Software Requirements Specification (SRS)

Project HFDS4

Authors:
Ethan Miller
Michael Dittman
Jason Harris
Jesse Barrett Stroster
Tianli Zhou

Customer: Andrew Davenport, Groundspeed Analytics Inc.

Instructor: Dr. Betty Cheng

1 Introduction

This document will lay out the requirements for a Hands-Free Driving System (HFDS). Each section will cover some of the purposes, requirements, descriptions, and diagrams within each system that is used throughout the process. The systems will be detailed in their relation to the overarching functionality of HFDS and illustrate the significance of each system and its ability to function together properly. The focus of the systems are specified and the importance of each system is highlighted in the models and diagrams. Each model and use case diagram will elaborate on the many potential situations given by the requirements list along with how each scenario will be handled. A prototype is also included to help visualize the system and its many functions, creating scenarios for the user to see how the system operates and how the many complicated systems come together.

1.1 Purpose

The purpose of this document is to describe the intricacies of the HFDS. Driver fatigue is often colloquially cited as one of the major causes of automobile accidents, but it is hard to empirically study the magnitude of its impact on accident causation due to survivorship bias [1]. While driver fatigue is often thought of as the user being drowsy on the road, the term actually encompasses various states of fatigue including what is known as "Highway Hypnosis". Highway Hypnosis is when the environment the user is operating in is monotonous and the layout of the road is relatively straight, leading the user to enter a state where they have a higher response latency and decreased memory formation [2]. Any attempt to alleviate this phenomenon by automating driving in such an environment can potentially save many lives, which is why the development of a (HFDS) is crucial to ensure highway safety. This document goes into detail about the systems used, the features needed and the requirements to develop such a system. This documentation is created for the purpose of ensuring requirements are agreed upon by all related parties and is intended for stakeholders and developers of said system.

1.2 Scope

HFDS is an embedded system within a vehicle that allows the user to travel along a highway without having to use the steering wheel or the pedals. One of the many goals of HFDS is the ability for the user to conveniently take their hands off the steering wheel and pedals, while still paying attention, and have the vehicle safely drive. Since the user still has to pay attention, the system will ensure that the user is not distracted while driving and can avoid the many hazards and dangers of driving, including Highway Hypnosis. The HFDS uses many subsystems to ensure not only the safety of the user, but also safe road conditions for operation. Said subsystems include sensors like GPS, LiDAR, and RADAR to establish a safe path for the vehicle to follow along. When HFDS is enabled and the user is driving over 20 miles per hour, eye and head movements are both monitored to confirm the user's attentiveness even when their hands and feet aren't controlling the vehicle. Additionally, the car monitors road conditions, construction zones, position of lanes and other cars, and calculates if HFDS can be enabled. If HFDS cannot be enabled or must be turned off, multiple warnings will ensue to attempt to get the user to regain control of the vehicle. If the multiple warnings are ignored and the user is found to be not attentive, the car will turn the vehicle's hazard lights on and stop the vehicle.

1.3 Definitions, acronyms, and abbreviations

- **Hands-Free Driving System (HFDS)** - System to control a vehicle without the need of the user touching the steering wheel
- **Global Positioning System (GPS)** - An accurate worldwide navigational and surveying facility based on the reception signals from an array of orbiting satellites
- **Light Detection and Ranging (LiDAR)** - A detection system which works on the principle of RADAR, but uses the light from a laser
- **Radio Detection and Ranging (RADAR)** - A system for detecting the presence of object by sending out pulses of high-frequency electromagnetic waves that are reflected off of the object back to the source
- **User Interface (UI)** - Input devices and software in which the user and computer subsystems interact
- **Human Machine Interface (HMI)** - The hardware or software through which an operator interacts with a controller.
- **Path Prediction System** - A specific path that a vehicle will travel along to not crash or switch lanes
- **Software Requirements Specification (SRS)** - A comprehensive description of the intended purpose and environment for software under development
- **Highway Hypnosis -** An altered mental state in which a person can drive a vehicle for long distances in a correct manner and have no recollection of consciously doing so
- **Cruise Control** - Electronic device in a motor vehicle that can be switched on to maintain a selected constant speed without the use of the accelerator
- **Driver Assist subsystem -** Technologies used to make motor vehicle travel safer by automating, improving, or adapting some or all of the tasks involved in operating vehicle
- **Driver Attention system -** System used to gauge the attentiveness of the user operating the vehicle
- **Vehicle Control system -** System used to control the speed and sometimes heading of the vehicle in accordance with signals received from a guidance system
- **Vehicle Position system -** System used to determine the geographical position of the vehicle using subsystems such as GPS and RADAR.
- **Red Green Blue Light (RGB Light) -** Lights responsible for flashing when a warning is initiated to the user
- **Portable Network Graphics (PNG) -** File type for the storage of video data in the system

1.4 Organization

The remainder of the Software Requirements Specification (SRS) highlights some of the key elements that make up the HFDS. Section 2 features an overall description of HFDS and will contain information about the functionality, constraints and assumptions. Section 3 focuses more on the requirements followed by section 4 which will dive deeper into the modeling and developmental aspects of the system. Section 5 will outline how to access the prototype, how to use it, and how it functions. Section 6 will contain citations to works referenced throughout the SRS and section 7 will provide the points of contact for developers and customers.

2 Overall Description

This section is an overall description of the HFDS. It will cover such as product perspective, functionality, constraints, user characteristics, any assumptions and dependencies, and also an apportioning of requirements at the end. Diagrams will be included to help represent the constraints and functionality of the HFDS.

2.1 Product Perspective

The product perspective identifies the context in which the product can be used. It will give an outlook of how the elements come together to create a larger system, and highlight constraints that the system has. Long road trips can be mentally taxing on a user and while on the highway it can be tedious work as you cruise in one lane for hours to constantly keep up with the flow of traffic, tilt the steering wheel to follow the wind of the road, and tap the brakes on occasion when you run up too close behind the user in front of you. With the HFDS it can allow a user to relax a little more as the system will take control of the vehicle. Unlike normal cruise control where the vehicle will only keep a consistent speed, the HFDS will not only control the acceleration of the vehicle, but also will be able to steer and brake as the user can be completely hands-off and an active passenger in their own car ready to take back the wheel whenever it is necessary. Figure 1 below represents the Data Flow Diagram of how the data flows from sensor to subsystem, subsystem to subsystem, and all of the data processed together to send a response back to the user. It gives an idea of how the elements of the HFDS work with each other by sharing information.

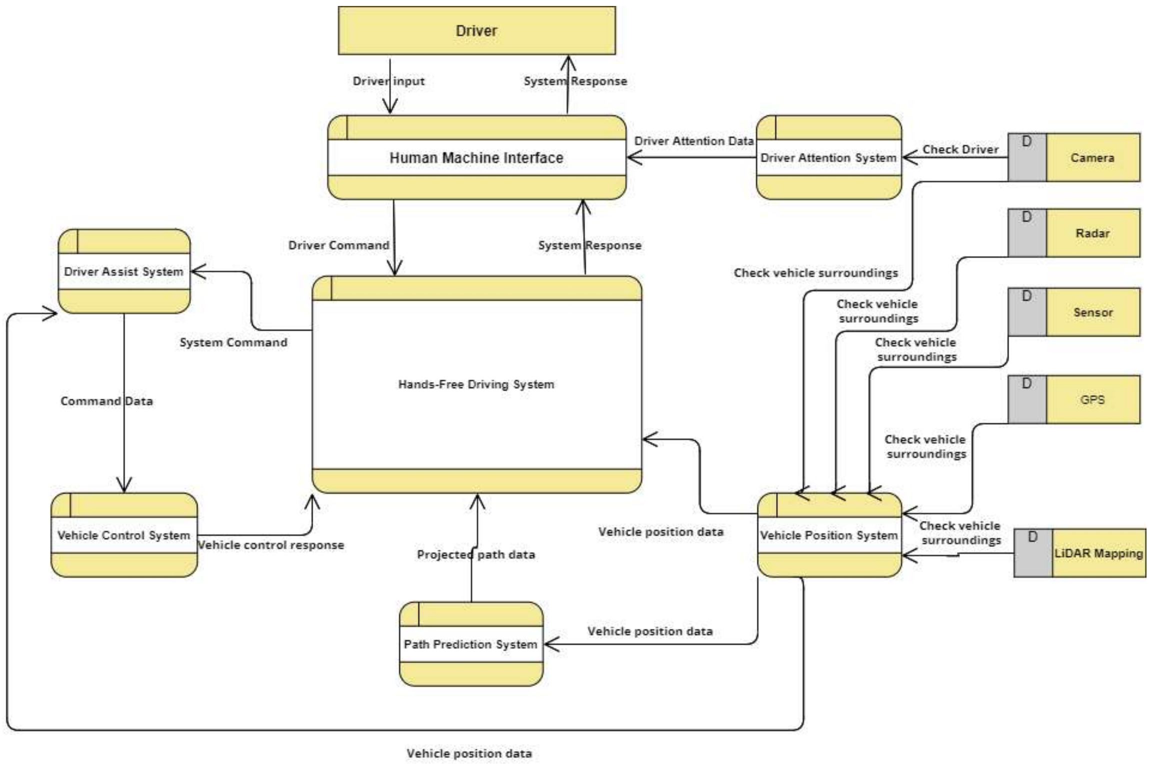

Figure 1. This Data Flow Diagram represents how information flows through each subsystem and sensor of the HFDS. The subsystems are represented in small yellow/white rectangles while the sensors are in small yellow/gray rectangles. The diagram shows once activated, how the sensors send sensor data to the Driver Attention and Vehicle Position Systems and the communication between all of the subsystems. All of the data sent from the subsystems is processed and allows the HFDS to issue the correct response back to the user of what action the vehicle will perform.

In regards to software constraints for the HFDS, the system will be created off the adaptive cruise control that already exists in the vehicle. For the system to be activated adaptive cruise control must be enabled and the environment must be deemed safe by HFDS standards. The user must set speed for adaptive cruise control, the HMI will take that user input and allow for adjustments by user whenever is necessary. Hardware constraints will include cameras inside and outside the vehicle. There will be a camera on the steering wheel to track the user's eye and head movement to ensure the user is actively paying attention to the road. Outside cameras will be used along with vehicle sensors, RADAR, and GPS to validate the vehicle's position in the real-world and the surroundings of the vehicle. With the position established the vehicle will use LiDAR mapping and information from the Vehicle Position system to continue driving in its current lane while being able to steer, brake, and accelerate automatically.

2.2 Product Functions

The product functions are the major functions that the HFDS will accomplish to achieve the correct functionality of the system as a whole. The HFDS gives the ability to steer, accelerate, and brake on a highway without the assistance of a user. In order to accomplish this, there must be goals reached at each level of the system to ensure functionality. The environment must be checked for safety, the user's head and eyes must be monitored for attention warnings, the vehicle position must be calculated, all these components are goals that must be reached to allow the system to function. Each of the major functions are listed below as well as a high-level goal diagram in Figure 2 that depicts how each major goal of the system corresponds with the system as a whole.

