# Software Requirements Specification (SRS)

Pedestrian Collision Avoidance System (PCAS) 3

**Authors:** David Levy Khankin, Shoimya Chowdhury, Dorian Florescu, Ricky Horan & Rikito Takai

**Instructor:** Dr. Betty H.C. Cheng

Michigan State University - CSE435

November 2022

## 1 Introduction

Autonomous driving is an area of intense interest by the automotive industry and the public. A self-driving car must be able to stay in its own lane, brake at intersections, and remain under control during these maneuvers. The system should be capable of exhibiting basic driving skills. Another fundamental aspect of the driving function is avoiding collisions in emergency situations. In this case this has to do with pedestrians. This document details the requirements, functionality, and elements and uses the Pedestrian Collision Avoidance System (PCAS) that assists automotive drivers from accidentally colliding with pedestrians. In section 1, the purpose of this document and the scope of the software to be developed for PCAS are discussed.

## 1.1 Purpose

What this document accomplishes is to provide details that are easily understandable of PCAS such that the document can satisfy the requirements of the customer while at the same time be a clear guide for designing and implementing the system. This document is intended for the developers to understand the PCAS before they start the designing and implementation phases. This paper is also meant to explain every component of the system to the project's stakeholders.

## 1.2 Scope

PCAS is an embedded system in an automotive vehicle whose primary function is to assist the driver in preventing collisions, injuries, and damages to pedestrians that come in range of the vehicle when it is traveling at lower speeds. PCAS obtains sensor values from onboard sensors, processes this information using a controller, and utilizes actuators to engage in appropriate responses based on the scenario it is placed in. PCAS uses sensors to detect obstacles in the path of the vehicle and will respond with warnings to the driver. If needed the system will automatically decelerate/stop the vehicle to prevent collision, but will automatically accelerate back to the most recent speed if the obstacle is no longer a danger. The driver has the ability to deactivate the system at any given moment.

## 1.3 Definition, acronyms, and abbreviations

**Pedestrian Collision Avoidance System (PCAS):** The PCAS is the entire embedded system that is described throughout the document.
**Dashboard:** The information display found behind the steering wheel where odometer and speedometer are located.
**Obstacle:** Anything that the vehicle could collide with when driving. The main example is a pedestrian.
**Autonomous Emergency Braking (AEB):** A system to engage the brakes automatically if the vehicle is moving above a set speed and an object is detected.
**Anti-lock Braking System (ABS):** A system to prevent brakes from locking up and skidding when engaged.

## 1.4 Organization

Section 2 of the SRS will be the overall description of the PCAS system. The overall description includes the product perspective, functions, user characteristics, and constraints on the system. It will also describe the assumptions and dependencies of the system, environment, and user interactions.

Section 3 describes specific requirements. It will contain an enumerated list of hardware, invariants, and system requirements.

Section 4 includes diagrams for the PCAS system and their descriptions. The first diagram is the use case diagram with use case descriptions. The second diagram will be the domain model with a data dictionary that has an explanation for all elements of the diagram. Next is sequence diagrams that outline important scenarios within the system. Lastly, State Diagrams are used to describe how the system changes in response to certain triggers.

Section 5 introduces the prototype with descriptions on how to use it followed by a description of scenarios on how the system behaves.

Section 6 provides a point of contact for further information regarding this document and project.

# 2 Overall Description

Section 2 describes the PCAS in detail. Firstly, the document summarizes the functions and perspective of the product. After that, the expectations for the users of the system are described. Then come the constraints that are set on the system. Wrapping it up the document explains the assumptions and dependencies of the requirements.

## 2.1 Product Perspective

The PCAS is responsible for obstacle detection, collision prevention and damage mitigation. Accident prevention and damage mitigation are both high-level goals of vehicles, the PCAS works specifically to prevent unintended collisions with pedestrians or objects during vehicle operation under 50 kph. The PCAS is a unique addition to a vehicle, it interacts with many other components of the vehicle. As a subsystem of the vehicle, the PCAS cannot exist independently as it relies on existing components on the vehicle. Examples include the dashboard, the sound system, and the haptic feedback system. These are used mainly to warn the driver of a pedestrian or obstacle. The dash alerts the driver through visual cues with messages. The sound system gives audible warnings of varying frequencies depending on the distance of the detected pedestrian or object. The haptic feedback system alerts the driver through vibrations in the steering wheel.

## 2.2 Product Functions

The subsystems of the PCAS system include automatic braking, automatic acceleration, object detection and reporting of pedestrians. These allow the system to react appropriately according to a given situation. The system uses sensors, cameras, a display, the accelerometer, and other data to control the vehicle. These allow the system to maintain a safe distance from pedestrians and avoid potential collisions.

The automatic braking system is provided by the customer. It is mentioned as the Brake by Wire system. Throughout our document we refer to this as the Automatic Braking System. We also reference Autonomous Emergency Braking and Anti-lock Braking Systems which refer to this same system.

## 2.3 User Characteristics

What is expected of the user of the PCAS system is described here. The driver of the vehicle with PCAS is expected to possess a driver’s license that is valid. The PCAS manual is provided to the driver and they are expected to have read and understand it. During operation of the vehicle and the PCAS system, the driver is responsible for following the traffic laws and ensuring the safety of the vehicle’s passengers and nearby pedestrians.

## 2.4 Constraints

The constraints of this system are as follows. This is only for use under the assumption that the existing conditions match the described conditions. No extreme external factors are accounted for.

All hardware must function properly always. It is assumed that if anything causes the failsafe state from faulty hardware, increasing the reaction time of the AES / ABS system from 200 ms to 900 ms will fix any issue.

## 2.5 Assumptions and Dependencies

The PCAS uses other systems and components of the vehicle to be developed. External hardware or software systems of the vehicle such as the braking system and sound system etc. are expected to be installed on the vehicle and working properly. When the PCAS requests a service provided by other systems, the corresponding systems are expected to perform the action requested by the PCAS. PCAS uses multiple sensors and feedback systems, all external communication formats are assumed to be uniform. The driver of the vehicle is expected to use the PCAS system as specified in the document.

## 2.6 Apportioning of Requirements

Several components of the PCAS system will be outside of the scope, they may be addressed in the future. The pedestrian or object detection algorithm used for object recognition is not considered in the document. The algorithm's implementation is abstract. The data from the sensors will not be fully detailed. Vehicle's OS and resource management will not be part of the project.

# 3 Specific Requirements

Section 3 describes the hardware, invariants, and system requirements for the PCAS to be developed.

## 3.1 Hardware Requirements

1. **Radar, Sonar, and/or Lidar Scanner**
   *For collecting information about the environment and alerting the driver if need be.*
   a. The sensors, whether sonar, radar, or both, must be active when the system is on.
   b. The sensors should alert the driver by providing a visual and audible alert if an object is detected.
   c. The sensors should not alert the driver if the vehicle is not in motion and the driver has already been alerted to the presence of an obstacle.
   d. The system should only report obstacles detected within 30 meters of the vehicle.

2. **360 Degree Coverage Cameras**
   *For both sensing obstacles and giving the driver a visual on their display.*
   a. The cameras must always be on, and on the drivers display, when the system is active.
   b. The cameras will alert the driver by providing a visual and audible alert if an object is detected.
   c. The camera should not alert the driver if the vehicle is not in motion and the driver has already been alerted to the presence of an obstacle.
   d. The system should only report obstacles detected within 30 meters of the vehicle.
3. **Autonomous Emergency Braking and Anti-lock Braking Systems**
   *Also called AEB and ABS, respectively. For use by the system to brake automatically if need be.*
   a. If you are within 30 meters from a detected object, apply brakes automatically to maintain a probable stopping distance (based on the current braking delay and speed of the vehicle).
   b. The automatic braking system should only work if the speed of the vehicle is below 50 kph.
4. **Digital User Interface**
   *For providing the driver with information about the environment behind them, as well as visual alerts.*
   a. If the vehicle is not in motion, and the driver has already been given a visual warning, the driver should not be warned again.
   b. The driver should be informed if we switch the system off.

# 3.2 Invariants

Invariants are a special subset of requirements that must always be true. The system has failed when an invariant is violated.

- Avoid All Collisions
- The brakes will always work
- The vehicle will not exceed 10 mph if an obstacle is detected within 10 ft in front of the car
- The alert system in the vehicle should go off when the sensors are triggered
- The hardware should always be working: camera, sensors, AEB, ABS
- The system will never suffer from cyber attacks
- The system can be deactivated manually at any moment
- The system will never attempt to steer the vehicle
- The system should always apply brakes in a way to not cause injury to the driver

## 3.3 System Requirements

**Set of requirements for a vehicle system that will avoid pedestrian collisions:**

**Overall System Requirements**

*These requirements are for the overall system itself.*

1. The system should only be active when the vehicle is turned on.
2. If the system is unable to operate nominally, possibly caused by damaged sensors, weather, or other unexpected conditions, the system should inform the driver as such and then operate in Failsafe Mode.
3. When the system is initially activated, if an object is detected, the driver should be warned.
4. The system should not alert the driver, or automatically apply brakes, if it has been manually disengaged, but should notify the driver that the system is off.

