Cruise Control Software Development Document

Team Puppies

David Liang, Aparajita Rana, Susmitha Shailesh, Max Shi

Version 2.0

30 March, 2020

Table of Contents

Section 1: Executive Summary **3**

Section 2: Introduction **4**

Section 3: Project Requirements **6**

Functional Requirements 6

Input 7

Output 8

System Requirements 8

Performance Requirements 9

Reliability Requirements 9

Security Requirements 9

Section 4: Use cases and UMLs **11**

Use Cases 11

Use Case Diagram 18

Class-Based UML + Index Cards 19

Activity Diagram 22

State Diagram 22

Section 1: Executive Summary

We aim to develop the software for a cruise control system that would include all the main features of cruise control a consumer would typically expect, such as the capability to set the desired speed or adjust the speed already set while in operation. It will incorporate the standard sensors and physical hardware most cruise control systems use in order to regulate the speed of the car. However, the system will also be robust enough to handle unexpected interruptions to the control equipment, given the ramifications of implementing the system into real-world scenarios in order to provide the safest and most user-friendly cruise control possible. Overall, we aim to develop a fully-operational cruise-control software with an agile design process to meet the needs of users of cruise-control in a fast, reliable manner in an exhaustive number of scenarios.

Section 2: Introduction

Today, cruise control systems have become commonplace in most vehicles. With the ability to maintain a certain speed without the need to keep the accelerator held down, cruise control has made the lives of drivers significantly easier. That being said, there are obviously pros and cons to cruise control. The obvious advantage to using cruise control is that it reduces driver fatigue when on highways or sparsely populated roads. It also avoids any potential of speeding and utilizes fuel more efficiently. In regards to drawbacks, cruise control cannot be used in certain weather conditions, winding roads, and could promote distracted driving. Therefore, in developing a cruise control, we would ideally seek to solve the listed cons as much as possible. With the recent popularity of autonomous vehicles, there are many more resources regarding cruise control than there were in the 1960s when cruise control just started.

While the main purpose of cruise control is simply to maintain a desired speed, there are also many other basic functionality requirements we must account for as well. For example, the cruise switch can typically consist of multiple commands including, but not limited to, an on and off switch, a way to change the set cruise control speed, and the ability to temporarily stop and resume cruise control. We must also consider how cruise control may interact with the car in its entirety. If the driver were to brake while the cruise control was still active, it would have to suspend itself until the driver resumes cruise control, such as by engaging the system again. Other features could be setting a minimum speed requirement to be able to use cruise control, or refusing to engage cruise control when the car's sensors report it is unsafe to do so, such as low tire pressure.

We aim to develop the software required to interpret input from human interfaces and data from internal automotive sensors in order to send signals to the car's powertrain and braking system to modulate speed. The physical interfaces outside of the software are outside the scope of this project, and this project will act as a black box for the inputs and outputs in the rest of the vehicle. In addition to the core systems of engaging and disengaging cruise control as well as maintaining speed, the system will be robust enough to handle unexpected situations, such as abrupt disconnection of power, damage to certain control systems, or rapid failure of systems, such as in a collision. Overall, we aim for zero failures, as this is a mission-critical piece of software with lives at stake, as well as steady and predictable operation of the cruise control when presented with multiple situations.

In order to succeed in this project, we will have to first have a thorough understanding of the mechanisms of embedded systems and cruise control in particular. We will need to decide what the main requirements and goals of our product must be, both for consumers and makers of the product. We will need to consider the use cases for cruise control, including what users expect from a cruise control system and how the system will be applied in practical cases. We will do research into similar systems already on the market and their strengths and weaknesses. Based on all of this information, we will begin the design of our product. When designing our product, we will keep in mind the requirements that we had set and the use cases that the product must satisfy. Finally, we will go into the testing of the product and restart the process until we create a viable final design. Overall, we will utilize this sort of "agile" design process when approaching this project, in order to keep costs down, be flexible to consumer demands, and ensure smooth progress in the project.

