

Software Requirements Specification (SRS)

Active Park Assist 2

Authors: Cameron Rooks, Will McGee, Emilio Castillo, Claury Mejia, Yumo Wang

Customer: Eileen Davidson, Ford Motor Company

Instructor: Dr. Betty H.C. Cheng

1 Introduction

This document provides the requirements for the Active Park Assist system (APA), an automatic parking subsystem to be installed in future Ford vehicles. This document is intended for both the customer and the developers of the system, containing models that show the structure and sub-elements of the system, as well as the expected behavior of the system through different events. This document also includes a link to both a visual presentation of the interface as well as an intractable prototype interface for the system.

1.1 Purpose

This document establishes the basis for an agreement between the customer and the developers on what the Active Park Assist system is required to do, as well as what it is not allowed to do. The intended audience are developers and customers.

1.2 Scope

This software system is an active park assist system. The software automates the parking process by establishing a potential spot, verifying it with the driver, and taking over the traditional tasks required to move into a parking spot such as reversing, accelerating, steering, and maneuvering the vehicle into position.

It will create less error when parking and takes stress away from the driver when driving in very densely populated cities. The system will only conduct parking and will not leave a parking spot for the user.

The system will communicate with an on-board set of sensors and cameras to perform the parking maneuvers. This information will be communicated through an onboard communication within the car.

1.3 Definitions, acronyms, and abbreviations

This section introduces and defines a set of acronyms, abbreviations and key phrases used throughout this document.

- Driver - The operator of the car, and participant in certain system interactions (also known as “User”)
- Active Park Assist (APA) - Name of the system, a system that performs parking maneuvers without interaction of the driver
- HW - Hardware, mechanical pieces of the car
- SW - Software, the programs used to perform system functions within the car
- Human Machine Interface (HMI) - Interface used by the user to interact with the system, used primarily to communicate with the driver for certain decisions
- Actuation Unit (ACU) - Component of a machine that moves or controls a mechanism or system
- Brake Control - Subsystem which controls the brakes
- Powertrain Management - Subsystem that allows us to adjust the movement speed of the car during a maneuver
- Steering Control - Subsystem that allows us to steer the car
- Controller Area Network (C.A.N.) - In-vehicle peer-to-peer electronic network that transfers data to multiple CAN devices
- Park Control - Subsystem to control the gearshift
- SYNC - An integrated communications and entertainment system common among vehicles (not an acronym).
- Legal Parking Spot - Available non-handicapped parking spot
- 3rd Party Attack - Unknown entities with the ability to break the security of a system and affect control it
- Autonomous Feature - A feature that has decision-making and the ability to show the user the world in real time 3D showing the varying scenarios
- Size Metric - dimensions of an object that will not damage or impede the functionality of the vehicle (unaffected by varying car sizes)

1.4 Organization

The following section explains the topics to be covered in this document starting from section 2 until section 7.

Section 2 is intended as a high level look at what this project entails for the system. This description includes product functionality, user characteristics, system constraints, assumptions made about the HW, SW, environment and user interactions, and the apportioning of the requirements. Section 3 presents a list of specific requirements of the APA using a hierarchical numbering scheme. Section 4 describes the specification portion of the requirements document, providing different diagrams explaining the bridge between the application domain and the machine domain. This section includes the use-case, class/domain, sequence and state diagrams. Section 5 introduces the prototype in terms of system functionality, explains how to run the prototype and different scenarios using the system. Section 6 provides the reader with a list of references. Section 7 provides the point of contact for questions regarding use of this document.

2 Overall Description

This section describes the overall functionality of the system. This includes details on the context of the system as well as an explanation on specific characteristics of an intended user of the system. The system functionality is also summarized in this section, specifying the major functions the software performs. There are a number of different types of constraints the system faces, and those are described in this section along with assumptions for the system.

2.1 Product Perspective

The APA is intended to be an autonomous feature that takes control of the vehicle once it is activated. The system executes the parking maneuver, taking the necessary steps and measurements to park the vehicle with minimal interaction required from the user starting from initialization of the system, selection and user verification of parking spot after which control of the vehicle is yield to the system until completion of parking maneuver returning the control of the vehicle to user and activating the parked gear. The overall goal is to improve already integrated technology and add new functionality and features to provide the user with the safety needed such as securing the safety pedestrians and user by avoiding collision with object/s detected by the system.

The APA system is formed by its integration with other subsystems: Brake Control, Steering Control, Park Control, Powertrain Management, Vehicle Position, and the HMI.

The user interacts with the system through the HMI in which the user activates the Park Assist in order to initiate the search of a parking spot. Cameras in the front and back of the vehicle detect available parking spots while Ultrasonic sensors are used to measure the space available between vehicles in which the user can park and shares the information obtained with the user by displaying the parking spot's rendering in the HMI.

The Park Control Subsystem receives the input from the user through the HMI and consults the Vehicle Position to start the maneuver and give instruction to the rest of subsystems.

Brake Control, Steering Control and Powertrain Management are manipulated by the Park Control Subsystem to complete the trajectory of the vehicle into the parking spot.

The system provides a simple, intuitive and easy to navigate interface. Once the Park Assist is activated, the user will be able to select the parking mode, verify a parking spot, cancel the maneuver, and extend searching time after reaching the limited amount of time to find a parking spot. If any failures or a manual abort were to occur, messages will be displayed on the screen to the user.

Each subsystem has its own unique ACU and all communicate through CAN, which is vulnerable due to its simple messages and likely they should be encrypted in the context of security and safety to avoid access from 3rd party attackers to sensitive data which once modified, altered or changed can affect the operation of the system. The system should check for “fresh” messages as a common attack is a spam of repeated messages.

2.2 Product Functions

This section will provide an overview of the major functions of the APA system. The main functionality the Active Park Assist system adds is the feature that allows a vehicle to automatically park itself. The system functions for both parallel and perpendicular types of parking spaces, and will require no driver interaction during the parking process. To activate the system’s functionality, the driver will hit a button on the SYNC interface, and the system will activate. The system will begin identifying potential available parking spots, and will ask the driver to verify the spot once found. Ultrasonic sensors on the sides of the vehicle, along with cameras in the front and back of the vehicle, will allow the system to detect potential parking spots and objects. If the identified spot is verified by the driver, the system will take over the driving of the vehicle and park itself in the spot. The cameras and sensors will continue to be used during the automatic parking function of the system to ensure the vehicle does not hit anything unintentionally. Software will detect when the parking process function is complete, and will shift the vehicle into park as well as alerting the driver that the system has completed its task. Other system functionality includes the ability to detect obstacles in the path of the vehicle during the parking process, and the ability for the driver to abort the system during the parking process by pressing the brake.

2.3 User Characteristics

The user for this system is a legal driver of a Ford vehicle. The driver is expected to understand what a legal parking spot is and what spots would be illegal for them to park in (such as a handicap spot or reserved spot), despite the system offering such a potential spot. The user is also expected to remain in the car during use of the system, as well as to be aware of their surroundings. For example, the user starts the system with nearby available spots, being mindful of the changes regarding parking and obstructions in the nearby area, so as to maximize the proper use of the APA system.

2.4 Constraints

The APA is constrained by several aspects, which may affect the performance of the system under certain situation. Listed below are those constraints.