## 3.4 Security Requirements

**While out of scope for what the customer has requested, as stated by the customer, we have several potential requirements that can help maintain a secure system.**

- Create a microchip that controls the system with built in security. This tends to cost more.
- Allow the manual disabling of certain automatic systems in case of a cyber attack.
- Allow the ability to disconnect the vehicle from any bluetooth or wifi connection that it may have in order to cut off outside signals.
- Create key fobs that have stronger RFID capabilities.

# 4 Modeling Requirements

Section 4 presents various diagrams and their corresponding descriptions to depict key elements, interactions, scenarios, and services of the PCAS system in detail.

## 4.1 Use Case Diagram and Descriptions

The sections below introduce the use case diagram for the PCAS and the corresponding use case descriptions. A use case diagram captures a user's visible functionality and interaction with the system. The use case descriptions describe each use case in the use case diagram.

### 4.1.1 Use Case Diagram

The use case diagram below functions to showcase three things: how the driver interacts with the system, how the system interacts with itself, and how the system interacts with a pedestrian. Details on how each use case operates are outlined in the use case diagram descriptions in section 4.1.2.

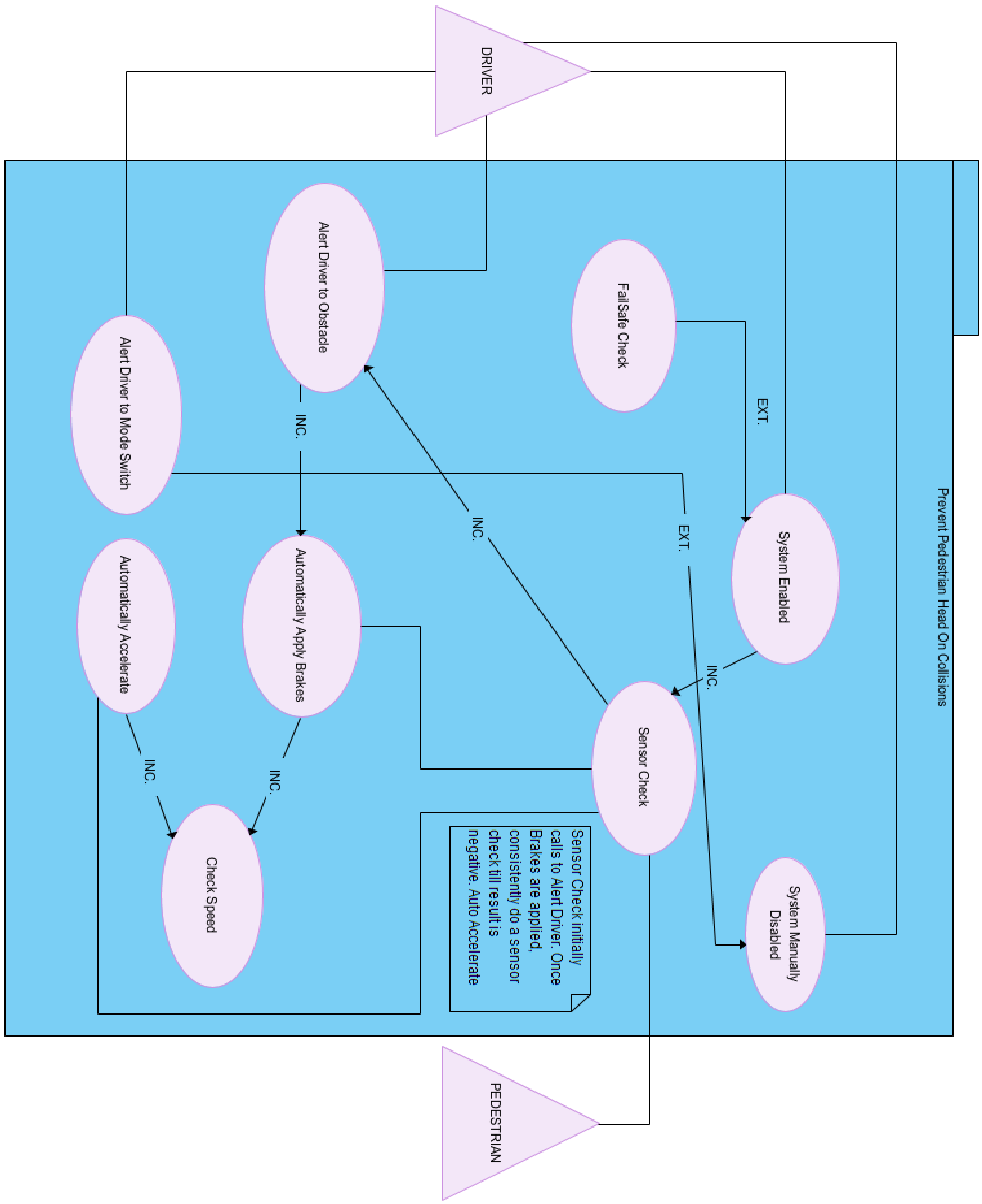



Figure 1: Use case diagram for PCAS

## 4.1.2 Use Case Descriptions

Tables 1 through 9 detail the use cases in Figure 1, providing descriptions of the use cases. The use case field includes the name of the use case. The actor denotes the actors involved in the use case. The description provides an overview of the use case. The type puts the case into either primary or secondary. Includes and extends list other use cases that are related to the use case. Cross-refs enumerates the invariants and requirements in Section 3 that are referenced in the use case. Lastly, the use case field describes the other use cases that must be satisfied for the current use case to be active.

| | |
|---|---|
| **Use Case:** | System Enabled |
| **Actors:** | Driver (initiator) |
| **Description:** | <u>*Turns on the system. This is only the case if several things are true. 1) The vehicle in question is turned on, and 2) the system has not been manually turned off. No other use cases should ever engage or be valid so long as this is not true. It should immediately perform a check for obstacles after this.*</u> |
| **Type:** | Primary and Essential |
| **Includes:** | *Sensor Check* |
| **Extends:** | *Failsafe Check* |
| **Cross-refs:** | 3.3(1), 3.3(3) |

Table 1: Use case description for *System Enabled*

| | |
|---|---|
| **Use Case:** | Failsafe Check |
| **Actors:** | System (initiator) |
| **Description:** | <u>*Checks whether or not the system is in peak condition. If it is not, the system will continue to operate, but will do so in* ***Failsafe Mode.***</u> |
| **Type:** | Secondary and Essential |
| **Cross-refs:** | 3.1(1a, 2a), 3.3(2) |
| **Use cases:** | Must have completed the Systems Enabled use case. |

Table 2: Use case description for *FailSafe Check*

| | |
|---|---|
| **Use Case:** | Sensor Check |

| | |
|---|---|
| **Actors:** | System (initiator), Pedestrian |
| **Description:** | *After the system is enabled and at every moment following activation, this reports if a pedestrian is detected or not, and at what distance they are from the center-front of the vehicle. Based on this, determines whether to accelerate, decelerate, or take no action.* |
| **Type:** | Secondary and Essential |
| **Includes:** | *Alert Driver to Obstacle, Automatically Apply Brakes, Automatically Accelerate* |
| **Use cases:** | Must have completed the Systems Enabled use case.<br>Must have completed the Failsafe Check use case. |

Table 3: Use case description for *Sensor Check*

| | |
|---|---|
| **Use Case:** | Alert Driver To Obstacle |
| **Actors:** | System (initiator), Driver |
| **Description:** | *Provides an alert to the driver after detecting a pedestrian within 30 meters of the vehicle. This alert is both visual and auditory. Emits an auditory beep to warn the driver, as well as provides an alert on the dashboard that a pedestrian has been detected. After the initial alert,* ***it will not continually provide an alert.*** *It will only alert an additional time if a new pedestrian is detected.* |
| **Type:** | Secondary and Essential |
| **Includes:** | *Automatically Apply Brakes* |
| **Cross-refs:** | 3.1(1b, 1c, 1d, 2b, 2c, 2d, 4a), 3.3(3) |
| **Use cases:** | Must have completed the Systems Enabled use case.<br>Must have completed the Failsafe Check use case.<br>Must have completed the Sensor Check use case. |

Table 4: Use case description for *Alert Driver To Obstacle*

| | |
|---|---|
| **Use Case:** | Automatically Apply Brakes |
| **Actors:** | System (initiator) |
| **Description:** | *When initially called, take note of the current speed of the vehicle (this will be the* ***steady state speed****. Apply brakes and decelerate as needed from the current speed. This will not occur if the vehicle exceeds the system speed limit (50 kph.)* |

| | |
|---|---|
| **Type:** | Secondary and Essential |
| **Cross-refs:** | 3.1(3a, 3b) |
| **Includes:** | *Check Speed* |
| **Use cases:** | Must have completed the Systems Enabled use case.<br>Must have completed the Failsafe Check use case.<br>Must have completed the Sensor Check use case. |