Section 3: Project Requirements

Functional Requirements

1. The cruise control software shall have two different states.
    a. The first state shall be when the software and system are turned on when the engine starts; the cruise control shall be ready to be set at a specific speed.
    b. The second state shall be when the software is activated; the cruise control shall maintain the set speed.
2. The Cruise Control system should check that basic functionalities of the car, through testing connections with sensors and input devices as well as the EMS, is operational for the safety of the passengers and driver on startup.
    a. The tire pressure sensors must report adequate tire pressure for driving. (This can be set in a configuration file by the manufacturer, as different cars require different tire pressures)
    b. The EMS must not report any critical errors from its own system checks when the car starts.
    c. The Cruise Control system should check that all levers and buttons to operate the cruise control system are connected properly.
3. The cruise control shall only activate if a minimum speed requirement of twenty miles per hour is met.
    a. This feature shall be able to be disabled or modified (the minimum speed) through the manufacturer of the car through the aforementioned configuration file.

4. The cruise control system will remain powered on through the car's battery for one minute after the engine has been shut off.
    a. The system will use this time to log deactivation of the system. The logging capabilities are detailed in the "Output" section of these requirements.
    b. The cruise control system will power off one minute after the engine has been shut off.
5. In case of unexpected events such as abrupt power shutdown, car engine issues, or running out of gas, the cruise control system shall display a warning to the driver that there is an issue with the system and it is about to turn itself off. Then, it will turn itself off.

Input

1. The cruise control system shall receive input from the driver through articles of the car's hardware, such as buttons or levers, that it should be turned on.
2. The cruise control shall receive the current speed as an input upon activation through the EMS, every time it activates.
3. The cruise control system shall increase or decrease the desired cruise speed by receiving input from the driver through two buttons that change the speed by 1 mph in the desired direction.
    a. The only increments in this cruise control system will be 1 mph up/down at a time.
4. The cruise control system shall receive information from sensors that the brake has been applied by the driver.

5. If the driver brakes, the software shall move back into the first state (ready for activation).
6. If the driver deactivates the system, the software shall also move back into the first state.

Output

1. When the driver activates the system, the system shall send an activation request to the EMS.
2. There shall be visual feedback to inform the driver that cruise control has effectively been turned on.
3. In the activated state, the cruise control shall communicate with the throttle through the EMS to speed up or slow down to maintain the set speed. (Design goal: within 2mph)
4. The cruise control shall display that it has been turned off and why it has been turned off (driver input, unexpected event) to the driver.
5. The cruise control will log and timestamp cruise control activation, the set speed, changes in speed, and the deactivation of the system.
6. When looking to deactivate, the cruise control should provide a deactivation request to the throttle EMS.

System Requirements

1. The cruise control system shall incorporate 1 GB of memory storage for logging capabilities.
2. The cruise control software requires the necessary connections to sensors observing the road and driving conditions, the engine management system via the throttle, and human interfaces (buttons, levers, etc) to function properly.

3. Cruise control hardware shall be able to receive power both from battery and alternator.
4. The cruise control hardware shall have a system clock for logging purposes.
5. An authorized technician shall be able to download the system logs from the software.
6. The system will activate itself within half a second after the activation signal has been sent by the user.
7. The cruise control software will follow the communication protocol specified by the manufacturer, industry, and government in order to keep things safe and secure.

Performance Requirements

1. The cruise control software shall turn on and be ready to set a speed within 2 seconds of receiving the input from the driver.
2. The cruise control shall begin maintaining the speed it received as an input at startup within 1 second of activation.
3. The cruise control shall acquire and begin maintaining the new desired speed within 1 second of receiving the driver's input for a speed change.
4. The cruise control system shall log information about the state of the system within 1 second after the shutdown signal or after deciding that shutdown is necessary due to unexpected events.

Reliability Requirements

1. The software shall meet 99.999% reliability, excluding the reliability of the hardware.

Security Requirements

1. The software shall not have any access to or interfaces with wireless systems such as the internet, Bluetooth, or radio.

2. The software shall be password protected to ensure that only authorized technicians can service the software and the system.

Section 4: Use cases and UMLs

Use Cases
Use Case 1: User sets cruise control.
Actor: Driver
Goal: Set cruise control to the current speed of the car.
Preconditions: Car and cruise control is activated and the car's speed is above 25 mph.
Trigger: The driver pushes the "Set" button.