- "The HFDS system allows a vehicle to automatically steer, accelerate and brake in certain highway conditions." (Davenport)
- "The Driver Assist System validates road conditions, current trajectory, sensor input, and predicted path. If the current trajectory is deemed safe, the user can opt to enter hands-free mode." (Davenport)
- " The vehicle will enter an adaptive cruise control state and stay within its existing lane for the duration of the session." (Davenport)
- Camera monitoring will ensure that the user is actively engaged on the road at all times
- Warnings will be sent to the user if the user is not properly engaged with the road. "If the system identifies the user as inactive, the system aborts hands-free mode and if needed, the vehicle will come to a stop." (Davenport)

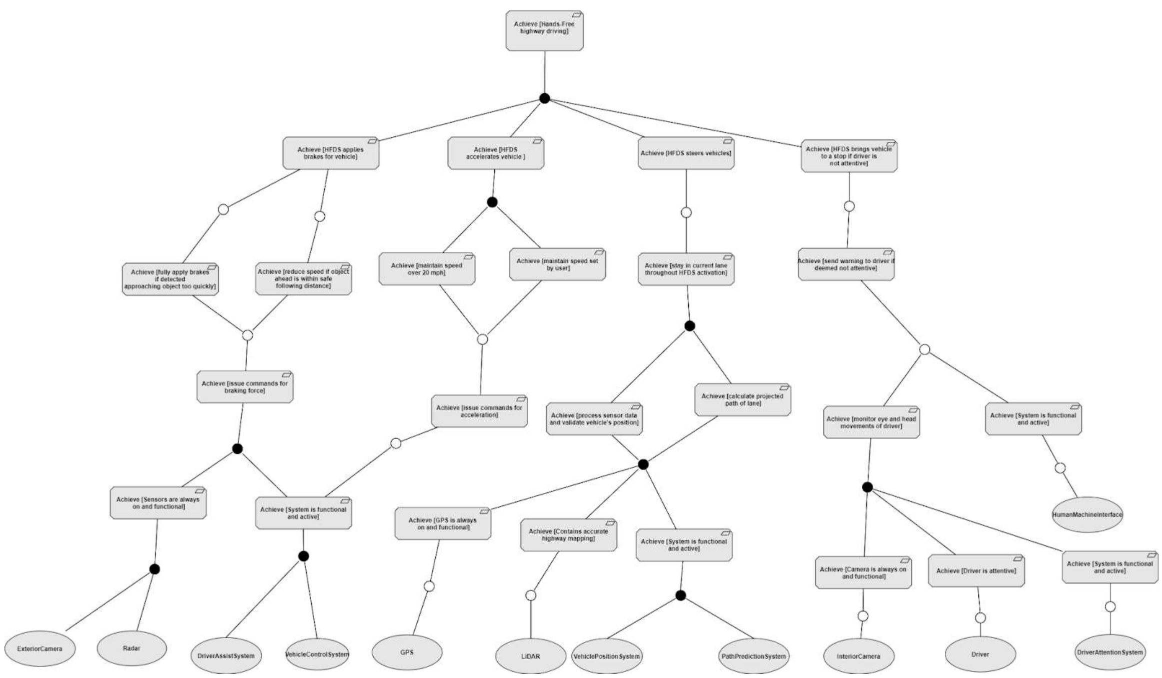

Figure 2. This is a High-Level Goal Diagram for the HFDS. It represents how each goal leads to the functionality of Hands-Free highway driving.

2.3 User Characteristics

User characteristics are the characteristics we are expecting users of the HFDS will contain. It is important to identify who the system is designed to be used by to ensure that there is quality user interaction to produce the desired results from the system. Users are expected to own a driver's license and be in a physical condition that allows them to operate a vehicle safely without any assistance. Before use, users are expected to have read the manual that contains the features, uses, safety precautions, and operations for the system. The average user should have a comfortability with highway driving though the system works with any level of expertise on the road making the HFDS easy to use for all users. Users are expected to be actively engaged with the road while the system is active in order to gain control at any time deemed necessary by the system or user.

2.4 Constraints

The constraints that the HFDS abides by are listed here. The constraints for the HFDS system will ensure the safety of the user and other users on the road. The HFDS cannot be activated unless the environment is deemed safe by the Driver Assist System and Adaptive Cruise Control is active. An environment is deemed safe if the vehicle can travel over 20mph, weather conditions are not obscuring the vision of the cameras or sensors, and road conditions are not causing wheel slippage that could cause the vehicle to slide when performing a maneuver. The user must always have the ability to regain control of the vehicle by controlling the steering wheel, braking, or throttle. If the system detects any single point failure the system must relinquish control back to the user. A single point of failure can be any of the following examples; sensors can not accurately collect data, lidar mapping is not aligning with the projected path ahead, or the lane the vehicle is in can not be detected. These are only a handful of examples, they are not the only possible single points of failure. If a feature is not operating as it is intended to, then that is a single point of failure. Relinquishing control back to the user is a safety precaution to give the user the ability to act if there is an unforeseen obstacle that the system encounters. "Both hardware and sensor redundancies must be in place to ensure safe operation and provide time for the user to become reengaged with the vehicle if a problem occurs." (Davenport). One of the previously stated single point of failure examples where the system will relinquish control back to the user, "lidar mapping is not aligning with the projected path ahead", accounts for scenarios such as when a vehicle drives into a construction zone where speed limits and lane closures can change on the fly. Security constraints will be in place to protect the system from software hacks while being able to accept over-the-air updates.

2.5 Assumptions and Dependencies

Here is what users can assume to always be true about the HFDS. It is assumed that Lane detection and adaptive cruise control are pre-existing features within the vehicle. The HMI system will be able to work with adaptive cruise control and accept user input from the user such as set speed and following distance. It can also be assumed that the vehicle is driving on the highway when activating the system. Another software assumption is LiDAR mapping of highways has been captured meaning paths are already in place for the vehicle to follow. The system should be able to work in moderate rain levels, any rain levels that obscure any sensors or cameras will be declared a fault and control will be relinquished back to the user. In traffic situations the system will be able to lower the vehicle speed to match the flow of traffic, but will not raise the speed to exceed the speed limit. Users are expected to be actively engaged with the road at all times and ready to take back control of the vehicle at any point while the HFDS is active. The system is assumed to be able to run without the use of the internet. The Internet will only be a component when dealing with over-the-air updates.

2.6 Approportioning of Requirements

In the future, the HFDS can include more features to ensure safety to users and to be able to adapt to new situations. The ability to detect emergency vehicles and safety constraints to handle over-the-air updates were discussed as points of interest to improve the current system's functionality.

3 Specific Requirements

This section contains a list of requirements for the HFDS. The requirements are split into four categories: Global Invariant Requirements, Hardware Requirements, System Requirements and CyberSecurity Requirements. In each of these subsections, requirements are listed by order of necessity for the safety of the user.

3.1 Global Invariant Requirements

The Global Invariant Requirements section provides an enumerated list of the invariants in the HFDS system. This list is organized by the importance of the requirement for the safety of the user. Global Invariants are requirements that must always be true in the system no matter what. Any violation of an invariant leads to a system failure.

3.1.1 The system must allow the user to consistently regain control of the vehicle in multiple scenarios via controlling the steering wheel, braking, or throttle regardless of the state of the software.

3.1.2 If there is any fault whatsoever, for example: sensor turns off, sensor is damaged, any subsystem is not active, and user is inattentive after 3 warnings, car should turn on hazard lights and come to a complete stop.

3.1.3 If the user is deemed inactive after 3 warnings, the system must abort hands-free mode, and if needed come to a complete stop with the hazard lights on.

3.1.4 Alert user and relinquish control of the car in cases when an error or fault cannot be corrected by the equipment.

3.1.5 Hardware and sensor redundancies must be in place to provide time for the user to become reengaged with the vehicle if a problem occurs.

3.1.6 The Driver Attention System must always be monitoring the user’s head movement and eye movements to ensure active engagement. Cameras must be able to monitor eye movements in all lighting conditions.

3.1.7 The system will initiate 3 warnings to the user. If the user is inattentive for 5 seconds, the next warning will initiate. The first two warnings consist of visual and audio queues. The HMI will vibrate the seat of the user for the 3rd warning.

3.1.8 The road which the user activates the HFDS must always have proper LiDAR mapping with the Path Prediction System (a pre-map of the highway).

3.1.9 The Vehicle Position System must always have the correct vehicle position, RADAR data, and camera data to detect, predict, and react to objects and cars in vicinity of the vehicle.

3.1.10 The HFDS must not be able to engage in significant levels of rain, fog, snow, and ice.

3.1.11 The adaptive cruise control must be active for HFDS to engage.

3.1.12 If the vehicle's sensors detect any deviation from the LiDAR map, the HFDS must give up control of the vehicle to the user.

3.2 Hardware Requirements

The following section outlines the hardware that is required for the system to operate. Hardware required for the HFDS is partially available on most vehicles. Other sensors not included with a vehicle can be sourced from various manufacturers. Equipment compatibility can be insured via the equipment manufacturer.

3.2.1 Brakes, accelerometer, steering wheel must be present in the system to allow the user to interact with the HFDS.

3.2.2 RADAR, LiDAR, GPS, and cameras must all be present to calculate the vehicle's blue path and validate vehicle position.

3.2.3 HMI display screen to turn on and off HFDS system, display warnings to the user, and display the vehicle's current trajectory.

3.2.4 Infrared camera to detect when a user may be distracted regardless of light levels or user apparel.

3.2.5 Separate backup sensors to account for any redundancy when a primary sensor fails.

3.2.6 Internet connection hardware to ensure the system receives over the air updates.

3.3 System Requirements

The System Requirements section gives a concise language description of how each subsystem performs and interacts with other subsystems. The HMI is considered as part of hardware in section 3.2.

3.3.1 The HFDS must use data from the Path Prediction System and precision LiDAR mappings to determine the course for the vehicle to steer, brake, and accelerate automatically.

3.3.2 The Vehicle Control System activates actuators (brakes, steering wheel, accelerator) to control the vehicle.

3.3.3 The Vehicle Position System records data from sensors to send to the Path Prediction System.

3.3.4 The Path Prediction System calculates the vehicle's blue path based on data from Vehicle Position System and LiDAR mappings.

3.3.5 The Driver Assist System sends commands to the Vehicle Control System to activate actuators.

3.3.6 The Driver Assist System uses sensor information to determine if road conditions are safe by determining following distance.

3.3.7 The Driver Attention System monitors the user to ensure attentiveness based on head angle and eye engagement.

3.4 CyberSecurity Requirements

The following section considers cybersecurity threats and the requirements to ensure that the system is operating safely.

3.4.1 The system should be able to receive over the air updates published only by the manufacturer or vendor.

3.4.2 Any and all connections to a network must remain encrypted.

3.4.3 The system must be regularly tested to prevent most DOS or other compromising attacks.

3.4.4 The system should be able to detect when it has been compromised and give control to the user.

4 Modeling Requirements

This section depicts several types of diagrams which model the requirements of the HFDS that are specified in Section 3. Illustrated in the following subsections are the key components, functionality and scenarios associated with the HFDS.

4.1 Use Case Diagram

This subsection outlines the use case diagram for the HFDS. Use case diagrams are used to depict the behavior and operation of a system. They describe what happens when the system is running. Use case diagrams display one feature of a system or one set of interactions between the system and the user. They can depict boundaries, actors, functionality and outcomes of actions. Figure 3 displays the Use Case Diagram for the HFDS. It includes use cases for turning on HFDS, monitoring the user attentiveness, sending commands to control actuators, and validating if HFDS can be enabled.