Table 5: Use case description for *Automatically Apply Brakes*

| | |
|---|---|
| **Use Case:** | Automatically Accelerates |
| **Actors:** | System (initiator) |
| **Description:** | <u>*Accelerate the vehicle until the current speed matches the speed that was indicated from Automatically Apply Brakes as the **steady state speed**.*</u> |
| **Type:** | Secondary and Essential |
| **Includes:** | *Check Speed* |
| **Use cases:** | Must have completed the Systems Enabled use case.<br>Must have completed the Failsafe Check use case.<br>Must have completed the Sensor Check use case.<br>Must have performed and no longer be performing the Automatically Apply Brakes use case. |

Table 6: Use case description for *Automatically Accelerates*

| | |
|---|---|
| **Use Case:** | Check Speed |
| **Actors:** | System (initiator) |
| **Description:** | <u>*Inform the system of the current speed of the vehicle.*</u> |
| **Type:** | Secondary and Essential |
| **Use cases:** | Must have completed the Systems Enabled use case.<br>Must have completed the Failsafe Check use case. |

Table 7: Use case description for *Check Speed*

| | |
|---|---|
| **Use Case:** | System Manually Disabled |
| **Actors:** | Driver (initiator) |

| | |
|---|---|
| **Description:** | *This will override the system but must be done manually by the operator, in this case the driver.* |
| **Type:** | Primary and Essential |
| **Extends:** | *Alert Driver to Mode Switch* |
| **Cross-refs:** | 3.3(4) |

Table 8: Use case description for *System Enabled*

| | |
|---|---|
| **Use Case:** | Alert Driver To Mode Switch |
| **Actors:** | System (initiator), Driver |
| **Description:** | *Inform the driver on the digital display that the system has been manually disabled.* |
| **Type:** | Secondary and Essential |
| **Use cases:** | Must have completed the System Manually Disabled use case. |
| **Cross-refs:** | 3.1(4b) |

Table 9: Use case description for *Alert Driver To Mode Switch*

## 4.2 Domain Model and Data Dictionary

The following section contains the domain model, object oriented diagram, and corresponding data dictionary, an explanation for all elements of the diagram. Each element within the box titled Pedestrian Collision Avoidance System is a part of the PCAS system. The two elements outside (Pedestrian and Operator) are not a part of the system but interact with it significantly. Each element interacts with the others and these interactions are represented by association lines. Each line is labeled to give a brief description of the relationship between the elements.

### 4.2.1 Domain Model

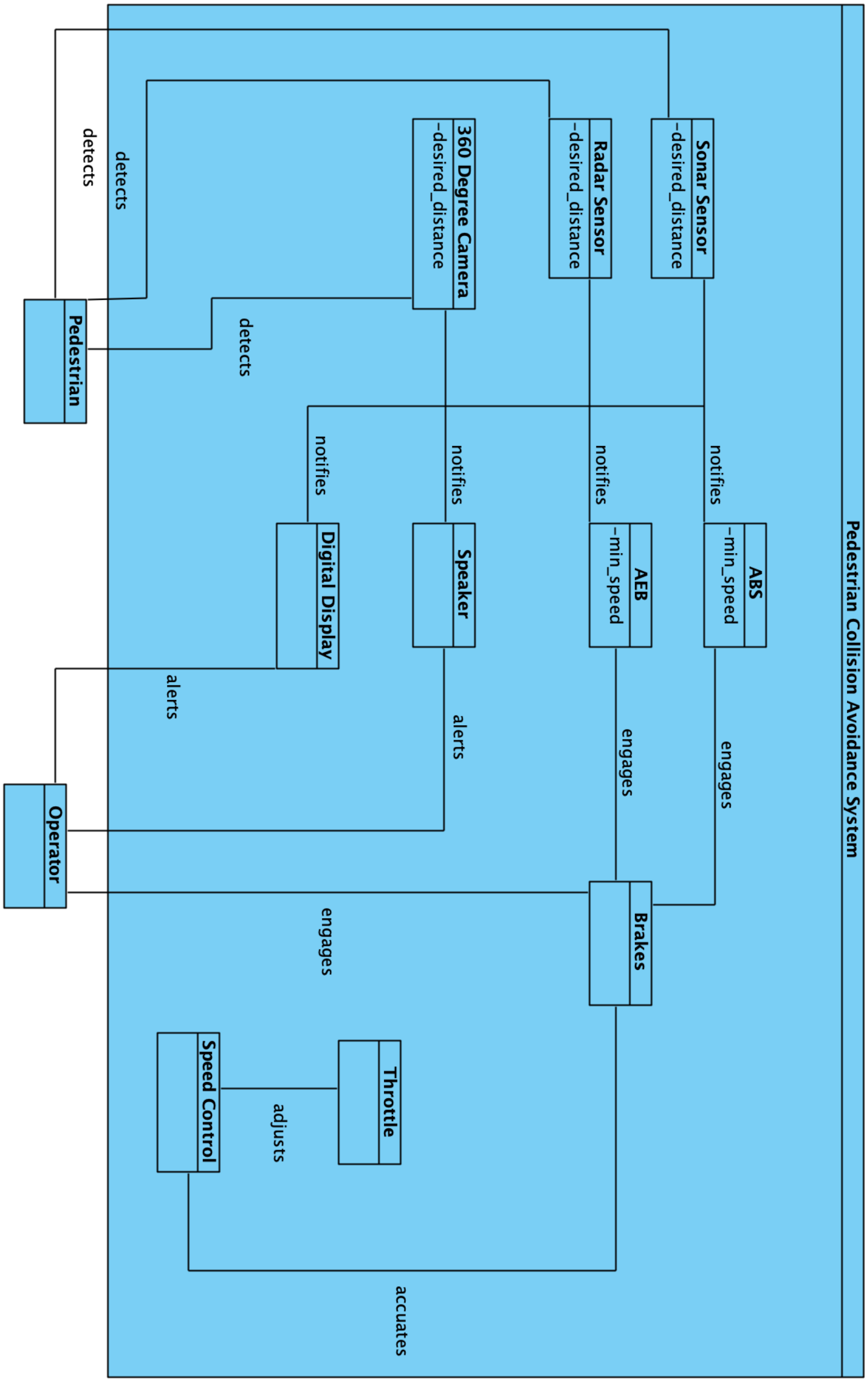



Figure 2: Domain Model for PCAS

### 4.2.2 Data Dictionary

| Element Name | | Description |
|---|---|---|
| ABS (Anti-lock Braking System) | | A system to prevent brakes from locking up and skidding when engaged. |
| **Attributes** | | |
| | minSpeed : int | The minimum speed the vehicle must be moving to resort to the AEB/ABS system. |
| **Relationships** | | |
| | Brakes | Engages the break |

Table 10: Data Dictionary entry for ABS

| Element Name | | Description |
|---|---|---|
| AEB (Autonomous Emergency Braking) | | A system to engage the brakes automatically if the vehicle is moving above a set speed and an object is detected. |
| **Attributes** | | |
| | minSpeed : int | The minimum speed the vehicle must be moving to resort to the AEB/ABS system. |
| **Relationships** | | |
| | Brakes | Engages the break |

Table 11: Data Dictionary entry for AEB

| Element Name | Description |
| --- | --- |
| Brakes | The vehicle's brakes used to slow down the vehicle. |

Table 12: Data Dictionary entry for Brakes

| Element Name | | Description |
| --- | --- | --- |
| Digital Display | | A LED display to alert the user when an object is detected. |
| **Relationships** | | |
| | Operator | Alerts the operator if a pedestrian is detected. |

Table 13: Data Dictionary entry for Digital Display

| Element Name | | Description |
| --- | --- | --- |
| Operator | | The person behind the wheel of the vehicle. Not a part of the system. |
| **Relationships** | | |
| | Brakes | Operator can engage the brakes manually. |

Table 14: Data Dictionary entry for Operator

| Element Name | Description |
| --- | --- |
| Pedestrian | The potential person the the detection field of the sensors. Not a part of the system. |

Table 15: Data Dictionary entry for Pedestrian

| **Element Name** | | **Description** |
| --- | --- | --- |
| Radar Sensor | | The radar sensor on the vehicle. |
| **Attributes** | | |
| | desiredDistance : int | The distance at which the sensor will start notifying the system if an object is detected |
| **Relationships** | | |
| | ABS | Sensor notifies the ABS if a pedestrian is detected. |
| | AES | Sensor notifies the AES if a pedestrian is detected. |
| | Speaker | Sensor notifies the Speaker if a pedestrian is detected. |
| | Digital Display | Sensor notifies the Digital Display if a pedestrian is detected. |
| | Pedestrian | Sensor detects a pedestrian. |

Table 16: Data Dictionary entry for Radar Sensor

| Element Name | | Description |
|---|---|---|
| Sonar Sensor | | The sonar sensor on the vehicle. |
| **Attributes** | | |
| | desiredDistance : int | The distance at which the sensor will start notifying the system if an object is detected |
| **Relationships** | | |
| | ABS | Sensor notifies the ABS if a pedestrian is detected. |
| | AES | Sensor notifies the AES if a pedestrian is detected. |
| | Speaker | Sensor notifies the Speaker if a pedestrian is detected. |
| | Digital Display | Sensor notifies the Digital Display if a pedestrian is detected. |
| | Pedestrian | Sensor detects a pedestrian. |