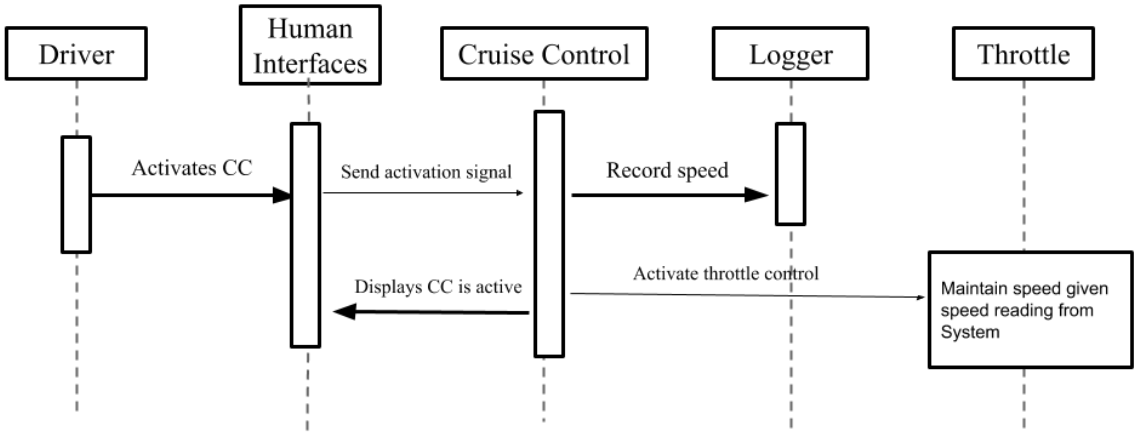
1.) The driver turns on cruise control through the "set" button.
2.) The speed sensor reads the current speed and logs it as the target cruise control speed (If there is a speed already set it overwrites it) and sends a signal back to the cruise control software.
3.) The actuator reads the logged speed and manipulates the throttle body to maintain the set speed and then returns a signal back to the cruise control software.
4.) The cruise control system provides visual feedback that cruise control has been successfully set.
5.) If the speed sensor reads a speed faster than the set speed, the actuator will be alerted and close the throttle body until the set speed is reached. If the car is too slow, then the throttle body will open and speed up the car.
6.) The speed sensor constantly reports the current speed and adjusts accordingly.

<u>Exception Use Case</u>

1.) The speed sensors do not send back a signal, cruise control will not activate and provide visual feedback that cruise control can not be set.
2.) The actuator does not send back a signal, cruise control will not activate and provide visual feedback that cruise control can not be set.

3.) Car has not met minimum speed and will provide visual feedback that the car must be driving at 25 mph or faster to activate.

4.) If the speed can't be logged, it will provide visual feedback that something is wrong and cruise control will not turn on



Use Case 2: User wishes to increase or decrease the set speed.
Actor: Driver
Goal: Increase the set cruise control speed
Preconditions: Cruise control is on and a speed has been set and the set speed must be above 25 always be above 25 mph.
Trigger: The driver selects the "+" or "-" buttons.

1.) The user will select "+" or "-" to increase or decrease the speed respectively.

2.) If the user taps "+", it will increase the current logged speed by 1 mph and return a signal back that it has successfully done so. Vice versa for "-".

3.) The speed sensor should be constantly reporting the speed and will report the current speed to the actuator.

4.) The actuator reads the set speed and adjusts the throttle body to achieve the new set speed and sends visual feedback that the cruise control has reached the new set speed.