- The process time of the system is limited due to requirement of real-time computing

- The vehicle may differ in size and carry additional attachments, such as bicycle, luggage, etc. These are likely to cause unpredictable consequences. The driver should ensure before activate the system that there are no additional objects of any kind attached to any part of the vehicle
- The system cannot function if the detectors are not working properly. In addition, A detector failure may occur under a bad weather condition, such as fog, thunderstorm, etc.
- The system is likely to be attacked intentionally by a third party. It should use a verification mechanism to prevent the operation from being tampered. Also, as a back-up measurement, the driver should be able to abort the system manually at any time
- The ACU should be able to accurately carry out the instructions issued by controller
- Any system critical to the use of this system must be available, including ACU, HMI, and all detectors

2.5 Assumptions and Dependencies

The system requires certain assumptions to operate correctly. Listed below are those assumptions, as well as a few dependencies the system requires to function.

- There are available parking spaces in the area when the system is activated
- The driver will select the type of parking to search for (perpendicular vs parallel)
- The driver will not have control of the vehicle during maneuvers, except to abort the system
- Should the driver require to abort the system, pushing the brake will do so
- The sensors will read in information, in real time, to perform a parking maneuver
- The driver should abort the system manually should a mistake be made during initiation
- The hardware (sensors/brakes/etc.) should be functional and undamaged
- The user will verify that the parking spot chosen by our system is a legal parking spot

2.6 Appropriation of Requirements

Future versions of the APA will include a feature to account for objects attached to the vehicle that changes the overall size of the vehicle, such as a bike rack. The user will be able to input those objects' dimensions.

Sensors will detect moving objects from non-moving objects. This will allow the system to differentiate between objects to avoid collision with such objects. Sensors will detect objects larger than the defined size metric for the system.

In vehicle architecture, each subsystem has its unique ACU in which all communicate though C.A.N., for this reason the system is vulnerable to attacks, thus in future versions it is likely that the messages that it produces will be encrypted as a measurement against hackers. Also, in addition to a verify button on the HMI, the system will be able to detect that there is a driver in the seat as a security measurement.

3 Specific Requirements

1. The system requires a method for the driver to verify the initiation of the parking maneuver, proving it was not the result of a 3rd party attack. The systems in place will be a sensor in the seat detecting a person is in fact in the car and interacting with the interface to initiate the system via touch screen
2. The system must be activated by the driver via HMI
 - 2.1. The driver must select the type of parking to perform (parallel or perpendicular)
3. The system will use the ultrasonic sensors and cameras to identify the parking spot, and its availability
 - 3.1. The driver must verify the system's analysis of the parking spot before the parking maneuver can be performed. They will do so after viewing a rendering on the HMI and validating that the spot is legal for the driver
 - 3.2. Should no spot be found within 2 minutes, the system will ask the driver if they would like to keep searching
4. The system will detect obstacles that appear during the maneuver and react to them by stopping the car if they are about the Size Metric of the car
5. The system can be cancelled and return control to the driver at any time by tapping the brake
 - 5.1. The system may be aborted due to communication failure, detected obstructions, HW failure, or in response to a 3rd Party Attack. The car will be put into park and control will be returned to the driver in this case
 - 5.2. When the maneuver has ended via the brake being pressed or the process is aborted, a notification will display through the HMI
6. The system will use both forward and reverse gears to perform the parking
 - 6.1. The driver does not need to participate in the parking performed by our system
7. At the end of the maneuver, the car will be put into the parked gear
8. Any sensor failures shall be detectable and prevent the system from being used

4 Modeling Requirements

This section includes models that are intended to specify a bridge between the system requirements and the intended system structure and behavior. There are four models in this section, which include a Use-Case Diagram, displaying the main activities of the system and the associated external actors, a Class Diagram which models the key elements of the system's structure, a Sequence Diagram which models the flow of activities between the key elements of the system, and a State Diagram which captures the numerous states the system can be in based on different events. Each model also includes a description describing their respective notation.

Use Case Diagram - Active Park Assist System

This model (*Figure. 1*) depicts the main activities of the system in regards to the external actors that interact with the system. These actors are represented by the figures outside the system boundary, which is the large square encompassing the many circles in the diagram. These circles represent the activities that the actors can do when interacting with the system. These circles may have arrows with dotted lines that show either an activity that the parent activity extends or includes, as denoted by the keywords. The arrows with the white arrow head denote the individual options for the parent activity. Each of these activities have a table below the diagram to clarify on what they specifically entail.

Use Case Templates

Use Case:	Activate Assist
Actors:	Driver
Description:	The user activates the system through the HMI.
Type:	Primary
Includes:	None
Extends:	Conduct Parking Into Parking Spot
Cross-refs:	Requirement 2
Use cases:	None

Use Case:	Press Cancel
Actors:	Driver
Description:	The user can cancel the maneuver at any moment by pressing the brakes.
Type:	Secondary
Includes:	Search Parking Spot
Extends:	None
Cross-refs:	Requirement 5
Use cases:	Deactivate Assist

Use Case:	Select Parking Mode
Actors:	Driver
Description:	The user proceeds to select the parking mode (parallel or perpendicular). Once the parking mode is selected the park control subsystem takes control of the vehicle and transmission shifts to reverse/drive and conducts the vehicle to the parking spot.
Type:	Primary
Includes:	None
Extends:	Conduct Vehicle Into Parking Spot
Cross-refs:	Requirement 2.1
Use cases:	Select Parallel and Select Perpendicular

Use Case:	Select Parallel
Actors:	Driver
Description:	The user selects parallel parking mode through the HMI.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirement 2.1
Use cases:	None

Use Case:	Select Perpendicular
Actors:	Driver
Description:	The user selects perpendicular parking mode through the HMI.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirement 2.1
Use cases:	None

Use Case:	Verify Parking Spot
Actors:	Driver
Description:	Once the HMI identifies an available spot, the user is able to verify it.
Type:	Primary
Includes:	None
Extends:	Ensure Security
Cross-refs:	Requirements 1 and 3.1
Use cases:	None

Use Case:	Deactivate Assist
Actors:	Driver
Description:	The system can be cancel at any time by tapping the brake or it could abort the parking maneuver due to communication failure, detected obstructions, HW failure or in response to a 3rd Party Attack.
Type:	Primary
Includes:	None
Extends:	None
Cross-refs:	Requirement 5
Use cases:	Press Cancel and Manual Abort

Use Case:	Manual Abort
Actors:	ACU and Driver
Description:	The system detects an unexpected behavior/activity in system that could potentially affect the operation of the system, such as communication failure, detected obstructions, HW failure or in response to a 3rd Party Attack and aborts maneuver.
Type:	Secondary
Includes:	Ensure Safety
Extends:	None
Cross-refs:	Requirements 5, 5.2 and 8
Use cases:	Apply Brake and Press Abort

Use Case:	Conduct Vehicle Into Parking Spot
Actors:	ACU
Description:	Once the parking mode is selected the park control subsystem takes control of the vehicle and transmission shifts to reverse/drive and conducts the vehicle to the parking spot.
Type:	Primary
Includes:	None
Extends:	Control ACU and Ensure Security
Cross-refs:	Requirements 6 and 6.1
Use cases:	None

Use Case:	Apply Brake
Actors:	Driver
Description:	The user manually aborts the parking maneuver by tapping the brake.
Type:	Secondary
Includes:	None
Extends:	None
Cross-refs:	Requirements 5 and 5.2
Use cases:	None

Use Case:	Press Abort
Actors:	ACU
Description:	The system is aborted due to communication failure, detected obstructions, HW failure, or in response to a 3rd Party Attack.
Type:	Secondary
Includes:	None
Extends:	None
Cross-refs:	Requirements 5.1, 5.2 and 8
Use cases:	None

Use Case:	Ensure Security
Actors:	ACU
Description:	Ensures that the parking spot to be selected is legal and that there is not unexpected behavior in the system.
Type:	Primary
Includes:	None
Extends:	Conduct Vehicle Into Parking Spot
Cross-refs:	Requirements 3.1 and 5.1
Use cases:	None

