Reverse Engineering of an OEM specific CANBUS and creating an application to control the instrument cluster.

Billy Irvin
C00251069
Richard Butler
South East Technology University

17 April 2023

Abstract

A vehicle's internal CAN network is made up of a number of parts e.g., sensors, actuators and ECUs. With this project, I want to take control of the CAN network and alter its message structure so that the vehicle can perform activities it wouldn't normally be able to. I'm then going to develop a desktop application that will enable a user to have the ability to control the CAN bus by sending their own CAN message onto the CAN bus and throughout the CAN network in order to accomplish this goal of reverse engineering and manipulating messages on a CAN network.

Table of Contents

Introduction

In this document I will outline the functional specification of my project and I hope to see more machine reverse engineering in year four of the cyber security course. I will discuss the components of my application and the system requirements that are needed. I will provide an overall idea of what I want my application to look like. I will furthermore design a use case diagram to explain more in-depth my application.

Functional Specification

A functional specification is a formal document used to describe a product's intended capabilities, appearance, and interactions with users in detail for software developers. The functional specification is a kind of guideline and continuing reference point as the developers write the programming code. (Rosencrance, 2019)

Project scope

To build an easy-to-use standalone instrument cluster control application which will allow an end user to control the instrument cluster of their vehicle

Functional Specification Scope

This document will outline how the developed application will function to control the instrument cluster of a vehicle and the functionalities of the application. I will outline the actors that would use this application. With the use case diagram, I will explain the step needed for the actor to use the application I will take examples from the application to show how the application will interact with the actors.

Requirements

System Requirements

This tool will be platform-independent tool meaning it can run on various systems

System

Context diagram

The Context Diagram shown defines and clarifies the boundaries of the proposed system. It identifies the flows of information between my system and external entities. The system is shown as a single process (University of Cape Town, 2011).

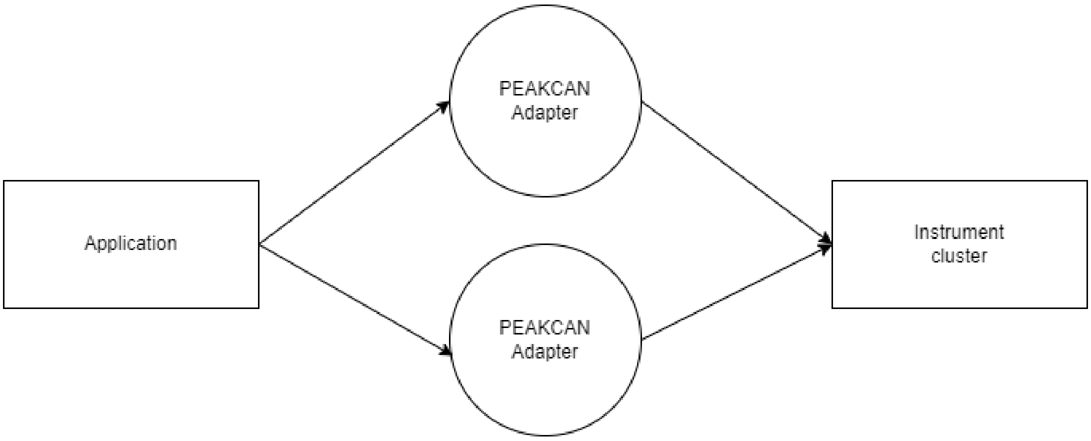

Figure 1: context diagram

Use Case Diagram

A use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system. (Contributor, 2020)

