Software Requirements Specification (SRS)

Active Park Assist (APA)

Authors: Amtullah Naalwala, Daniel Adu-Djan, Emma Hettinger, George Pauls, Joe Hayes

Customer: Ms. Eileen Davidson

Instructor: Dr. Betty Cheng

1 Introduction

The sections within this Software Requirements Specification document provide specifications of the various features for the Active Park Assist (APA) software. The APA is a system that allows a vehicle to automatically park itself in either a parallel or perpendicular parking space. The document first poses the purpose, scope and details of the vehicle. Following this, model related requirements such as domain models, use case diagrams, and state diagrams are included. Finally, research references and contact information for the course instructor have been included.

1.1 Purpose

The purpose of this SRS document is to provide the customer, Ms. Eileen Davidson and the Ford Motor Company, the requirements for the APA system. The intended audience for this document is also those who are familiar with software for automotive systems.

1.2 Scope

The Active Park Assist locates parking spots and parks in them. A benefit to parking autonomously is that it allows a user who fears parking in certain spots to have the vehicle do it. This reduces the fear users have of hitting an object if they were to be parking the vehicle themself. With regard to this benefit, the vehicle is able to autonomously park in both parallel and perpendicular parking spots. In order to find a parking spot, the APA must be able to detect that a car is on each side of the empty spot using its sensors and cameras. The APA is embedded in an automotive system. In order to park safely, it detects any obstacles and brakes before hitting them. During a parking maneuver, the vehicle will steer into the spot and shift the car into park. The system also gives the option to transfer the control of the vehicle to the Ford Pass application.

1.3 Definitions, acronyms, and abbreviations

This section defines some common acronyms and technical terms that will be used throughout the rest of the document.

APA: Active Park Assist is the feature for parking automatically that this project creates.
Park Control System: The system that coordinates the active park feature and sends commands to all other subsystems.
HMI: The Human Machine Interface is the interface where the customer initiates the parking event and sends any other inputs to the APA. It also displays information about the parking to the customer.
Ultrasonic Sensors: Ultrasonic sensors detect objects near the vehicle.
Powertrain Management Subsystem: This system controls the acceleration and gear shift of the vehicle.
Brake Control Subsystem: This system controls the braking of the vehicle.
Steering Control Subsystem: This system controls the vehicle steering.
Vehicle Position Subsystem: This system processes information from the cameras and sensors in order to find the vehicle's position and identify parking spots.

1.4 Organization

The remainder of this document describes the APA feature in more detail, starting with section 2 describing general features such as constraints, assumptions, and the users of the feature. Section 3 describes the requirements in detail, and section 4 models the system through use case diagrams, state diagrams, and domain models. Section 5 provides the instructions for running an early version of the prototype. Descriptions of various scenarios that could occur when using the APA feature are also provided in relation to the prototype. Sections 6 and 7 complete the document with references and contact information.

2 Overall Description

This document describes the purpose and functionality of the APA system. This consists of a self-parking vehicle that uses various subsystems to perform automatic parking maneuvers. The project also describes the intended users for the vehicle and the constraints of the environment for the system.

2.1 Product Perspective

Parking a vehicle can become a stressful situation for many drivers. Parallel parking can specifically be a great challenge. External factors in the environment such as pedestrians and other surrounding vehicles can introduce further complexity for the driver. The APA system is designed to automate the parking operation by automating vehicle maneuvering with minimal interaction. Furthermore, by automating the parking procedure, APA eliminates human driver error by relying on sensors to park without hitting anything.

There are both hardware and software constraints for the APA. For the hardware constraints, the vehicle will require its sensors and cameras to detect the vehicle's position and parking space. Software constraints include requiring any calculations to be nearly instantaneous, since all parts of the vehicle must respond immediately to any changes in the environment.

2.2 Product Functions

For the system to be activated, the driver must provide some input to the system through the Human Machine Interface (HMI). Once the system receives the driver's parking preference, the cameras and sensors will recognize nearby available spaces and return the data to the system to display on the HMI for the user to choose one. Once a parking space has been selected by the user, the system will utilize its cameras and sensors to determine the location of the vehicle relative to the parking space. The system will concurrently be calculating the location of the vehicle while controlling the steering functionality, the speed of the vehicle, and will also brake if needed. While parking, the system provides an option for the user to transfer control to the Ford Pass App. The objective is to automatically park the vehicle safely.

The automated parking procedure is controlled by the Park Control Subsystem. The Park Control Subsystem interacts with the Powertrain Management Subsystem, HMI, Brake Control Subsystem, Steering Control Subsystem, and the Vehicle Position Subsystem to successfully park the vehicle.

2.3 User Characteristics

The user of the APA system should have a valid driver's license from the respective country or state/region they are driving in. Users include any drivers of Ford vehicles, especially those who often park in crowded parking lots.

2.4 Constraints

For the system to work, all of the sensors must properly function for the vehicle to be able to detect an obstacle in its surrounding view. This is a very safety-critical application, since a failure could cause the vehicle to hit an object or a pedestrian. The APA system must be able to park the vehicle within 20 seconds of the start of the parking procedure. The APA system must check all inputs into it and verify that they are from the user otherwise it should shut down.

2.5 Assumptions and Dependencies

It is to be assumed that the hardware will contain working camera sensors, as well as a functioning automobile complete with transmission, among other components. For the environment, an assumption could be that there is no more than 1 empty space between cars, such that the system will exclusively detect open spots between other vehicles. For the user interactions, it is to be assumed that once they have opened the app, they are in search of, again, an empty spot contained between two inactive (occupied) parking spaces.

2.6 Approportioning of Requirements

As the system recognizes patterns in user input and familiarity in its environments, it is possible that the algorithm will continue to be optimized so that safety-wait periods could potentially be reduced, the parking maneuver itself could be performed at higher speeds to save time.

As improvements to the hardware technology within the system are introduced, the software programs related to such hardware technologies will require slight modifications and updates.

Additionally, some features were out of the scope of the requirements. These features include parking accommodations for bad weather conditions and validation of messages from the FordPass app.

3 Specific Requirements

This section enumerates the requirements for the APA system.

1. The system must detect available parking space and avoid any parking attempts for which there is not enough space.

The following rules must be followed to accomplish the requirement.

1.1 APA must only allow parking in a safe size spot within which a U.S. standard vehicle can park.

APA must acknowledge the following to determine a valid parking spot.

1.1.1 A space a minimum of 22 feet to 26 feet in length and 8.6 feet in width is a valid spot for parallel parking [2].
1.1.2 A space a minimum of 18 feet long and 9 feet wide is a valid parking spot for perpendicular parking.