Use Case:	Search Parking Spot
Actors:	ACU
Description:	The system will use ultrasonic sensors and cameras to identify the parking spot and if no spot is found would prompt the user to extend the section and keep searching.
Type:	Primary
Includes:	Press Cancel
Extends:	Conduct Vehicle Into Parking Spot
Cross-refs:	Requirements 3-3.2
Use cases:	None

Use Case:	Check Detectors
Actors:	ACU
Description:	The system will check if the cameras in the vehicle have detected any obstacle.
Type:	Secondary
Includes:	None
Extends:	Search Parking Spot and Ensure Safety
Cross-refs:	Requirement 4
Use cases:	None

Use Case:	Ensure Safety
Actors:	ACU
Description:	The system will check if the cameras in the vehicle have detected any obstacle that could lead to a collision.
Type:	Secondary
Includes:	Manual Abort
Extends:	Conduct Vehicle Into Parking Spot
Cross-refs:	Requirement 4
Use cases:	None

Use Case:	Control ACU
Actors:	ACU
Description:	Controls the system by using the Brake Control, Powertrain Management, Steering Control, and Park Control subsystems.
Type:	Primary
Includes:	None
Extends:	Conduct Vehicle Into Parking Spot
Cross-refs:	Requirements 4 - 7
Use cases:	None

Use Case:	Obstacle Detected
Actors:	Obstacle
Description:	Obstacles are detected and shown in the screen. if an obstacle is detected the vehicle stops.
Type:	Secondary
Includes:	None
Extends:	None
Cross-refs:	Requirement 4
Use cases:	None

Use Case:	Response to Obstacle
Actors:	ACU
Description:	If an obstacle is detected the vehicle will stop.
Type:	Secondary
Includes:	Obstacle Detected
Extends:	Ensure Safety
Cross-refs:	Requirement 4
Use cases:	None

Use Case:	Abort Maneuver
Actors:	ACU
Description:	If an obstacle is detected the vehicle will stop.
Type:	Secondary
Includes:	None
Extends:	Response to Obstacle
Cross-refs:	Requirements 4 and 5.1
Use cases:	None

Class Diagram - Active Park Assist System

This model (*Figure.2*) depicts the key elements that are present in the system as well as the associations they have with other elements. Each box represents one of these elements with its data values and operations listed below the element's name in that respective order. The arrows connecting the elements represent the associations with the activities that connect them listed on the arrow. Numbers near the end of an association describe the number of elements connected by the association and if no number is present, the number is assumed to be one. For specifics on the displayed elements, a data dictionary is included after the diagram.

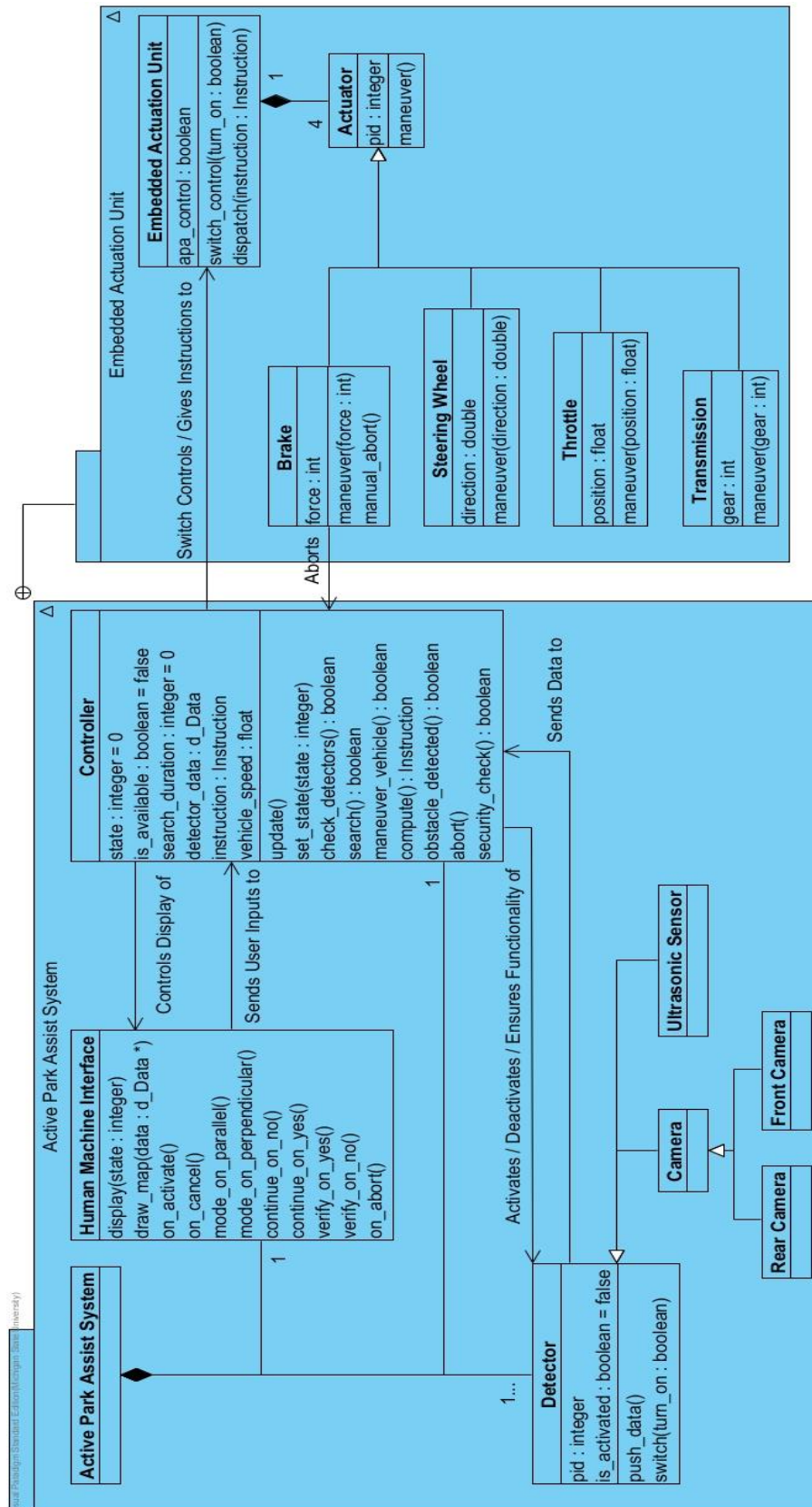


Figure 2: The class diagram for the APA system

Element Name	Description
Active Park Assist System	Overall system, composed of the HMI, controller, and detectors
Relationships	Composed of Human Machine Interface, Controller, and Detector

Element Name		Description
Human Machine Interface		Interface between driver and controller, which is responsible for screen display, as well as collecting user input
Operations		
	display (stage: integer)	The essential operation of the HMI, which takes the state number and display the corresponding interface to user
	draw_map(data : d_Data*)	Draws the map on the screen according to the given data during spot search and maneuver. Once invoked, it will dynamically generate map and display it, until the HMI stage being changed
	on_activate()	A user input handler that is to be invoked when the user presses activate button.
	on_cancel()	A user input handler that is to be invoked when the user presses cancel button
	mode_on_parallel()	A user input handler that is to be invoked when the user presses parallel button
	mode_on_perpendicular()	A user input handler that is to be invoked when the user presses perpendicular button
	continue_on_no()	A user input handler that is to be invoked when the user presses “no” button in corresponding situation
	continue_on_yes()	A user input handler that is to be invoked when the user presses “yes” button in corresponding situation
	verify_on_yes()	A user input handler that is to be invoked when the user presses “yes” button in corresponding situation
	verify_on_no()	A user input handler that is to be

		invoked when the user presses “no” button in corresponding situation
	on_abort()	A user input handler that is to be invoked when the user presses abort button
Relationships	<ul style="list-style-type: none"> Sends user inputs to Controller 	