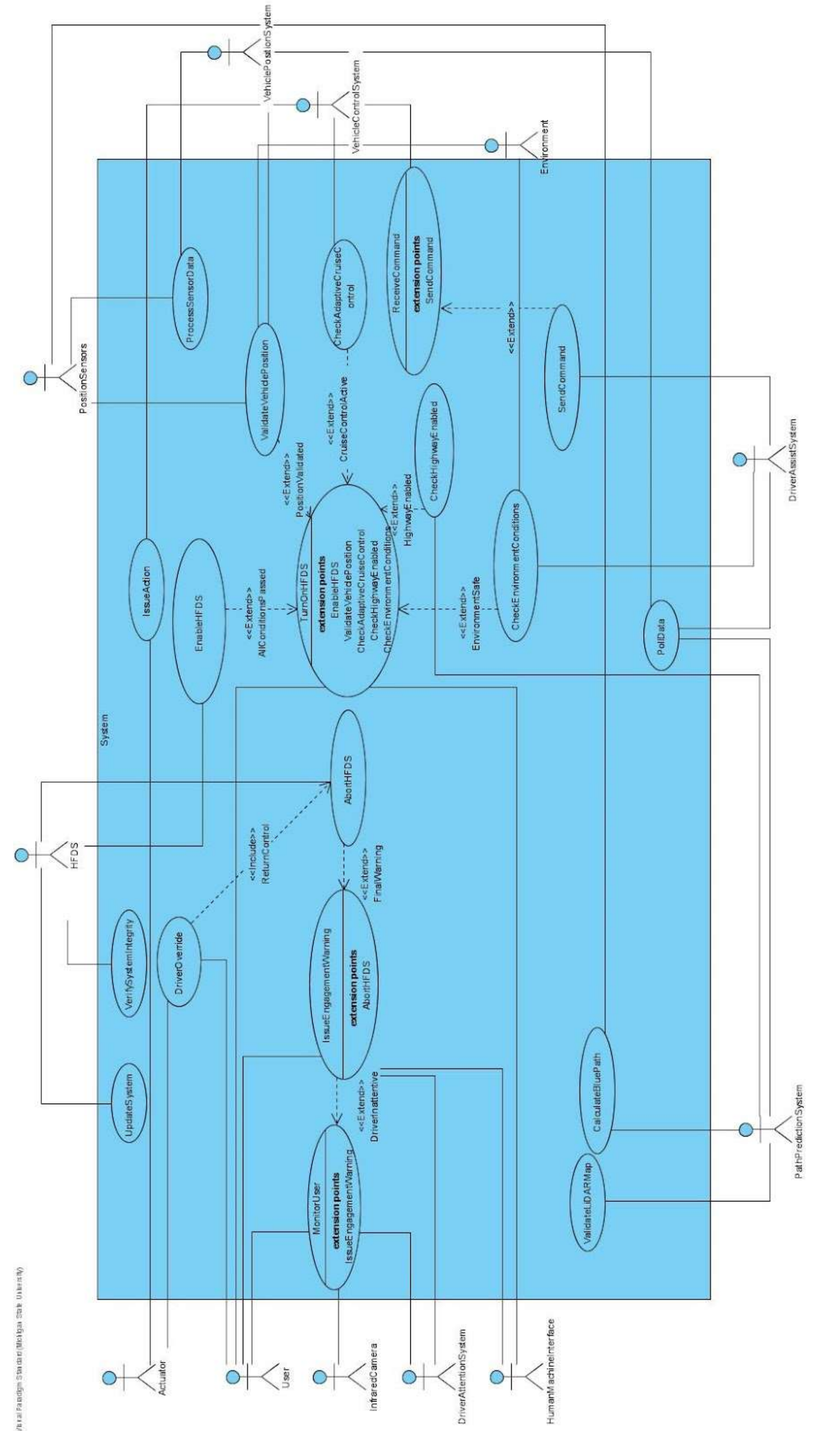

Figure 3. Use Case Diagram for HFDS

4.1.2 Use Case Descriptions

The following tables detail each of the use cases found in Figure 3. In each provided table the following are provided: a brief description of the use case, the type of use case (Primary or Secondary), inclusions and extensions, and the requirements the use case satisfies.

Use Case:	AbortHFDS
Actors:	HFDS
Description:	Use case responsible for aborting the HFDS and returning control to the user.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.1.1, 3.1.2, 3.1.3, 3.1.4, 3.1.5, 3.1.12
Use cases:	DriverOverride

Table 1. Use Case Description for *AbortHFDS*

Use Case:	CalculateBluePath
Actors:	PathPredictionSystem, PositionSensors
Description:	Use case responsible for calculating the vehicles projected path (blue path) given data from the vehicles position sensors.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.2.2, 3.2.5, 3.3.4
Use cases:	None

Table 2. Use Case Description for *CalculateBluePath*

Use Case:	CheckAdaptiveCruiseControl
Actors:	VehicleControlSystem
Description:	Use case to ensure that the adaptive cruise control is enabled first before HFDS can enable.
Type:	Primary
Includes:	None
Extends:	TurnOnHFDS
Cross-refs:	Requirements 3.1.11
Use cases:	None

Table 3. Use Case Description for *CheckAdaptiveCruiseControl*

Use Case:	CheckEnvironmentConditions
Actors:	DriverAssistSystem, Environment
Description:	Use case that ensures the HFDS is not enabled in significant levels of rain, fog, snow, or ice.
Type:	Primary
Includes:	None
Extends:	TurnOnHFDS
Cross-refs:	Requirements 3.1.10
Use cases:	None

Table 4. Use Case Description for *CheckEnvironmentConditions*

Use Case:	CheckHighwayEnabled
Actors:	PathPredictionSystem
Description:	Use case that ensures the highway is mapped and enabled by the Path Prediction System before HFDS can be turned on.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.1.8
Use cases:	None

Table 5. Use Case Description for *CheckHighwayEnabled*

Use Case:	DriverOverride
Actors:	Actuator, User
Description:	Use case that depicts the ability for the user to use an actuator (brake, steering wheel, accelerator) to override the HFDS and regain control of the vehicle.
Type:	Primary
Includes:	AbortHFDS
Extends:	None
Cross-refs:	Requirements 3.1.1, 3.2.1
Use cases:	None

Table 6. Use Case Description for *DriverOverride*

Use Case:	EnableHFDS
Actors:	HFDS
Description:	Use case that activates upon once the user turns on HFDS and the following conditions are met: valid vehicle position, adaptive cruise control enabled, highway enabled by Path Prediction System and environment determined safe.
Type:	Primary
Includes:	None
Extends:	TurnOnHFDS
Cross-refs:	Requirements 3.1.8, 3.1.9, 3.3.1
Use cases:	None

Table 7. Use Case Description for *EnableHFDS*

Use Case:	IssueActuatorAction
Actors:	Actuator, VehicleControlSystem
Description:	Use case that allows the Vehicle Control System to control the actuators of the car (brake, steering wheel, accelerometer).
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.3.1, 3.3.2
Use cases:	None

Table 8. Use Case Description for *IssueActuatorAction*

Use Case:	IssueEngagementWarning
Actors:	DriverAttentionSystem, HumanMachineInterface, User
Description:	Use case that depicts the HMI issuing an attentiveness alert to the user after the infrared camera and Driver Attention System found the user to be inattentive.
Type:	Primary
Includes:	None
Extends:	MonitorUser
Cross-refs:	Requirements 3.1.2, 3.1.3, 3.1.6, 3.1.7, 3.2.3, 3.2.4, 3.3.7
Use cases:	AbortHFDS

Table 9. Use Case Description for *IssueEngagementWarning*

Use Case:	MonitorUser
Actors:	DriverAttentionSystem, InfraredCamera, User
Description:	Use case that depicts the Driver Attention System and the infrared camera monitoring the attentiveness of the user.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.1.3, 3.1.6, 3.1.7, 3.2.4, 3.3.7
Use cases:	IssueEngagementWarning

Table 10. Use Case Description for *MonitorUser*

Use Case:	PollData
Actors:	DriverAssistSystem, PathPredictionSystem, VehiclePositionSystem
Description:	Use case that displays the Driver Assist System polling sensor data from the Vehicle Position System and Path Prediction System.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.1.9, 3.1.12, 3.2.2, 3.3.3, 3.3.6
Use cases:	None

Table 11. Use Case Description for *PollData*

Use Case:	ProcessSensorData
Actors:	PositionSensors, VehiclePositionSystem
Description:	Use case that depicts the Vehicle Position System communicating with the position sensors to acquire camera, LiDAR and RADAR, and GPS data.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.1.9, 3.2.2, 3.3.3
Use cases:	None

Table 12. Use Case Description for *ProcessSensorData*

Use Case:	ReceiveCommand
Actors:	VehicleControlSystem
Description:	Use case that shows the Vehicle Control System receiving a command from the Driver Assist System in order to issue actions to actuators.
Type:	Primary
Includes:	None
Extends:	SendCommand()
Cross-refs:	Requirements 3.3.2, 3.3.5
Use cases:	IssueActuatorAction

Table 13. Use Case Description for *ReceiveCommand*

Use Case:	SendCommand
Actors:	Driver Assist System
Description:	Use case that illustrates the Driver Assist System sending a command to the Vehicle Control System.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.3.2, 3.3.5
Use cases:	ReceiveCommand

Table 14. Use Case Description for *SendCommand*

Use Case:	TurnOnHFDS
Actors:	User, HumanMachineInterface
Description:	Use case that illustrates the HMI displaying an option to turn HFDS on.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.1.8, 3.1.9, 3.1.10, 3.1.11, 3.2.3
Use cases:	EnableHFDS, ValidateVehiclePosition, CheckAdaptiveCruiseControl, CheckHighwayEnabled, CheckEnvironmentConditions

Table 15. Use Case Description for *TurnHFDSOn*

Use Case:	UpdateSystem
Actors:	HFDS
Description:	Use case that outlines the ability of the HFDS to receive secure, wireless, updates.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.4.1, 3.4.2
Use cases:	None

Table 16. Use Case Description for *UpdateSystem*

Use Case:	ValidateLiDARMap
Actors:	PathPredictionSystem
Description:	Use case that depicts the Path Prediction System validating the LiDAR map with additional sensor data.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.1.8
Use cases:	None

Table 17. Use Case Description for *ValidateLiDARMap*

Use Case:	ValidateVehiclePosition
Actors:	Environment, PositionSensors, VehiclePositionSystem,
Description:	Use case that displays the Vehicle Position System validating the position of the vehicle with respect to all of the position sensor data.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.1.9
Use cases:	TurnOnHFDS

Table 18. Use Case Description for *ValidateVehiclePosition*

Use Case:	VerifySystemIntegrity
Actors:	HFDS
Description:	Use case that illustrates the system testing itself to prevent DOS attacks and whether it has been compromised to hand control to the user.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirements 3.4.3, 3.4.4
Use cases:	None

Table 19. Use Case Description for *VerifySystemIntegrity*

4.2 Domain Model and Data Dictionary

This subsection outlines the domain model and data dictionary for the HFDS. The domain model is used to focus on the structure of a system as opposed to outcomes of actions. Domain models are used to highlight the key elements of a system.

4.2.1 Domain Model

The domain model in Figure 4 outlines the structure of the HFDS. Each of the blue boxes in Figure 4 represents a class in the system. Each class in the diagram is an object that interacts and has a role to play in the system. Relationships between classes are drawn as a line between them. The relationships indicate that the two objects interact with each other in the system. The number of objects that exist in a given association is indicated by the number written near the endpoints of the relationship line. For instance there can only be one of every system and subsystem, but there can be any number of warning outputs and actuators which is indicated by the asterisk. The relationships between objects can be further specified by aggregation and composition. Aggregation shows that an object is part of another object. This is shown as an open diamond at the endpoint of the object that has other objects as part of itself. As an example, four subsystems are part of the HFDS. Composition shows that an object is owned by another object. This is shown as a shaded diamond terminated at the owning class. Composition is not featured in this domain model. Further, classes can exhibit generalization which depicts the relationship between two objects when one object, the child, can be said to be the same type of object as the parent object. Classes can have any number of attributes and operations. Attributes are named values of classes and operations are services that can be requested of the class. The HFDS is composed of six main subsystems: the Human Machine Interface, the Driver Attention System, the Driver Assist System, the Vehicle Position Subsystem, the Path Prediction System, and the Vehicle Control System as shown in Figure 4. The HFDS itself only has relations to four of the subsystems. The diagram further features classes for actuators, position sensors, and warning outputs.