Table 17: Data Dictionary entry for Sonar Sensor

| Element Name | | Description |
|---|---|---|
| Speaker | | A speaker in the vehicle to alert the user when an object is detected. |
| **Relationships** | | |
| | Operator | Speaker notifies the operator if a pedestrian is detected. |

Table 18: Data Dictionary entry for Speaker

| **Element Name** | | **Description** |
|---|---|---|
| Speed Control | | A device used to store speed information about the vehicle. This allows the vehicle to return to an appropriate speed after handling a pedestrian detection. |
| **Relationships** | | |
| | Brakes | Speed control accuates the brakes. |
| | Throttle | Speed Control adjusts the throttle. |

Table 19: Data Dictionary entry for Speed Control

| **Element Name** | **Description** |
|---|---|
| Throttle | The device used to regulate fuel to the engine and therefore the speed of the vehicle. Updated with information from the sensors to determine when to slow down and reaccelerate in the event of a pedestrian detection. |

Table 20: Data Dictionary entry for Throttle

| Element Name | | Description |
|---|---|---|
| 360 Degree Camera | | A 360 degree camera mounted on the vehicle. |
| **Attributes** | | |
| | desiredDistance : int | The distance at which the sensor will start notifying the system if an object is detected |
| **Relationships** | | |
| | ABS | Camera notifies the ABS if a pedestrian is detected. |
| | AES | Camera notifies the AES if a pedestrian is detected. |
| | Speaker | Camera notifies the Speaker if a pedestrian is detected. |
| | Digital Display | Camera notifies the Digital Display if a pedestrian is detected. |
| | Pedestrian | Camera detects a pedestrian. |

Table 21: Data Dictionary entry for 360 Degree Camera

## 4.3 Sequence Diagrams

In this section multiple sequence diagrams are used to describe each scenario that can occur to the system over time. The large blue boxes are the object and the small bar below is the object lifeline. Bold line arrows between the lifelines are used to indicate that a message was sent from one object to another. The dotted line arrows are used to indicate that a response is needed from the object that the message was sent to.

## 4.3.1 Turning On the System

The figure below is the sequence diagram of turning the system on. First the driver turns on the engine which activates the PCAS system. Once the PCAS system is activated the system triggers the turnOn method to all sensors, cameras, and the dashboard. When the system is on, the camera displays the view to the dashboard at all times.

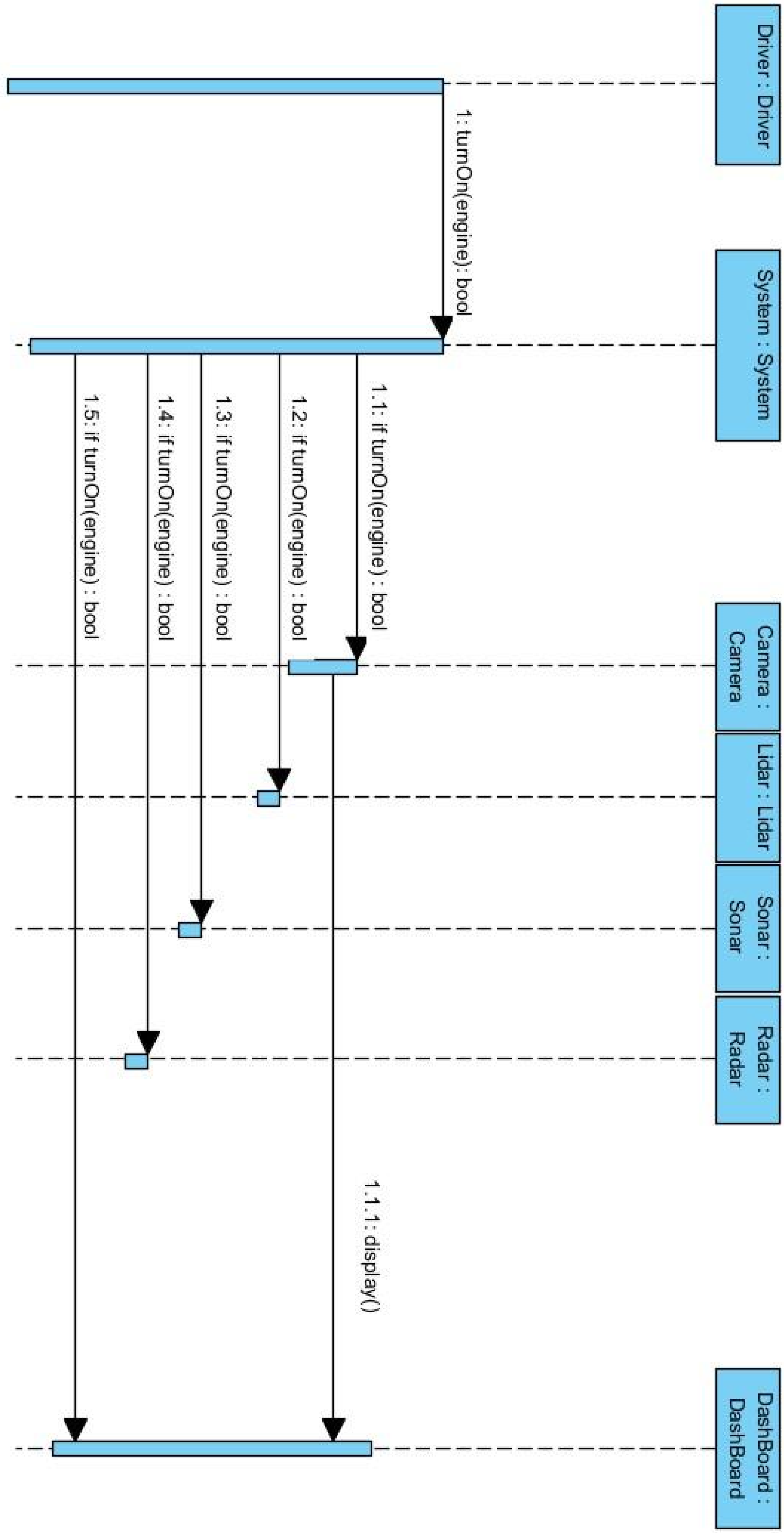



Figure 3: Sequence diagram for Turning On the System

### 4.3.2 Object Detected by Sensors

Figure 4 is the sequence diagram of when our sensors detect an object. Once the system is turned on, it will trigger the camera, lidar, radar, and sonar sensors to send the information to the system. Then the system will check if the object is either 15m, 2m, or 1m away from the car. If the object is only 15m away then the system will only display the camera view to the dashboard. When an object is detected 2m away it will emit warning sounds to the driver, once the warning sounds are activated it will generate a warning message on the dashboard for the driver to see. Lastly if an object is detected 1m away, the system will automatically apply the emergency brakes to prevent collisions.

