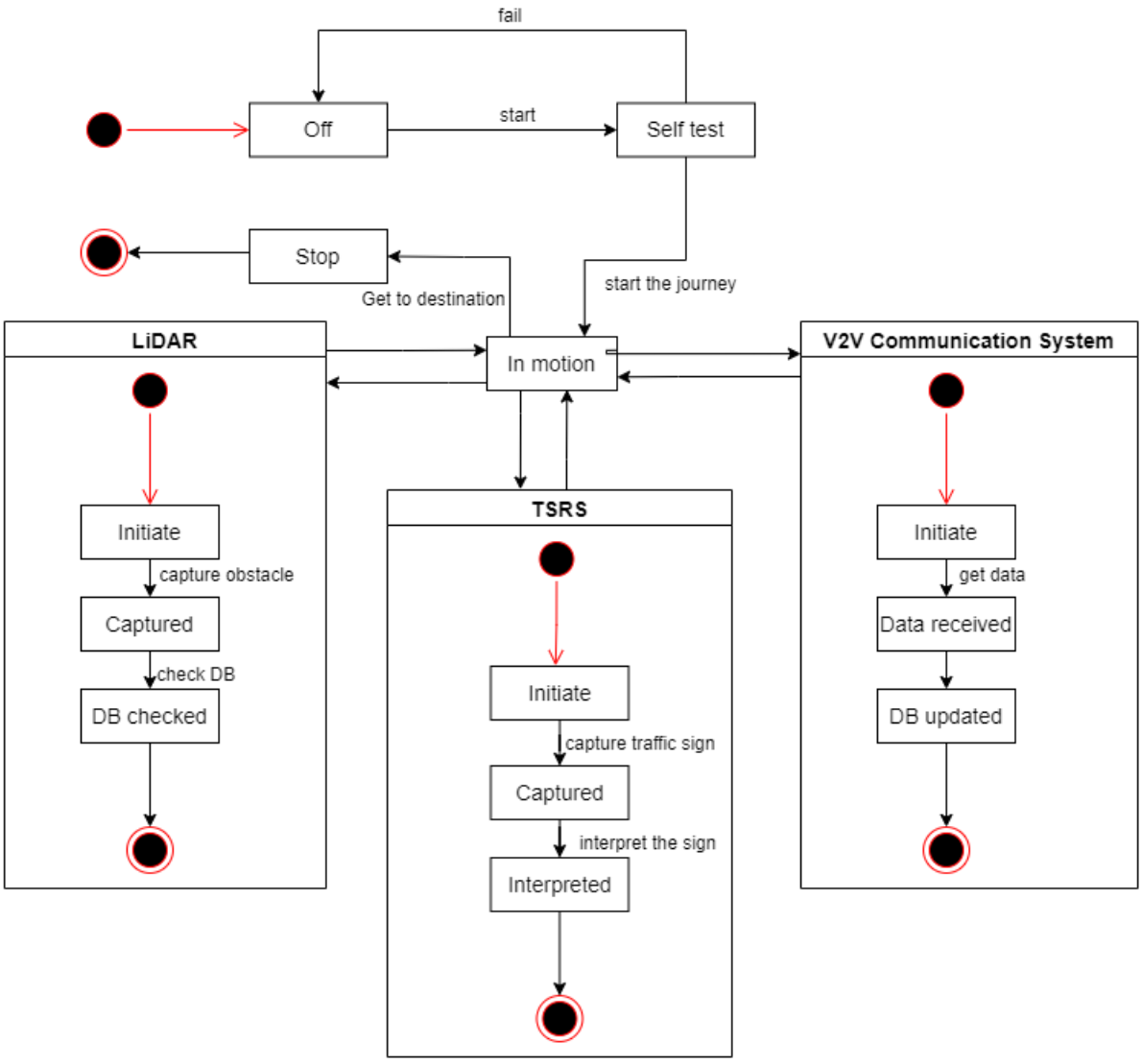


Sequence Diagram for Detecting Traffic signs



Sequence Diagram for V2V Communication

State Transition Diagram



Data Structures

- **Lists:**

Multiple elements can be kept in a single variable using lists, as shown in this software. For example, vehicles[], signs[] etc.

- **Stacks:**

Stacks are also implemented in this software as messages to the user. For example, the car log, which stores the messages that the user gets, is implemented as a stack.

- **Queues:**

A queue is a linear data structure that uses FIFO (first-in, first-out) ordering to store objects. The only queue implemented in this software is obstacles[]. Because the first barrier spotted is avoided before the others, all obstacles discovered by the LiDAR will be added to the obstacles[] queue.

- **Dictionaries:**

Dictionaries enable us to store values in key-value pairs. In this software, the sorts of obstacles and traffic signs are kept in dictionaries.

Reference

Aghdam, H. H., Heravi, E. J. & Puig, D. (2016) A practical approach for detection and classification of traffic signs using Convolutional Neural Networks. *Robotics and Autonomous Systems* 84: 97-112

Dahlman, E., Parkvall, S., & Sköld, J. (2021) 5G NR (Second Edition) The Next Generation Wireless Access Technology. *Science Direct* 39: 39-56.

Liu, Y., Peng, J., Xue, J., Chen, Y., & Fu, Z. (2021) Neurocomputing. *Science Direct* 11: 10-22.

Lutkevich, B. (2019) Self-driving Car (Autonomous Car or Driverless Car). 30 Oct 2019. *Search Enterprise AI.* Available from: https://www.techtarget.com/searchenterpriseai/ [Accessed 26 October 2022]

Reddy, P. P. (2019) Driverless Car: Software Modelling and Design using Python and Tensorflow. Available from: https://easychair.org/publications/preprint_open/k7wj [Accessed 17 October 2022]

WSJ (2017), "Why Your Next Car May Look Like a Living Room," *Wall Street Journal* (www.wsj.com); at http://on.wsj.com/2tlCvYp [Accessed 26 October 2022]

Zhou, Z. Q. & Sun, L. (2019) Metamorphic testing of driverless cars. *Communications of the ACM* 62(3): 61–67. DOI: https://doi.org/10.1145/3241979