# Software Requirements Specification (SRS)

# Project Active Park Assist: Team 4

**Authors:** Elizabeth Stevens, Chris Cardimen, Cody Lowen, Shaheer Hasan
**Customer:** Ms. Eileen Davidson, Ford Motor Company
**Instructor:** Dr. Betty H C Cheng, Michigan State University

## 1 Introduction

The Ford Motor Company is an American based automotive manufacturing company. Each year new features are developed, tested, and rolled out into vehicle manufacturing. As high capability hardware has become more available, additional autonomous features can be developed and integrated. The Active Park Assist Feature (APA) is an autonomous feature that will allow drivers to have a hands-off experience while the vehicle autonomously parks itself in either parallel or perpendicular parking spaces. The topics outlined in this document include the purpose and scope of the project as well as definitions of terms that are widely used throughout the document. This document will also provide details such as the product perspective, functionality, user characteristics, and limitations as well as assumptions about the environment and approportioned requirements. These requirements are hierarchically structured and modeled based on user cases and respective scenarios. This document also contains information about two executable prototype versions that visually represent different scenarios and system responses of the Active Park Assist Feature. For more information, references and a point of contact are given at the end of this document.

## 1.1 Purpose

The purpose of the Software Requirements Specification (SRS) document is to outline and model the requirements of the Active Park Assist feature. This document is intended to provide more information about the feature for the further development of a system and in general for users or stakeholders to review.

## 1.2 Scope

The Active Park Assist feature is a system that is integrated into the operating system of an automotive vehicle as embedded software. This feature allows the vehicle to autonomously park itself in either a parallel or perpendicular parking space by the discretion of the driver through either the HMI (Human Machine Interface) or the FordPass App. The objective of this feature is to develop a system that uses built in safety technology for allowing a vehicle to autonomously park with very minimal driver interaction. The benefits of this system allow parking maneuvers to be completed in an accurate and safe manner while promoting safer driving by relieving the possibility of driver error.

When the Active Park Assist Feature is enabled, the driver will be able to choose an appropriate parking spot from a list of available parking spots nearby and have the choice to either stay or exit the vehicle before the completion of the parking maneuver. During this maneuver, the driver will also have the ability to regain full control of the vehicle again through the braking system or steering wheel. If the driver chooses to exit the vehicle, the maneuver can be monitored through the FordPass App with the ability to manipulate vehicle speed remotely. The system will use the hardware safety equipment to monitor surroundings while directing the vehicle into the desired spot. In the case that the safety equipment experiences failure or an obstruction was encountered during the maneuver the system will stop the vehicle until the driver presses the brake and is again given full control of the vehicle. If the parking maneuver is completed successfully the vehicle is placed into parking gear and the feature is then disabled.

## 1.3 Definitions, acronyms, and abbreviations

*Prototype-User:* Person who is using the prototype
*User* : Anyone who is operating the autonomous parking feature.
*Driver* : The human who is physically operating the automobile.
*Active Park Assist* : The defined name of the autonomous parking feature.
*HMI* : Acronym for the Human Machine Interface which is a touch screen available in the vehicle used for the Active Park Assist feature activation and interaction.
*Ford Pass Application* : The mobile device application that can be synchronized with a vehicle for use of the Active Park Assist feature remotely.
*Ultrasonic Sensor* : Safety equipment built on to the vehicle for detecting objects in the vehicle's surroundings.
*APA:* Active Park Assist.

## 1.4 Organization

The following Software Requirements Specification (SRS) document outlines seven sections that outline the capabilities and limitations as well as the functionality of the Active Park Assist System. The purpose, scope, terminology and organization will be defined in Section 1 of this document. Section 2 will provide an overall description of the product such as the system's functionality, user characteristics, limitations and stated assumptions. Section 3 provides a hierarchical structure of enumerated requirements in accordance with the customer as we see are modeled in specific use cases and scenarios in Section 4. Section 5 is a description of two prototype versions to present the product's use cases and scenarios. A list of references and a point of contact is also provided at the end of this document in Section 6 and Section 7.

# 2 Overall Description

The following subsections describe the product in terms of functionality, user, constraints, and assumptions. This will give a better understanding of how the product is to be used in a real life situation and how it works.

## 2.1 Product Perspective

The active park assist system is a compilation of physical and software components that will assist the user by parking the vehicle autonomously. The user of this system can activate the APA via the HMI or the FordPass app. This system makes use of subsystems such as the braking, powertrain, camera, ultrasonic sensors, throttle control, and steering systems.

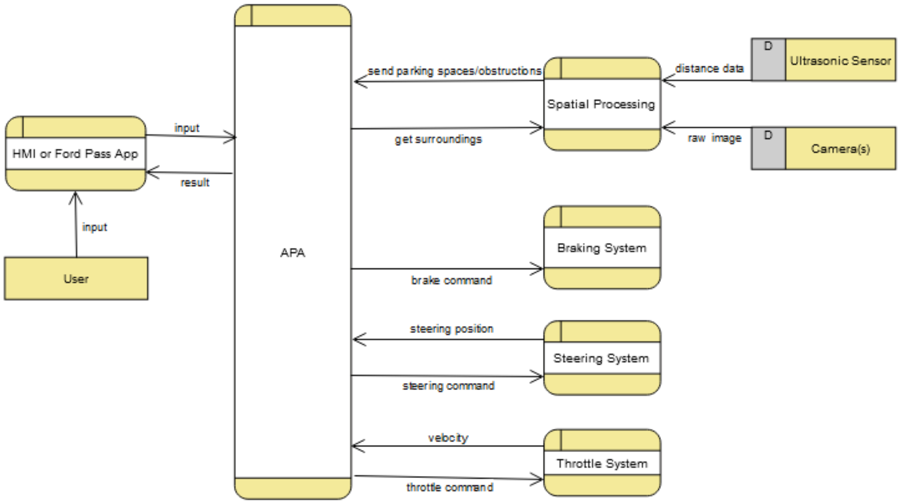


Figure 2.1: Data Flow Diagram

**The system also has the following interface constraints:**

**System Interfaces:** If any system required for the APA to operate fails or malfunctions the system will be non functional.