1.2 The system must decide which available parking spots can be used.

2. APA must park the vehicle both horizontally and in parallel fashion based on customer input through the HMI.

3. APA must use the camera and lidar sensors on the vehicle to detect other cars or obstacles, moving or stationary, in real time, that obstruct the confirmed parking attempt.

3.1 If the system detects an obstacle and it is within 2 feet of the vehicle, the system should brake.

3.2 Sensors must continue to monitor the environment to make sure there are no obstacles.

4. APA must not take too long to provide a verdict on available parking spots and should complete the parking maneuver within a reasonable duration.

5. APA must seek verification from the driver about the parking spot identified, if any.

6. User must start APA through the HMI.

6.1 APA must verify that the request was initiated by the driver.

7. APA must accept user inputs only from the HMI, FordPass app
and brake pedal.

7.1 APA must receive controls from the FordPass app when the control is transferred to the app by the user.

8. APA must send all warnings to the HMI Subsystem.

9. APA must verify that sensor readings and other inputs are credible.

10. APA must always verify that all sensors are functional.

10.1 APA must detect and report defective sensors.
10.1.1 APA must halt the parking procedure if no functional backup sensors for defective sensors are available.

11. APA must detect potential issues and alert the driver on the HMI display, such as if the vehicle stops moving while under system control or any sensor fails.

12. APA must cancel, at any point, on request by the user, and return to human control.

12.1 The vehicle must be put in park position on cancellation.

13. APA must be able to maneuver forward or backward, brake, and steer the vehicle in order to get into the spot and avoid obstacles.

14. APA must react to changes in speed initiated by the customer though the brake pedal or the Ford Pass app.

14.1 APA must ensure vehicle speed is always less than 5 mph.

14.2 The vehicle must maintain a maximum speed by default during the parking maneuver.

14.3 The vehicle must adhere to braking by the customer during the parking maneuver.

15. APA must put the automatic transmission into "Park" position after parking.

16. APA must indicate to the customer that the parking process has completed.

17. APA must be inactive after indication to the customer that the process is completed.

18. APA must have a fail-safe state to revert to.

19. APA must acknowledge only the Ford Pass App as source of remote control.

19.1 APA must accept only authenticated requests or commands from the Ford Pass App.

4 Modeling Requirements

4.1 Use Case

Use-Case Diagram

The use case diagram in figure 1 shows how a driver interacts with the APA system. Everything inside the rectangle is part of the system. Each oval is a use case.

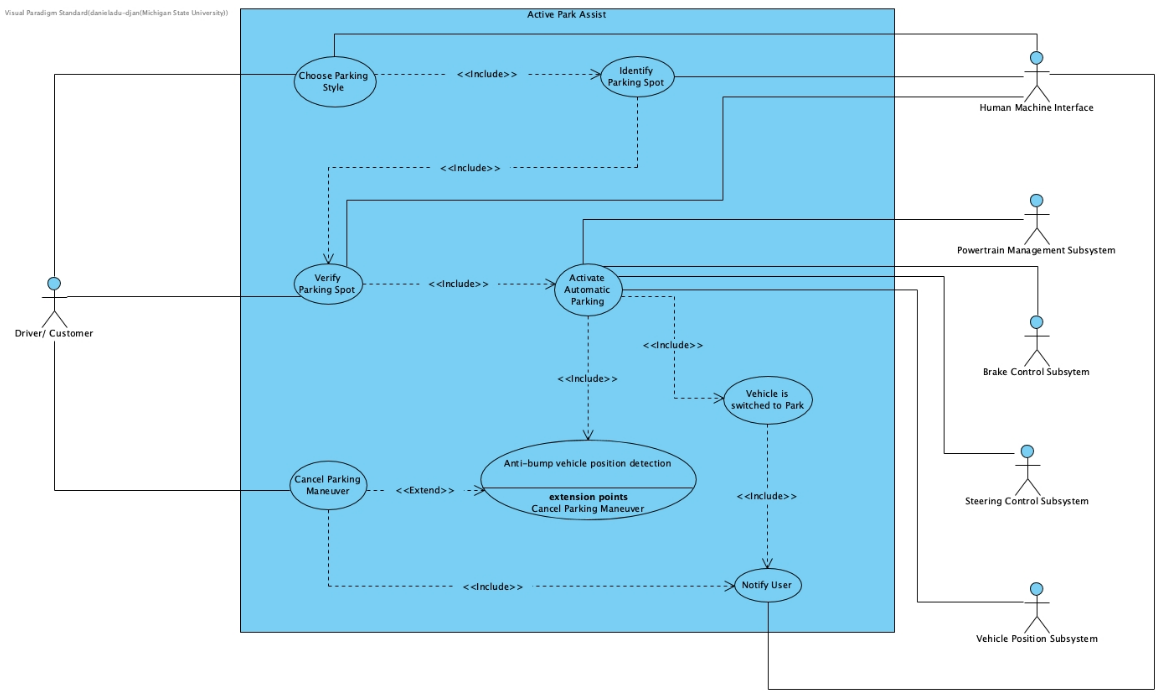

Figure 1: Use case diagram for a driver interacting with the Active Park Assist feature

Use-Case Descriptions

Use Case:	**Choose parking style**
Actors:	***User, HMI***
Description:	***The system asks the user to choose between a parallel or perpendicular parking spot style through the HMI. After a selection is made, the Identify Parking Spot use case is triggered.***
Type:	***Primary (essential)***
Includes:	***Identify Parking Spot***
Extends:	***None***
Cross-refs:	***1,2,6,7,8,9***
Use cases:	***None***

Use Case:	***Identify Parking Spot***
Actors:	***User, Vehicle Position Subsystem***
Description:	***The system identifies a spot in view using the sensors and the Vehicle Position Subsystem. The spots are then presented to the user through the HMI.***
Type:	***Primary (essential)***
Includes:	***Verify Parking Spot***
Extends:	***None***
Cross-refs:	***1,2,4***
Use cases:	***Choose Parking Style***

Use Case:	***Verify Parking Spot***
Actors:	***User, HMI***
Description:	***The user must verify that the spot the vehicle has identified is a valid parking spot on the HMI***
Type:	***Primary (essential)***
Includes:	***Activate Automatic Parking Maneuver***
Extends:	***None***
Cross-refs:	***5,7***
Use cases:	***Identify Parking Spot***

Use Case:	***Activate Automatic Parking Maneuver***
Actors:	***System, Powertrain Management Subsystem, Brake Control Subsystem, Steering Control Subsystem, Vehicle Position Subsystem***
Description:	***The system takes over control of the vehicle from the user. The system has control over the speed, brakes, transmission, and steering wheel. The system then attempts to park the vehicle for the user.***
Type:	***Primary (essential)***
Includes:	***Anti-Bump Vehicle Position Detection, Vehicle is Switched to Park***
Extends:	***None***
Cross-refs:	***2,12,13,14,15,18,19***
Use cases:	***Verify Parking Spot***