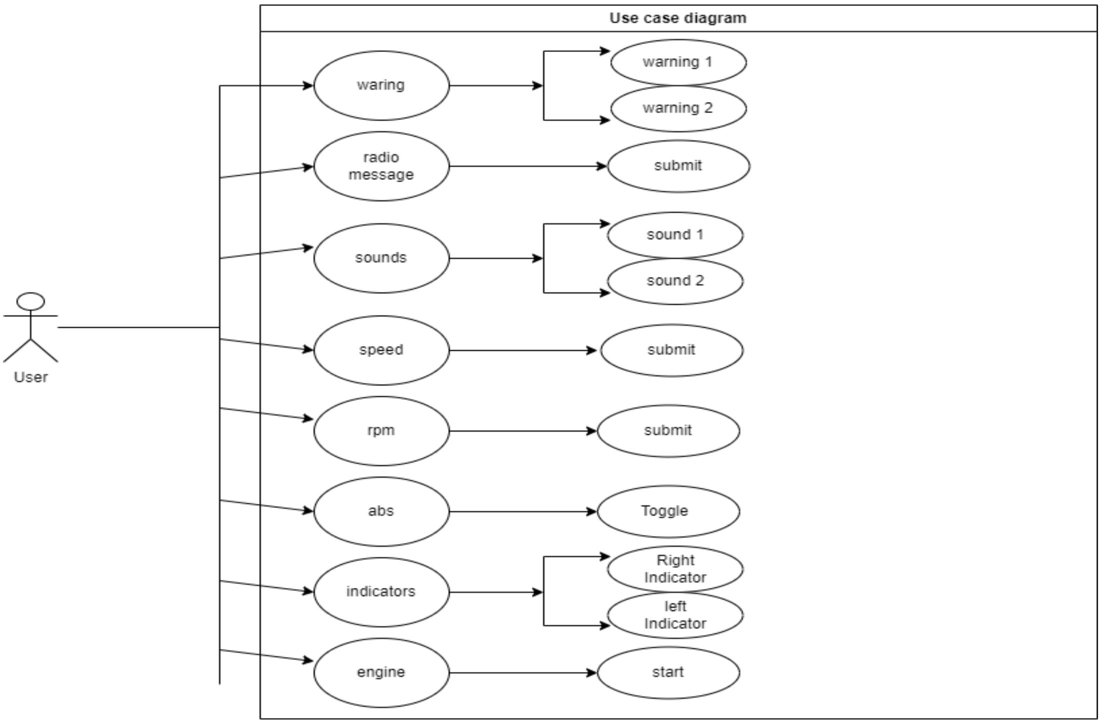

Figure 2: Main use case

Functional Specification

Multiple Specifications are implemented in the overall system they are classified with two headings Core and Non-Core, as shown below.

Name	Classification
Warning	Core
Radio Message	Core
Sounds	Core
speed	Core
Rpm	Core
abs	Core
Indicators	Core
Engine	Core