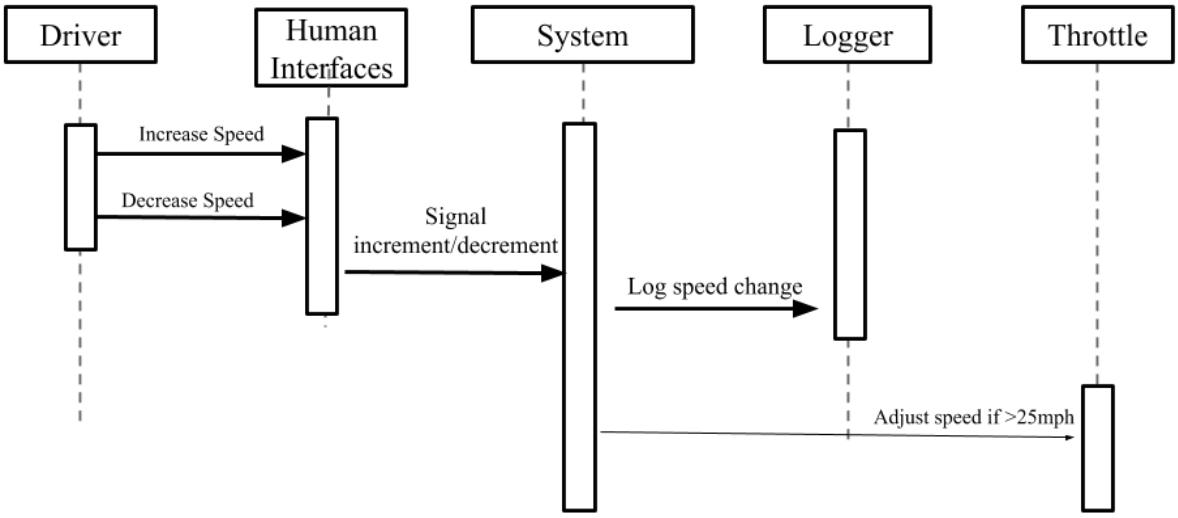
5.) Cruise control should respond the same for "-" except the set speed decreases by 1 mph per button push.

6.) Speed sensors continuously return current speed and adjusts accordingly.

Exception Use Case

1.) Actuator or speed sensor does not send a signal back, cruise control will remain at its current speed and immediately provide feedback that it could not increase/decrease the speed.

2.) If the driver attempts to decrease speed while the set speed is 25 mph, it will not decrease and will provide visual feedback that cruise control is already set to the minimum speed.

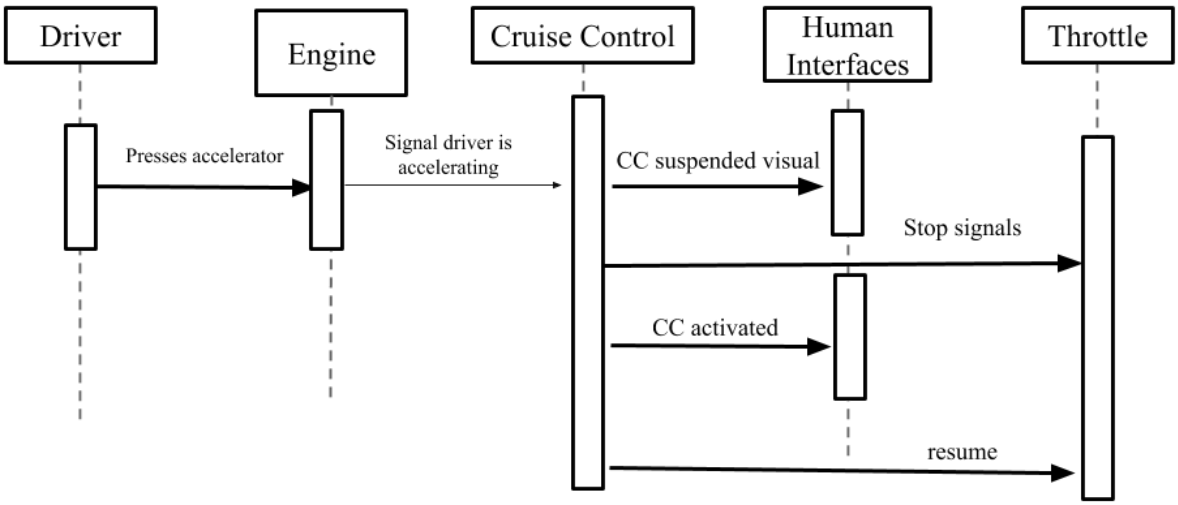


Use Case 3: User accelerates while cruise control is on
Actor: Driver
Goal: Cruise control will suspend itself and save the previous set speed
Preconditions: Cruise control is on and a speed has been set.
Trigger: User accelerates

1) User pushes the accelerator while cruise control is active
2) Cruise control stops sending signals to the throttle to allow the driver to accelerate freely
3) Cruise control provides visual feedback that it has been suspended
4) Cruise control remains on however, and the previous set speed is still logged.
5) If the accelerator is released, cruise control once again resumes at the previous logged speed and the throttle adjusts until the previous set speed is met.
6) If cruise control is turned off, or the brake is pressed at any time, refer to use case 4 and 5 respectively.