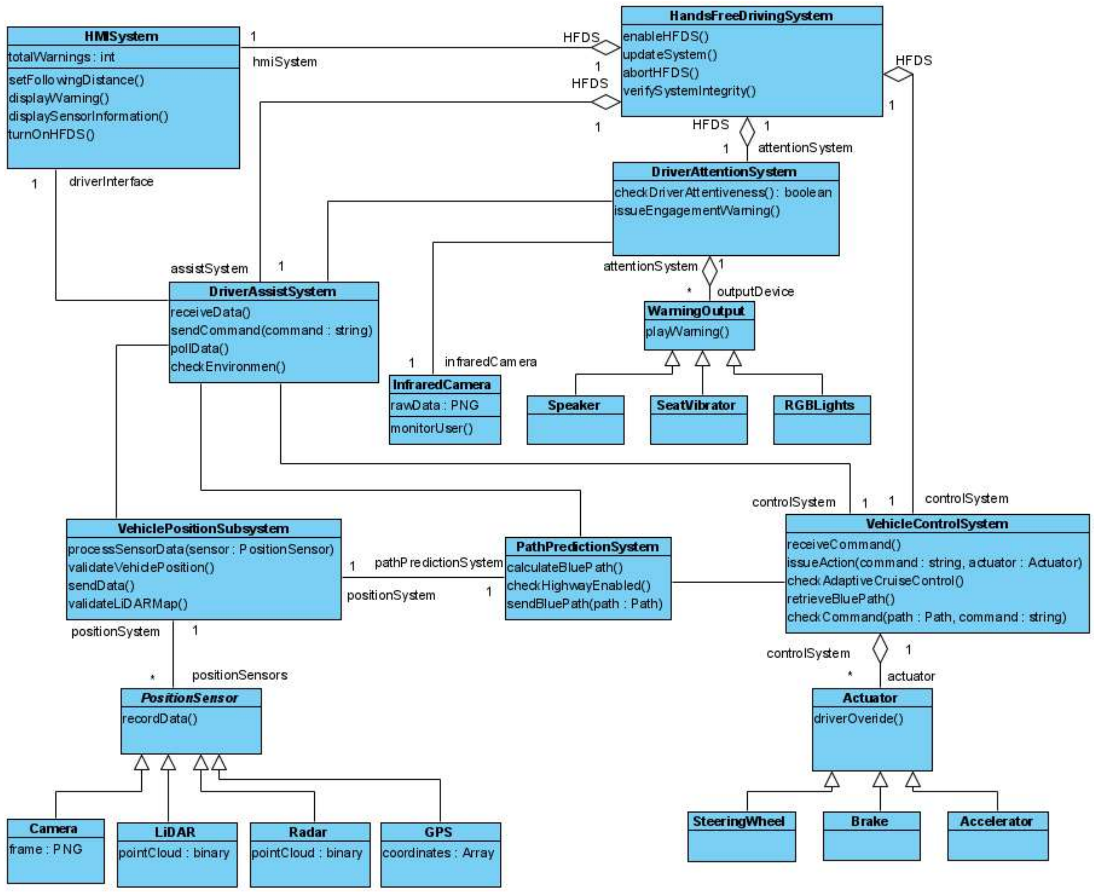

Figure 4. High-Level Class diagram (Domain Model) for HFDS

4.2.2 Data Dictionary

The following tables elaborate each of the classes that exist in Figure 4. Classes will be elaborated to show their relationships, operations, and attributes. The dictionaries will depict: aggregation, composition, attributes, operations, and generalization. Composition is not featured in this domain model. Operations and attributes will be displayed with their return types and data types respectively if applicable. A brief description of each class is given for clarity.

Element Name		**Description**
Accelerator		An actuator that is physically responsible for the acceleration of the vehicle.
Attributes	None	
Operations	None	
Relationships	Associations : None Aggregations : None Composition : None Generalization: Is a type of actuator	
UML Extensions	None	

Table 20. Data dictionary entry for *Accelerator*

Element Name		Description
Actuator		A device that the user can interact with to change the state of the Hands-Free Driving System.
Attributes	None	
Operations	driverOveride()	All actuators can be interacted with to disengage the Hand-Free Driving System.
Relationships	Associations : None Aggregations: Is part of the VehicleControlSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 21. Data dictionary entry for *Actuator*

Element Name		**Description**
Brake		An actuator that is physically responsible for the deceleration of the vehicle.
Attributes	None	
Operations	None	
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of actuator	
UML Extensions	None	

Table 22. Data dictionary entry for *Brake*

Element Name		**Description**
Camera		Position sensor responsible for recording visual data about the surrounding environment.
Attributes	frame : PNG	A single frame of visual data stored as a png file.
Operations	None	
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of PositionSensor	
UML Extensions	None	

Table 23. Data dictionary entry for *Camera*

Element Name		**Description**
DriverAssistSystem		System that polls necessary information to determine safe conditions. Responsible for issuing commands to DriverControlSystem.
Attributes	None	
Operations	receiveData()	Service that receives data from VehiclePositionSubsystem and PathPredictionSystem.
	pollData()	Service that requests data from the VehiclePositionSubsystem.
	sendCommand(command : string)	Service that sends commands to the VehicleControlSystem.
	checkEnvironment()	Function responsible for checking the safety of the environment surrounding the vehicle.
Relationships	Associations : HMISystem, DriverAttentionSystem, VehicleControlSystem, PathPredictionSystem, VehiclePositionSubsystem Aggregations: Is part of HandsFreeDrivingSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 24. Data dictionary entry for *DriverAssistSystem*

Element Name		**Description**
DriverAttentionSystem		Subsystem that monitors the user's engagement with the road.
Attributes	None	
Operations	checkDriverAttentiveness() : boolean	Service that checks the user's engagement with the road.
	issueEngagementWarning()	Service that tells the warning outputs to activate.
Relationships	Associations : DriverAssistSystem, InfraredCamera Aggregations: Is part of HandsFreeDrivingSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 25. Data dictionary entry for *DriverAttentionSystem*

Element Name		Description
GPS		System that monitors the position via satellites.
Attributes	coordinates : Array	An array of the coordinates of the vehicle.
Operations	None	
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a PositionSensor	
UML Extensions	None	

Table 26. Data dictionary entry for *GPS*

Element Name		**Description**
HMISystem		System that displays sensor information and issues various warnings when needed.
Attributes	totalWarnings : int	The current total number of attention warnings that have been issued to the driver.
Operations	setFollowingDistance()	Service that allows the user to specify the following distance.
	displayWarning()	Service that displays warnings to the user.
	displaySensorInformation()	Service that displays sensor information to the user.
	turnOnHFDS()	Service that allows the user to enable HFDS.
Relationships	Associations : DriverAssistSystem Aggregations: Is part of HandsFreeDrivingSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 27. Data dictionary entry for *HMISystem*

Element Name		**Description**
HandsFreeDrivingSystem		The main system that keeps the vehicle in a highway lane automatically.
Attributes	None	
Operations	enableHFDS()	Function that enables the HFDS after all system checks return as passed (is environment safe, is adaptive cruise control enabled, is user attentive, is position validated, is highway enabled).
	abortHFDS()	Function responsible for aborting the HFDS if any single point of failure is encountered.
	updateSystem()	Function responsible for securely retrieving updates for the HFDS over the internet.
	verifySystemIntegrity()	Function responsible for testing if the system is compromised to any DOS or other compromising attacks.
Relationships	Associations : DriverAssistSystem Aggregations: Is made up of : HMISystem, DriverAssistSystem, DriverAttentionSystem, VehicleControlSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 28. Data dictionary entry for *HandsFreeDrivingSystem*

Element Name		**Description**
InfraredCamera		Device responsible for measuring the attentiveness of the user.
Attributes	rawData : PNG	The raw analog data the camera collects.
Operations	monitorUser()	Function responsible for initiating the infrared camera to monitor the user.
Relationships	Associations : DriverAttentionSystem Aggregations: None Compositions : None Generalization: None	
UML Extensions	None	

Table 29. Data dictionary entry for *InfraredCamera*

Element Name		**Description**
Lidar		Position sensor responsible for collecting and providing light detection data.
Attributes	pointCloud : ADC	The raw data stored as a point cloud in an ADC format.
Operations	None	
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a PositionSensor	
UML Extensions	None	

Table 30. Data dictionary entry for *LiDAR*

Element Name		**Description**
PathPredictionSystem		Subsystem responsible for calculating the projected path based on information from the VehiclePositionSubsystem.
Attributes	pointCloud : ADC	The raw data stored as a point cloud in an ADC format.
Operations	calculateBluePath()	Function that calculates the projected path based on data from the VehiclePositionSubsystem.
	checkHighwayEnabled()	Function that verifies if the highway has been enabled by the Path Prediction System.
	sendBluePath(path: Path)	Operation responsible for sending the blue path to the Vehicle Control System.
Relationships	Associations : VehiclePositionSubsystem, DriverAssistSystem, VehicleControlSystem Aggregations: None Compositions : None Generalization: None	
UML Extensions	None	

Table 31. Data dictionary entry for *PathPredictionSystem*

Element Name		**Description**
PositionSensor		Class that defines the functionality for all types of position sensors available to the HFDS.
Attributes	None	
Operations	recordData()	Operation responsible for recording the data from a specific position sensor on the vehicle.
Relationships	Associations : VehiclePositionSubsystem, DriverAssistSystem Aggregations: None Compositions : None Generalization: None	
UML Extensions	None	

Table 32. Data dictionary entry for *PositionSensor*

Element Name		Description
RGBLights		Responsible for controlling the flashing warning lights for the HFDS.
Attributes	None	
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of WarningOutput	
UML Extensions	None	

Table 33. Data dictionary entry for *RGBLights*

Element Name		**Description**
RADAR		Position sensor responsible for collecting and delivering radar data.
Attributes	pointCloud : ADC	The raw data stored as a point cloud in an ADC format.
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of Position Sensor	
UML Extensions	None	

Table 34. Data dictionary entry for *RADAR*

Element Name		**Description**
SeatVibrator		Warning output responsible for vibrating the seat of the user.
Attributes	None	None
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of WarningOutput	
UML Extensions	None	

Table 35. Data dictionary entry for *SeatVibrator*

Element Name		**Description**
Speaker		Warning output responsible for playing warning sounds to the user.
Attributes	None	None
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of WarningOutput	
UML Extensions	None	

Table 36. Data dictionary entry for *Speaker*

Element Name		Description
SteeringWheel		The actuator responsible for the angle along the transverse axis of the vehicle.
Attributes	None	None
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of Actuator	
UML Extensions	None	

Table 37. Data dictionary entry for *SteeringWheel*

Element Name		**Description**
VehicleControlSystem		The system responsible for issuing commands to the vehicle actuators.
Attributes	None	None
Operations	receiveCommand()	Function that receives a command from the DriverAssistSystem.
	checkAdaptiveCruiseControl()	Function that checks to see if Adaptive Cruise Control is enabled and speed is set to at least 20 miles per hour.
	issueAction(command : string, actuator: Actuator)	Function that tells an actuator the action to take.
	retrieveBluePath()	Operation responsible for retrieving the blue path from the path prediction system.
	checkCommand(path: Path, command : string)	Operation that checks the validity of a command with respect to a given blue path.
Relationships	Associations : DriverAssistSystem, VehicleControlSystem Aggregations: Is part of HandsFreeDrivingSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 38. Data dictionary entry for *VehicleControlSystem*

Element Name		Description
VehiclePositionSubsystem		The system responsible for processing sensor data from the vehicle's position sensors.
Attributes	None	None
Operations	processSensorData(sensor: PositionSensor)	Function responsible for processing any sensor data from any of the position sensors.
	validateVehiclePosition()	Function responsible for validating the vehicles position with respect to the position sensors and calculated blue path.
	validateLiDARMap()	Function that validates the LiDAR map of the highway with current position sensor data.
	sendData()	Function responsible for sending data from the position sensors to the DriverAssistSystem.
Relationships	Associations : PathPredictionSystem, DriverAssistSystem Aggregations: None Compositions : None Generalization: None	
UML Extensions	None	

Table 39. Data dictionary entry for *VehiclePositionSubsystem*

Element Name		Description
WarningOutput		The device which defines all actions of all warning outputs.
Attributes	None	
Operations	playWarning()	Method responsible for playing a warning output on a speaker, RGB light, or the vibration of the seat.
Relationships	Associations : None Aggregations: Part of DriverAttentionSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 40. Data dictionary entry for *WarningOutput*

4.3 Sequence Diagrams

This section will introduce sequence diagrams for specific scenarios in the HFD system. Sequence diagrams are used to capture the step by step procedure by which the system executes one of its features. Sequence diagrams show how objects interact with each other via the ordering of messages. The blue boxes along the top of the sequence diagram represent objects. Arrows between them indicate a message being sent between objects. The sequence diagram shows progression from top to bottom, messages towards the top of the diagram happen first and messages towards the bottom happen last.