Element Name		Description
Controller		Receiver of all sensor input and the HMI, which it utilizes to compute the situation and send out actions to other parts of the system
Attributes		
	stage : integer = 0	State of the system, where the integer is used for keeping track of state that APA is currently on, as well as determining the behavior of <i>update()</i> operation. It can be also regarded as an enumeration shown in <i>table. 2.1</i> . Valid value of the argument is from 0 to 8.
	is_available : Boolean = false	Whether the system is available, determined by vehicle speed the controller holds. The value “true” means the system is available
	search_duration : integer = 0	Time elapsed in second since the spot search is initiated
	detector_data : d_Data	The collection of data sent from detectors, of which the type is denoted as d_Data
	Instruction : instruction	Current instruction that is to be implement by ACU
	vehicle_speed : float	Current vehicle speed
Operations		
	update()	The essential operation of the controller, which is repeatedly invoked throughout the whole process. Its functionality varies depending on state of the controller. It is a typical state machine
	set_state(state : integer)	Sets the state of the system, where the argument is the state number, valid from 0 to 8

	check_detectors() : boolean	Perform a test of detectors. it returns true if all detectors are able to work properly
	search() : boolean	Performs the searching algorithm on detected data dynamically. It returns "true" if a spot is found. "false" if it is unable to locate a spot within 2 minutes
	security_check():boolean	Check if the request from the driver is secure and not a malicious attack. Returns true if the system is secure.
	obstacle_detected():boolean	Detect obstacles based on the data collected by controller. Returns true if it detects an obstacle.
	maneuver_vehicle() : boolean	Coordinates all data flows during maneuver, as well as switch the detectors and ACU
	compute()	Invoked repeatedly during maneuver to apply algorithm on detected data, which generates an instruction at a time
	abort()	Invoked when abort of the system is needed. it will brake the vehicle and set the system unavailable
Relationships	<ul style="list-style-type: none"> Controls display of HMI Activates, deactivates, and ensures the functionality of detectors Switches the control of the ACU as well as giving instructions to it 	

integer	state
0	System Unavailable
1	System Available
2	Pending Selection on Parking Mode
3	Searching Parking Spot
4	Pending Selection on Whether to Continue Searching
5	Pending Verification on Parking Spot
6	Maneuvering vehicle
7	Maneuver Aborted
8	Maneuver Completed

(Table 2.1 : Enumeration of states)

Element Name		Description
Detector		A base class that describes a detector, which is responsible for sending detected information to controller
Attributes		
	pid : integer	A unique process id assigned to each detector, used for identification.
	is_activated : boolean = false	The detector is activated. Value “true” means the detector is activated, otherwise is not.
Operations		
	push_data()	Send the real-time information recorded by detector to the controller.
	switch(turn_on : boolean)	Used to switch the detector on and off
Relationships	<ul style="list-style-type: none"> • Sends data to Controller • Generation of Camera and Ultrasonic Sensor 	

Element Name		Description
Embedded Actuation Unit		A class actually maneuvers all kinds of actuators, which is composed of Brake, Steering Wheel, Throttle, and Transmission
Attributes		
	apa_control : boolean = false	Whether the APA is currently taking control of the ACU.
Operations		
	switch_control(turn_on : boolean)	Switch the direct control of ACU between APA and driver. Input “true” will transfer the control to APA.
	dispatch(instruction : Instruction)	Dispatches the instruction generated by controller and maneuver actuators accordingly
Relationships	<ul style="list-style-type: none"> • Composed of all actuators 	

Element Name		Description
Actuator		A base class that describes an actuator
Attributes		
	pid : integer	A unique process id assigned to each actuator, used for identification.
Operations		
	maneuver()	A virtual function that changes the behavior of the actuators respectively

Relationships	<ul style="list-style-type: none"> Part of Embedded Actuation Unit
---------------	---

Element Name		Description
Brake		A class maneuvers the brake specifically
Attributes		
	force : integer	The force applied to brake
Operations		
	maneuver(force : integer)	Updates the force applied on the brake. The argument is supposed to be the value of new force
	manual_abort()	Invoked when user apply brake during maneuver, which in turn aborts the whole system
Relationships	<ul style="list-style-type: none"> Derived class of Actuator Aborts the Controller 	

Element Name		Description
Steering Wheel		A class maneuvers the steering wheel specifically
Attributes		
	direction : double	The direction of the steering wheel
Operations		
	maneuver(direction : double)	Updates the direction of the steering wheel. The argument is supposed to be the value of new direction
Relationships	<ul style="list-style-type: none"> Derived class of Actuator 	

Element Name		Description
Throttle		A class maneuvers the throttle specifically
Attributes		
	position : float	The position of the throttle
Operations		
	maneuver(position : float)	Updates the position of the throttle. The argument is supposed to be the value of new position
Relationships	<ul style="list-style-type: none"> Derived class of Actuator 	

Element Name		Description
Transmission		A class maneuvers the transmission specifically
Attributes		
	gear : int	The gear of the vehicle
Operations		
	maneuver(gear : int)	Updates the gear of the vehicle. The argument is supposed to be the value of new gear, which can

		be regarded as an enumerator as shown in <i>table.2.2</i> . Argument is valid from 0 to 2.
Relationships	<ul style="list-style-type: none"> Derived class of Actuator 	

integer	gear
0	park
1	drive
2	reverse

(table 2.2 Enumeration of gear)

Sequence Diagram - Active Park Assist System

This model (*Figure.3.1-3.3*) depicts the sequence of events through the interactions between the elements of the system from the previous class diagram. The y-axis represents time starting from the top, with each element having a vertical dotted line representing its lifeline within this system. The blue rectangles on the lifeline represent activity blocks for that given element, meaning that part of the system is conducting some sort of work at that time. The arrows show the interactions between the elements, either through messages or by calling methods in the receiving element. A horizontal dotted-line indicates a flow of information to the receiver.

