Autonomous Vehicle System Design

**Overview**

In this document I’ll design the software which supports object detection, auto pilot and navigation for self-driving cars through a series of UML diagrams. Before creating the diagrams, I researched self-driving cars and object detection systems.

**Background Research**

Rutvik Shah suggested using sensors, cameras, computer vision and machine learning for object detection and lane detection using the YOLO algorithm created by Joseph Redmon. (Shah, 2020).

Muhammad Tahir suggested using AI, machine learning, LIDAR, Radar, GPS, and computer vision to avoid obstacles and obey road signs (Tahir, 2019).

Sarangi Lekamani created an autonomous trajectory planning system for an autonomous vehicle and suggested to uses separate microprocessors to manage different functions of a self-driving car. (Lekamani, 2018)

**Object Detection System**

The object detection system will use cameras, sensors and machine learning to detect, analyse and react to road signs, and objects. Once the object is detected its added to a queue of detected objects. I decided to store the detected objects in a queue because it’s a FIFO data structure allowing the vehicle to react to objects in the order they are detected. After a collision occurs sensors check to see if anyone is injured.

**Navigation System**

Once the destination is set the GPS will store a list of possible routes then depending on the driver’s preferences and traffic, it will decide the best possible route. The directions for the chosen route will be stored in a queue because it’s a FIFO data structure. Since a route is an order dependent series of steps to reach a destination, a queue is the optimal data structure.