4.3.1 Normal Operation

Figure 5 shows what happens when the system is fully operational with proper road conditions. It demonstrates the user activating the system when the vehicle is on a supported highway. Also demonstrated is how the system interacts with the DriverAttentionSystem. First, the user turns on HFDS. Then the system checks if the Adaptive Cruise Control is on and ensures that the conditions are safe. If the user is not paying attention, green lights are displayed to ensure that they are paying attention. If the user continues to not pay attention, the next warning is displayed with yellow lights.

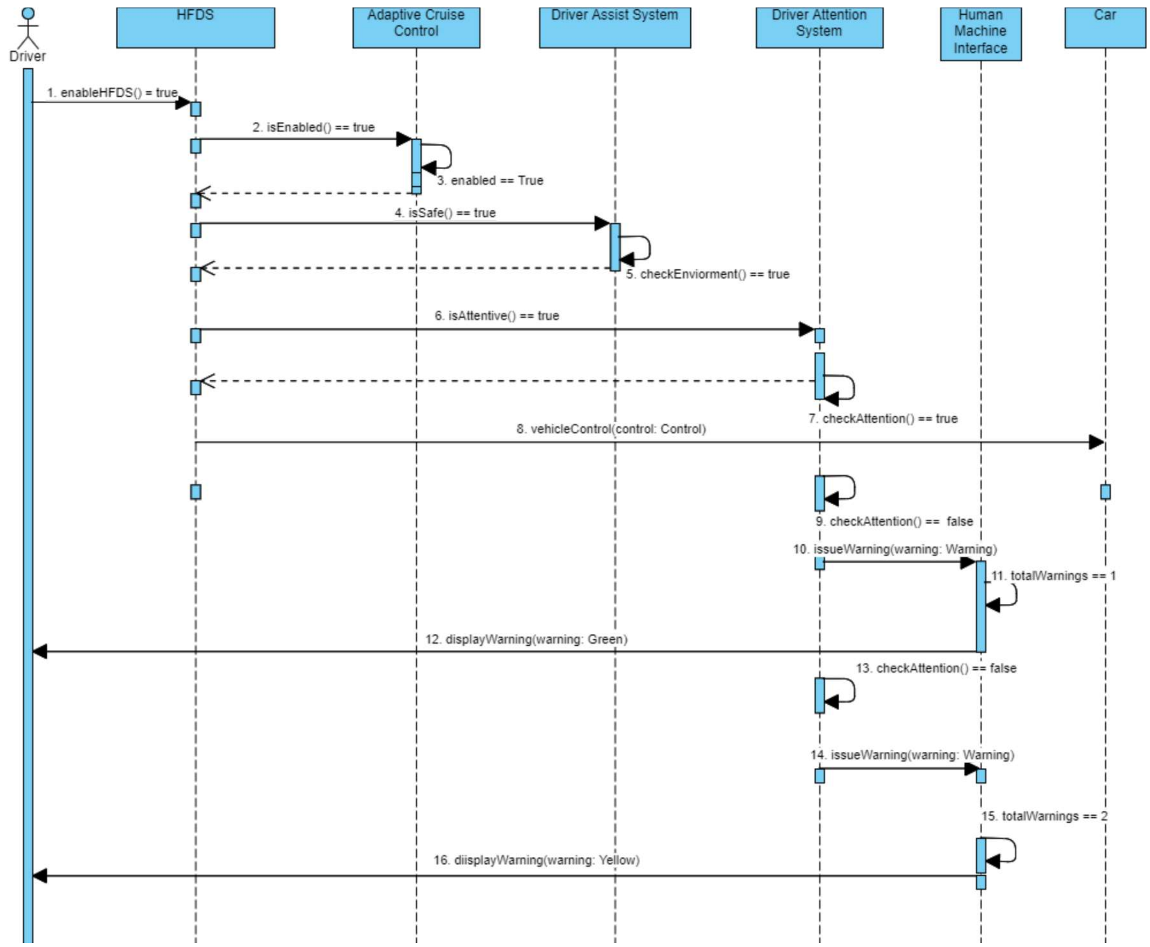

Figure 5. Sequence Diagram for Turning On the System and under normal operation and issuing two warnings to the driver when the driver is deemed not attentive.

4.3.2 System Failure

Figure 6 shows what happens when there is a system failure. It demonstrates that the system is already active while the vehicle is on a supported highway. The system then goes to collect data from one of its cameras and the camera data indicates that there was failure. When a failure occurs the system immediately warns the user what has failed and HFDS deactives - bringing the car to a complete stop with hazard lights active.

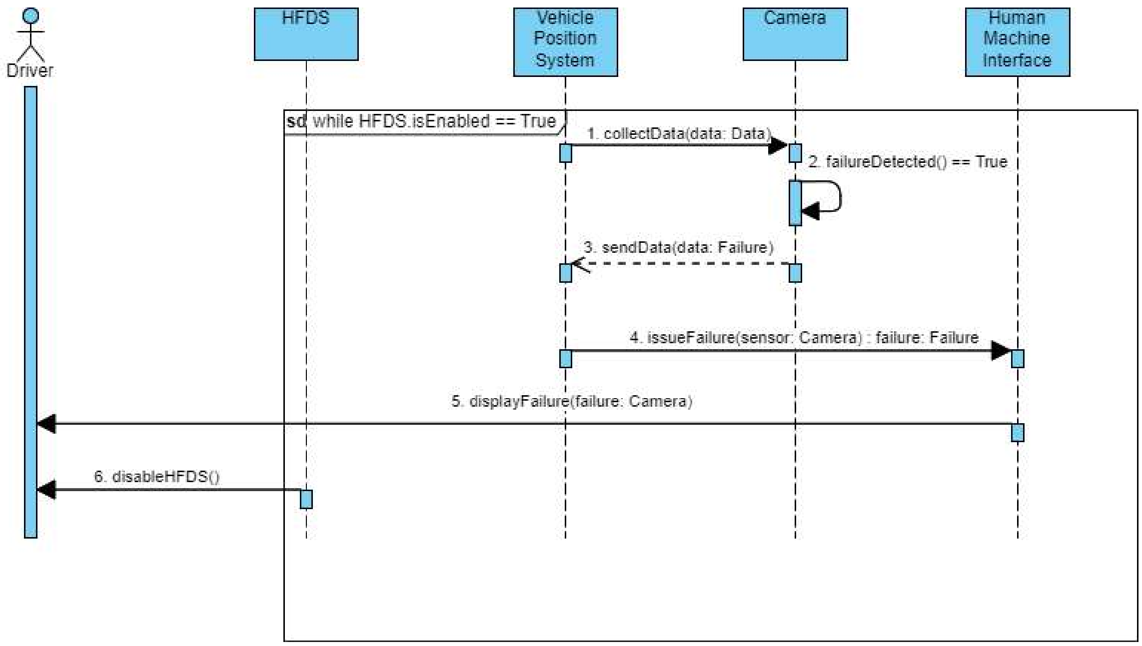

Figure 6. Sequence Diagram for A Single System Failure

4.3.3 Driver Disengagement

Figure 7 depicts what happens when the user becomes permanently disengaged from the road. First, if the user is found to not be attentive for 5 seconds the first warning is issued. If the user is found to not be attentive for another 5 seconds the third warning is issued. After another 5 seconds the third and final warning is issued, vibrating the seat of the user. After another 5 seconds if the user is still not attentive the car will come to a complete stop.

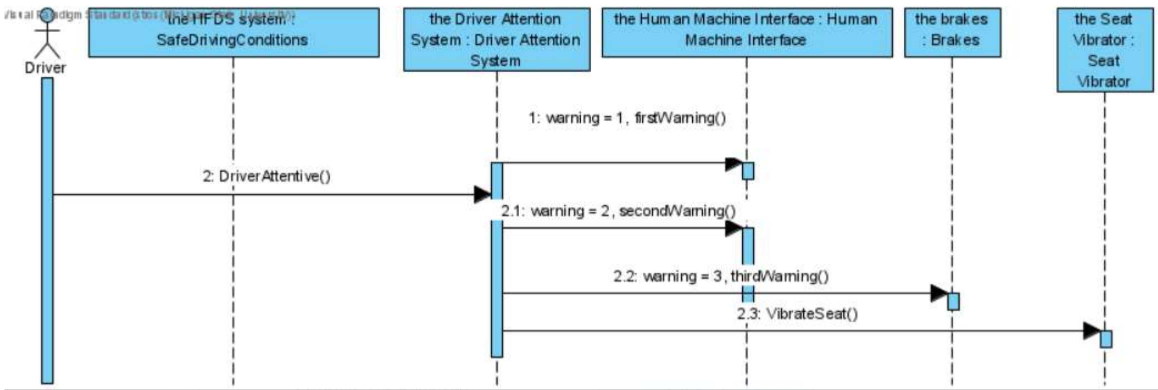

Figure 7. Sequence Diagram for Permanent User Disengagement

4.3.4 Continuous Control Change

Figure 8 shows what happens when there is a continuous change of control back and forth between the user and HFDS. First, the user enables HFDS which tells the HFDS system to turn on. If enableHFDS() is false, the HFDS reverts control to the user. If the enableHFDS() function is true then the HFDS is started. The HFDS issues commands to the actuators to control the vehicle. If there is a fault, the HFDS reverts control to the user. Also shown is when there is a user override on one of the actuators. This results in HFDS being disabled and control being reverted back to the user.