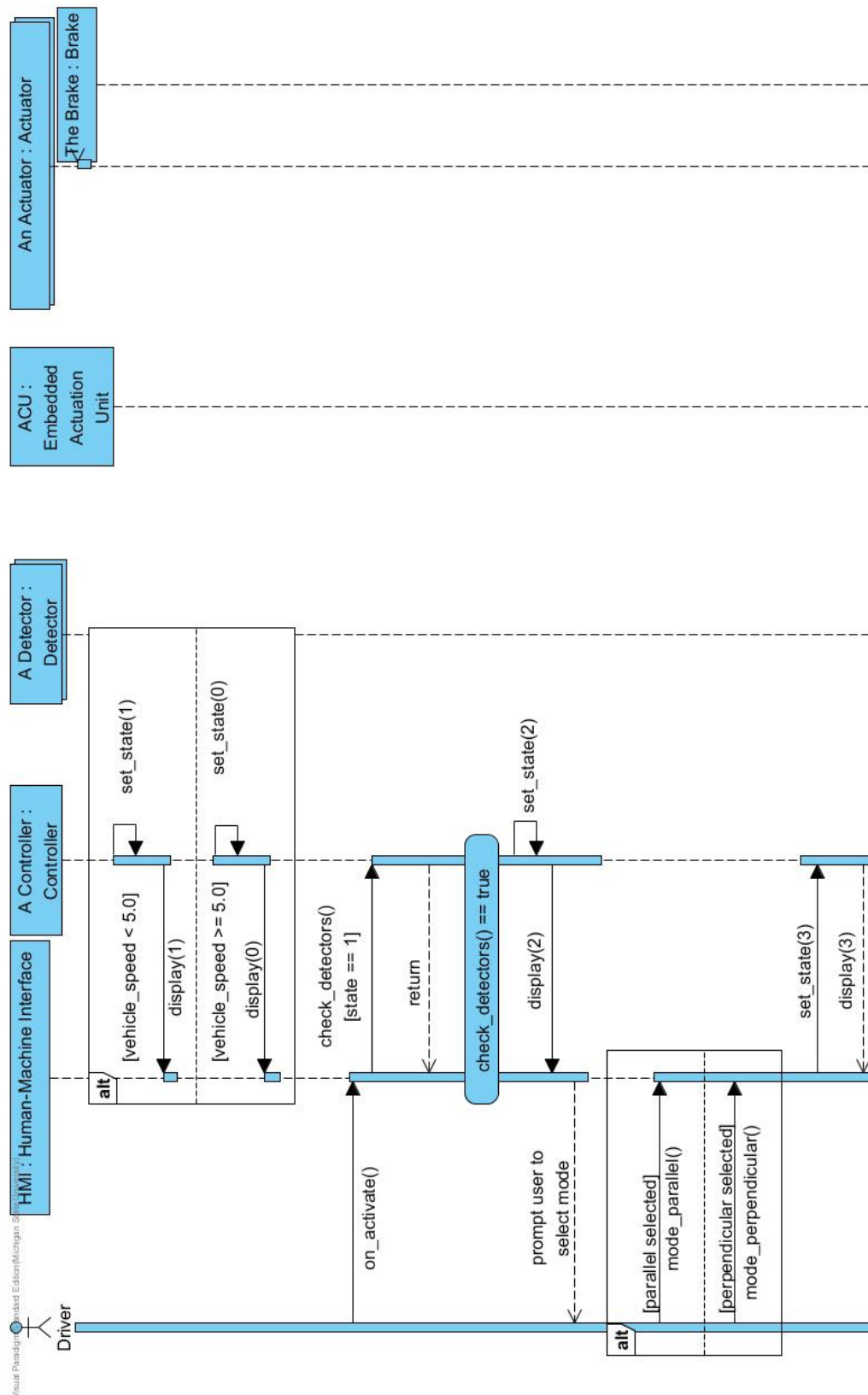


Figure 3.1: The sequence diagram for the APA system. (part 1)

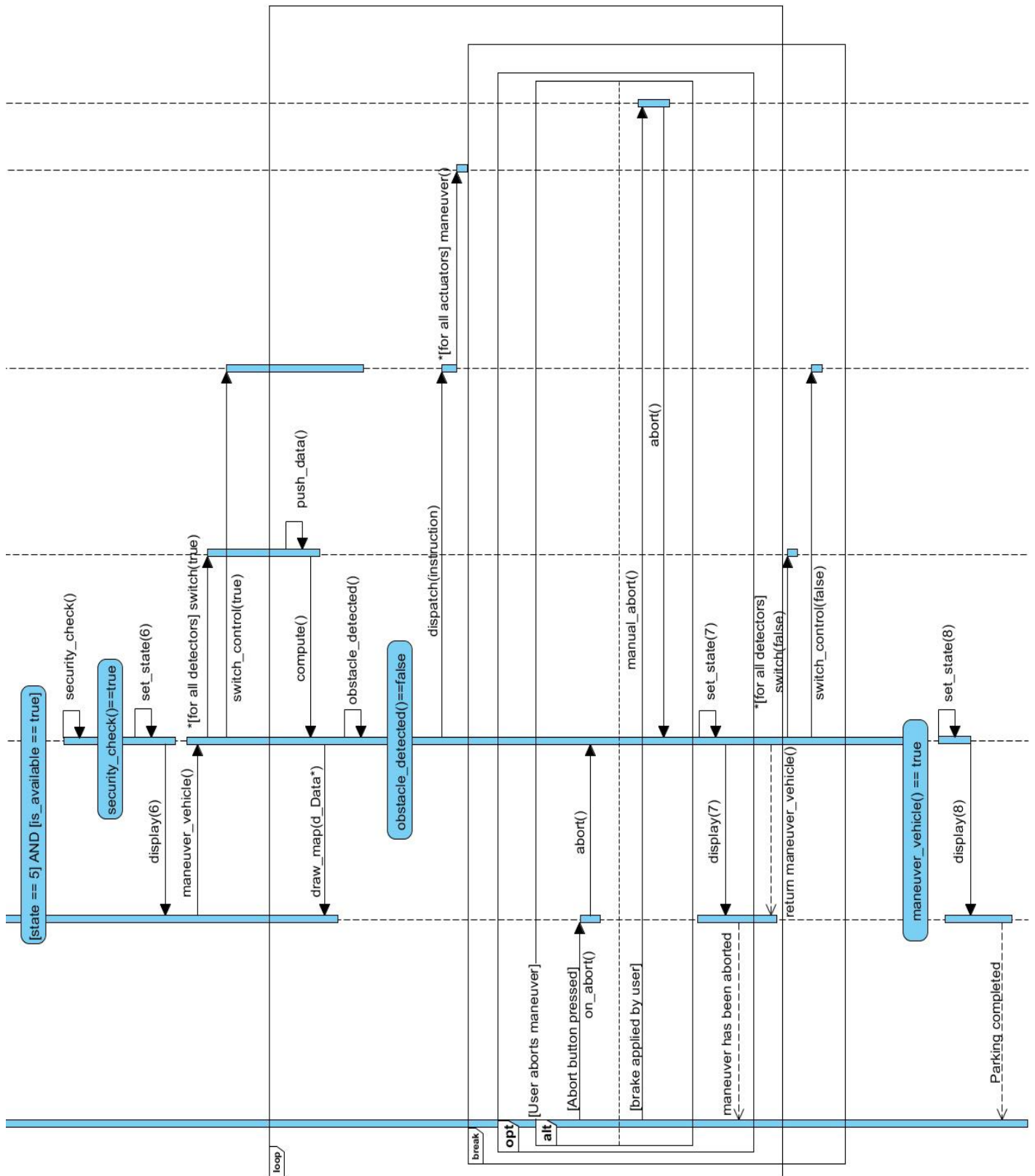


Figure 3.3: The sequence diagram for the APA system. (part 3)

State Diagram - Active Park Assist System

This model (*Figure.4*) captures the many states the system can be in as it moves through events triggered by the system or external actors. The blue sections represent these specific states which can have internal actions that occur on entry or exit. The arrow between the boxes are the transitions that transfer the system between the states based on conditions that are labeled near the arrow. Once a condition is met, a transfer between states can occur, but only to a state with a connecting arrow. Descriptions are listed after the diagram for specifics on what each state entails.