Warning

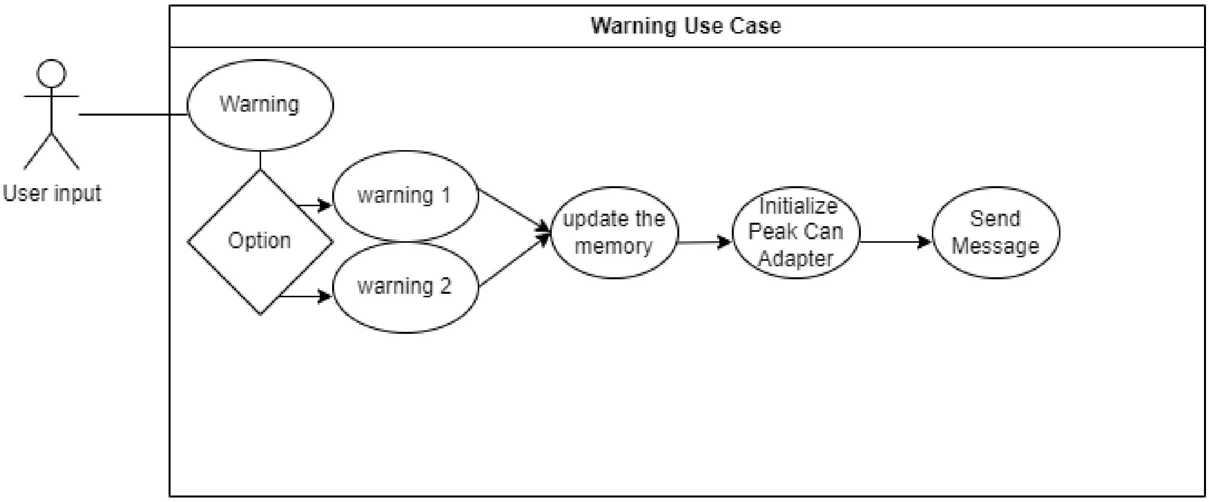

Figure 3: warning use case

Use case

UC-1	Warning
Primary Actor	User
Stakeholders	User
Trigger	User input
Pre-conditions	Instrument cluster is blank
Post-conditions	Message display on Instrument cluster
Main Success Scenario	Message is executed on instrument cluster
CAN Message Priority	Low
Priority	High

Radio Message

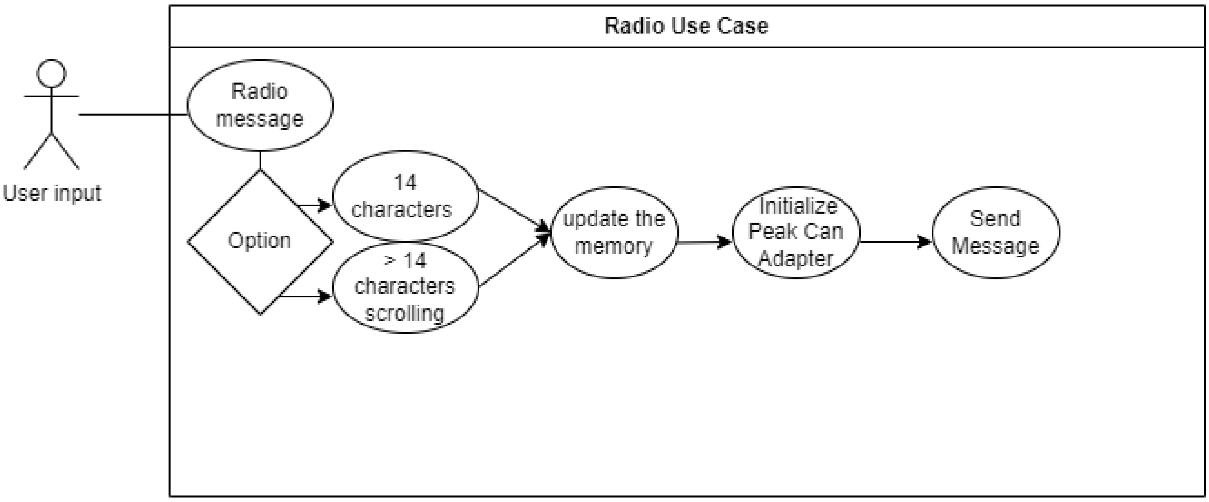

Figure 4: Radio use case

Use Case

UC-2	Warning
Primary Actor	User
Stakeholders	User
Trigger	User input
Pre-conditions	The instrument cluster is blank
Post-conditions	Message display on the radio the of Instrument cluster
Main Success Scenario	Displays text entered on the radio of the instrument cluster
CAN Message Priority	Low
Priority	High