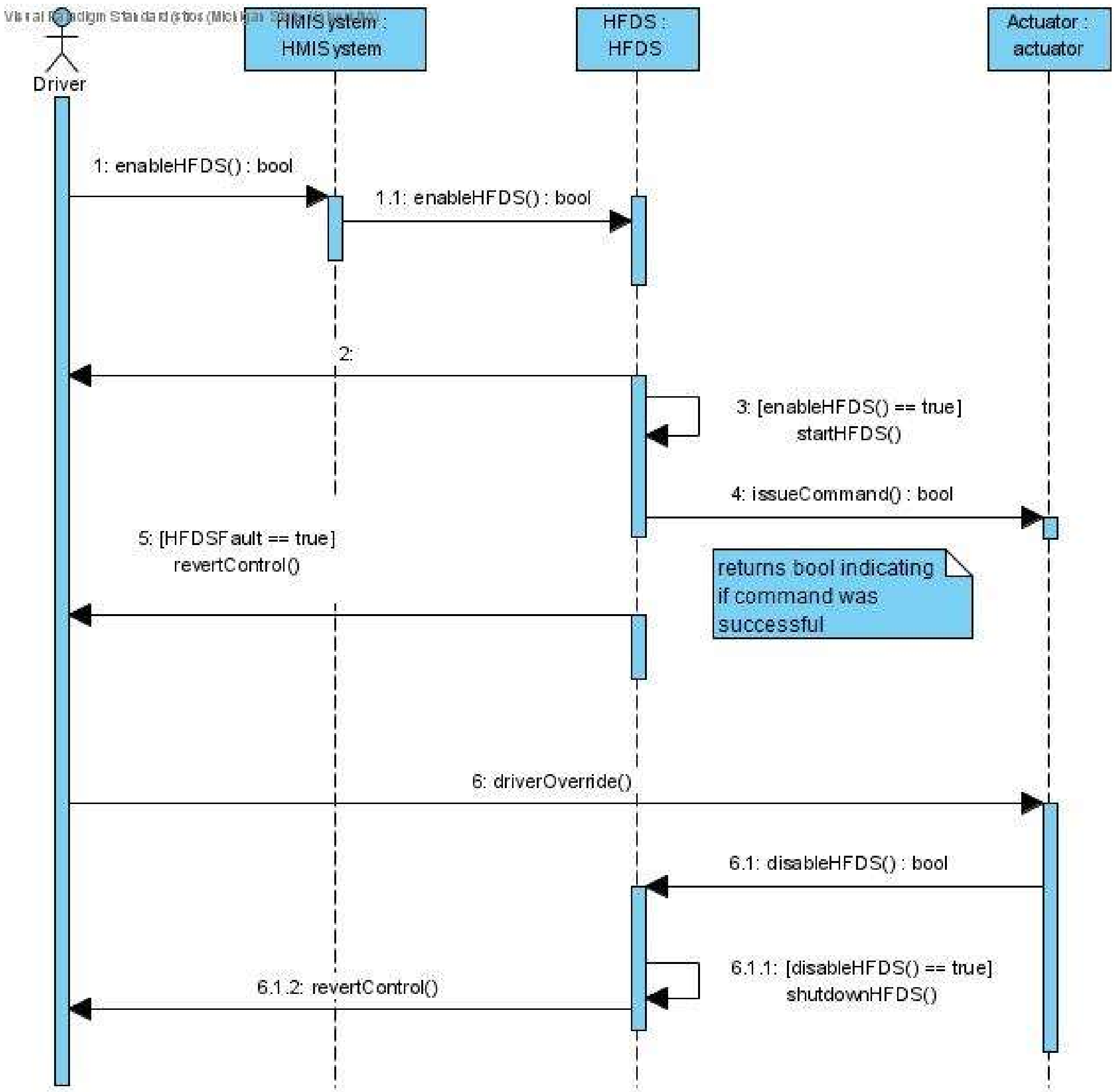

Figure 8. Sequence Diagram for Continuous Control Change

4.3.5 System Security

Figure 9 displays what measures can be taken to ensure only authorized commands are sent to the VehicleControlSystem. First, under normal conditions a command is sent from the DriverAssistSystem to the VehicleControlSystem. The VehicleControlSystem then gets the blue path and checks to see if there is a conflict between the path and the command. If this conflict is true, the VehicleControlSystem receives a command from the DriverAssistSystem. The VehicleControlSystem then checks the commands against each other. If there is no conflict, and the commands were checked the control system can send actions to the actuator. If there is a conflict or the commands were not checked, the actuator is sent a halt command.

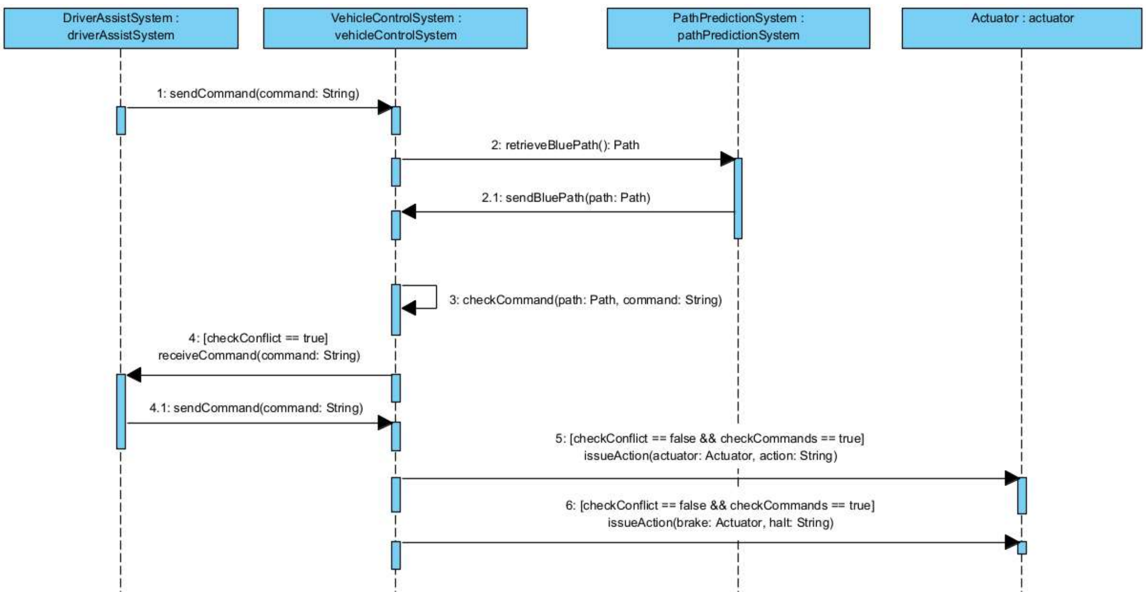

Figure 9. Sequence Diagram for Security Measures

4.4 State Diagram

This section contains the state diagram that shows how the system changes to different states when enabling HFDS. States are represented by the blue rectangles and transitions between them are represented by the named arrows. The name on the arrow represents what event took place to cause the state to change.

Figure 10 displays the process by which the HFDS turns on. The initial state is HFDS being disabled. If the user intiaizalies the HFD system the first procedure which is followed is the surroundings being checked. If the surroundings are deemed not safe, HFDS is not allowed to engage. If the surroundings are deemed safe, the system then moves into the next procedure of checking user attention. If the user is not attentive HFDS is not allowed to engage. If the user is deemed attentive then the system moves into its next procedure of checking if the cruise control is enabled. If the cruise control is not enabled HFDS is not allowed to engage. If cruise control is enabled then the system begins vehicle position validation. If the position is invalid, HFDS is not allowed to engage. If this state is valid, HFDS is finally allowed to engage.

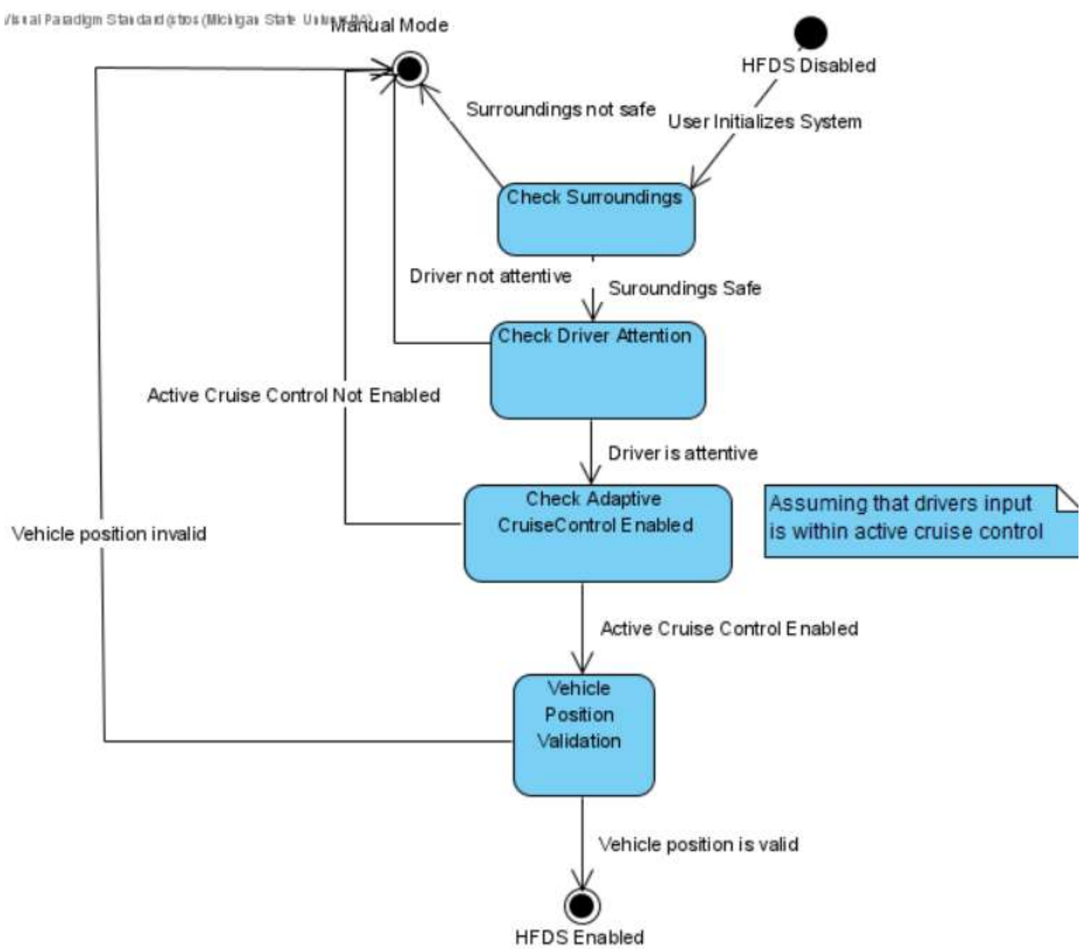

Figure 10. State Diagram for HFDS

4.4.1 Normal Operation

Figure 11 demonstrates how the system issues warnings with the DriverAttentionSystem. While HFDS is active the system checks the user's attention in the DriverAttentionSystem. It monitors the user's head and eye movements and declares the user attentive. A warning is issued by the DriverAttentionSystem and the HumanMachineInterface displays a warning message. If this is the first warning then it will display the warning specified for the first warning. If a second warning is issued the HumanMachineInterface will display the warning specified for the second warning.

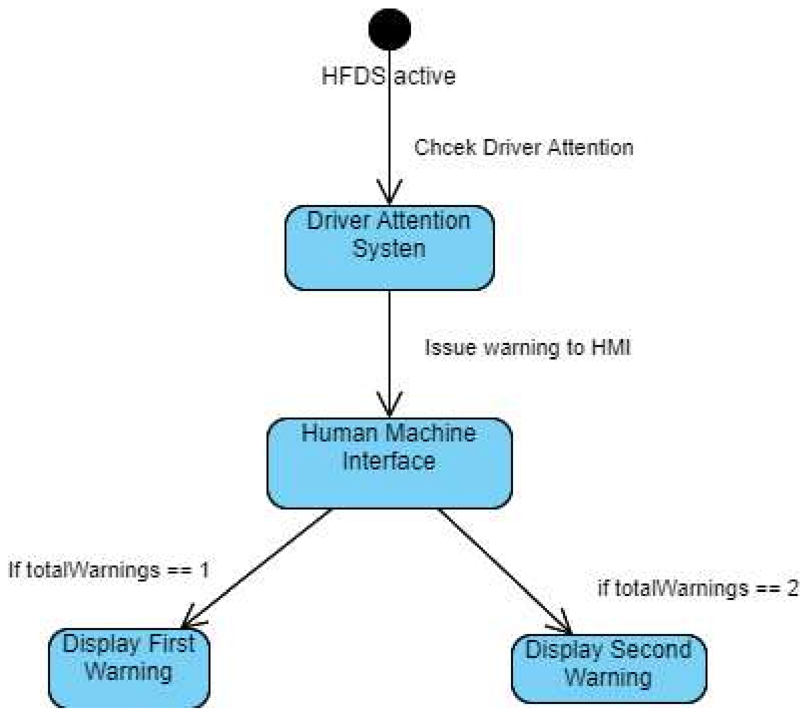

Figure 11. State Diagram for Turning On the System and under normal operation and issuing two warnings to the driver when the driver is deemed not attentive.

4.4.2 System Failure

Figure 12 displays what happens when there is a system failure, in this case when one of the vehicle cameras detect a failure. When HFDS is enabled, the VehiclePositionSystem proceeds to collect data from the vehicle cameras. While collecting data, a camera failure is detected. This failure is issued to the HumanMachineInterface where it will display system failures , similar to the third driver attention warning. The warning vibrates the seat of the user to re-engage them with the vehicle. If the driver is attentive and takes control of the vehicle, the system will disable and revert back to user control. If the driver is not attentive during the warning phase, the system will follow the emergency shutdown procedure. The procedure will begin by activating the vehicle's hazard lights and the brakes will be applied until the vehicle comes to a complete stop and HFDS will be disabled.