**User Interfaces:** Only the vehicle's operator can initiate and interact with the APA system.

**Hardware Interfaces:** Ultrasonic sensors, throttle, steering, brake, cameras, and the vehicle's on-board computer.

**Software Interfaces:** The cars dashboard which will act as the HMI, as well as the Ford Pass app.

## 2.2 Product Functions

There are many major functions that will be performed when using the app. A user will be able to activate and interact with the Active Park Assist feature within the vehicle or upon exit from the vehicle through the FordPass App. First, a dialog notification will appear on the screen to indicate if a parking spot is available. If it is, the user will have the option to decide whether the car should park parallel or perpendicular. Depending on the situation, there might only be one of these options; but if applicable, both will appear. If there is at least one parking spot available it will appear on the screen to indicate that the spot is available. The user will then be able to click the parking spot to begin the parking maneuver.

While the car is parking, the user will also have the ability to control the speed of the car as well as the brakes. Additionally, there are multiple different cameras and sensors that monitor the vehicle’s position and surroundings when parking. If the user does not feel comfortable or a human recognizable inconvenience is encountered during the autonomous parking, there is also an option that stops the parking at their convenience by either pressing the foot brake or turning the steering wheel.

## 2.3 User Characteristics

A typical user of this system must have valid government issued documentation such as a driver's license in order to legally operate a vehicle. The user must be familiar with the general operation of an automotive vehicle and will be knowledgeable about how to operate using generalized equipment that can be found in every vehicle such as the transmission, brake, and steering wheel. It is expected that a typical user also has a sufficient background in driver education and can manually perform parallel and perpendicular parking. The user must also be familiar with using Automatic Transmission but does not necessarily need to be familiar with Manual Transmission. The user should have general expertise and common knowledge about safety precautions and state laws and can actively acknowledge safety concerns as they arise during the operation of a vehicle. All users are expected to navigate a touchscreen and must be familiar with interpreting prompts when interacting with touchscreen technology.

## 2.4 Constraints

1) Regulatory policies: The government has forced restrictions on some activities with the APA system that may not be permitted in some states.
2) Hardware limitations: Cameras and sensors will be up to date with the latest vehicle models. Some features may not be compatible with older vehicle versions at Ford Motor Company.
3) Interfaces to other applications:
    a) Park Control Subsystem: masters the Active Park Assist feature. It accepts the customer input from the HMI subsystem, calculates the vehicle trajectory based on information from the Vehicle Position Subsystem, and issues commands to the other subsystem.
    b) Powertrain Management Subsystem: accepts inputs from the Park Control Subsystem to accelerate the vehicle and select the gear lever position in order to meet the required trajectory.
    c) HMI Subsystem: accepts customer inputs, displays camera information, and handles telltales / warnings.
    d) Brake Control Subsystem: accepts inputs from the Park Control Subsystem to brake the vehicle in order to meet the required trajectory.
    e) Steering Control Subsystem: accepts inputs from the Park Control Subsystem to steer the vehicle in order to meet the required trajectory.
    f) Vehicle Position Subsystem: processes data from the vehicle's cameras / radar in order to identify parking spots and verify vehicle position throughout the duration of a parking event.
    g) Customer Cell phone: allows the customer to control the parking maneuver remotely. [2]
4) Criticality of the application: In case of an emergency, dial 911 or your local number immediately.
5) Safety and security considerations: Although the APA is built to park a vehicle, its top priority is to ensure the safety of the passengers inside the vehicle and pedestrians around the vehicle.

## 2.5 Assumptions and Dependencies

- The driver will have the ability to interact with the APA system via the HMI and or the FordPass application at all times during the parking procedure and must be in close proximity to the vehicle when using the FordPass application.
- All hardware necessary to implement the APA system as defined will be selected and implemented in the vehicle design by Ford Motor Company.
- Hardware chosen by Ford Motor Company must possess the ability to send and receive data in a timely manner to assure safety.
- Ford Motor Company will encrypt all updates sent to both the APA system and any other systems within the vehicle to ensure security of the APA system.
- Ford Motor Company will only use electronic steering racks and steering control systems in vehicles which implement APA.
- The system will operate in any weather condition as long as ultrasonic sensors and cameras are not obstructed or malfunctioning.
- If the parking spot is perpendicular the spot only exists as perfectly perpendicular to the immediate road. Similarly a parallel parking spot exists perfectly parallel to the immediate road.
- The vehicle will not have any hitch attachments for the APA system to be operational.
- The vehicle's APA system will have the ability to adapt to vehicle modifications such as increase or decrease in tire size.
- The APA system will not be capable of parallel or perpendicular parking on the opposing side of an identifiable street with double yellow lines.
- All hardware required by the APA system must be operational for the APA system to be activated and continue to operate.
- The vehicle has systems in place to detect the presence of the driver and their passengers in the vehicle, and whether or not their seat belt is on. If APA is activated via the HMI both the driver and passengers must have their seat belts on if local, state or federal law dictates they are required for vehicle operation. In the case of the APA being activated via the FordPass app the vehicle must detect that there is neither driver nor passengers in the vehicle.
- The vehicle has an automatic transmission.