Sounds

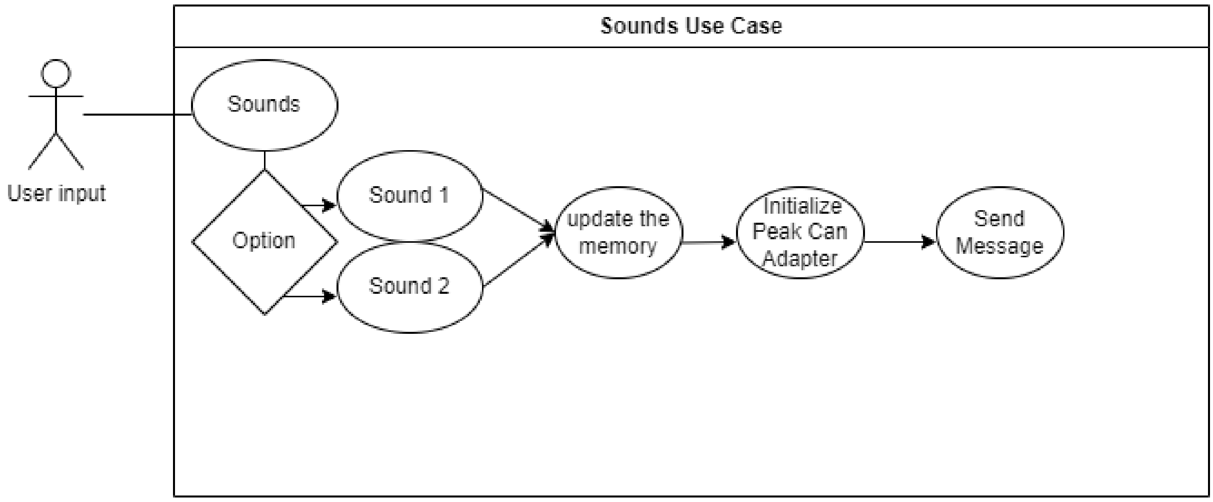

Figure 5: sounds use case

Use Case

UC-3	Warning
Primary Actor	User
Stakeholders	User
Trigger	User input
Pre-conditions	The instrument cluster is blank
Post-conditions	The message is executed on the instrument cluster
Main Success Scenario	Noise comes out of the Instrument cluster
CAN Message Priority	Low
Priority	High

Speed

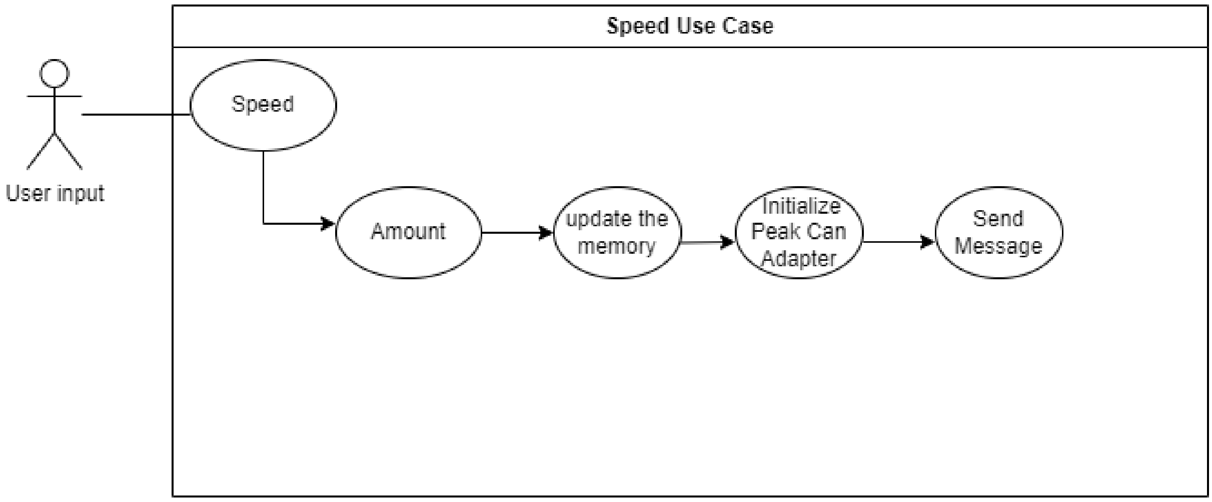

Figure 6: speed use case

Use Case

UC-4	Warning
Primary Actor	User
Stakeholders	User
Trigger	User input
Pre-conditions	The instrument cluster is blank
Post-conditions	The message is executed on the instrument cluster
Main Success Scenario	Speed increases on the Instrument cluster
CAN Message Priority	High
Priority	High