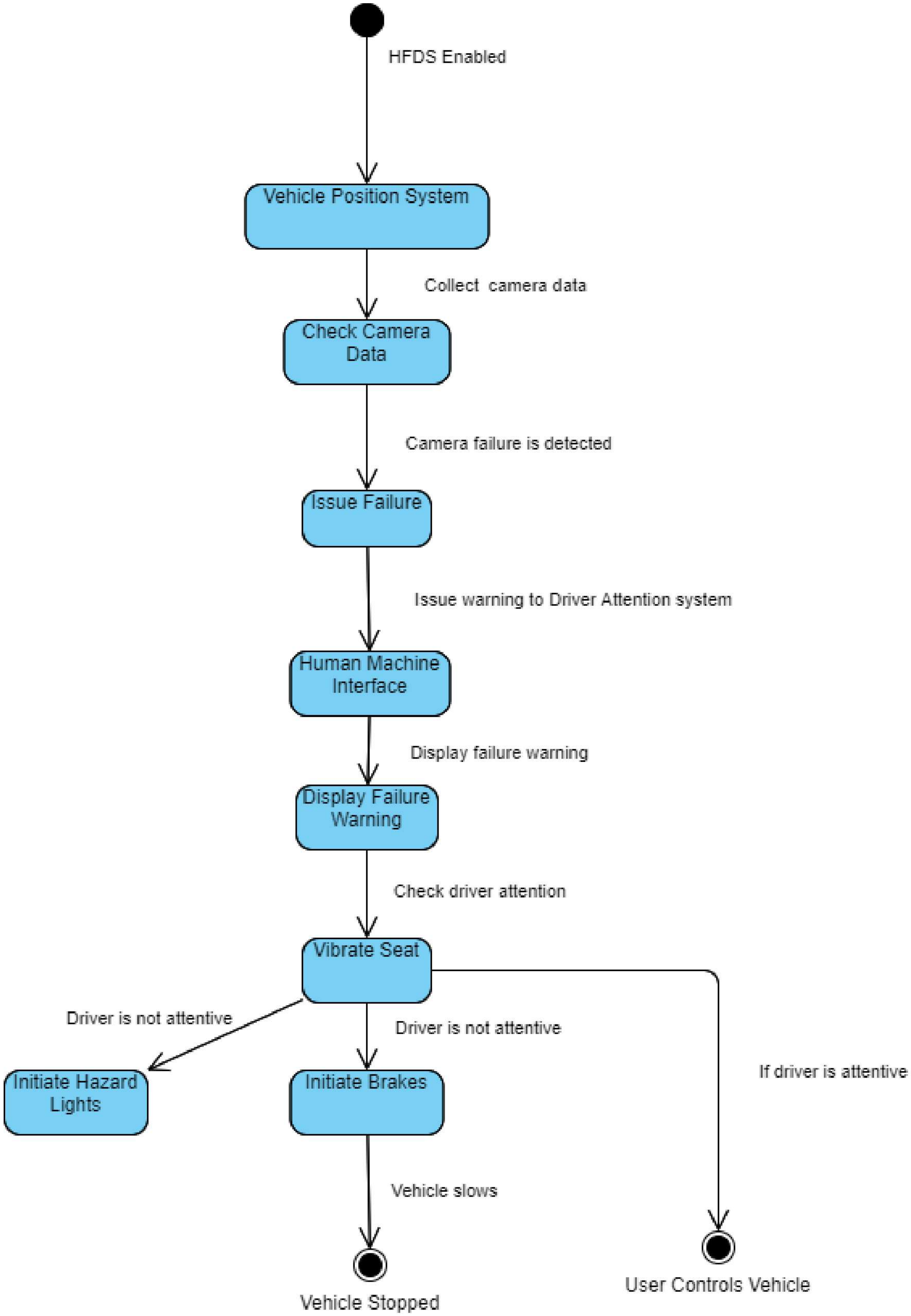

Figure 12. State Diagram for A Single System Failure

4.4.3 Driver Disengagement

Figure 13 displays what happens when the user becomes permanently disengaged from the road. The DriverAttentionSystem checks for attentiveness from the driver. The user is declared not attentive, therefore, the first attention warning is displayed. The user is again declared not attentive after 5 seconds from the first warning, so the second attention warning is displayed. The user is still not attentive after the second warning which will commence the third and final warning. The final warning will vibrate the seat of the user to again attempt to re-engage them with the vehicle. The attempt to re-engage the user was unsuccessful, now the system will follow the emergency shutdown procedure. The vehicle's hazard lights will be activated and the brakes will be applied until the vehicle comes to a complete stop.

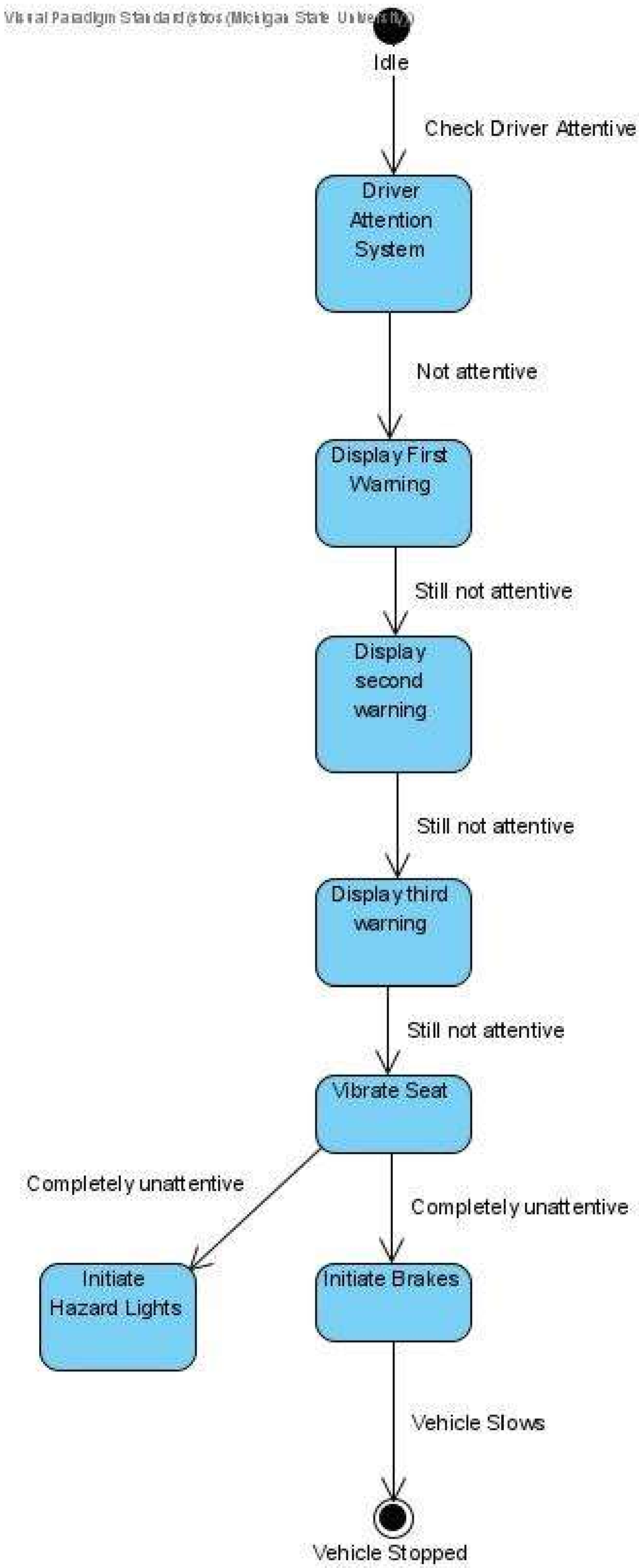

Figure 13. State Diagram for Permanent User Disengagement

4.4.4 Continuous Control Change

Figure 14 demonstrates what happens when there is a continuous change of control back and forth between the user and HFDS. The user initiates the power switch to enable HFDS. Following the initiation of the HFDS a command is sent to the actuators of the system. There is an actuator fault detected which will cause the system to revert control back to the user. The user takes back control of the vehicle and proceeds to enable HFDS again. The HFDS will follow the same process as stated earlier, a fault is detected and control is reverted back to the use.

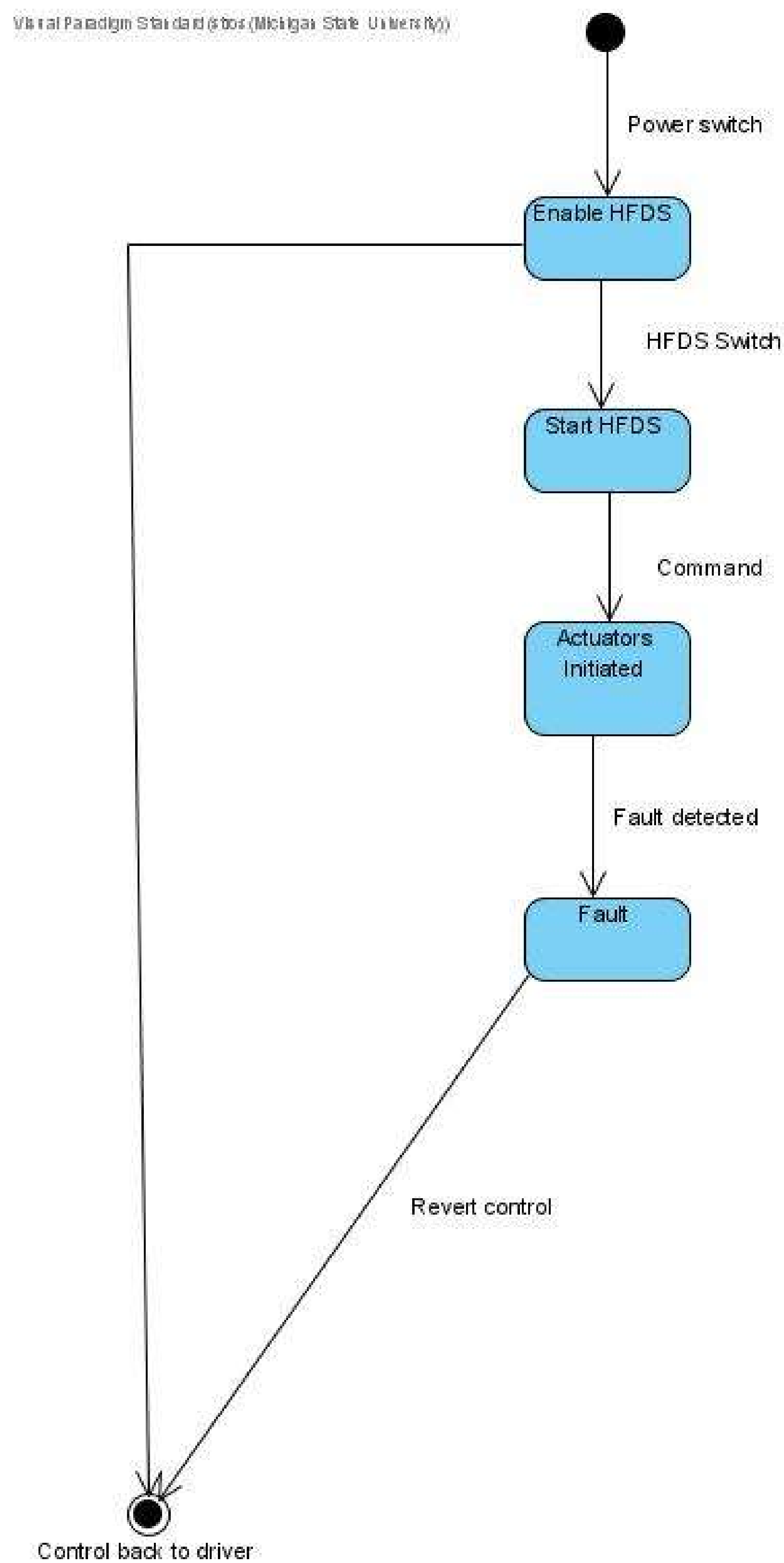

Figure 14. State Diagram for Continuous Control Change

4.4.5 System Security

Figure 14 displays what measures are taken to ensure only authorized commands are sent to the VehicleControlSystem. While HFDS is active, the VehicleControlSystem receives commands from the DriverAssistSystem. The VehicleControlSystem then receives the blue path to analyze for any conflicts between the path and command. If there is a conflict found, an actuator is sent a command to halt. If there is no conflict found, an initiation command is sent to an actuator.