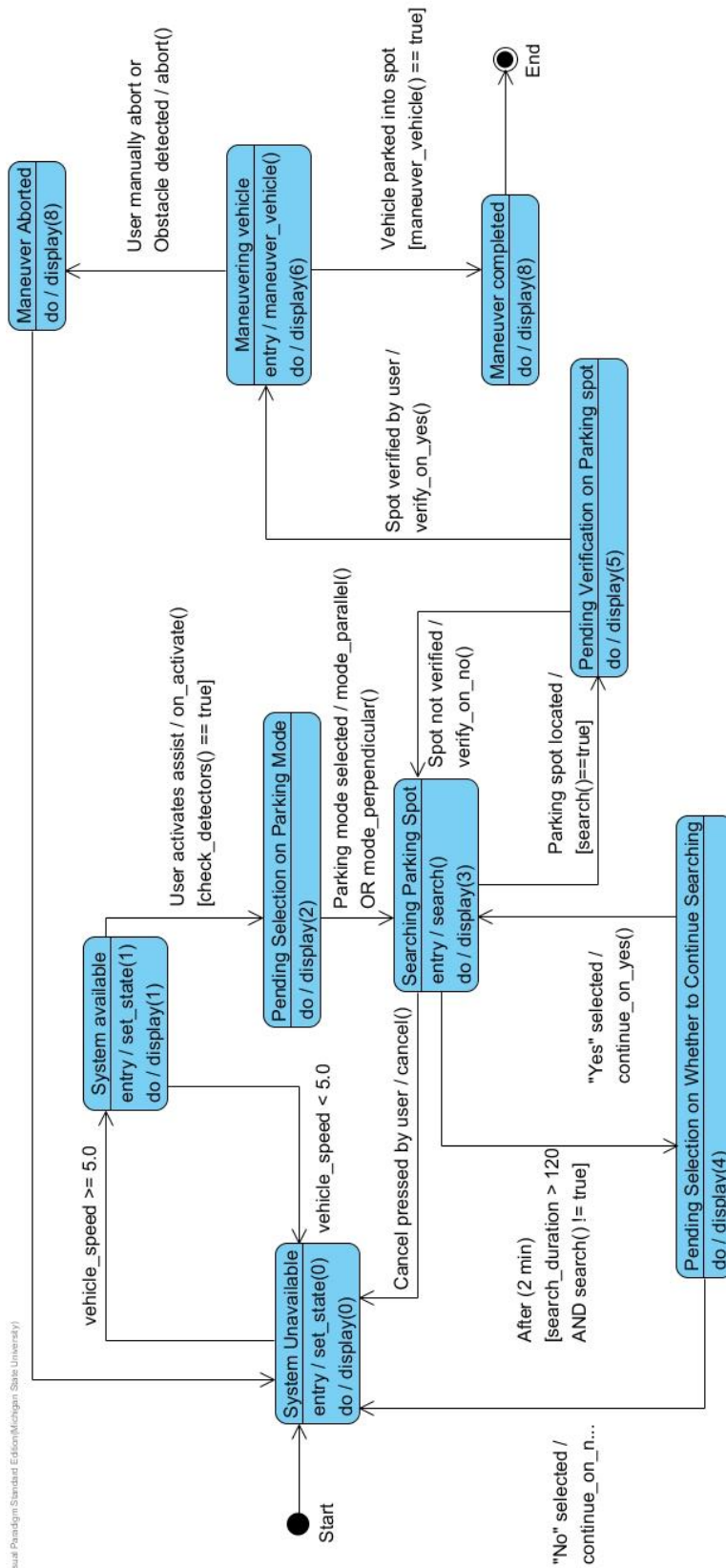


Figure 4: The state diagram for the APA system

Template based on IEEE Std 830-1998 for SRS. Modifications (content and ordering of information) have been made by Betty H.C. Cheng, Michigan State University (chengb at chengb.cse.msu.edu)

System unavailable: In this state, the system is inactive, until the speed of vehicle less than 5 mph.

System available: In this state, the system is idle, the user can initiate a searching at any time, unless the speed of vehicle exceeds 5 mph.

Pending selection on parking mode: In this state, the system is waiting for user's selection on whether to search a parallel or a perpendicular spot.

Searching parking spot: While in this state, the system invokes the detectors to search nearby available parking spot.

Pending on selection whether to continue searching: In this state, the system is waiting for user's selection on whether to continue searching.

Pending selection on parking spot: In this state, the system is waiting for user's choice whether to park into the spot which is currently provided.

Conducting vehicle into parking spot: In this state, the system is automatically maneuvering the vehicle into the parking spot, based on the information provided by controller.

Maneuver aborted: In this state, the system ceases to maneuver and becomes deactivated.

5 Prototype

The APA prototype will show exactly how a user will interact with the system, while demonstrating what function the system will be performing during that action. It will show an interactive SYNC interface, that will be similar to the interface used on current SYNC enabled Ford vehicles. The user of the prototype will be able to run the system by pressing a button to activate Active Park Assist. The interface will then be a live simulation of what would happen if the user was using this in an actual vehicle. The user can select a type of parking spot, and the prototype will demonstrate that the system will begin looking for that type of parking spot. The prototype will then have interfaces that the user can interact with, and show different interfaces based on the user interaction until the simulation is complete, or the system is aborted.

5.1 How to Run Prototype

To run the Prototype V1, an internet connection and modern browser is all that is needed. The prototype can be found through the team's website under the *About* tab. The link will open up to the visual presentation made through "Prezi". No account is needed to use this service, simply click through the presentation with the arrow keys on screen, and the slides have detailed descriptions for the actions on screen. No additional plugins or configuration settings are needed.

Prototype V1: <https://prezi.com/7rydocluq21p/prototype-apa-2/#>

Prototype V2 can also be found on the team's website under the *About* tab. This link will open up a web page containing a series of buttons and a centered image of the SYNC interface screen. The user of the prototype can run a simulation of the APA system by interacting with the buttons and observing the changes in the SYNC interface screen. The prototype buttons are designed to simulate different scenarios within APA, and represent what would actually happen when the buttons on the SYNC interface screen are pushed.

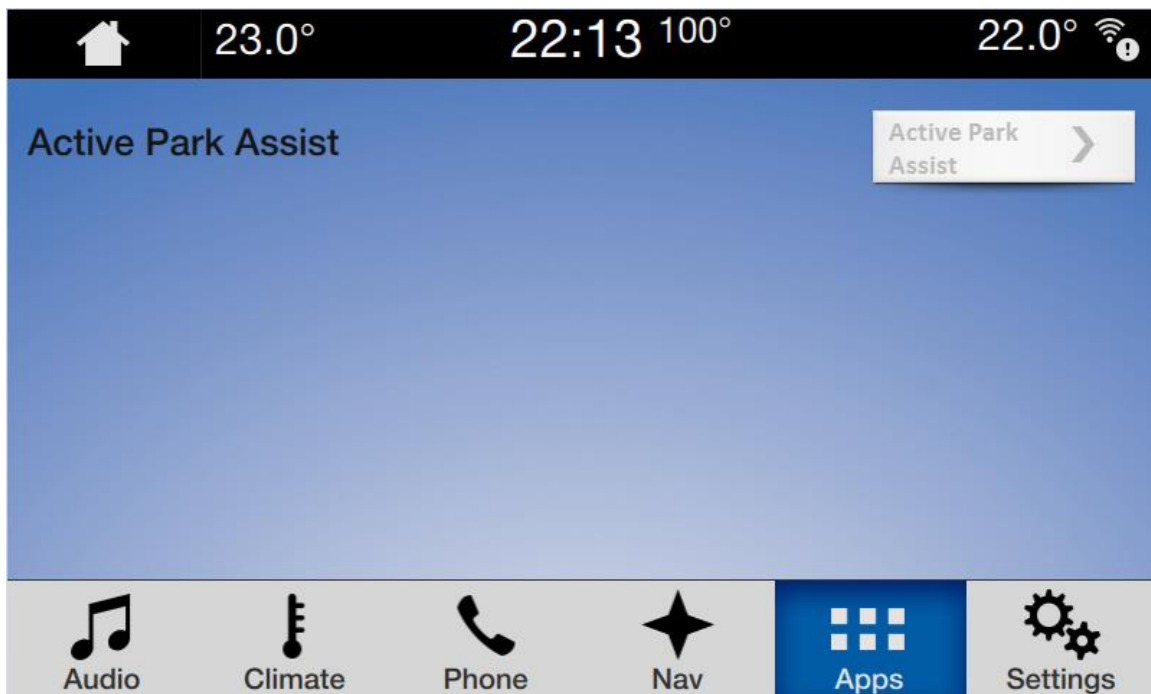
5.2 Sample Scenarios

The driver of the car wishes to park and slows to a speed under 5 mph near some available parking spots. Being below the maximum speed threshold of 5 mph, the button to activate the system is now active on-screen to touch. After the driver presses it to activate the APA system, the driver is prompted to choose the type of parking: parallel or perpendicular. Depending on the selection, the ultrasonic sensors and cameras are used to automatically locate nearby parking spots, up to a duration of two minutes.