Use Case:	***Notify User***
Actors:	***System, HMI***
Description:	***The system will notify the user when the parking maneuver is completed, or when the system fails before it is able to complete the maneuver.***
Type:	***Secondary (essential)***
Includes:	***Vehicle is Switched to Park***
Extends:	***None***
Cross-refs:	***16,17***
Use cases:	***Cancel Parking Maneuver, Anti-Bump Vehicle Position Detection***

Use Case:	***Vehicle is Switched to Park***
Actors:	***System, Powertrain Management Subsystem***
Description:	***The system puts the vehicle in park if the parking maneuver is finished, if the system fails to complete the parking procedure, or if the customer canceled the parking maneuver.***
Type:	***Secondary (essential)***
Includes:	***Activate Automatic Parking Maneuver***
Extends:	***None***
Cross-refs:	***15,16,17***
Use cases:	***Cancel Parking Maneuver, Anti-Bump Vehicle Position Detection***

Use Case:	**Cancel Parking Maneuver**
Actors:	***User and System, HMI***
Description:	***The user is able to cancel the parking maneuver at any time during the parking maneuver. The user can only cancel the operation via the HMI.***
Type:	***Primary (essential)***
Includes:	***Notify User***
Extends:	***None***
Cross-refs:	***1,3,7,8,9,10,11,18,19***
Use cases:	***Activate Automatic Parking Maneuver***

Use Case:	***Anti-Bump Vehicle Position Detection***
Actors:	***System, Vehicle Position Subsystem***
Description:	***The system uses its sensors to detect any obstacles that come in the path of the vehicle during the parking maneuver.***
Type:	***Secondary (essential)***
Includes:	***Activate Automatic Parking Maneuver***
Extends:	***Cancel Parking Maneuver***
Cross-refs:	***1,3,6,9,12***
Use cases:	***Activate Automatic Parking Maneuver***

4.2 Domain Model

Domain Model

The UML domain model (Figure 2) depicts the class structures and associations needed to implement an object-oriented programming paradigm to provide an adequate solution to the APA system design challenge.

The main APA system is composed of several subsystems that all use features from the Park Control subsystem. The APA system also includes a class that represents environmental obstacles that may potentially be in the vehicle's path. The APA system also includes a class that represents features of the vehicles that will recognize objects and parking spots (i.e sensor, camera).

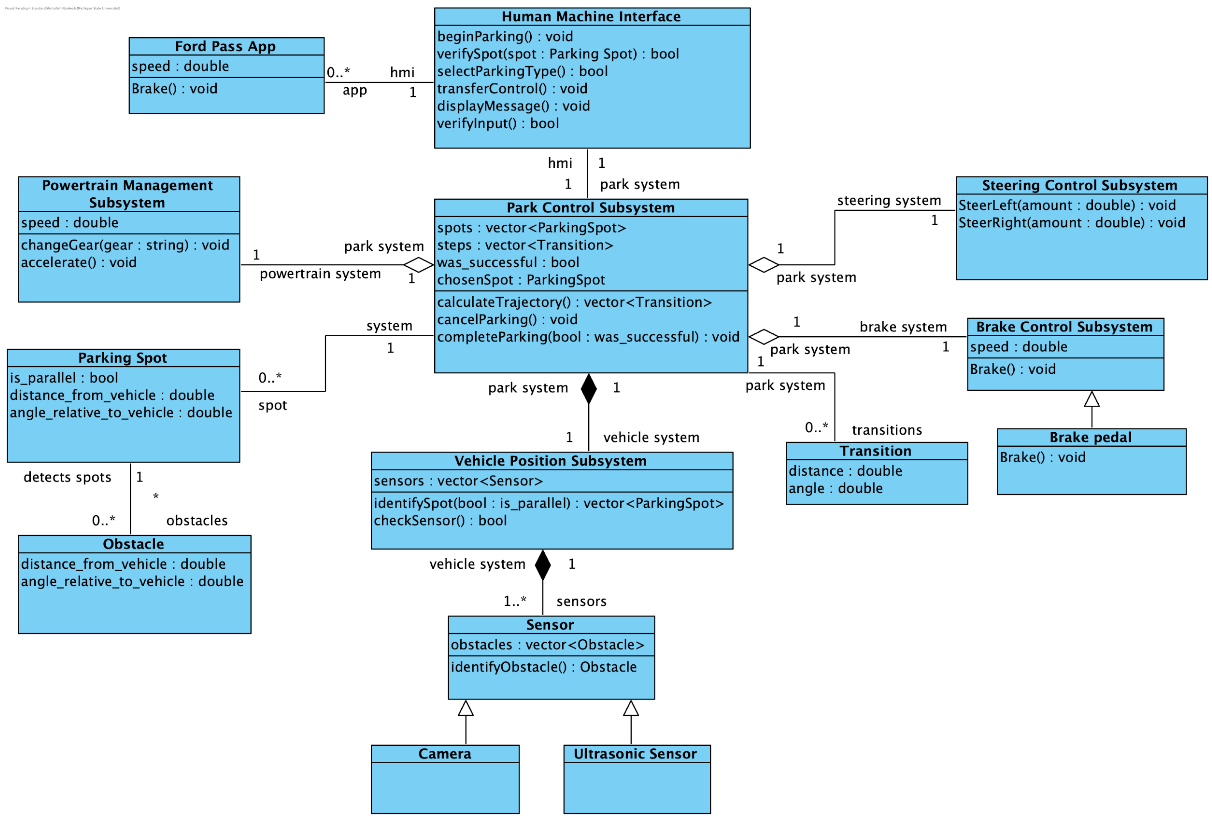

Figure 2: Model of object classes in the Active Park Assist feature

Data Dictionary:

Element Name		Description
Ford Pass App		A mobile application that can control the braking functionality of the vehicle once the customer transfers control to it.
Attributes		
	speed:double	Speed of the vehicle

Operations		
	Brake(): void	Allows the customer or system to apply the brakes to the vehicle and slow down the speed.
Relationships	Has an association with Human Machine Interface	

Element Name		Description
Brake Control Subsystem		System that is responsible for braking the car to keep it below a certain speed.
Attributes		
	speed : double	Speed of the vehicle.
Operations		
	Brake(): void	Allows the customer or system to apply the brakes to the vehicle and slow down the speed.
Relationships	Has an aggregate association with the Park Control Subsystem and is the parent class of the Brake Pedal class (generalization relationship).	