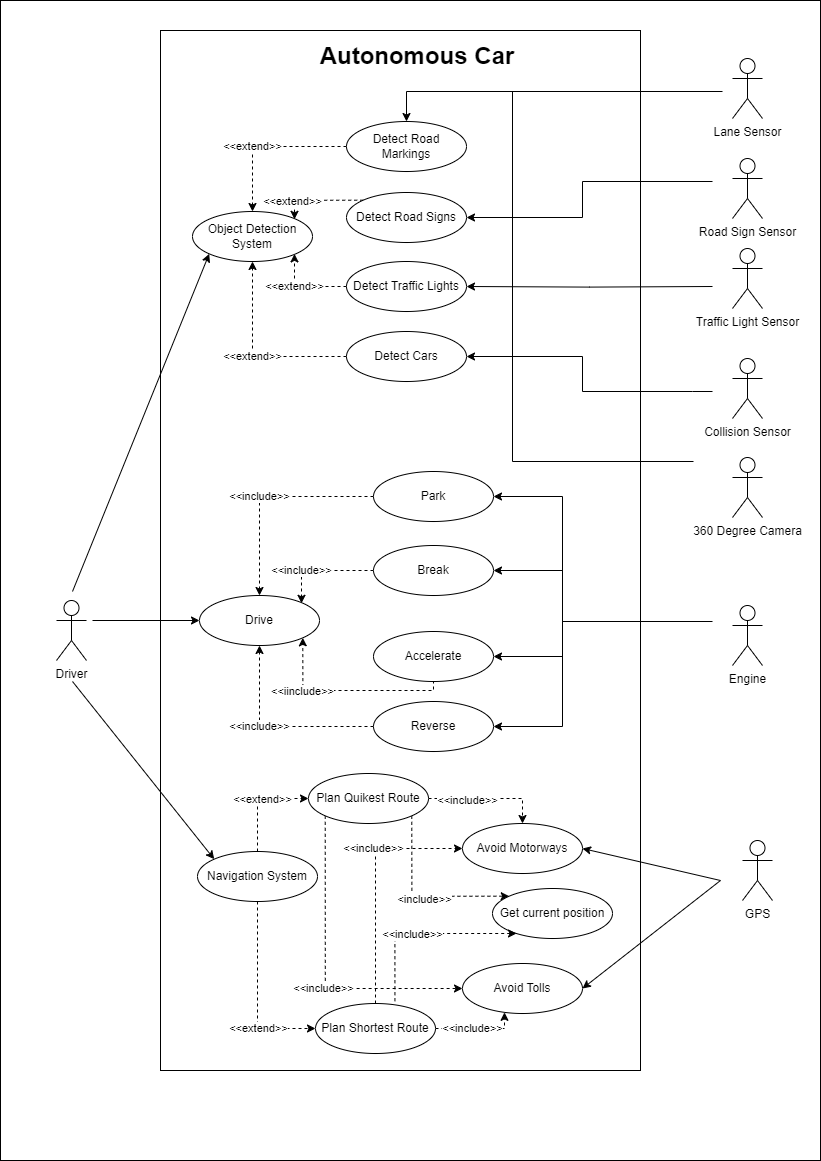
**Auto Pilot**

The driver is authenticated based on their biometrics which is stored in a dictionary. A dictionary is used to store key value pairs making it suitable for checking their biometrics match what is stored on the system. After the driver is authenticated, auto pilot is enabled, and the destination is set the vehicle sets off. If you’re running low on fuel the auto pilot system will redirect you to the closest petrol station, then once you’ve refuelled it’ll take you to your destination. Logs will be stored in a stack (a LIFO data structure) so the latest log information can be retrieved.