Rpm

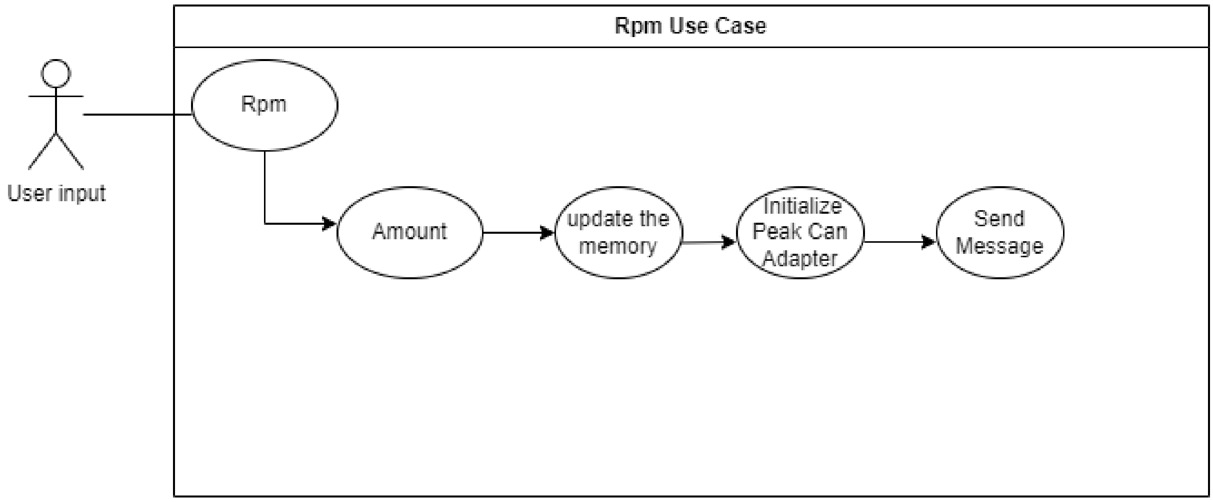

Figure 7: Rpm use case

Use Case

UC-5	Warning
Primary Actor	User
Stakeholders	User
Trigger	User input
Pre-conditions	The instrument cluster is blank
Post-conditions	The message is executed on the instrument cluster
Main Success Scenario	Rpm increases on the Instrument cluster
CAN Message Priority	High
Priority	High