Exception Use Case

1.) The actuator does not respond and cruise control is turned off
2.) If speed can not be logged, it will provide visual feedback that the speed could not be logged and cruise control does not resume after the accelerator is released.



Use Case 4: User brakes while cruise control is set
Actor: Driver
Goal: Cruise control is temporarily suspended
Preconditions: Cruise control is on and a speed has been set
Trigger: Driver brakes

1.) The user brakes while cruise control is active

2.) Cruise control recognizes that the car is braking and sends a signal to the actuator.

3.) The system stops maintaining speed by letting go of the throttle, leaving the car to coast.

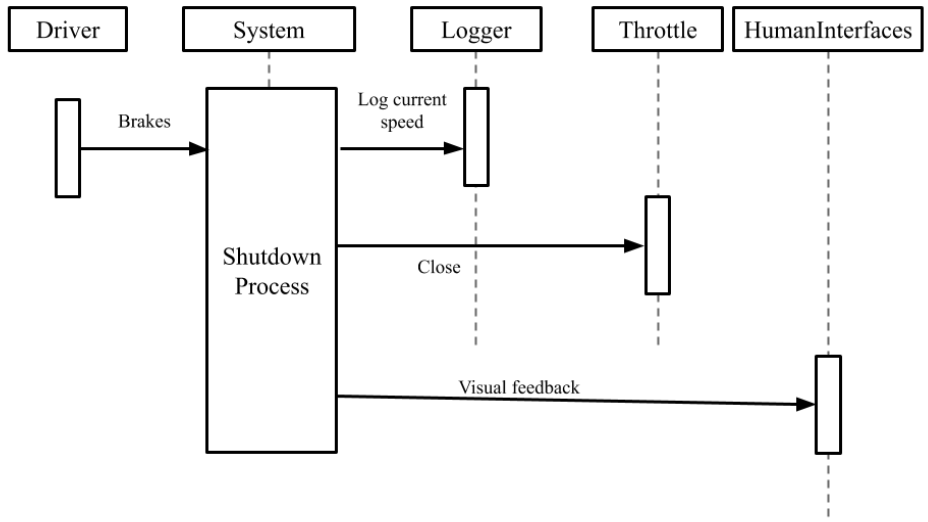
4.) The speed is logged in the memory.

5.) Cruise control sends visual feedback that it has been deactivated.

6.) If the driver selects “resume” cruise control turns back on at the previously set speed.

Exception Use Case

1.) None, if the driver brakes, cruise control should always deactivate itself



Use Case 5: User turns off cruise control
Actor: Driver
Goal: Cruise control turns off
Preconditions: Cruise control is on
Trigger: Driver turns off cruise control

1) The user turns off cruise control.

2) Cruise control logs data about the session before shutoff in the system memory.

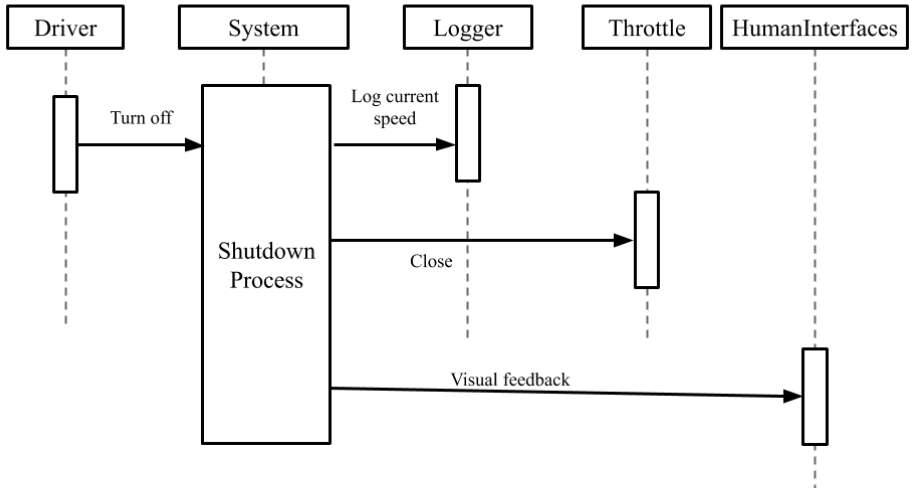
3) The system stops maintaining speed by letting go of the throttle, leaving the car to coast.