## 2.6 Approportioned Requirements

Based on negotiations with the customer, it was determined that there are requirements beyond the scope of the current project and may be addressed in future versions. Some of these requirements include but are not limited to:

- Independent features outside of driving such as the hands-free bluetooth system to be disabled when the Active Park Assist feature is enabled.
- Allow the user to choose the direction of vehicle entry into a parking spot.

# 3 Specific Requirements

1. Vehicle must be able to automatically park itself.
2. Vehicle can only park in parallel or perpendicular parking spots.
3. Interaction with the vehicle can only happen through the HMI or authorized Ford mobile app.
4. Direction of entry must be chosen for the vehicle prior to park maneuver.
5. Sensors must be able to accurately determine parking spots, and whether or not the chosen spot is large enough.
6. Vehicle can adjust steering to a specified location if parked on a hill (Should roll into the curb if the parking brake fails).
7. Requires pressure and seat buckle in drivers' seat to activate and operate if using in-vehicle HMI.
8. System could be exploited as it is software driven, so a security system must be in place.
9. Security system must be able to detect whether any unauthorized software changes have been made.
10. Vehicle uses Lidar and Sonar for parallel and redundant distance detection.
11. If sensors are failing due to coverage such as snow or other factors, the system should
    disable and warn the driver. Redundant systems that are nonfunctional may be ignored (until they become necessary, if need be).
12. The customer must be able to use the brake pedal to slow or stop the vehicle.
13. The customer must be able to control the vehicle's speed and position using the app.
14. The app must also be able to communicate with the vehicle.
15. Parking must be completed in a reasonable amount of time.
16. Vehicle can detect if a trailer or other towable is attached and disable the self parking feature.
17. Vehicle must use power steering to maneuver into the parking spot.
18. System must notify the user through HMI when no maneuvers are possible.
19. System must identify and notify the user during active parking maneuvers when new obstacles have been presented.
20. System will indicate to the user when the maneuver begins and ends.
21. System will place the vehicle into parking gear and the feature becomes inactive when the maneuver has completed.

# 4 Modeling Requirements

## Use Cases:

| | |
|---|---|
| Use Case: | Ford Account Authentication |
| Actors: | Ford User (initiator) |
| Type: | Secondary |
| Description: | The user will need to provide information that confirms their identity as well as the vehicle in use for the active park assist. This verification will be done through the Ford App. On completion, the user will be able to activate the active park assist feasibly. |
| Includes: | Proper documentation of user and vehicle. |
| Extends: | Make sure that authentication is correct for the user. |
| Cross-refs: | 1: Ford App |
| Use cases: | User must have Ford App downloaded |

| | |
|---|---|
| Use Case: | Obstacle Identification |
| Actors: | Ford User, Vehicle |
| Type: | Secondary |
| Description: | The objective of this use case is to identify any obstacles that may disturb the vehicle when parking. This could include shopping carts, people, other vehicles, etc. The vehicle will avoid the situation and possibly change parking spots. On completion, the vehicle will have avoided the obstacle and completed the parking, if possible. |
| Includes: | Camera and sensor input is up to date with newer models of vehicles. |
| Extends: | Ensures safety of vehicles to avoid any damage that might occur. |
| Cross-refs: | 1: Ford App |
| Use cases: | Users and vehicles must be authenticated. |

| | |
|---|---|
| Use Case: | Using Active Park Assist App |
| Actors: | Ford User (initiator) |
| Type: | Primary and essential |
| Description: | The user arrives at a parking lot with their vehicle and app downloaded. The user exits the vehicle and opens the app. The parking selections will then be chosen by the user. After finishing the available spots and option of parallel or perpendicular parking, the vehicle will park itself. Upon completion, an indication of completion will be received through the app. |
| Includes: | Validate user identity |
| Extends: | Ability to choose specific parking style and parking spot at location |
| Cross-refs: | 2: Ford App and Obstacle Identification |
| Use cases: | Users must complete the Ford account authentication. |