**Use case diagram**

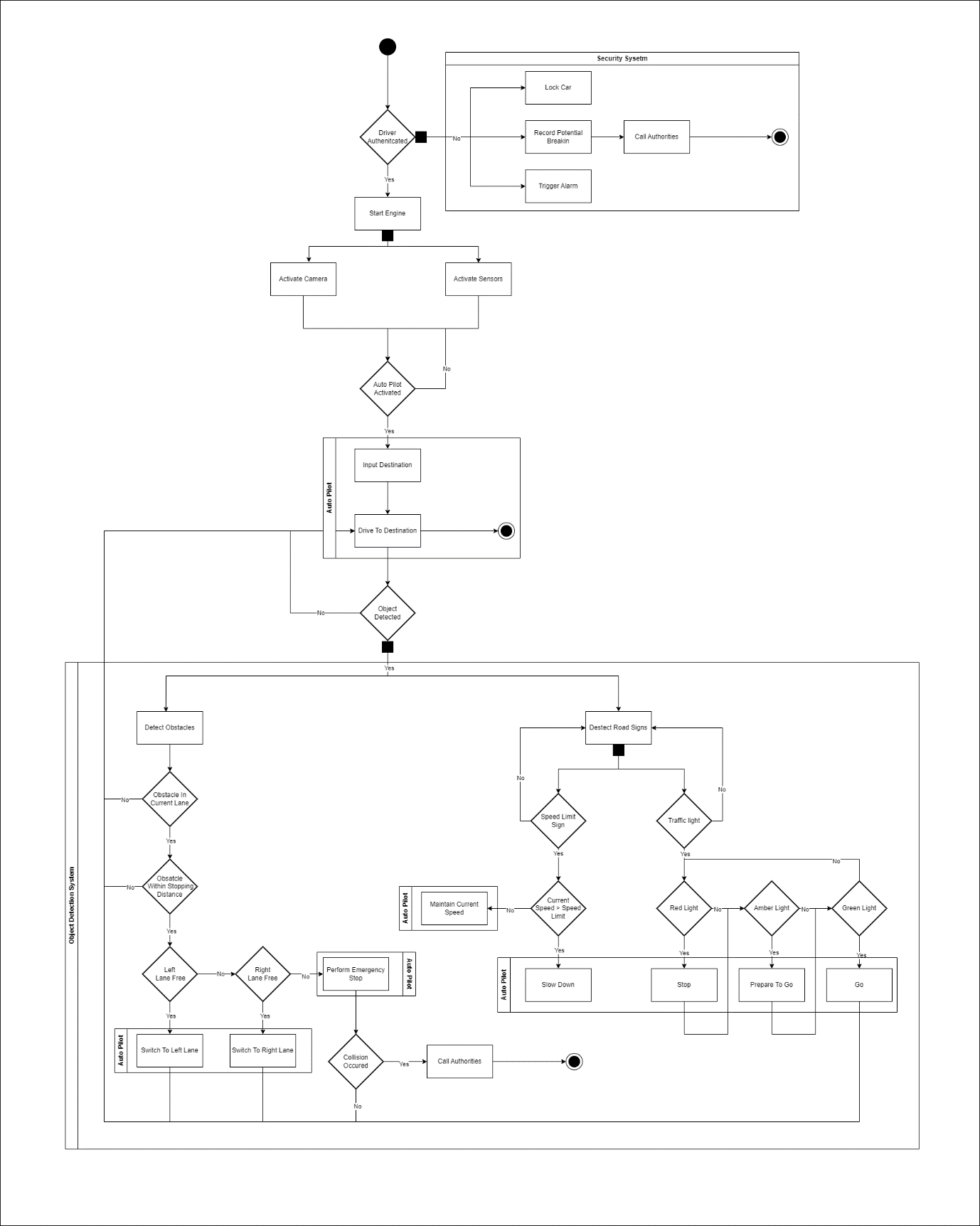
A diagram of a car

Description automatically generated



The use case diagram shows how the driver interacts with the system on a self-driving car for auto pilot, collision detection and object detection.

**Activity diagram**



The driver is authenticated then once the car is started the sensors and cameras are activated. Then the system waits for auto pilot to be turned on and for the destination to be set before setting off. Whilst on route to the destination the sensors and cameras scan for objects and road signs.