Element Name	**Description**

Brake Pedal		This is the pedal used to slow the vehicle down.
Operations		
	Brake(): void	Allows the driver or system to apply the brakes to the vehicle and slow down the speed.
Relationships	Inherited from the Brake Control Subsystem	

Element Name	Description
Camera	Responsible for transferring visual information to the system. It checks the surroundings for obstacles that may obstruct the parking process and checks for possible parking spots.
Relationships	Inherited from Sensor

Element Name		Description
Human Machine Interface		Responsible for receiving customer input, displaying warnings, and camera information.
Operations		

	beginParking(): void	Starts the parking process for the vehicle.
	displayMessage(): void	Displays a series of messages to assist users with parking selection. Also displays warnings when the system encounters an issue.
	verifySpot(spot: Parking Spot) : bool	Checks to see if the customer verifies the spot found by the system. Returns true if the customer verifies it.
	selectParkingType(): bool	A Boolean that controls what kind of parking style the customer needs. The two choices are parallel and perpendicular. If return true then it is parallel.
	transferControl(): void	Allows the customer to choose if they would like to transfer control to the Ford Pass App.
	verifyInput(): bool	Checks if the inputs given to the system are from the customer instead of an outside entity.
Relationships	Association with Ford Pass App and Park Control Subsystem	

Element Name	**Description**
Obstacle	An object that a sensor has identified as being in the way of the parking maneuver.

Attributes		
	distance_from_vehicle: double	The x position of the obstacle. The distance from the vehicle.
	angle_relative_to_vehicle: double	The y position of the obstacle. The angle relative to the vehicle.
Relationships	Association with Parking Spot	

Element Name		**Description**
Park Control Subsystem		The system responsible for accepting customer inputs from the HMI and issuing commands to other subsystems.
Attributes		
	spots: vector<Parking Spot>	List of available parking spots
	chosenSpot: ParkingSpot	The parking spot that has been chosen and verified by the customer.
	steps: vector<Transition>	
	was_successful: bool	Determines whether parking was able to be successfully completed.
Operations		

	calculateTrajectory(): vector<Transition>	Finds the trajectory needed to get the vehicle in the parking spot.
	completeParking(bool: wasSucessful) : void	Finds out if the parking maneuver has been completed. Takes in a boolean called wasSuccessful to check if parking was successful.
	cancelParking(): void	Cancels the parking maneuver if the system can not complete parking for whatever reason.
Relationships	Association with Human Machine Interface and Parking Spot. Aggregation relationship with Steering Control Subsystem, Brake Control Subsystem, and Powertrain Management Subsystem. Composition relationship with Vehicle Position Subsystem.	

Element Name		**Description**
Parking Spot		The spot that the vehicle can park in.
Attributes		
	distance_from_vehicle: double	The x position of the parking spot. The distance from the vehicle.
	angle_relative_to_vehicle: double	The y position of the parking spot. The angle relative to the vehicle.
	is_parallel	Returns true if the vehicle will be parallel parking. Returns false if the vehicle will be parking in the default perpendicular parking mode.
Relationships	Association with Obstacle and Park Control Subsystem	

Element Name		**Description**
Powertrain Management Subsystem		Responsible for accelerating the vehicle and selecting the gear level position necessary for completing the required trajectory.
Attributes		
	speed: double	The speed of the vehicle.
Operations		
	changeGear(gear: string): void	Changes the gear of the vehicle so it can complete the parking maneuver.
	accelerate(): void	Accelerates the vehicle to the speed needed to complete the parking maneuver.
Relationships	Association with Park Control Subsystem	

Element Name	**Description**
Sensor	Sensors used in the vehicle. Includes cameras and ultrasonic sensors. Used to check for available spaces and if any obstacles get in the way of the parking maneuver.

Attributes		
	obstacles: vector<Obstacle>	List of potential obstacles near the vehicle.
Operations		
	identifyObstacle(): Obstacle	Identifies an obstacle that is in the way of the vehicle.
Relationships	Inherited by Ultrasonic Sensor and Camera	

Element Name		**Description**
Steering Control Subsystem		Responsible for steering the vehicle to make the required trajectory needed to complete the parking maneuver.
Operations		
	steerLeft(amount: double): void	Allows the vehicle to steer left.
	steerRight(amount: double): void	Allows the vehicle to steer right.
Relationships	Aggregation association with Park Control Subsystem	

Element Name	**Description**
Ultrasonic Sensor	Ultrasonic sensor used to determine the distance between the vehicle and an obstacle.
Relationships	Inherited from Sensor

Element Name		**Description**
Vehicle Position Subsystem		Responsible for processing data from the vehicle’s sensors in order to identify parking spots and verify the position of the vehicle throughout the parking maneuver.
Attributes		
	sensors: vector<Sensor>	A vector list of all the sensors on the vehicle.
Operations		
	checkSensor(): bool	Checks to see if a sensor is working correctly and sending reliable data to the system.
	identifySpot(bool: isParallel) : vector<Parking Spot>	Finds an available parking spot and adds it to a vector of parking spots and returns it.

Relationships	Aggregation association with Park Control Subsystem and composition association with Sensor

4.3 Scenarios

This section includes example vehicle maneuver scenarios based off of examples provided by the client, Ford Motor Company. These example scenarios include instances such as a standard park procedure, when a sensor has failed, and when an instruction from the Ford Pass App is not a legitimate instruction for the vehicle to follow.

Scenario 1: This scenario depicts the “Standard Park Procedure” where a user would follow steps to perform a typical parking maneuver. The user activates the APA system through the HMI and the system is fully operational. The user then chooses their parking type. The system then finds a parking spot and has the user verify it via the HMI. The system then attempts the parking procedure and completes it with no obstacles encountered.