Abs

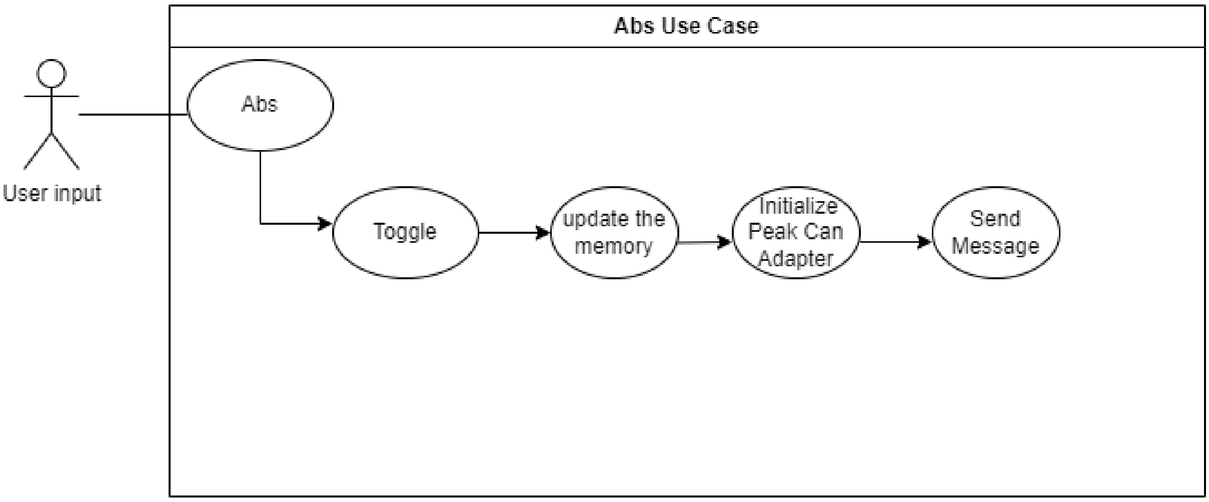

Figure 8: Abs use case

Use Case

UC-6	Warning
Primary Actor	User
Stakeholders	User
Trigger	User input
Pre-conditions	The instrument cluster is blank
Post-conditions	The message is executed on the instrument cluster
Main Success Scenario	Abs is toggled on or off on the Instrument cluster
CAN Message Priority	Low
Priority	High

Indicators

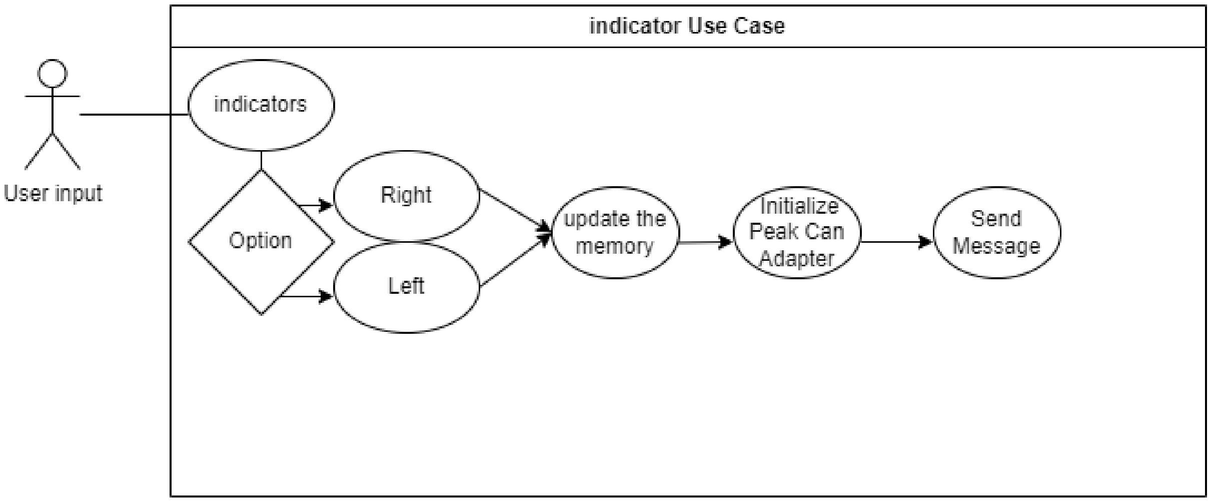

Figure 9: indicator use case

Use Case

UC-7	Warning
Primary Actor	User
Stakeholders	User
Trigger	User input
Pre-conditions	The instrument cluster is blank
Post-conditions	The message is executed on the instrument cluster
Main Success Scenario	The indicator is displayed on the Instrument cluster
CAN Message Priority	High
Priority	High