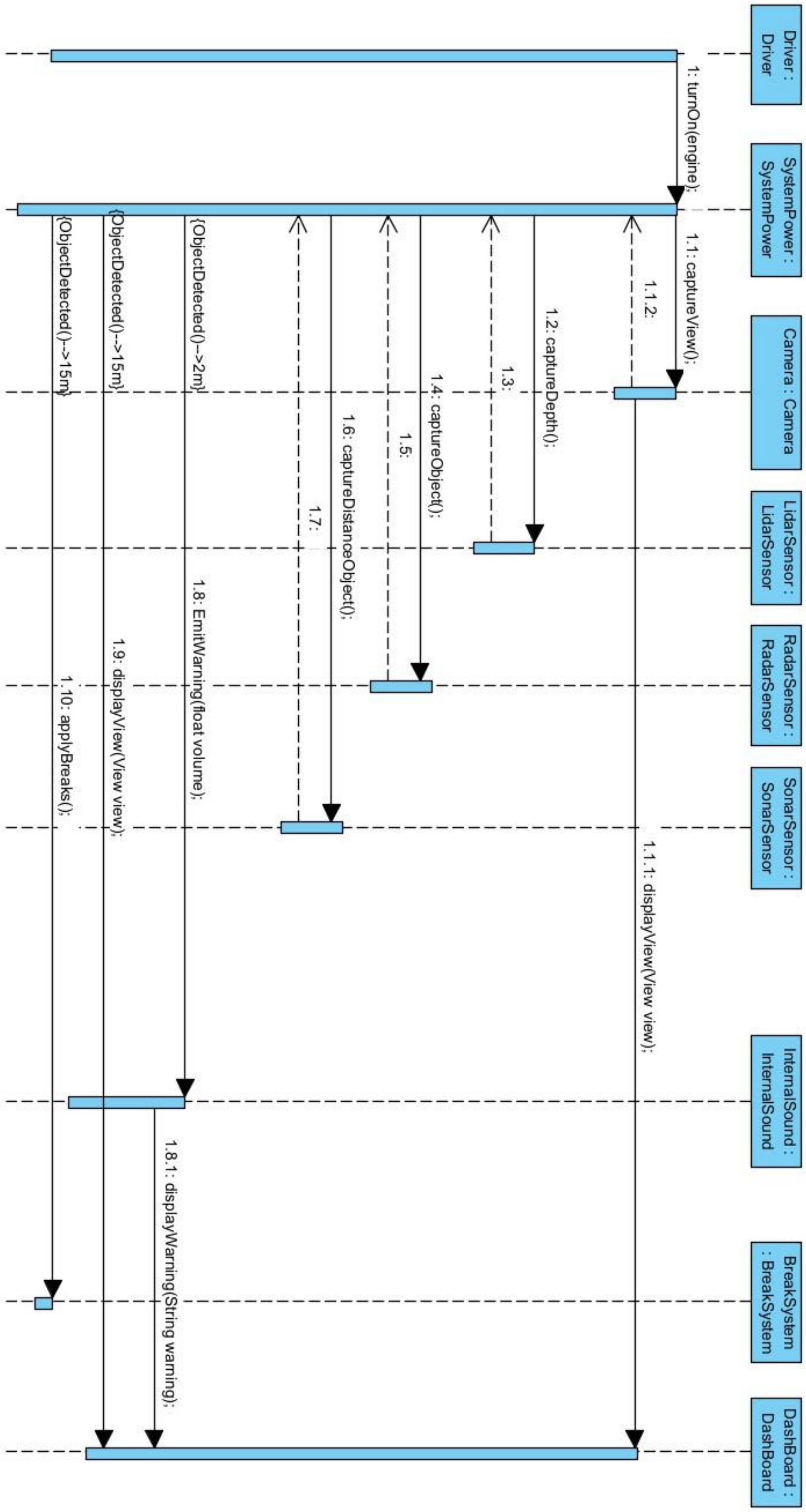



Figure 4: Sequence diagram for Object Detected by Sensors

### 4.3.3 Object No Longer Detected by Sensors

Figure 5 is a sequence diagram of the system no longer detecting any objects. The scenario in this diagram is that our sensors did detect an object and applied the emergency brakes, but now it does not detect any object so it accelerates back to the previous speed that it was in before the emergency brakes. First our cameras and sensors will capture the information outside of the vehicle and our system will check if there are any objects in the way. If the objectDetected function returns false then it will display a safe string message to the dashboard. Then the system will send the previous speed before the emergency brakes to acceleration and the car will automatically accelerate back to its previous speed.

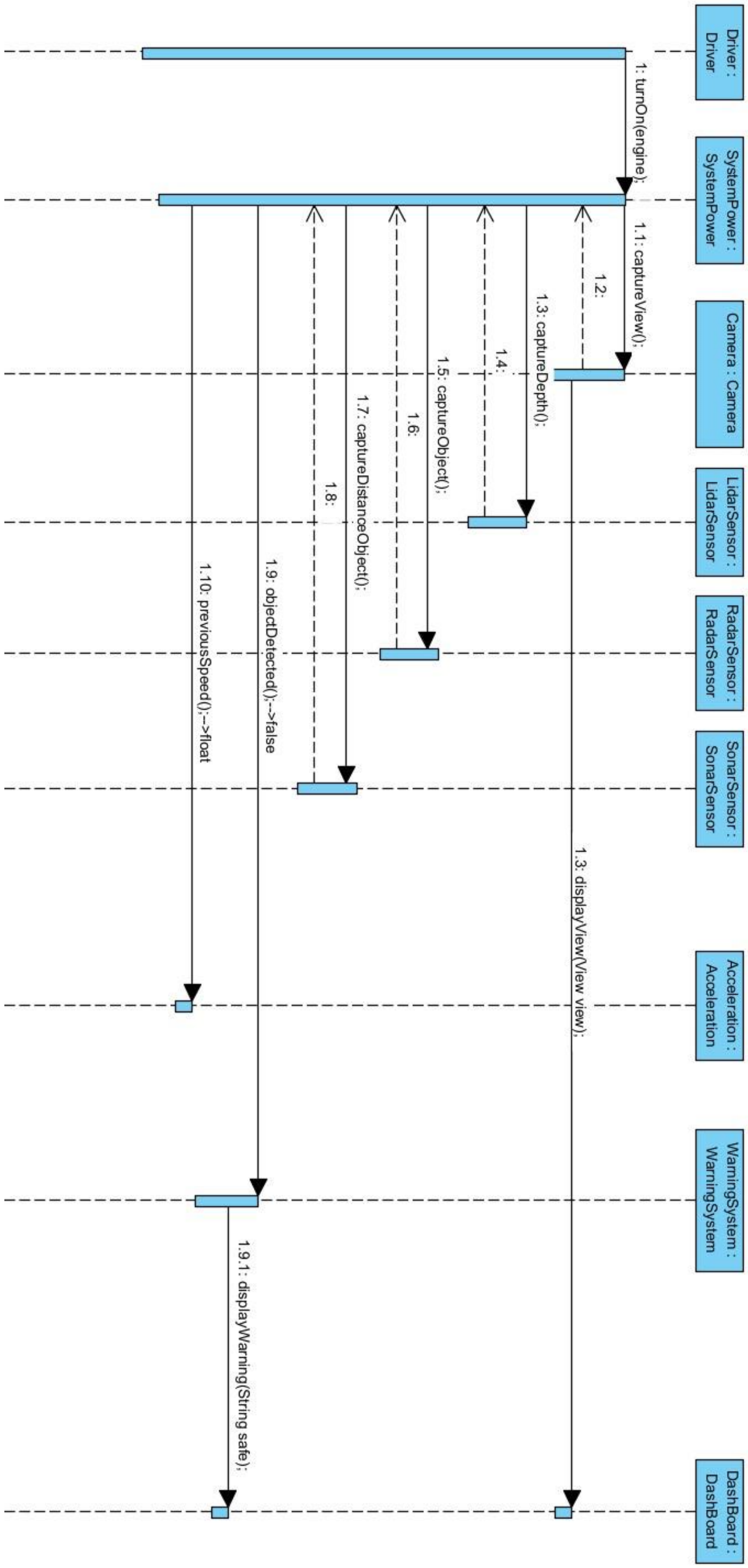



Figure 5: Sequence diagram for Object No Longer Detected by Sensors

### 4.3.4 System Override

Figure 6 is a sequence diagram of when the driver initiates the system override. In this scenario cameras and sensors will work the same as when the sensors detect an object. The driver will have the option to press the override button, if the override button is pressed then the dashboard will trigger the user override to the system. Once the system receives that the user override boolean is true, then the system will call the turnoff function to disable the automatic brake system and the warning system.