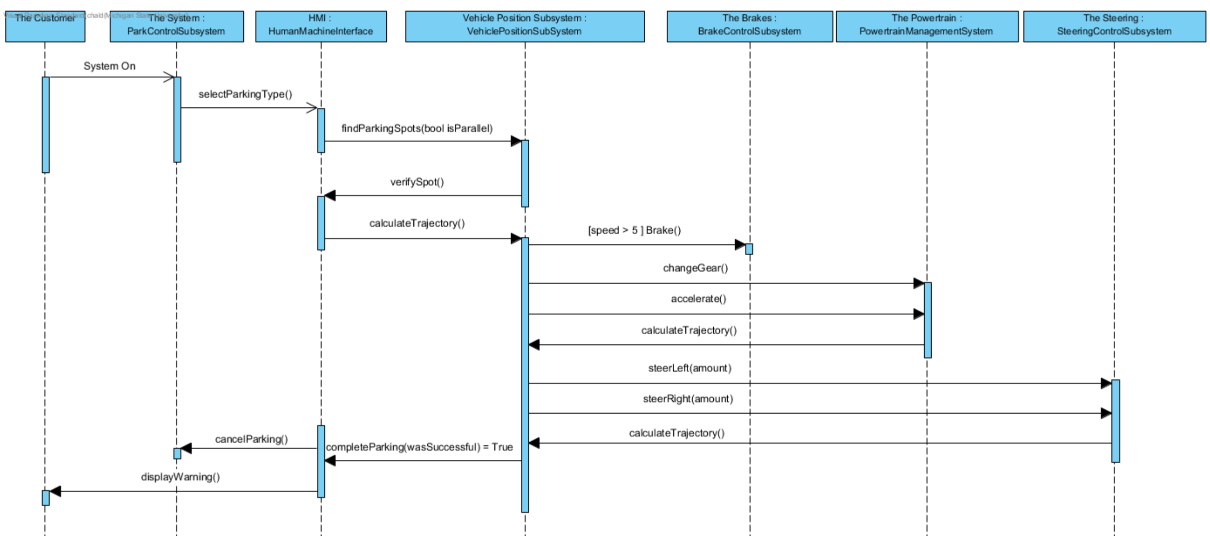

Figure 3: Sequence diagram showing a scenario where the system completes the parking maneuver with no interruptions.

Scenario 2: This scenario depicts what occurs when a sensor fails as the vehicle is attempting to park. The user activates the APA system through the HMI and chooses what parking type they would like via the HMI. The system then checks all of its sensors to see if they are all working correctly. The system also proceeds to verify the parking spot. If the vehicle detects that one or more of the sensors is not functional, the APA system notifies the user by displaying a warning. The parking procedure is canceled by turning off the system and prompting the user to return to the main menu.

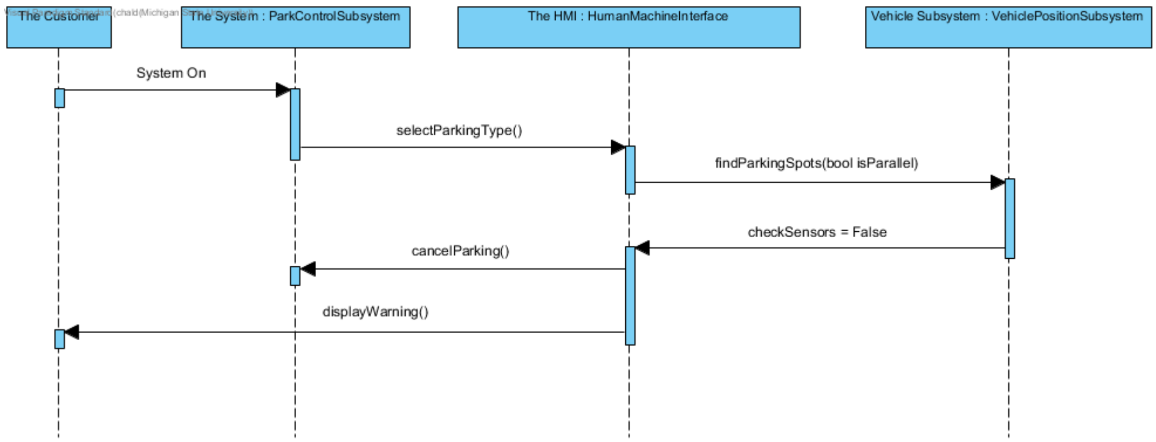

Figure 4: Sequence diagram for a scenario where a fault occurs during parking

Scenario 3: The customer activates the APA system through the HMI. They are then prompted by the HMI for which parking type they need. A parking spot is then found by the Vehicle Position Subsystem and the user verifies it via the HMI. The customer then chooses to use the app to complete the parking procedure. The system then checks each input to determine if the inputs are from the driver and not an outside entity. If one of the inputs is not from the customer then the parking procedure is cancelled and control is given back to the customer.

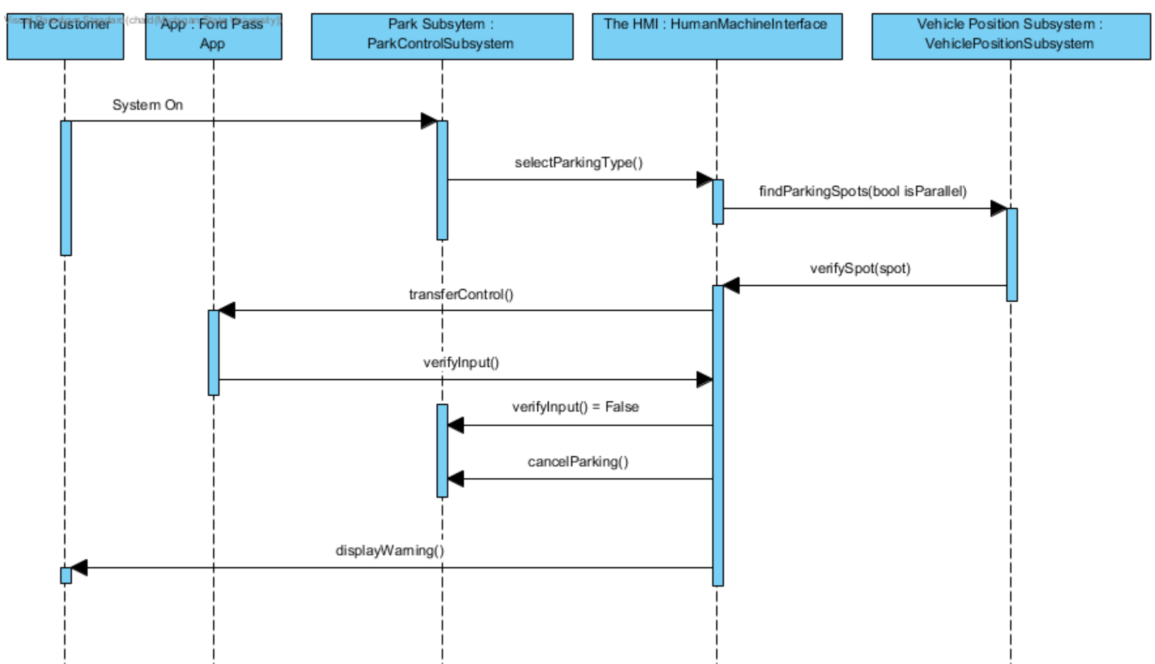

Figure 5: Sequence diagram for a scenario where the input from the Ford Pass App is illegitimate

State Diagram

The state diagrams represent possible states of each component of the system. The diagram incorporates the major phases of the parking maneuver and focuses on describing the various transition logic and conditionals for each state.

The states are represented by the blue boxes. State transitions are represented by one-way arrow lines that are labeled with a transition clause. The initial states are marked by black circles.