Engine

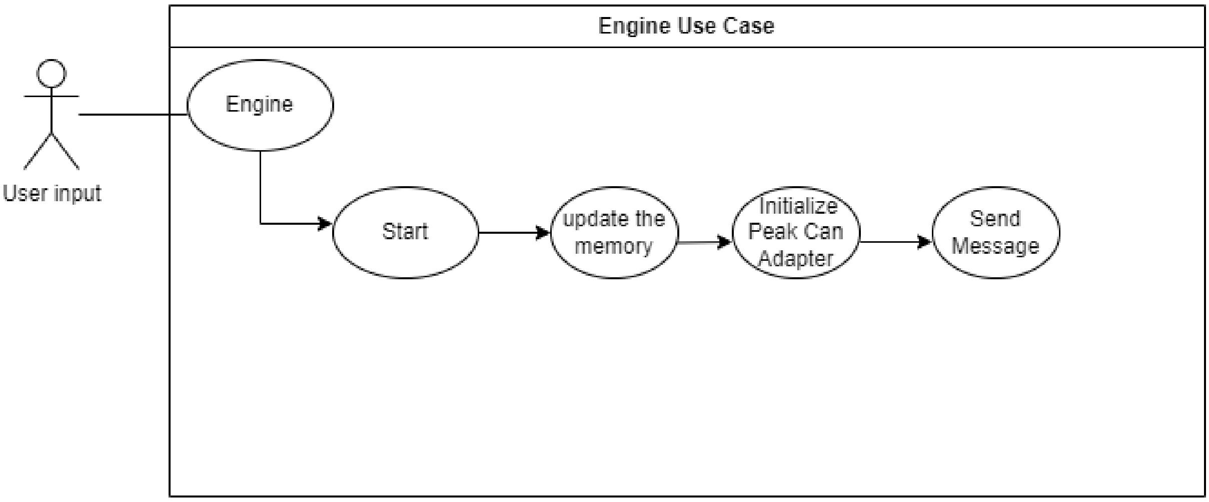

Figure 10: Engine use case

Use Case

UC-8	Warning
Primary Actor	User
Stakeholders	User
Trigger	User input
Pre-conditions	The instrument cluster is blank
Post-conditions	The message is executed on the instrument cluster
Main Success Scenario	Lights clear off on the Instrument cluster
CAN Message Priority	High
Priority	High