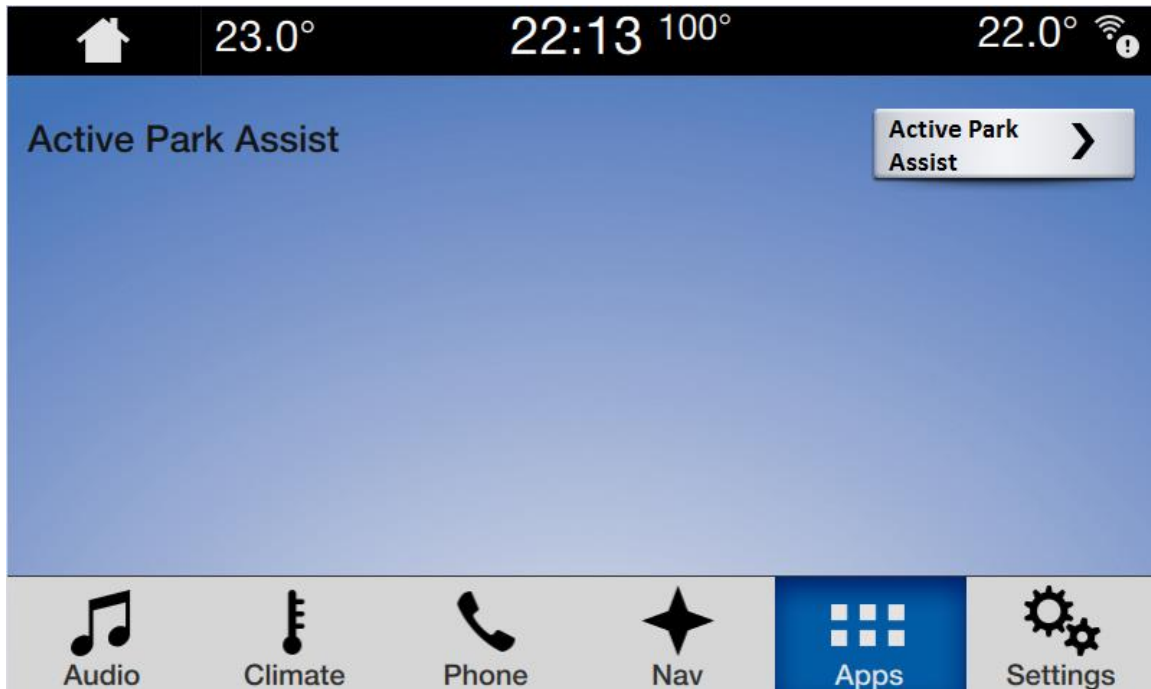
After locating a spot, the driver sees a rendering of the spot and must validate it by pressing a button on the display. Or, if the spot is not preferable, the driver may select to be offered another spot. This continues until a spot is chosen or there are no spots left in the nearby area, in which the driver is prompted to continue or exit the system.

Once a spot is chosen, the system maneuvers the vehicle into the spot, stopping for any obstacles that may appear in the path. During this maneuver, the driver may press the brake pedal to stop the maneuver and abort the system, otherwise the system completes the parking, leaving the car in park and notifying the driver that the process has been completed, after which it will close.

Screenshots from Prototype v2:



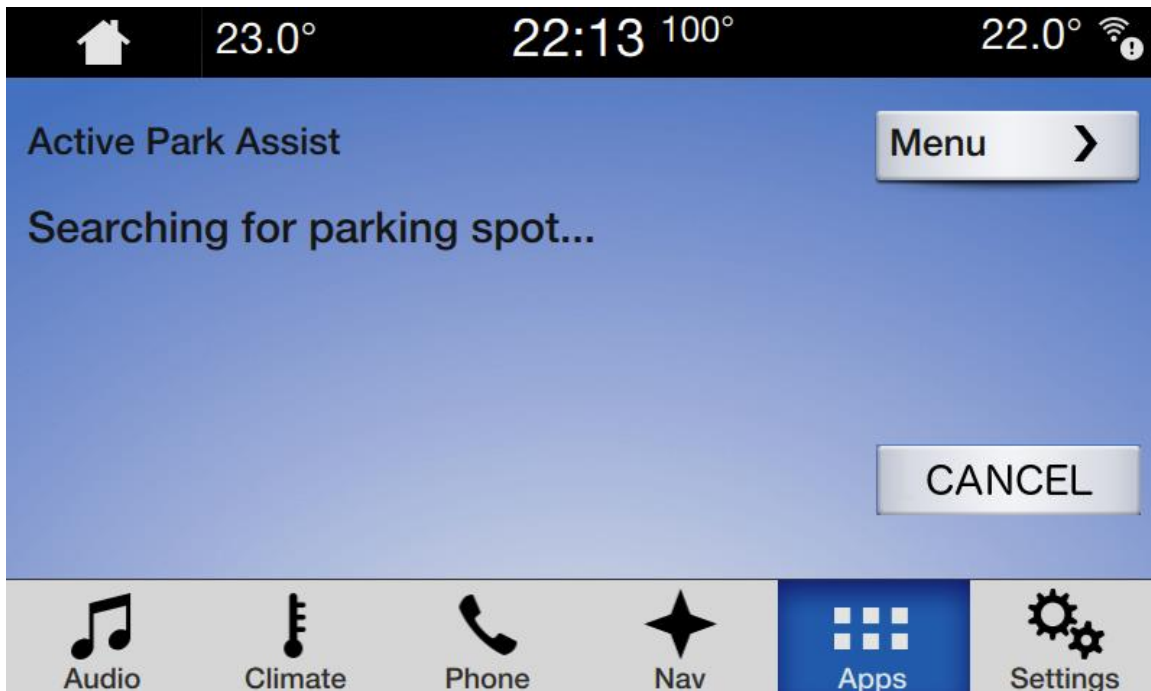
The SYNC interface screen shown when the vehicle is traveling over 5 MPH. The button to activate APA is grayed out in this scenario.