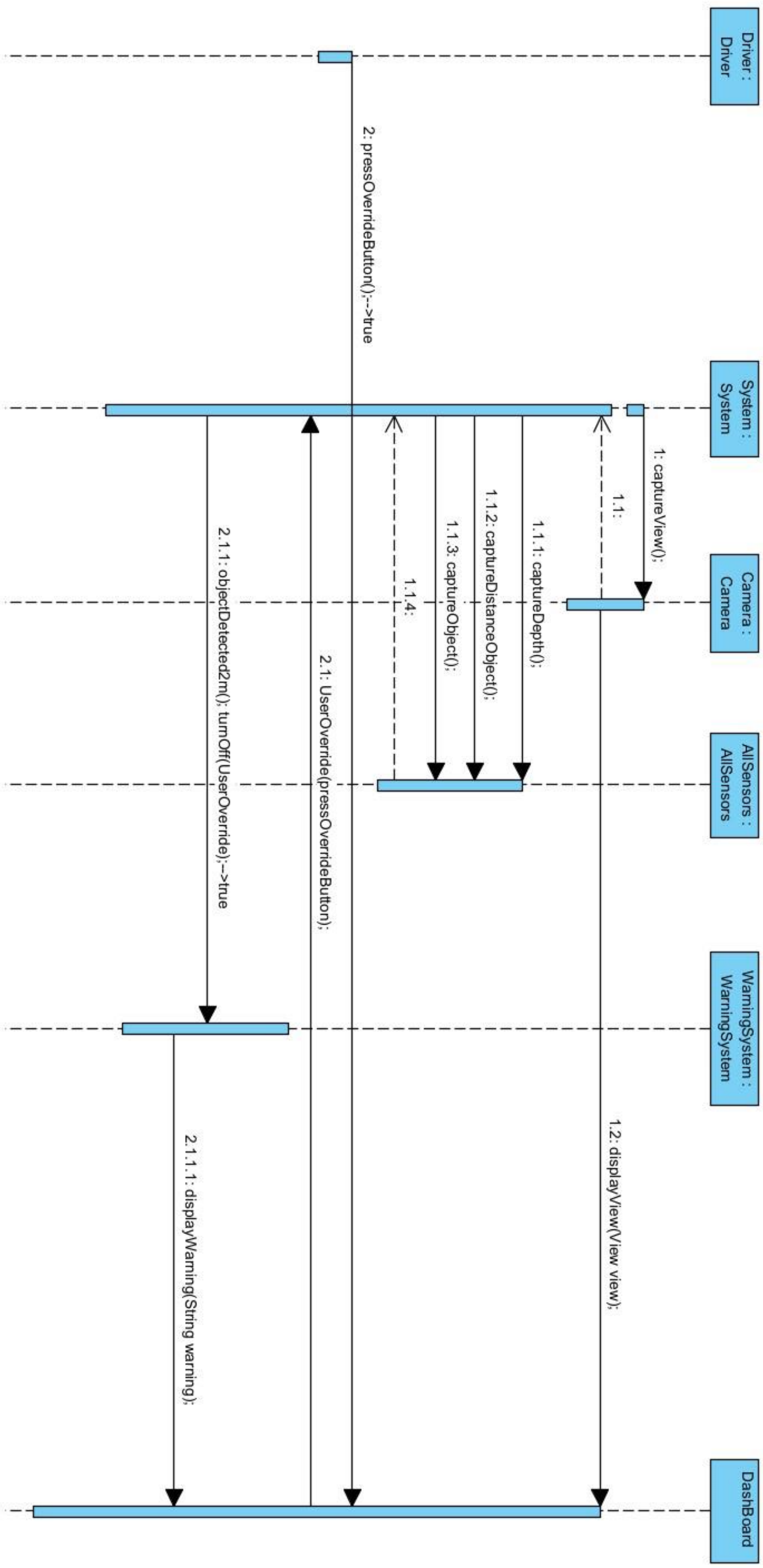



Figure 6: Sequence diagram for System Override

## 4.3.5 Failsafe Conditions Detected

Figure 7 is a sequence diagram of when the failsafe conditions are detected.  In this scenario, when the failsafe conditions are detected the system notifies the ABS with a deceleration request, when in fail safe the driver can not override as the system is already in failsafe mode. Once the request is received, the ABS increases the response time to 900ms from 200ms and notifies the system.

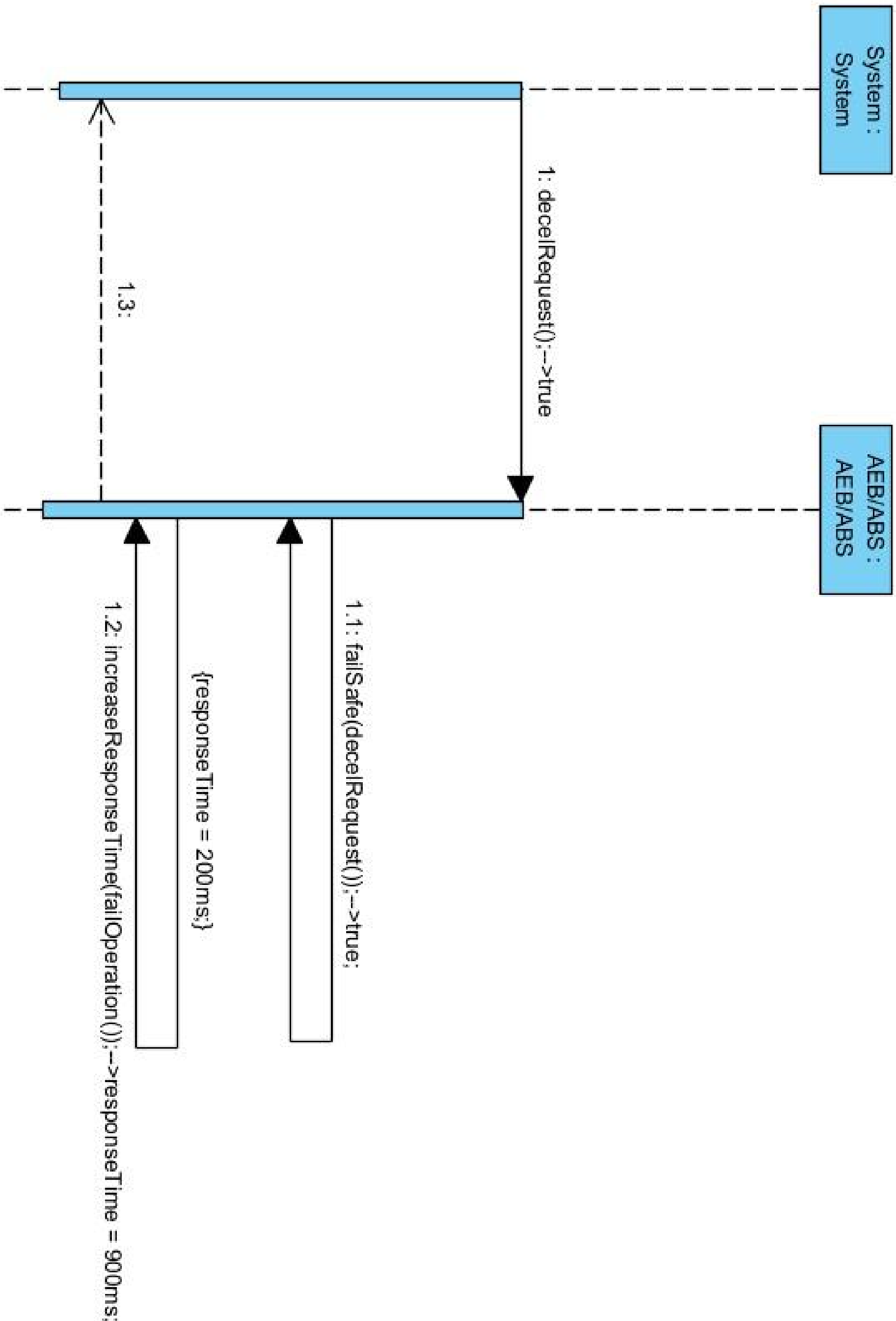



Figure 7: Sequence diagram for Failsafe Conditions Detected

## 4.4 State Diagrams

This section shows the state diagrams to model all of the possible states while also showing the behavior of one object over many use cases. In a state diagram, the start of the state is shown by a black-filled dot with an arrow leading to a state. States of the system are represented by the blue boxes. Arrows that connect the two states represent state transitions, these will also have descriptions.

### 4.4.1 Detection System

Figure 8 below shows the different states of the detection system. In this scenario the initial state of the detection system is off, when the engine turns on the detection system transitions to on state. When in on state, the camera and the sensors are also on and may communicate with the braking system as per the situation.

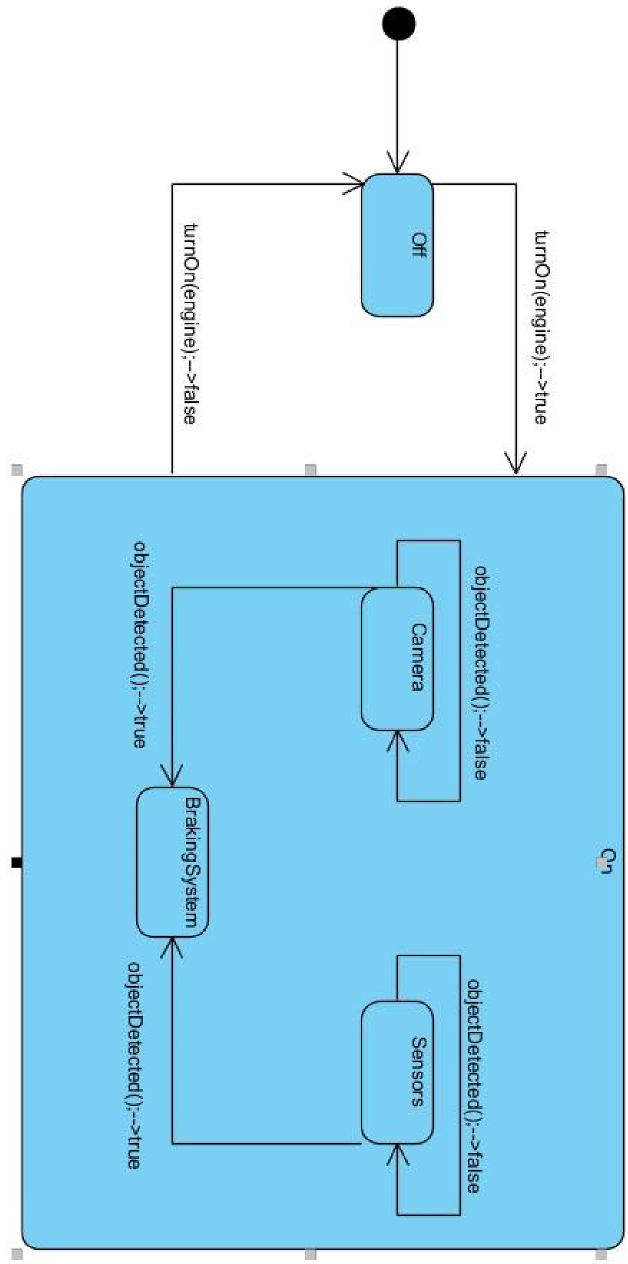



Figure 8: State Diagram for Detection System

## 4.4.2 Braking System

Figure 9 below shows the different states of the braking system. The braking system is initially inactive, when the brakes are applied or pedestrian is detected the system state transitions to an active state. Once the system is active the AEB and ABS are also activated, the distance parameter passed into the system will determine if the AEB will keep applying brakes. The ABS will communicate with the active system to check if speed is below 50kph.  If the parameter passed through applyBrakes is 0 and if pedestrians are not detected anymore then the active state transitions to an inactive state.