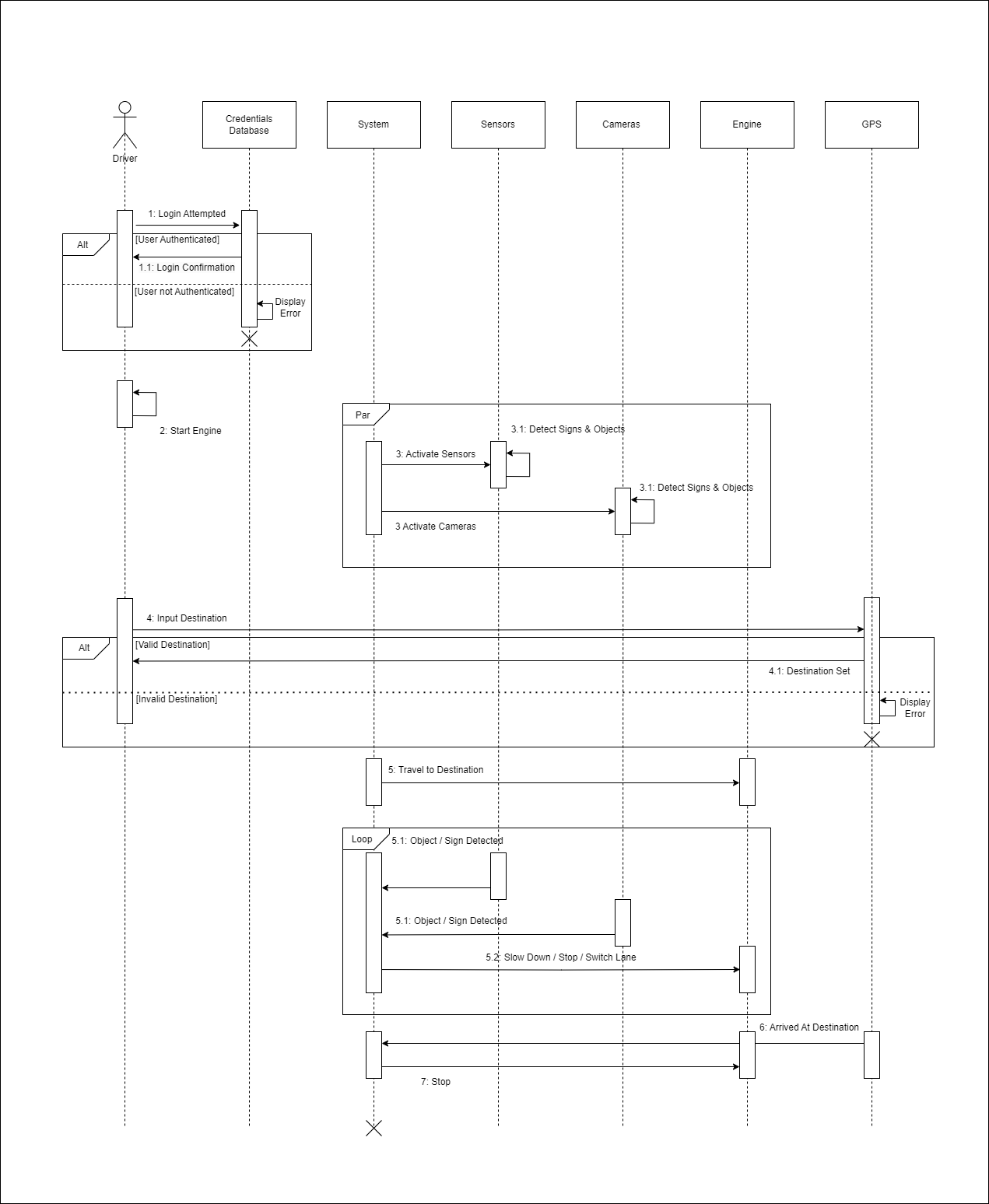
**Class diagram**

A white paper with black text

Description automatically generated

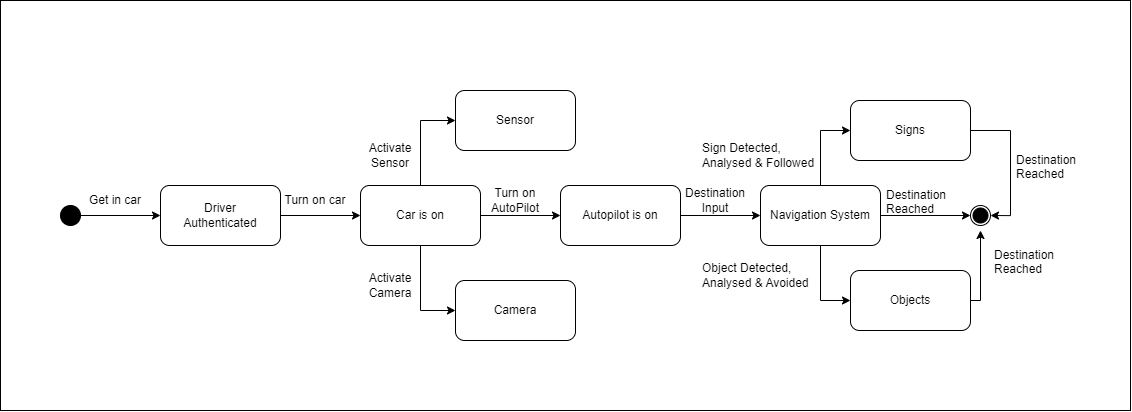
I used the vehicle class as the main class that controls the self-driving car. The vehicle class is made up of smaller classes representing different parts of the self-driving car. The reason I have designed it this way is to follow the SOLID design principles.

**Sequence diagram**



The vehicle can’t be started if its unable to verify the driver’s identity. The sensors and cameras are activated at the same time and are continuously looking for objects. If the destination isn’t valid the vehicle won’t go anywhere.

**State Transition diagram**



**Reference List**

Shah, R. (2020) L4 - Self Driving Car. Bachelor of Technology dissertation, Gujarat Technological University. Available from: <https://bvmengineering.ac.in/NAAC/Criteria1/1.3/1.3.4/16CP061_CP446_L4-SELF_DRIVING_CAR.pdf> [Accessed 07 January 2024].

Tahir, M. (2019) Driverless Car. Bachelor of Technology dissertation, Comsats University. Available from: <https://www.academia.edu/42580472/Driverless_car> [Accessed 03 January 2024]

Lekamani, S.V. (2018) Model Based Systems Engineering Approach to Autonomous Driving. Thesis, KTH Royal Institute of Technology. Available from: <https://kth.diva-portal.org/smash/get/diva2:1335887/FULLTEXT01.pdf> [Accessed 05 January 2024]

Visual Paradigm (N.D.) What is Activity Diagram? Available from: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/> [Accessed 03 January 2024]

Visual Paradigm (N.D.) What is Use Case Diagram? Available from: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/> [Accessed 03 January 2024]

Visual Paradigm (N.D.) What is Class Diagram? Available from: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/> [Accessed 03 January 2024]

Visual Paradigm (N.D.) What is Sequence Diagram? Available from: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/> [Accessed 03 January 2024]

Visual Paradigm (N.D.) All You Need to Know about State Diagrams. Available from: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/about-state-diagrams/> [Accessed 03 January 2024]