4) Cruise control provides visual feedback that it has been turned off.

5) The car returns to the ready state to activate cruise control on the activation command.

Exception Use Case

1.) None, cruise control should always turn off no matter what happens



Use Case 6: User turns off car
Actor: Driver
Goal: Turn off cruise control
Preconditions: Cruise control is on
Trigger: The driver pushing the "On/off" button

1) The user shuts off the engine of the car.

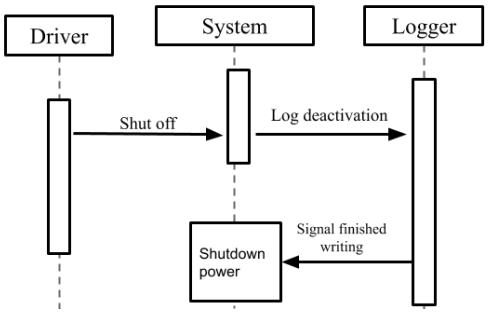
2) Cruise control receives power from the car battery temporarily.

3) Cruise control logs data about the session before the engine shuts off in the system memory.

4) Cruise control completely disconnects from power after 1 minute.

Exception Use Case

1.) None



Use Case 7: Technician Access
Actor: Admin
Goal: Allow admin access to the software
Precondition: Admin must enter a password
Trigger: Admin must physically connects to car

1.) Admin connects to cruise control via usb cable

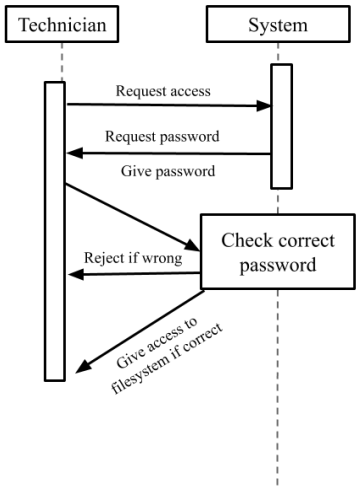
2.) Admin must enter a password to gain full access

3.) Admin can download and view data previously logged in the memory.