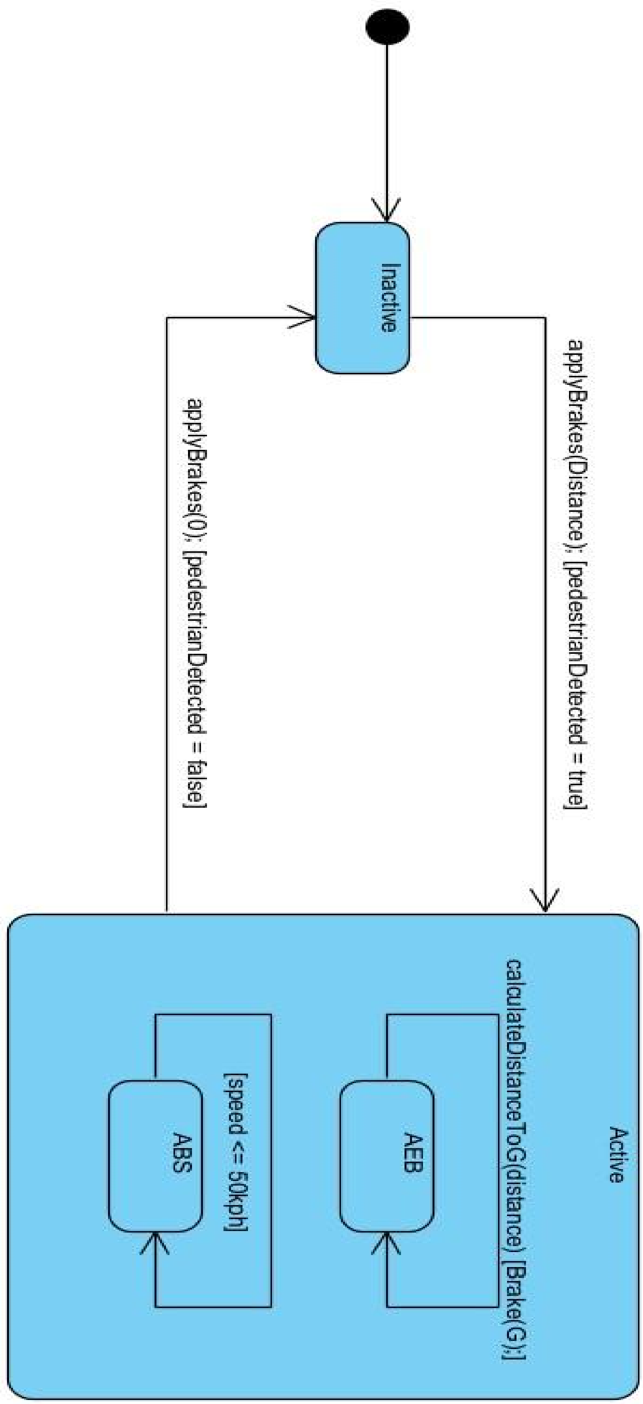



Figure 9: State Diagram for Braking System

### 4.4.3 Acceleration System

Figure 10 below shows the different states of the acceleration system. Initially the system is in an inactive state, but when an accelerate message is sent and there are no pedestrians detected the state transitions to an active state.  Once the system is active the states can either transition to reaccelerate or accelerate, if the previous speed is greater than 0 the state will transition to reaccelerate to the speed before needing acceleration. If the previous speed is 0 then the state will transition to accelerate to the speed it is wanted. Lastly if the system detects a pedestrian the state transitions to an inactive state.

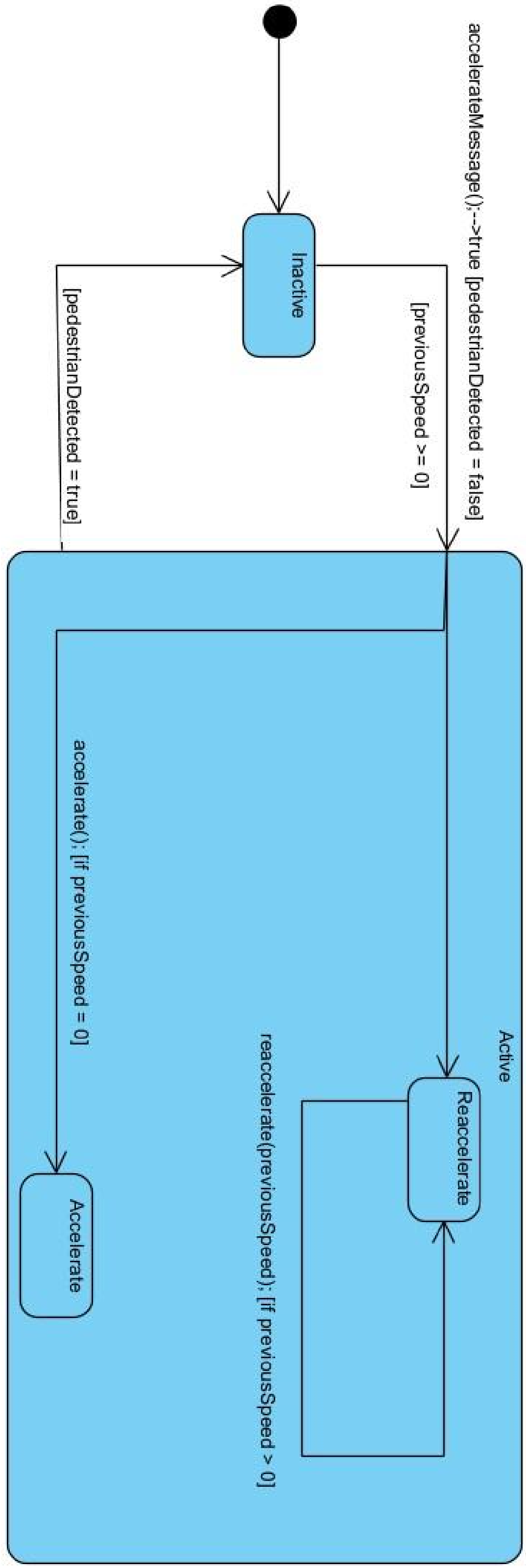



Figure 10: State Diagram for Acceleration System

### 4.4.4 Override Button

Figure 11 below shows the different states of the override system. Initially the system is in an inactive state. Once the override button is pressed, the override state is true and the system transitions to an active state where the PCAS is no longer functioning. The system will transition to an inactive state once the button is pressed again.