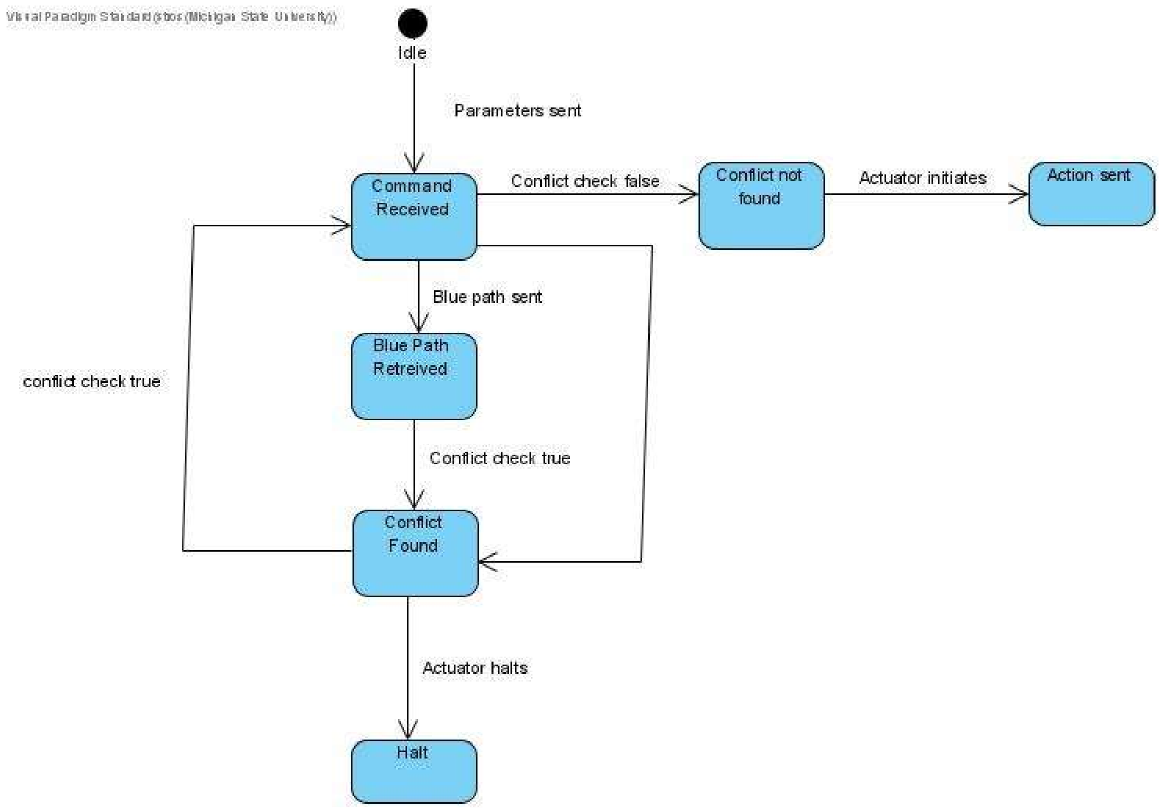

Figure 15. State Diagram for Security Measures

5 Prototype

A prototype is a rudimentary representation of a product. To help stakeholders visualize the HFDS that will be developed, a prototype in the form of a game was created that can account for elements outlined in the design documents.

Figure 16. Web demo of Prototype

As shown in Figure 16, the prototype can be interacted with through select keys outlined in the instructions. The interactive elements of the prototype include the HUD on the top left that indicate if the system has received input from the user, the speedometer above the central console, the central console, and the rest of the world.

5.1 How to Run Prototype

This subsection will outline how stakeholders will access the prototype and how they will interact with the prototype.

The prototype is accessible through this link: https://cse.msu.edu/~zhouti17/HFDS4/prototypes/prototype1.html.

The Phaser.js (3.55.2) library was used to create the prototype, it has the function to switch between WebGL rendering and canvas (HTML5) rendering depending on the browser used to view the webpage. All browsers that support the HTML5 standard are capable of running the prototype, that is, all browsers made/updated past 2020 will support the prototype. Independent copies of the libraries required are hosted on the site linked above, but should the stakeholder wish to run the prototype independently on their system, they are recommended to download Phaser 3.55.2. Performance will depend on the viewer's hardware.

As the controls are performed using a keyboard, one will be required to fully test the prototype.

5.1.1 Controls

The prototype starts with the car in a stopped state. It can be controlled using 14 keyboard keys:

- W increases the car's velocity by 5MPH
- S decreases the car's velocity by 5MPH
- A turns the car a bit to the left
- D turns the car a bit to the right
- B completely stops the car in 2 seconds
- H toggles HFDS depending on various factors
- F will trigger a single point of failure in the HFDS
- T will add/remove light rain from the environment
- Y will add/remove heavy rain from the environment
- U will add/remove another car on the road
- I adds in a pedestrian on the road
- O creates a construction site on the road
- P toggles the user's attention between attentive and not attentive
- R restarts the prototype in case of a crash

5.2 Sample Scenarios

This subsection will outline some of the possible scenarios the prototype can account for.

In ideal operating condition (i.e. no rain and clear path ahead), the HFDS has the following states, represented by a text output on the central console:

1. User is not paying attention and attempts to activate HFDS

Figure 17. Console display prompting the user to pay attention

2. The car's speed is below 20MPH and user attempts to activate HFDS

Figure 18. Console display instructing the user to reach 20MPH

3. The car is moving and the user attempts to activate the HFDS, so HFDS turns on

Figure 19. Console display showing that HFDS has been activated

4. The car is moving and HFDS is turned on, the user activates brakes which stops the car and turns off HFDS

Figure 20. Console display showing that HFDS has been deactivated

5. The car is moving and HFDS is turned on, HFDS faults, the car continues moving while checking if the user is attentive. Red lights inside the car will start flashing and an audio prompt telling the user to pay attention will play. If the user is attentive, the HFDS will just give control back to the user.

Figure 21. Sequence of console displays when the HFDS faults and the user is attentive

If the user is not attentive, the HFDS will pull over to the side of the road.

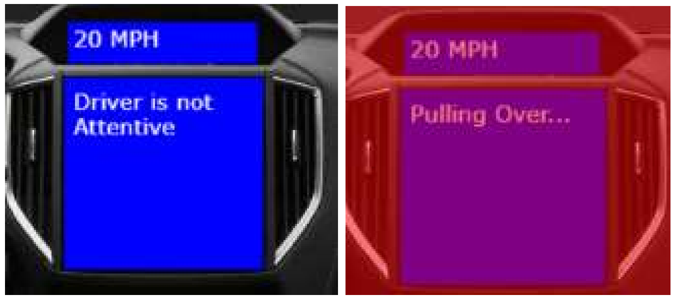

Figure 22. Sequence of console displays when the HFDS faults and the user is not attentive

Figure 23. Final state of the car after it faults and the user is not attentive

6. The car is moving but more than half of the car is beyond the rumble strip, the HFDS cannot calculate a blue path and fails to activate

Figure 24. Blue path cannot be calculated when the car is not on the road

With user adjustable settings such as weather and other cars, more scenarios are introduced.

1. HFDS is on, light rain is detected, and the car is cruising at a safe speed. The console will flash a warning but otherwise proceed as usual.

Figure 25. HFDS detects light rain at safe speed

2. HFDS is on, light rain is detected, and the car is going too fast. The console will display the same warning seen in Figure 25 but also slow the car down to 60MPH

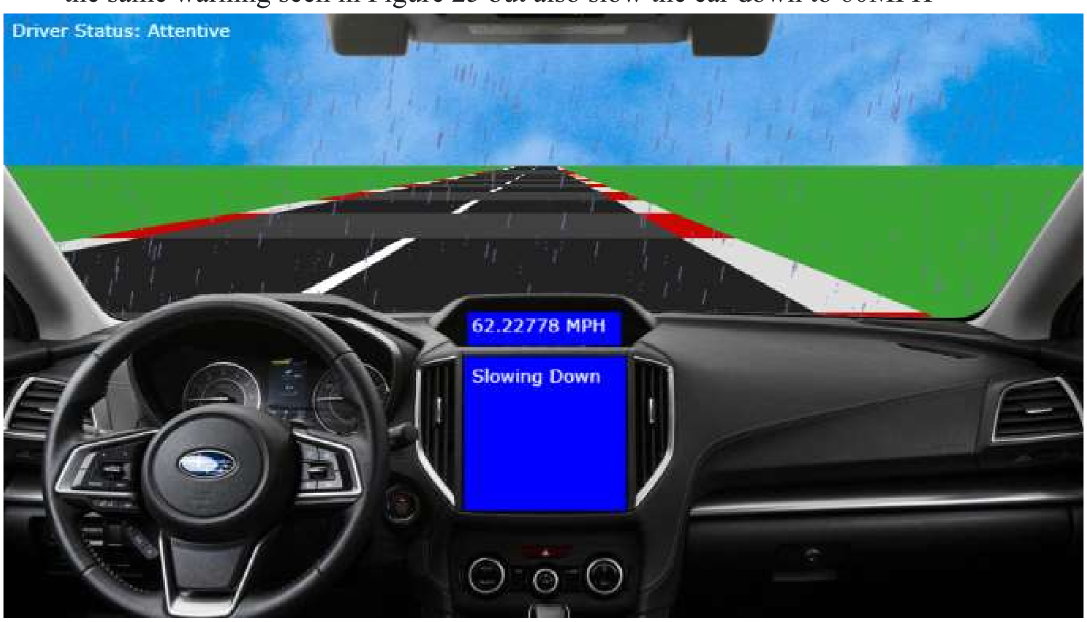

Figure 26. HFDS in the process of decelerating due to rain

3. HFDS is on and heavy rain is detected. The console will display a heavy rain warning and relinquish control to the user

Figure 27. HFDS detecting heavy rain

4. HFDS is on and a car is detected ahead of the user's car, the car will slow down and ensure a safe distance between the two cars

Figure 28. HFDS matching the other car's speed of 20mph

6 References

[1] I. D. Brown, “Driving fatigue,” *Endeavour*, vol. 6, no. 2, pp. 83–90, 1982.

[2] P. Thiffault and J. Bergeron, “Monotony of road environment and driver fatigue: A simulator study,” *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 381–391, 2003.

[3] D. Thakore and S. Biswas, “Routing with Persistent Link Modeling in Intermittently Connected Wireless Networks,” Proceedings of IEEE Military Communication, Atlantic City, October 2005.

[4] T. Zhou, “CSE 435: Hands-Free Driving System (HFDS) team 4,” *HFDS4*. [Online]. Available: https://cse.msu.edu/~zhouti17/HFDS4/homepage.html.

[5] J. Gordon, “javascript-racer: How to build an outrun-style racing game in javascript,” *GitHub*. [Online]. Available: https://github.com/jakesgordon/javascript-racer.

7 Point of Contact

For further information regarding this document and project, please contact **Prof. Betty H.C. Cheng** at Michigan State University (chengb at msu.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.