## Class Diagram:

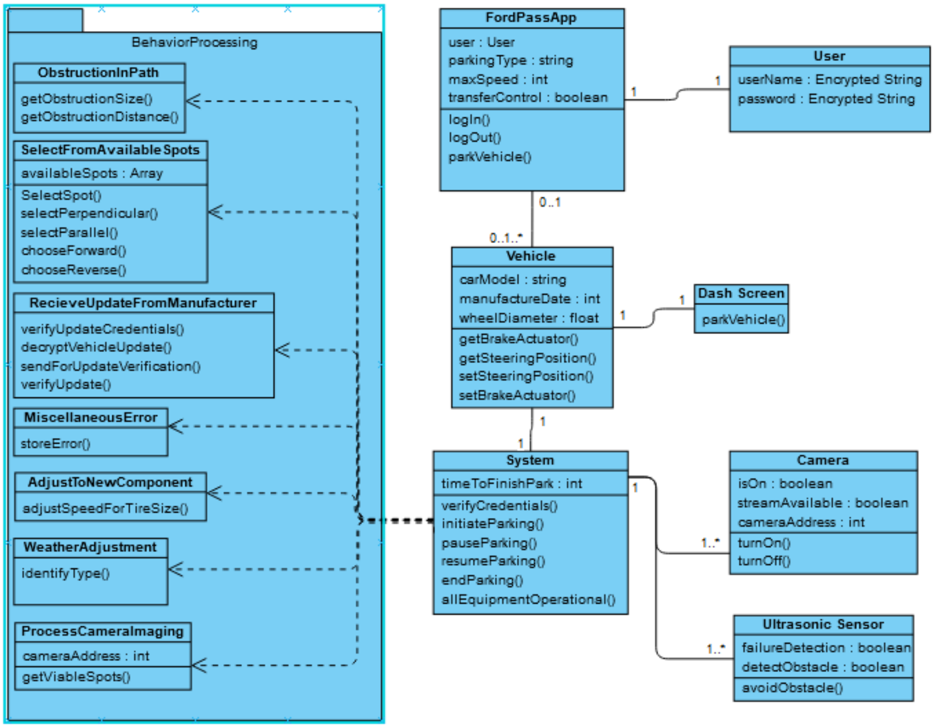


Figure 4.1:Class Diagram

| **Element Name** | | **Description** |
|---|---|---|
| Camera | | Class representing the hardware Camera on the vehicle. Available for detection. |
| Attributes | isOn : boolean | Boolean variable to determine if the Camera is on |
| | streamAvailable : boolean | Boolean variable to determine if the Camera stream is available |
| | cameraAddress : int | Integer variable to indicate the camera's address |
| Operations | turnOn() | Turns the camera on. |
| | turnOff() | Turns the camera off. |
| Relationships | Has a one-or-many-to-one relationship with the System. There can be one or more Camera instances to one System. | |
| UML Extensions | System | |

| Element Name | | Description |
|---|---|---|
| DashScreen | | Represents the dash screen located in the vehicle, also known as the HMI. |
| Attributes | Has no attributes. | N/A |
| Operations | parkVehicle() | Operation to indicate to the Vehicle that it can begin parking. |
| Relationships | Has one-to-one relationship with the Vehicle. There can only be one DashScreen to one Vehicle. | |
| UML Extensions | Vehicle | |

| Element Name | | Description |
|---|---|---|
| FordPassApp | | Class that represents the FordPass Application. |
| Attributes | user : User | The user that is associated with the FordPass application. |
| | parkingType : string | Variable that indicates the type of parking spot, parallel or perpendicular. |
| | maxSpeed : int | Integer variable that indicates the maximum speed that vehicle can attain. |
| | transferControl : boolean | Boolean that indicates whether the control has been transferred from the HMI to the FordPass App |
| Operations | logIn() | Operation that allows login for a specific User. |
| | logOut() | Operation that allows logout for a specific User. |
| | parkVehicle() | Operation to indicate to the Vehicle that it can begin parking. |
| Relationships | Has a one-to-one relationship with User and a none-or-many-to-none-or-many relationship with Vehicle. | |
| UML Extensions | Vehicle, User | |

| Element Name | | Description |
|---|---|---|
| System | | Class representing the Active Park Assist Feature System. |
| Attributes | timeToFinishPark : int | Integer indicating the amount of seconds it takes for a Vehicle to complete parking. |
| Operations | verifyCredentials() | Operation used to verify that credentials have been authorized from the FordPass App with this System. |
| | initiateParking() | Operation to initiate parking for a vehicle. |
| | pauseParking() | Operation to pause parking for a vehicle. |
| | resumeParking() | Operation to resume parking for a vehicle. |
| | endParking() | Operation to end parking for a vehicle. |
| | allEquipmentOperational() | Operation to make sure all equipment is operational. |
| Relationships | Has a one-to-one relationship with Vehicle. Has a one-to-one-to-many relationship with Camera. Has a one-to-many relationship with Ultrasonic Sensor | |
| UML Extensions | Vehicle, Camera, Ultrasonic Sensor | |

| Element Name | | Description |
|---|---|---|
| Ultrasonic Sensor | | Distance sensor capable of accurate distance measurement within a short range |
| Attributes | failureDetection : boolean | detects failures within the ultrasonic sensors |
| | detectObstacle : boolean | obstacle distance |
| Operations | avoidObstacle() | returns the distance recorded as float |
| Relationships | One to many relationship with System | |
| UML Extensions | System | |

| **Element Name** | | **Description** |
|---|---|---|
| User | | Class which represents the person using the Ford app. |
| Attributes | userName : encrypted string | Account name associated with the user |
| | password : encrypted string | Secret word or phrase used to gain admission to app per user |
| Operations | No operation available | N/A |
| Relationships | Has a one-to-one relationship with FordPassApp. One user will have one app downloaded. | |
| UML Extensions | FordPassApp | |

| **Element Name** | | **Description** |
|---|---|---|
| Vehicle | | Class which represents an overview of the vehicle in which the active parking assist system is being utilized |
| Attributes | carModel : string | Model of the car |
| | manufactureDate : int | Date that the car was manufactured |
| | wheelDiameter : float | Diameter of the currently attached wheels |
| Operations | getBrakeActuator() | returns the current state of the brake activity |
| | getSteeringPosition() | returns the current steering position |
| | setSteeringPosition() | sets the current steering position |
| | setBrakeActuator() | sets the current state of brake activity |
| Relationships | A FordPassApp has 0...1…* vehicle classes. Each vehicle class has 1 System class association and 1 DashScreen association. | |
| UML Extensions | FordPassApp, DashScreen, System. | |