Components of the Application

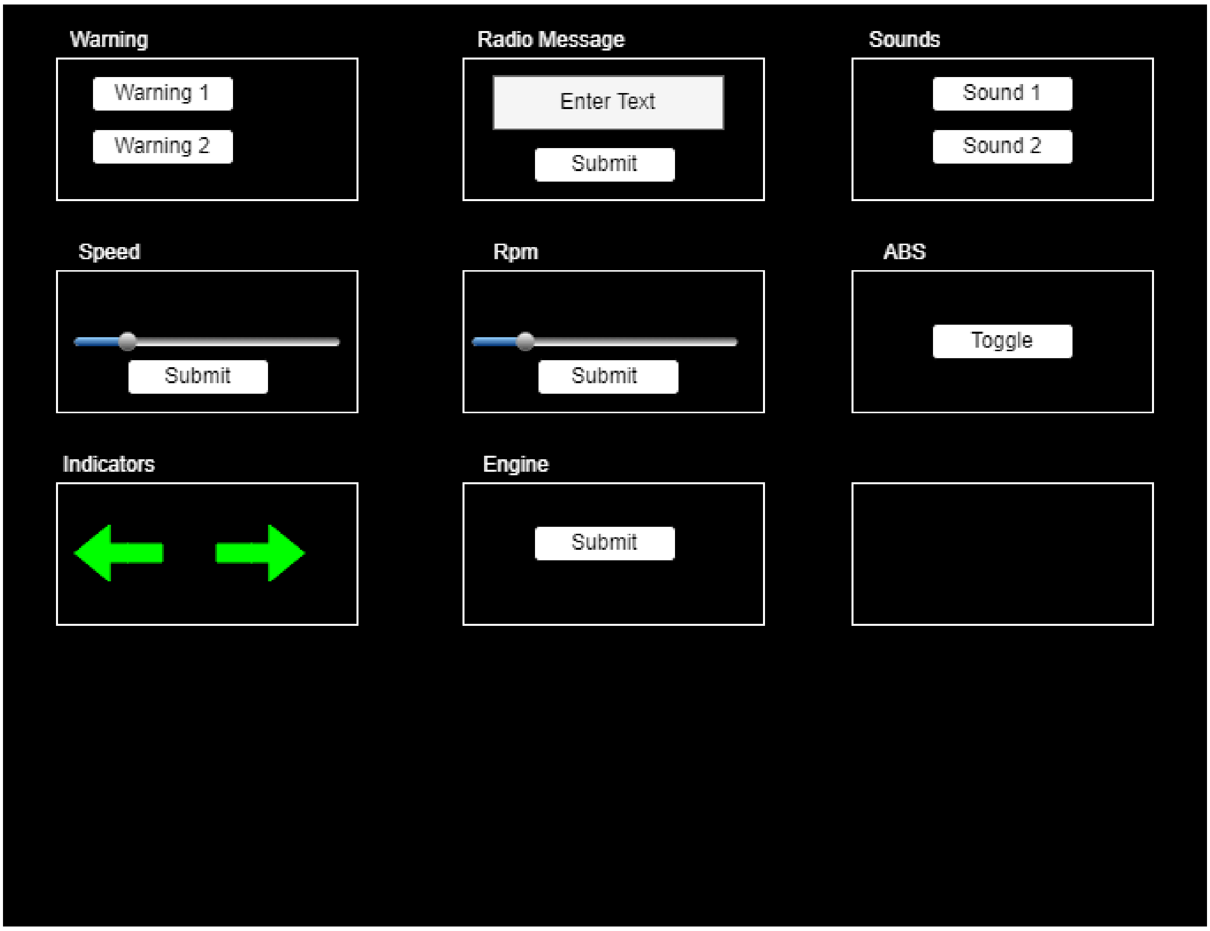

Figure 11: application

The above is an example of what I would like my application to look like it should have the following functionality.

Name	Description
Maintainability	The application will be fully supported and easy to update.
Usability	The application will be user-friendly.
Availability	The application will have a 95% availability rate.
Reliability	The application will always perform the tasks it has to do.

External tools/libraries

The following are needed in the development of the application:

- Peak Can USB adapter – This adapter allows you to read CAN messages being sent across the network and send CAN messages into the network.

- Peak Can basic API – This allows the python code to communicate with the adapter and gain all the functionality e.g., sending messages, and reading messages.
- PyCharm – This is the IDE I will be using to code the application.

Project Plan

Hardware/software requirements

Follow is the software and hardware I used for the implementation of this project:

Software
Python
PyCharm
Peak Can basic API
Windows 11

Hardware
Peak Can Adapter
Mazda Dashboard
16g ram
Core i7

Table below is the project time frame:

Task	Start Date	End Date	Duration (days)
Research Manual	17/10/2022	25/11/2022	40
Functional Spec	25/11/2022	19/12/2022	25
Presentation	19/12/2022	19/12/2022	1
Learning testbed and adapters	28/12/2022	6/01/2023	10
Reverse engineering	7/01/2023	20/02/2023	45
Learning Python	21/02/2023	28/02/2023	8
Learning to implement API	22/02/2023	03/03/2023	10
App Development	21/02/2023	31/03/2023	39
Final Report	01/04/2023	17/04/2023	17

References

Rosencrance, L. (2019) *What is a functional specification document?*, *Software Quality*. TechTarget. Available at: https://www.techtarget.com/searchsoftwarequality/definition/functional-specification (Accessed: December 12, 2022).

University of Cape Town (2011) Chapter 6. Data-Flow Diagrams. Available at: https://www.cs.uct.ac.za/mit_notes/software/htmls/ch06s06.html (Accessed: December 12, 2021)

Contributor, T.T. (2020) *What is a use case diagram?*, *WhatIs.com*. TechTarget. Available at: https://www.techtarget.com/whatis/definition/use-case-diagram (Accessed: December 15, 2022).