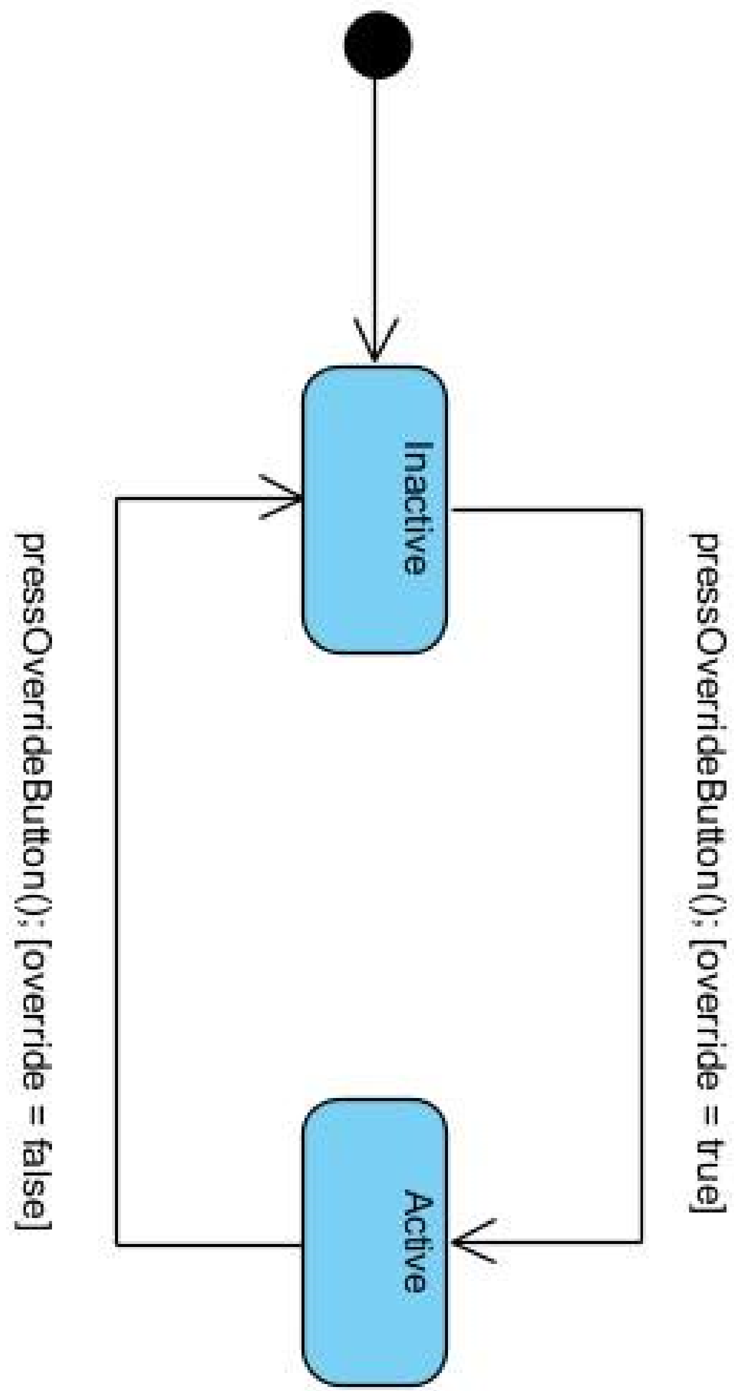
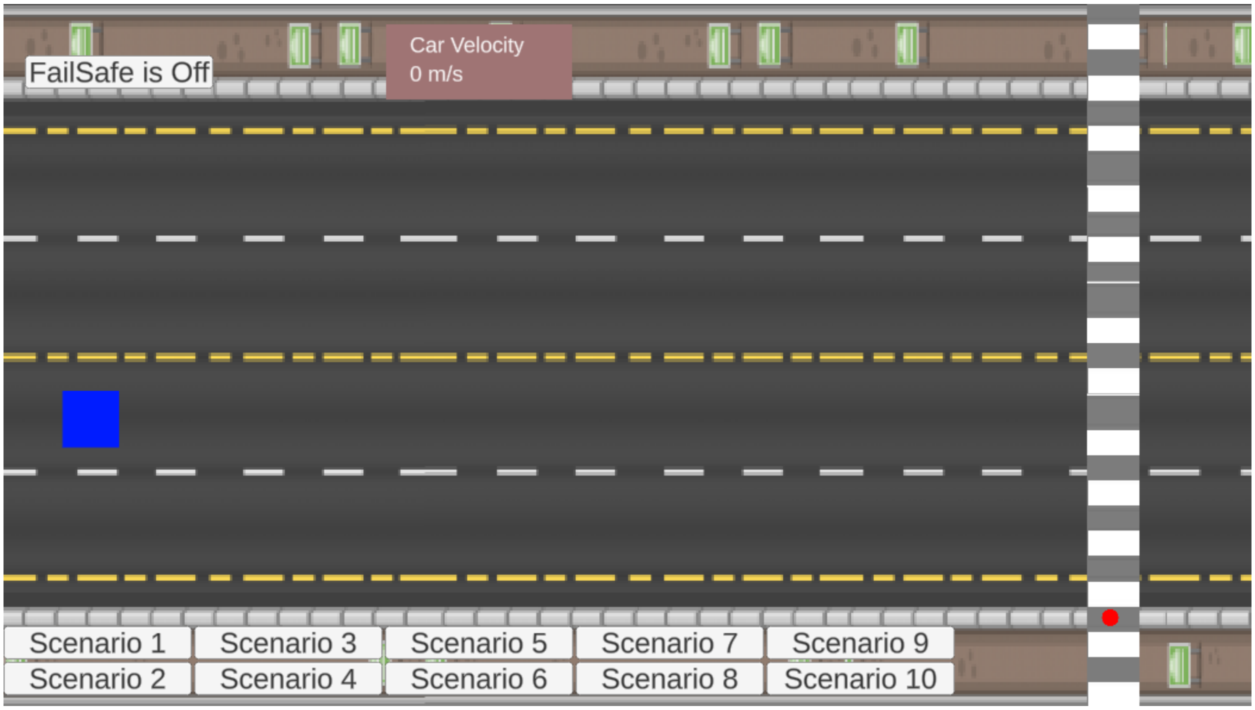



Figure 11: State Diagram for  Override Button

# 5 Prototype

As a proof of concept for the PCAS to be developed, we constructed a prototype framework demonstrating the features and functionalities of the PCAS system. This section introduces and demonstrates the prototype.




## 5.1 How to Run the Prototype

Select a scenario button from the bottom of the page. The vehicle (blue) and pedestrian (red) will move according to the preset scenario. Observe the interaction between the two, along with the reported statistics. The vehicle is the blue square shaped object. The pedestrian is the red circle located on the crosswalk.

## 5.2 Types of Scenarios

There are 3 different types of scenarios. In each scenario, the pedestrian acts differently, causing the car to respond appropriately. They are explained below in each subsection, and following this is a table representing these same scenarios.

Scenarios 1-4 are of the type “Moving then Stopped Pedestrian.”

Scenarios 5-7 are of the type “Static then Moving Pedestrian.”

Scenarios 8-10 are of the type “Static Pedestrian.”

### 5.2.1 Moving then Stopped Pedestrian

These scenarios can be seen in table 22. In these scenarios, the pedestrian begins a distance of -7 meters away from the vehicle. The pedestrian begins moving immediately but comes to a halt at a position of y meters away from the vehicle. This value y varies based on the selected scenario.

In scenario 2, the system makes the vehicle come to a full stop. This is because the system has determined that the pedestrian is capable of moving into the path of the vehicle at any moment. In order to bypass this, the driver should manually accelerate the vehicle.

| Moving Then Stopped Pedestrian | | | | |
|---|---|---|---|---|
| Scen # | Initial Position, Yi | End Position,Yf | Initial Speed | Final Speed |
| | (m) | (m) | (kph) | (kph) |
| 1 | -7 | 0 | 10 | 0 |
| 2 | -7 | -2 | 10 | 0 |
| 3 | -7 | -3 | 10 | 0 |
| 4 | -7 | -5 | 10 | 0 |

Table 22: Scenarios 1-4, Moving Then Stopped Pedestrian

### 5.2.2 Static then Moving Pedestrian

These scenarios can be seen in table 23. In these scenarios, the pedestrian begins a distance of -y meters away from the vehicle. The pedestrian begins moving after a delay of s seconds at a speed of 10 kph. These values for y and s vary based on the selected scenario.

| Static then Moving Pedestrian | | | | |
|---|---|---|---|---|
| Scen # | Initial Position, Yi | Delay Before Moving | Initial Speed | Final Speed |
| | (m) | (s) | (kph) | (kph) |
| 5 | 0 | 1.5 | 0 | 10 |
| 6 | -2 | 1.8 | 0 | 10 |
| 7 | -4 | 1.1 | 0 | 10 |

Table 23: Scenarios 5-7, Static then Moving Pedestrian

### 5.2.3 Static Pedestrian

These scenarios can be seen in table 24. In these scenarios, the pedestrian does not move. They remain stationary at a distance of y meters away. This value y varies based on the selected scenario.

In scenario 9, the system makes the vehicle come to a full stop. This is because the system has determined that the pedestrian is capable of moving into the path of the vehicle at any moment. In order to bypass this, the driver should manually accelerate the vehicle.

| Static Pedestrian | |
|---|---|
| | (m) |
| 8 | 0 |
| 9 | -2 |
| 10 | -4 |

Table 24: Scenarios 8-10, Static Pedestrian

## 5.3 Failsafe Mode

This button switches how the car interacts in each scenario. It switches between regular and failsafe mode (in reality triggered by hardware issues or inclement weather). In this mode, it changes the response time to reach a requested deceleration from 200 ms to 900 ms.

The visible result of this is a minor increase in the overall time it takes to decelerate.

## 5.4 Statistics

At the top of the prototype, you are able to see the current speed of the vehicle. The pedestrian is always either static or moving at 10 kph, so this value is excluded from the displayed statistics.

## 5.5 Prototype Constraints

There are several other factors occurring behind the scenes in this prototype. It accounts for the following information:

### 5.5.1 ABS / AES

The deceleration accuracy is +/- 2%

The response time to reach the requested deceleration is 200 ms normally, and 900 ms in failsafe mode.

The release time is 100 ms.

The maximum deceleration is 0.7 g, where 1g = 9.81 m/s^2

### 5.5.2 Vehicle

Normal steady state speed is 50 kph

Accelerating to steady state speed is .25 g

The vehicle is 2m wide and long, 1m in each direction from the origin

Always initially at steady state speed

Always moves along the X axis

Initially at x,y = 0m,0m

### 5.5.3 Pedestrian

Can change velocity with infinite acceleration
The size of the pedestrian is a circle with a 0.5m diameter
The pedestrian can only move at a right angle to the vehicle path
Initially at x,y = 35m,-7m

# 6 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.

# References

Cheng, Betty. "*Project Overviews and Team Roles*" Online. Michigan State University, 20 Oct. 2022, Michigan State University. Lecture.

Capaldi, C. (Ed.). (n.d.). Automated Vehicle Algorithm for Pedestrian Collision Avoidance Function.

Intellias Automotive (2018, November 29). *The ultimate sensor battle: Lidar VS Radar*. Medium. https://medium.com/@intellias/the-ultimate-sensor-battle-lidar-vs-radar-2ee0fb9de5da