## Sequence Diagram(s):

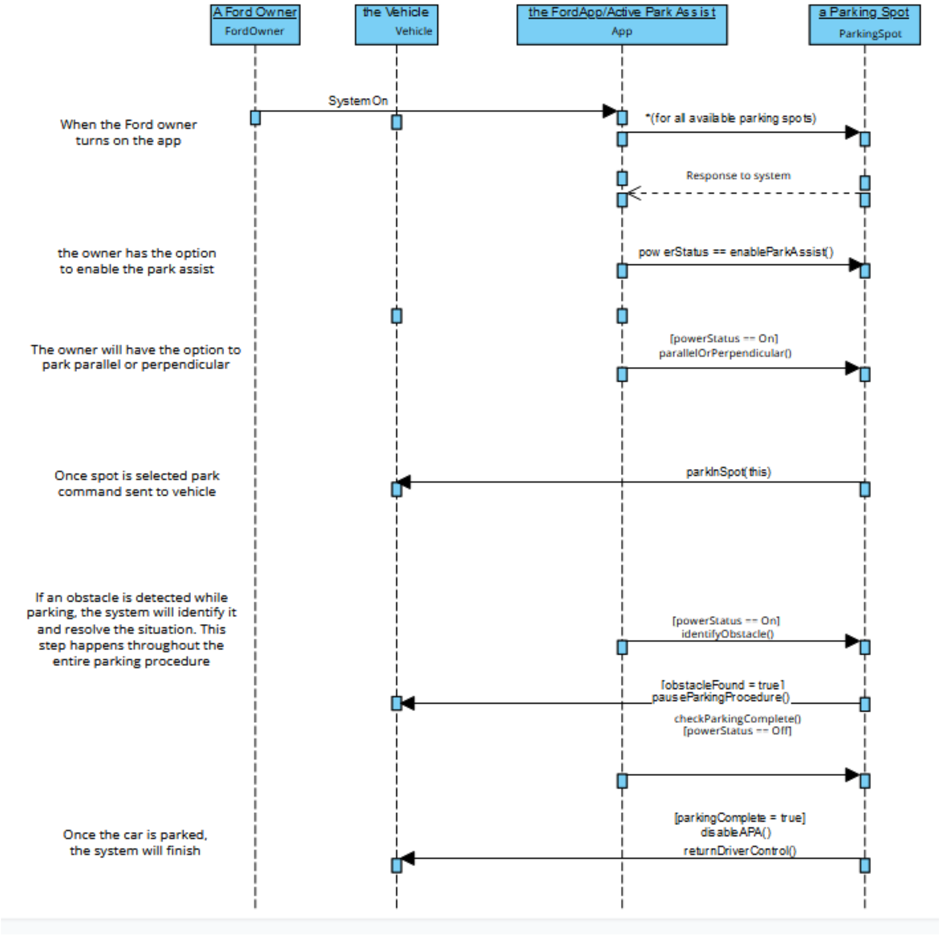

A Ford Owner
FordOwner
the Vehicle
Vehicle
the FordApp/Active Park Assist
App
a Parking Spot
ParkingSpot
SystemOn
When the Ford owner turns on the app
*(for all available parking spots)
Response to system
the owner has the option to enable the park assist
powerStatus == enableParkAssist()
The owner will have the option to park parallel or perpendicular
[powerStatus == On]
parallelOrPerpendicular()
Once spot is selected park command sent to vehicle
parkInSpot(this)
If an obstacle is detected while parking, the system will identify it and resolve the situation. This step happens throughout the entire parking procedure
[powerStatus == On]
identifyObstacle()
[obstacleFound = true]
pauseParkingProcedure()
checkParkingComplete()
[powerStatus == Off]
Once the car is parked, the system will finish
[parkingComplete = true]
disableAPA()
returnDriverControl()

Figure 4.2 Sequence Diagram

## Scenario 1:

In this scenario the user activates the APA system via the FordPass application. The application will display parking locations found by the vehicle and prompt the user to select a parking location. Once the user has selected a parking location the vehicle will begin its parking maneuver. The vehicle throughout this maneuver will look for obstructions and pause (stop the vehicle) if an obstruction is found. Once the obstruction clears or the APA system times out the vehicle will continue its parking procedure or in the case of timeout it will return control to the driver. If at any point in the parking maneuver the user presses the “Brake” button on the app the vehicle will halt. Once the vehicle has successfully parked in the desired spot the control of the vehicle will be returned to the user. In this scenario the user cannot perform any other autonomous actions from outside of the vehicle so the APA system will disable, and no further input will be accepted.

## State Diagram:

The state diagram describes the behavior of the system's classes and how they respond to input by both the driver and the APA systems command-response structure.

### Diagram entity descriptions:

- Blue round rectangles represent the states of the system
- Arrows represent the transition of the states given, and the text describing them represents the required parameters for the transition to exist.
- Blue diamonds represent a decision. An arrow into a blue diamond represents input into the decision from the previous state. Arrows out represent the resulting decisions transition path between states.