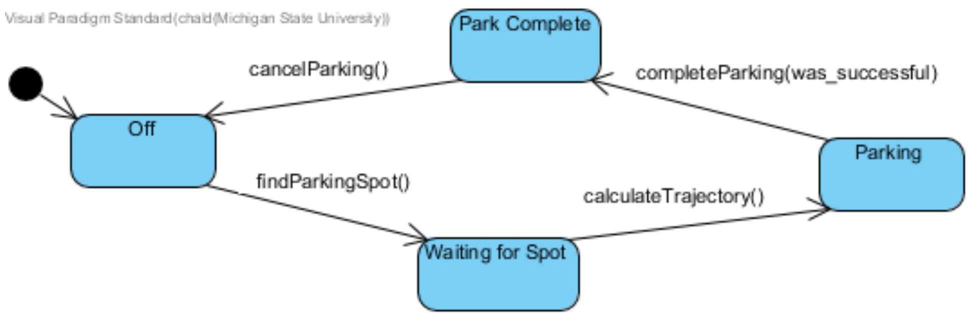

Figure 6:State diagram for the Park Control Subsystem

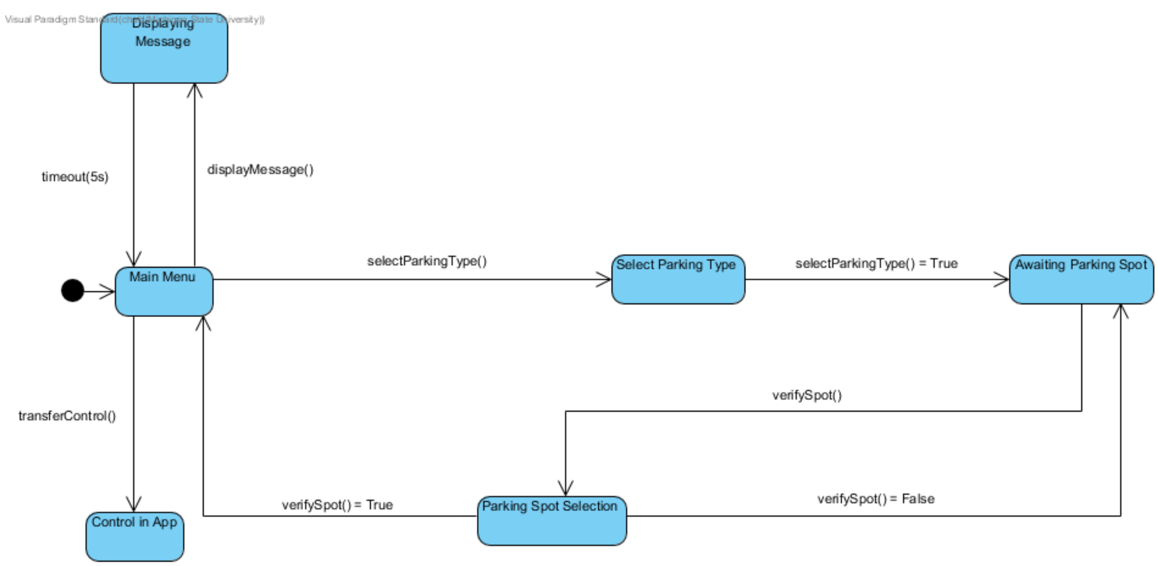

Figure 7: State diagram for the Human Machine Interface

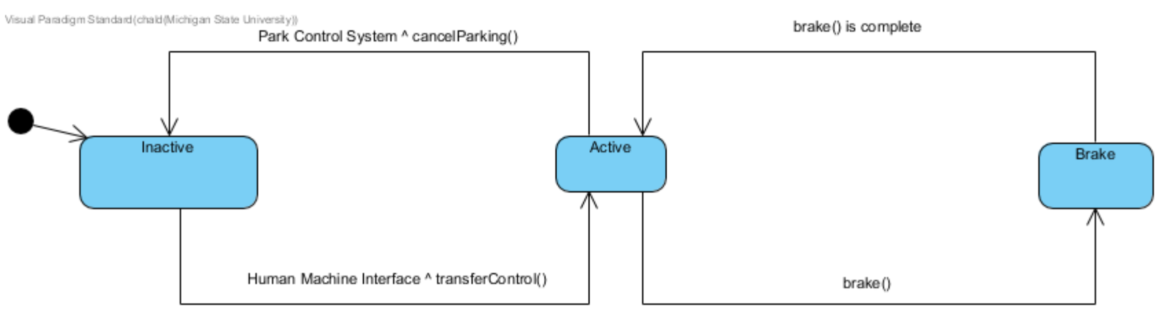

Figure 8:State diagram for the Ford Pass App

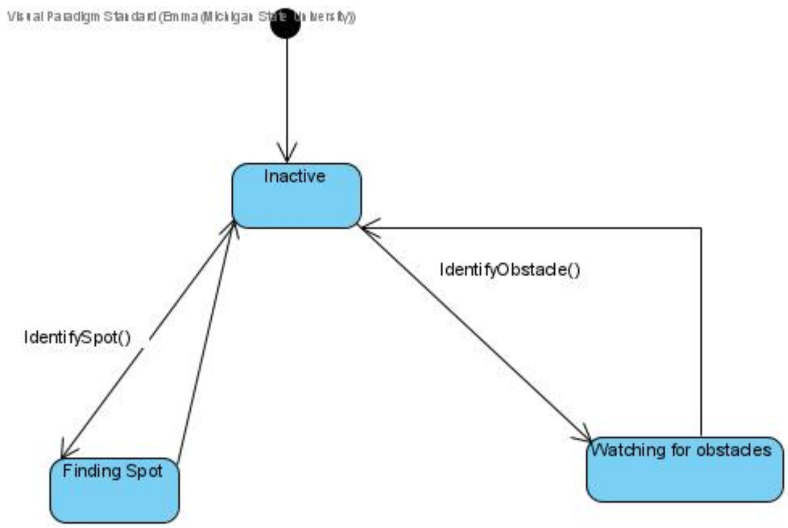

Figure 9: State diagram for the Vehicle Position Subsystem

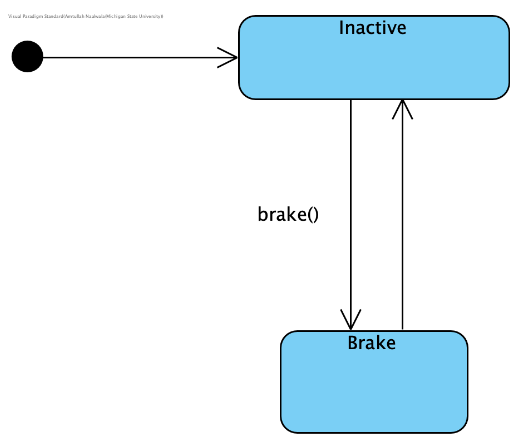

Figure 10: State diagram for the Brake Control Subsystem

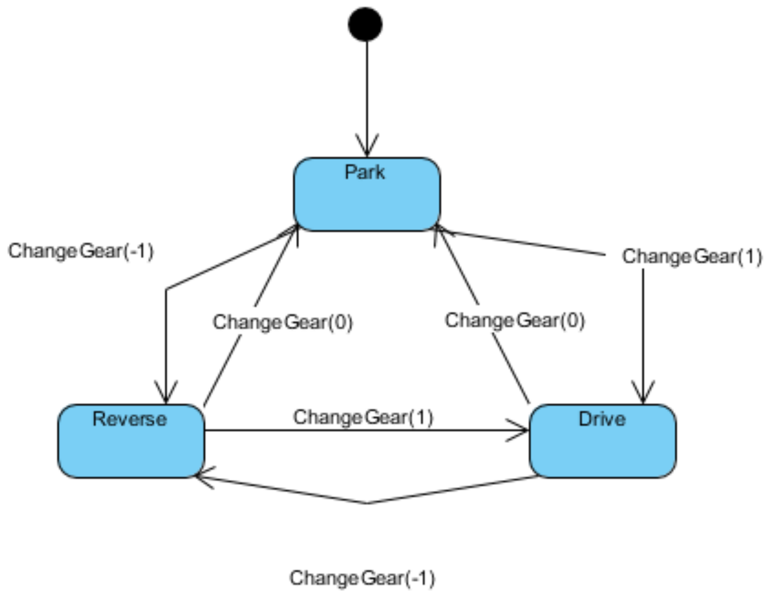

Figure 11: State diagram for the Powertrain Management Subsystem

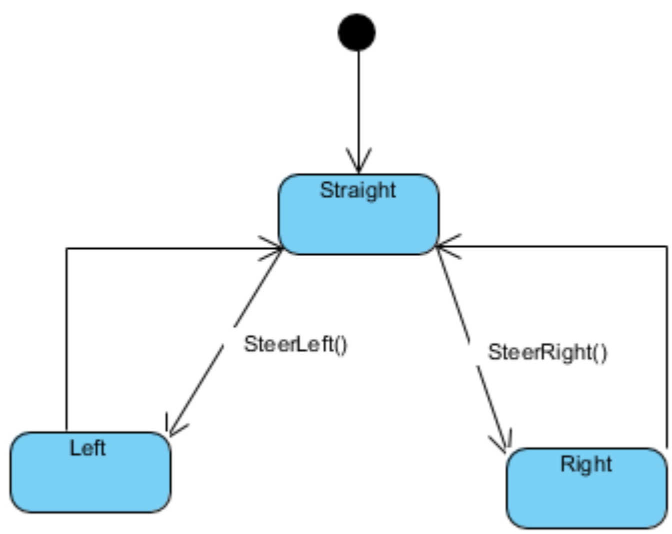

Figure 12: State diagram for the Steering Control Subsystem

5 Prototype

5.1 How to Run Prototype

- Any modern web browser is acceptable to run the prototype.
- The prototype is accessible through the web.
- The prototype can be found at: https://apa3.msu.luzcode.com/prototype

5.2 Sample Scenarios

Scenario 1: No Spots Found
This scenario depicts a situation where the user is displayed "no spots found" upon using the prototype. As seen in Figure 13, the user selects "Start APA" in the main HMI display, upon selection, the user is prompted to select a parking style - perpendicular or parallel (Figure 14). The system then searches for any nearby available parking spaces and when it cannot find one, it displays to the user that there are "no available parking spaces" (Figure 15).