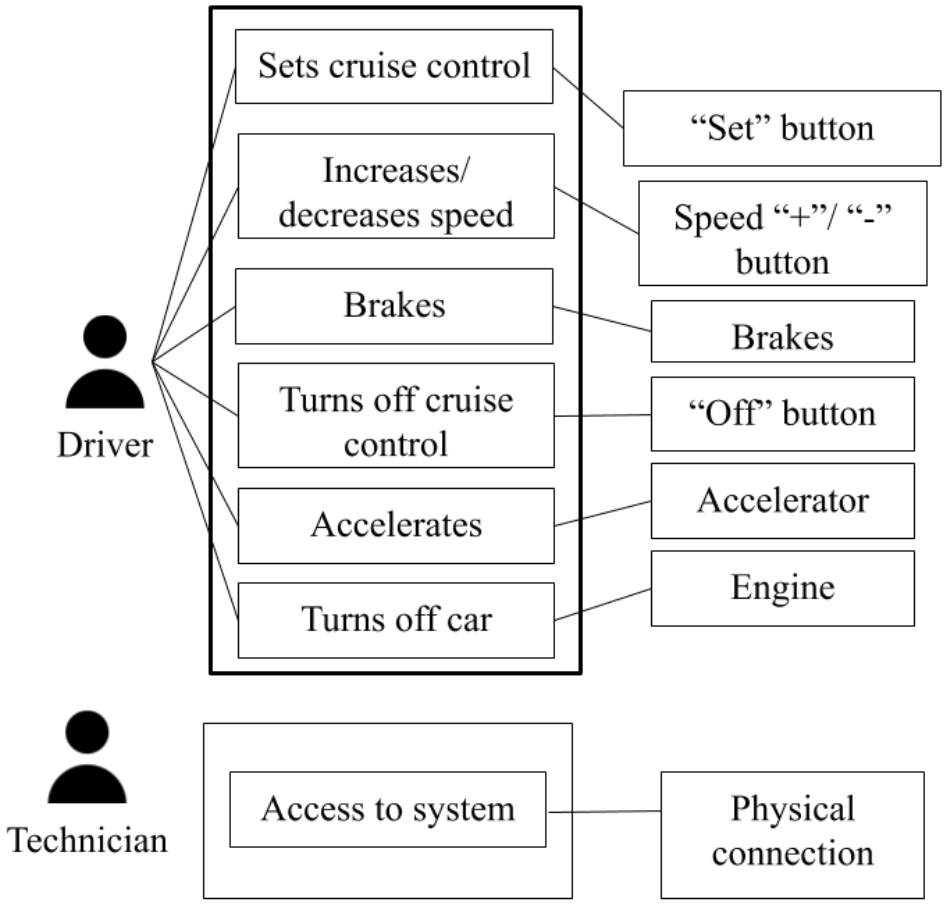
4.) Admin can update the software

Exception Use Case

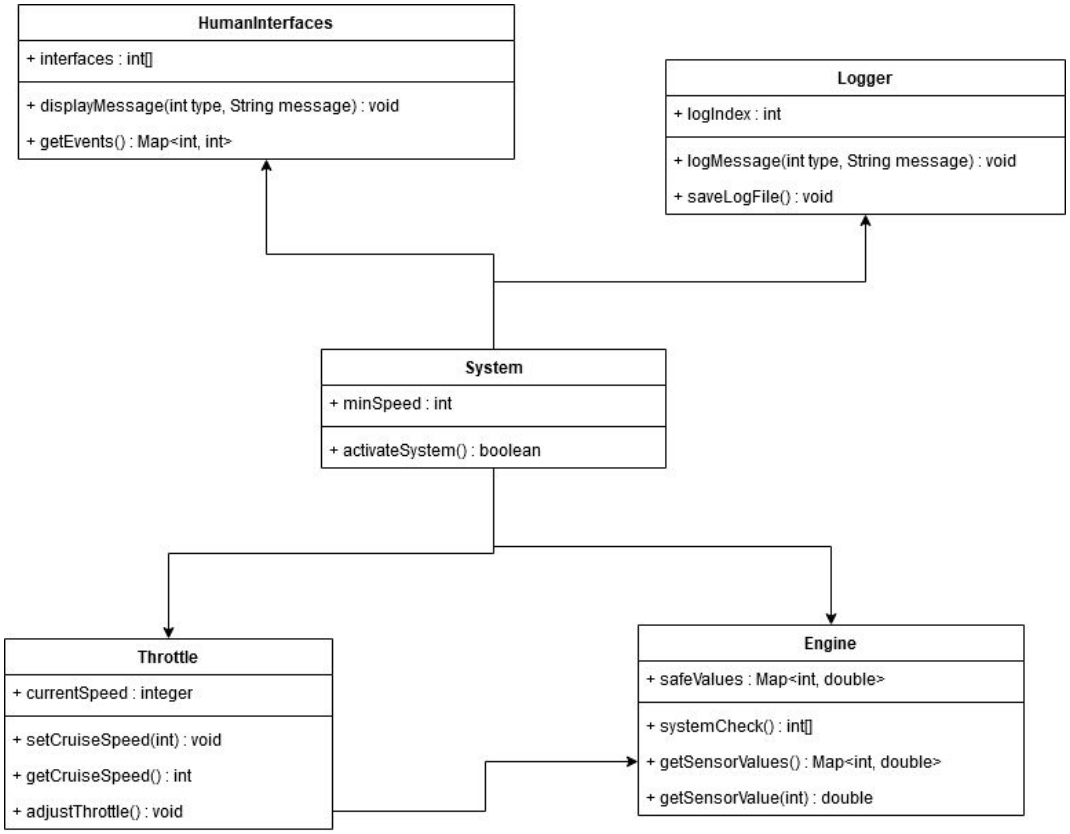
1.) 5 unsuccessful attempts to login will block the device connected and lock the system for 20 minutes.