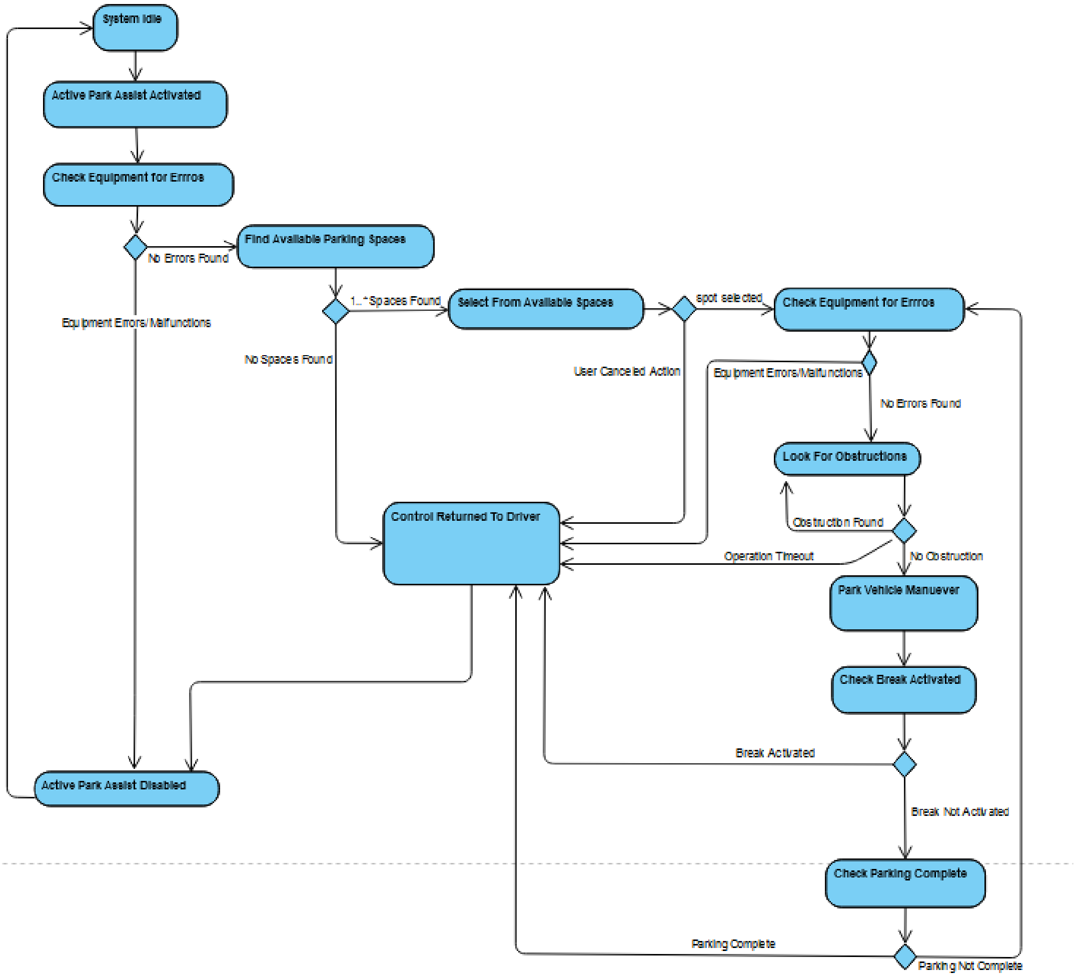


Figure 4.3 State Diagram

## State Diagram States:

**Idle:** When the vehicle is not using the APA system the system is idling, waiting for user input to tell it to activate.

**Active Park Assist Activated:** By either the HMI or the FordPass app the APA system has been activated.

**Check Equipment for Errors:** Equipment required by the APA system is checked for errors. This state occurs once when the system is first enabled, and multiple times throughout the parking procedure state loop.

**Find Available Parking Spaces:** The apa system looks for a parking space. If no parking space is available then control is returned to the driver and the APA system is disabled.

**Select From Available Parking Spaces:** If a parking space is found in the previous step the user has the option to select the parking space from the available spaces. If the user cancels the action then the control is returned to the driver.

**Look For Obstructions:** Once a parking space is selected and the second error check occurs the APA system will look for obstructions in the path of the vehicle. If obstructions are found it will circle back and restart this process until either the obstruction is cleared or the APA system times out.

**Park Vehicle Maneuver:** If there are no obstructions for a given parking maneuver (small movement or maneuver required for complete parking procedure) then the parking maneuver will execute and the vehicle will complete this maneuver.

**Check Brake Activated:** If the brake has been activated after a very small movement generated in the Park Vehicle Maneuver then control will be returned to the driver. If not, the state proceeds to Check Parking Complete.

**Check Parking Complete:** Vehicle checks to see if it has secured its desired location in the selected parking spot. If it has not then the loop begins again by checking for equipment errors.

**Control Returned to Driver:** Control of the vehicle is returned to the driver, then the APA system is disabled immediately after.

**Active Park Assist Disabled:** APA system is disabled and the vehicle's APA system goes into an idle state.

# 5 Prototype

Our prototype will show a variety of general use scenarios that a driver could encounter when using the Active Parking Assist functionality. It demonstrates a user's ability to control vehicle speed, brakes, and the decision on where to park. Additionally, multiple camera angles are included for the user so that they can obtain a better understanding of not just the vehicle, but how its surroundings are being used to calculate the autonomous parking capabilities.

Our prototype is built upon the Unity engine, rendered in three-dimensional graphics, allowing for the closest possible representation of how the active parking assist system would function in real-life situations. The prototype controls are embedded within a mobile phone application user interface, which represents an implementation of the Ford pass mobile application, and how the system would integrate into its existing capabilities. However, the same controls would also be embedded inside the actual vehicle - allowing for our prototype to demonstrate both use cases: a driver controlling the system from inside and outside the vehicle.