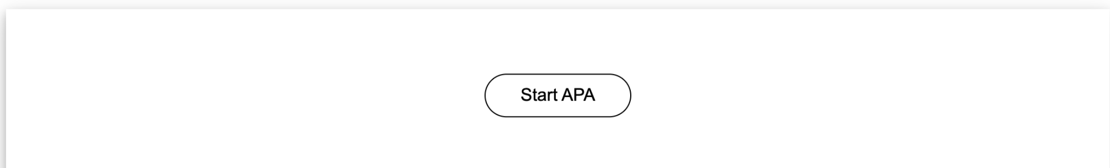

Figure 13: User selects start APA

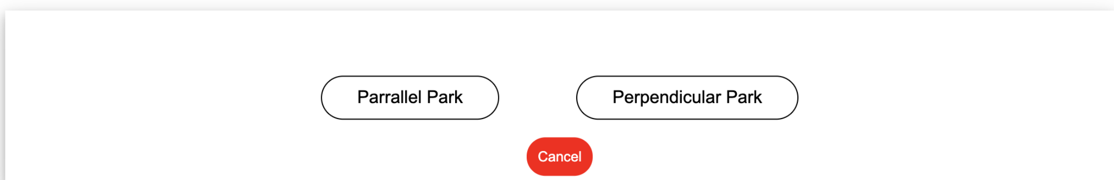

Figure 14: User is prompted to select either parallel or perpendicular

Figure 15: System indicates no parking spots available

Scenario 2: Successful Parking
This scenario depicts a situation where the user is displayed “successful parking” upon using the prototype. As seen in Figure 13, the user selects “Start Parking” in the main HMI display. Upon selection, the user is prompted to select a parking style - perpendicular or parallel (Figure 14). The system then searches for any nearby available parking spaces and the user is given the option to verify an available parking space from the list of nearby open spots (Figure 16). The system asks the user to select between staying in the vehicle or transferring control to the Ford Pass app (Figure 17). Once the vehicle completes parking in the spot without interruptions, the vehicle displays “Parking Successfully Complete” and prompts the user to return to the start of the HMI menu (Figure 18).