Use Case Diagram



Class-Based UML + Index Cards



System	
Responsibilities	**Collaborators**
Gets results of system engine check and interprets them to allow activation of cruise control	Engine
Reads configuration file for minimum speed requirement to activate cruise control	
Activates/deactivates cruise control on user input from HumanInterface and when Engine	HumanInterface
Activate throttle control on interface input	HumanInterface, Throttle
Monitor engine events for dangerous events to halt cruise control	Engine, Throttle

Display reasons for activation and deactivation to driver's display	HumanInterface
Log all actions	Logger

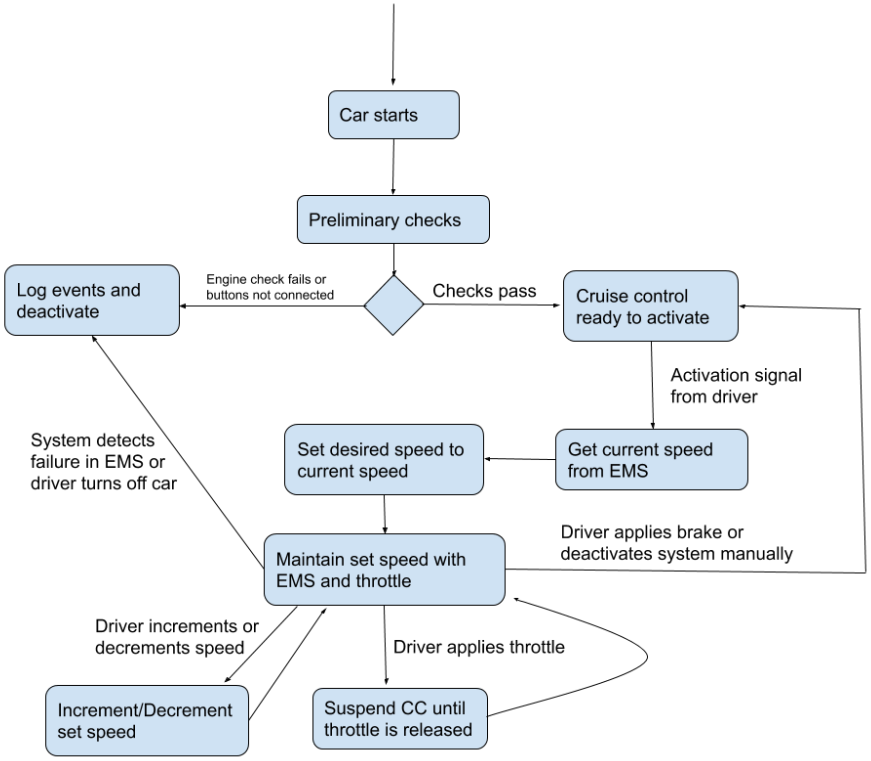
Logger	
Responsibilities	**Collaborators**
Logs output passed from other classes	All other classes
Writes output to system memory	
Timestamps log output with the system clock	

Engine	
Responsibilities	**Collaborators**
Load configuration files to determine safe values from sensors for driving.	
Interfaces with engine sensors to interpret safe and dangerous values.	
Raise an event to the System in case of failure.	System
Conduct overall system checks.	
Be able to get EMS sensor values when requested by other classes	System, Throttle

Throttle	
Responsibilities	**Collaborators**
Communicates with Engine sensors to modulate throttle and control speed.	Engine
Receives activation/deactivation signals from System.	System

HumanInterface	
Responsibilities	**Collaborators**
Monitor input from human interfaces (levers, buttons, throttle, brakes)	
Display messages to onboard car displays from System.	System
Send any inputs from driver to the System to handle.	System

Activity Diagram



State Diagram