## 5.1 How to Run Prototype

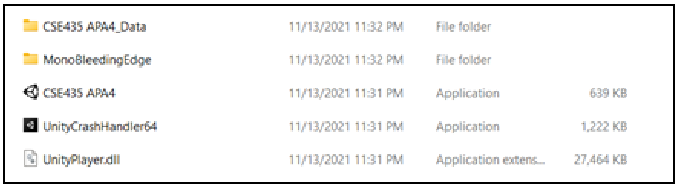
There are two possible ways to run our prototype: in-browser, under the "Deliverables" section of our team's webpage, or through a Windows OS executable file. Unless otherwise specified, assume that no additional plugins, constraints, or other conditions are required.

**Web-browser:**

1. Go to the CSE435 APA4 Website
2. Navigate to the "Deliverables" section and enter the username/password
   a. Username: *apa4*
   b. Password: *apa4*
3. The simulation will start automatically

**Executable file:**

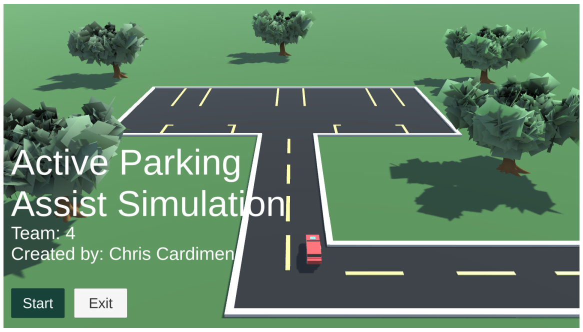
1. Download the executable by either clicking here, to directly download the executable, or by visiting the CSE435 APA4 Website and clicking on the *Prototype* hyperlink found in the *Links and resources* section.
2. Unzip the downloaded file
3. Open the unzipped folder
4. Navigate to the folder labeled "Windows" and open it
5. You will see files that look like this:



| | | | |
|---|---|---|---|
| CSE435 APA4_Data | 11/13/2021 11:32 PM | File folder | |
| MonoBleedingEdge | 11/13/2021 11:32 PM | File folder | |
| CSE435 APA4 | 11/13/2021 11:31 PM | Application | 639 KB |
| UnityCrashHandler64 | 11/13/2021 11:31 PM | Application | 1,222 KB |
| UnityPlayer.dll | 11/13/2021 11:31 PM | Application extens... | 27,464 KB |

6. Double click the Application file named *CSE435 APA4*
   Note: You may be prompted by Windows that this is an unverified/unknown file. There's nothing dangerous contained in it, navigate through the windows prompt to show more information & run the application.

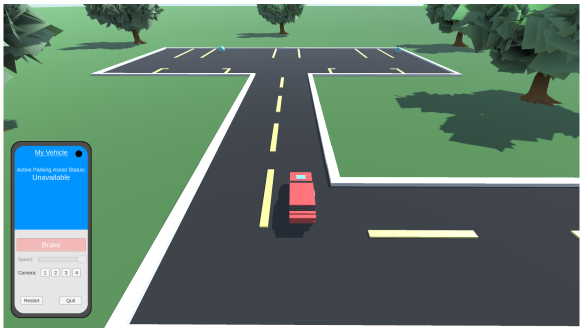
**Startup screen:**
As displayed in *Figure 1*, upon running the prototype, the startup screen will be displayed. From here, the prototype-user can elect to either begin the simulation by clicking the *Start* button, or exit the program by clicking the *Exit* button. Both buttons can be found in the bottom-left portion of the screen.



Active Parking Assist Simulation
Team: 4
Created by: Chris Cardimen

Figure 1: Prototype Startup Screen

**Simulation interface:**
Once the simulation has been started, the prototype-user will be introduced to the scenario through a beginning sequence of an automobile approaching a parking lot, as displayed in *Figure 2*. At this point in the simulation, no user-input is required. This scene simply serves to introduce the prototype-user to the world in which the simulation is occurring and the interface controls
.



My Vehicle
Active Parking Assist Status:
Unavailable

Figure 2: Introduction to the simulation

Once the automobile has come to a stop, the prototype-user's attention should be directed toward the mobile phone interface, illustrated in *Figure 3*, which has changed. For the purposes of this simulation, this interface will serve as the HMI. However, in a real-world scenario, drivers and users of the active parking assistance system would either use the mobile app while remaining outside the vehicle *or* an in-vehicle HMI while remaining physically present inside of the vehicle.

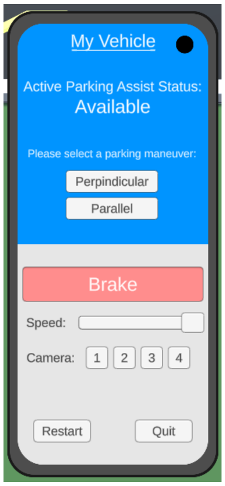


Figure 3:  Mobile phone interface

Now, the prototype-user's input is required.
The user-interface is broken into two distinct parts: the top half, shown in *Figure 4*,which houses the controls for parking spot selection, and the bottom half, in *Figure 5*, which has viewing, speed, braking, and scenario controls.