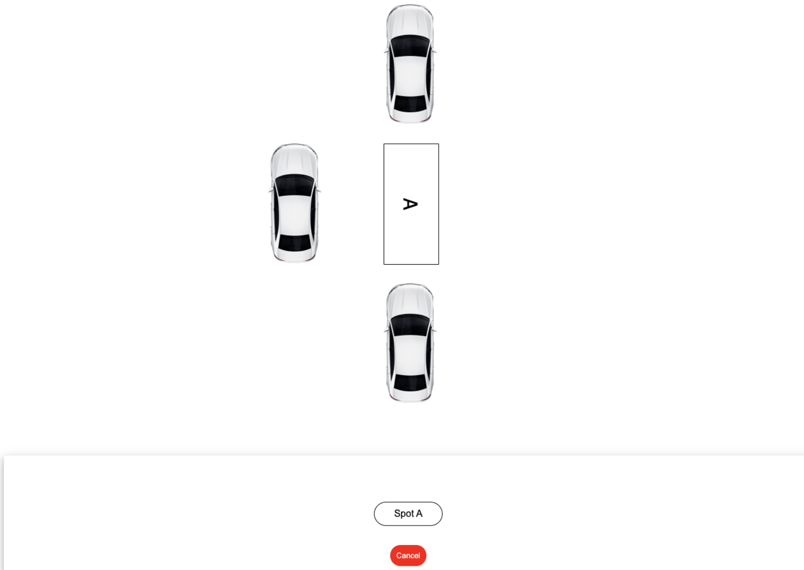

Figure 16: System prompts user to select from available spots

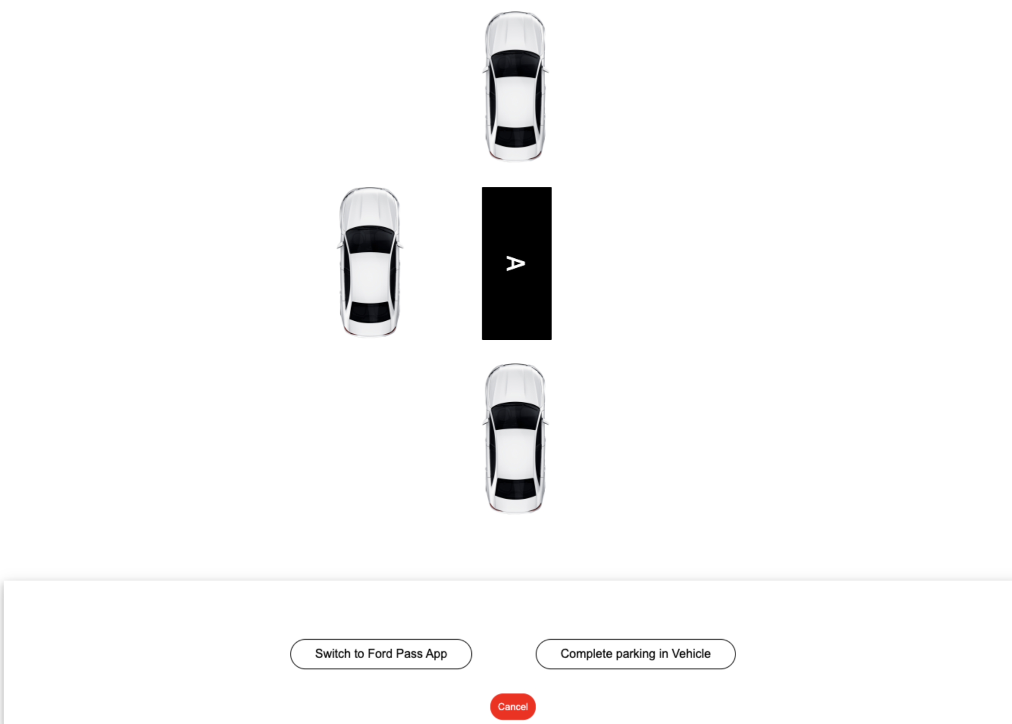

Figure 17: System prompts user to select between switching to ford pass app or complete parking in vehicle

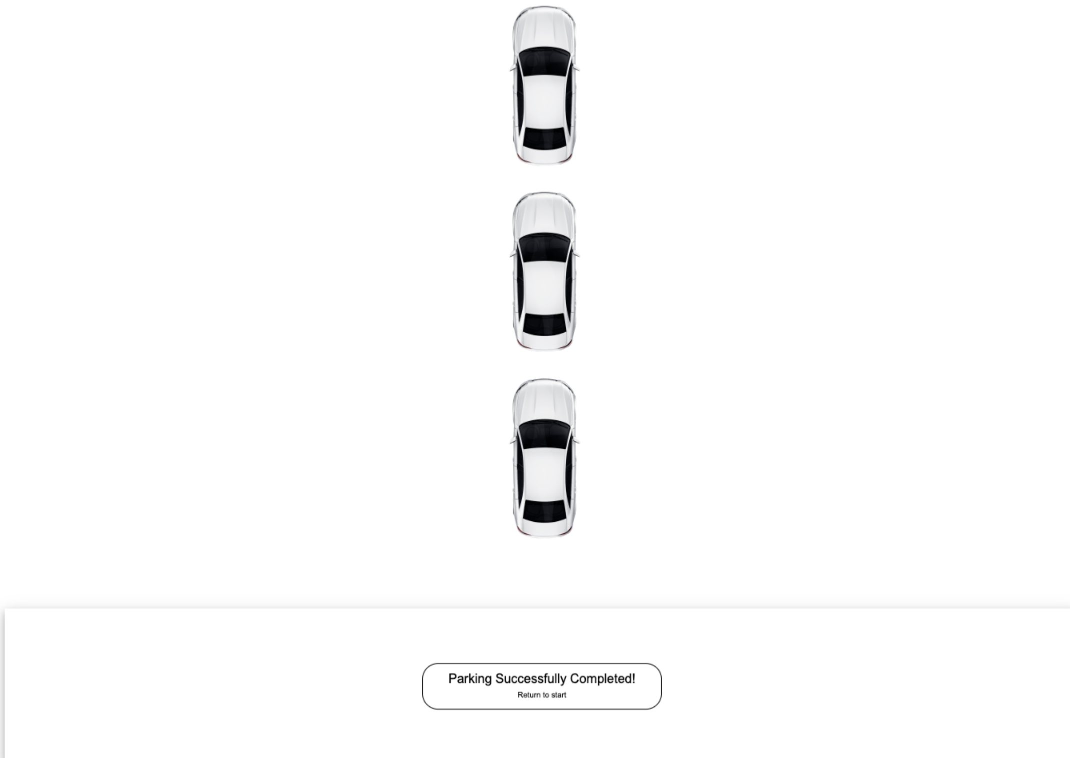

Figure 18: Parking successfully complete message

Scenario 3: Interrupted Parking
This scenario depicts a situation where the vehicle is interrupted during its parking process. As seen in Figure 13, the user selects “Start Parking” in the main HMI display. Upon selection, the user is prompted to select a parking style - perpendicular or parallel (Figure 14). The system then searches for any nearby available parking spaces and the user is given the option to verify an available parking space from the list of nearby open spots (Figure 16). The system allows the user to select between staying in the vehicle or transferring control to the Ford Pass app (Figure 17). The system detects that the chosen spot is now unavailable to park in potentially if a pedestrian walks into the spot. The system displays an error message indicating “Parking Unsuccessful.” This is represented in Figure 19. The system then returns control of the vehicle back to the user and returns to the main parking menu.

Figure 19: System detects obstacle and indicates to user that it cannot safely park

6 References

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

[2] "Parking Spaces Dimensions & Drawings." RSS, www.dimensions.com/element/parking-spaces [Accessed: October 10, 2020].

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.