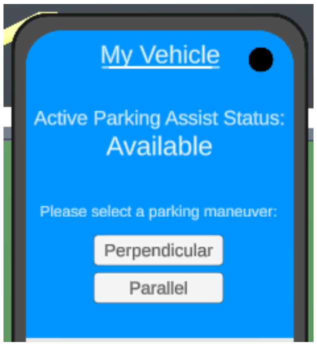

My Vehicle
Active Parking Assist Status:
Available
Please select a parking maneuver:
Perpendicular
Parallel

Figure 4:  Top controls

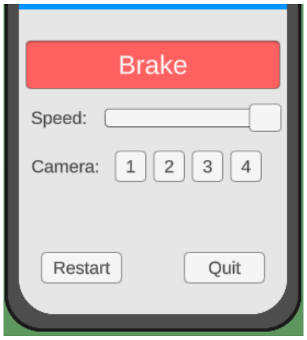


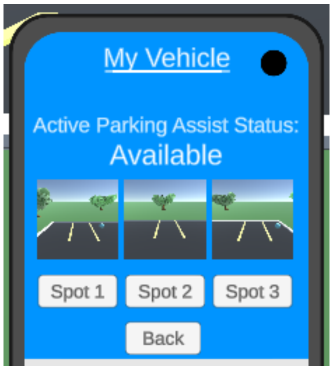
Figure 5:  Bottom controls

Usage of the bottom controls is illustrated below. Please note, these controls are inaccessible if the system is turned off, and will become greyed-out to represent this.

- Braking: To enable the brakes on the vehicle, click and hold the red button labeled "Brake". This will lower the speed of the vehicle, eventually bringing it to a stop - the change in speed will be represented by the speed slider, so that the prototype-user can view everything in real-time.
- Speed: The speed of the automobile can be controlled by clicking and dragging the speed slider to the desired speed.
- Camera: Four different cameras are available to monitor the simulation. Clicking any of the buttons ("1" through "4") will bring up the corresponding camera.
  - 1: Bird's eye view
  - 2: Top of automobile view
  - 3: Front/perspective view
  - 4: Back/rear view

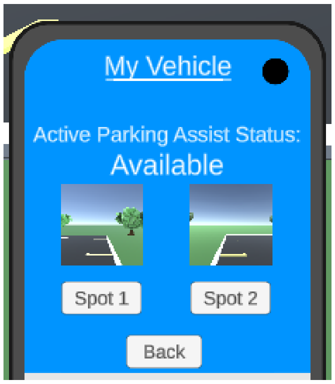
- ❖ Restart: To restart the simulation from the beginning, click the “Restart” button.
- ❖ Quit: By clicking the “Quit” button, the simulation will end.

To use the top controls, the prototype-user will first be greeted with the option to select an either perpendicular or parallel parking maneuver, as shown in *Figure 4*. Clicking on either of these buttons will bring up available options for the corresponding maneuver, with real-time rendered images of the available spot - both of these are referenced in *Figures 6*-7.



Active Parking Assist Status:
Available

Figure 6: Perpendicular spots

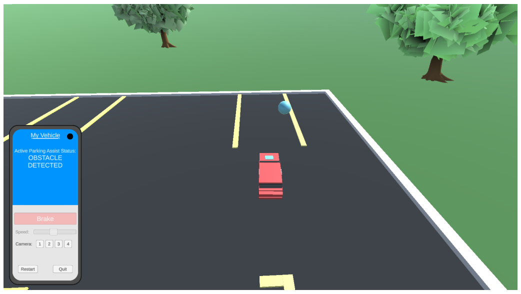


Active Parking Assist Status:
Available

Figure 7: Parallel spots

After selecting a type of maneuver, the user can either opt to select a parking spot from the list of available spots, or choose to go back to the maneuver selection menu to choose a different maneuver to bring up different spots.

- ❖ Spot selection: Click on one of the buttons labeled “Spot x”, where x is the corresponding numerical value to the spot.
- ❖ Back button: Click on the button labeled “Back” to return to the maneuver selection menu.



Active Parking Assist Status:
OBSTACLE
DETECTED

Figure 8: Obstacle detection

After selecting a spot, the vehicle will autonomously park itself. No user input is required, however, the user does still have the ability to control the brakes, speed, and cameras, as long as the vehicle is not currently encountering an obstacle. If the vehicle encounters an obstacle, it stops and waits until the obstacle has passed, then returns control to the user, as illustrated in *Figure 8*.

Once the obstacle, if present, has passed, and the vehicle has come to a stop in its selected parking spot, the simulated scenario is over, and the active parking assist system will turn itself off. At this point, the prototype-user can elect to either restart the simulation, to choose a different scenario or replay the current scenario, or exit the program.

## 5.2 Sample Scenarios

There are a variety of scenarios presented in this simulation:

**3 Perpendicular parking scenarios:**

1. A parking spot found on the left, including an obstacle.
2. A parking spot found directly in front of the vehicle.
3. A parking spot found on the right, including an obstacle.

**2 Parallel parking scenarios:**

1. A parallel parking spot found on the right of the vehicle around a corner.
2. A parallel parking spot found on the left of the vehicle around a corner.

# 6 References

[1] C. Cardimen, S. Hasan, C. Lowen, and E. Stevens, “APA4 Project Website,” *Active Parking Assist Team 4*, Nov-2021. [Online]. Available: http://cse.msu.edu/~cardime2/index.html. [Accessed: 16-Nov-2021].

[2] B. H. C. Cheng, “Project Description.” Michigan State University, East Lansing, Nov-2021.

[3] E. Davidson, “Active Park Assist.” Michigan State University, East Lansing, Nov-2021.

[4] P. Greenberg, “Legislation Related to Artificial Intelligence,” *Legislation related to Artificial Intelligence*, 15-Sep-2021. [Online]. Available: https://www.ncsl.org/research/telecommunications-and-information-technology/2020-legislation-related-to-artificial-intelligence.aspx. [Accessed: 16-Nov-2021].

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

# 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.