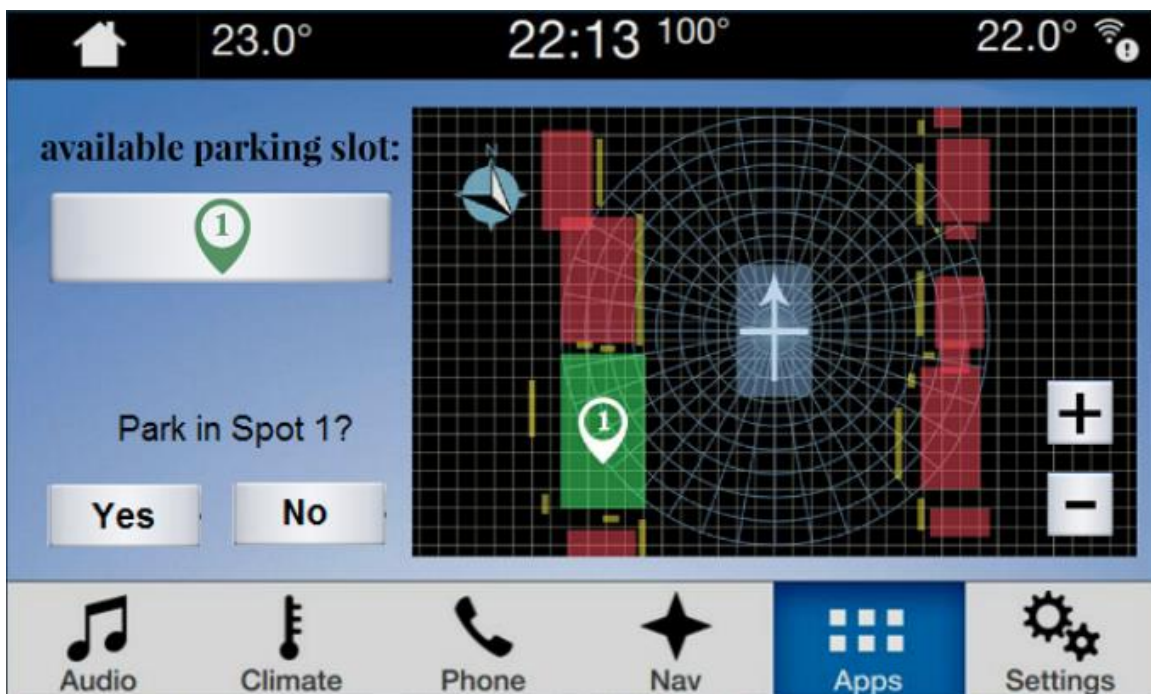
The SYNC interface shown when the vehicle is traveling 5 MPH or less, allowing the APA button to be active.



The SYNC interface screen shown when APA is activated. The driver can then select the type of parking spot.



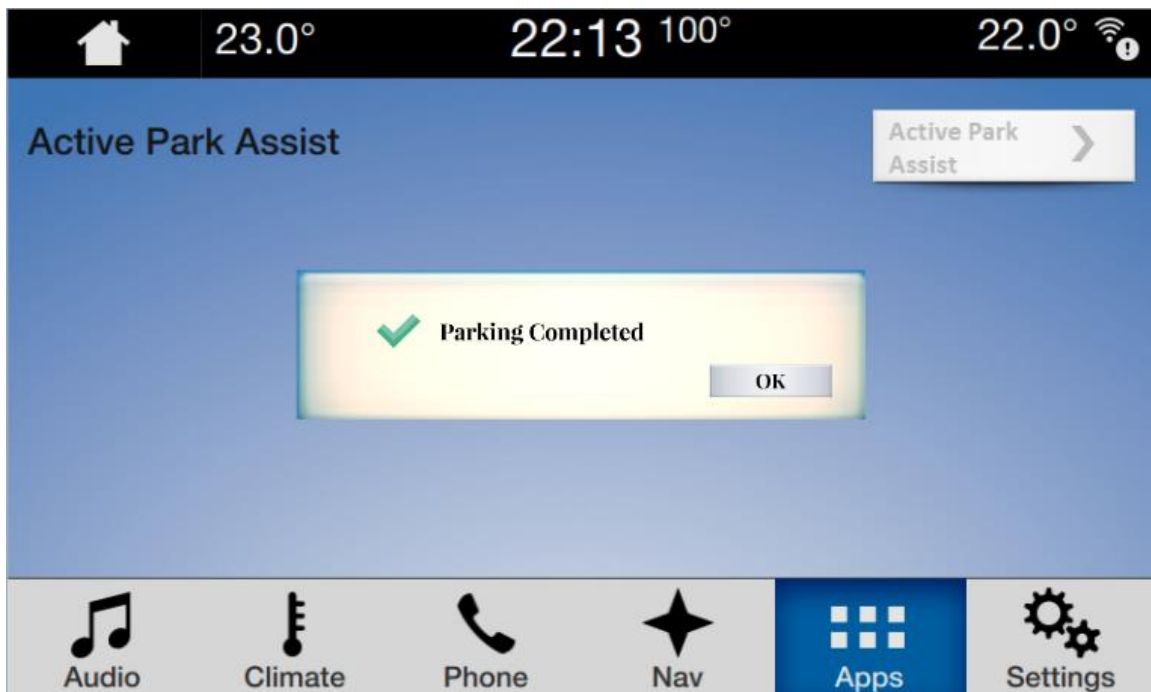
The SYNC interface screen shown when a type of parking spot is selected. APA will begin searching for that type of parking spot.



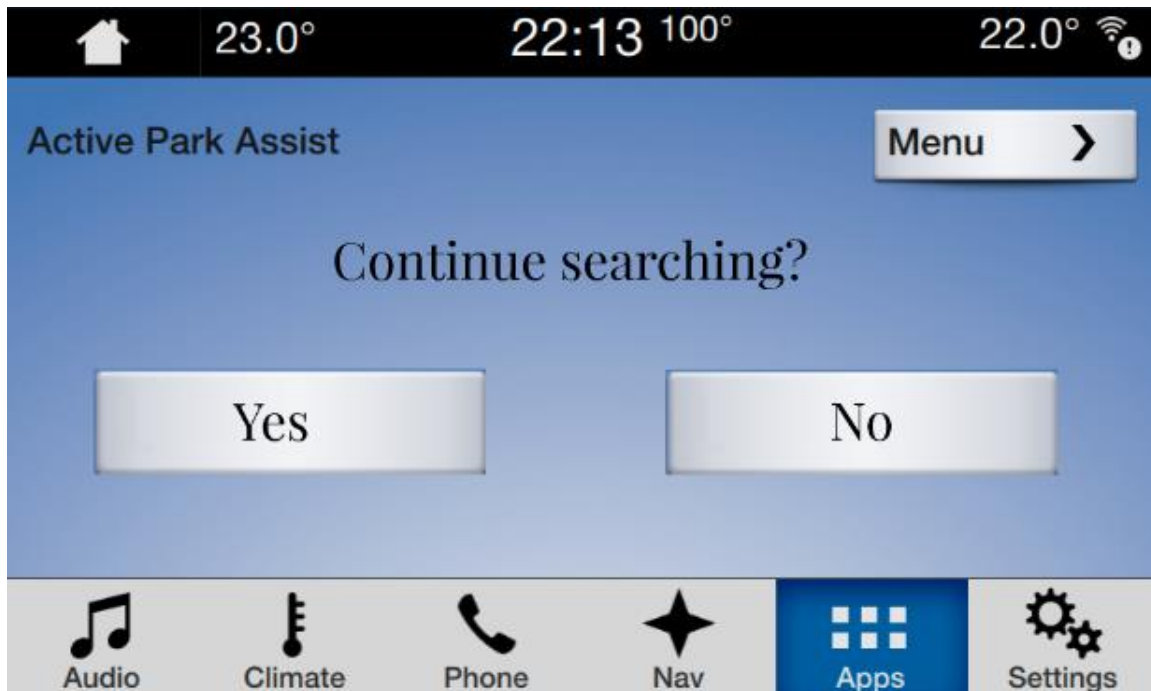
The SYNC interface screen shown when APA finds an available parking spot. The driver can then validate or deny the parking spot.



The SYNC interface screen shown when the driver validates an available parking spot. APA will then maneuver into the parking spot.



The SYNC interface screen shown when maneuvering into the parking spot is complete. The system will then abort.



The SYNC interface screen shown when APA is unable to find an available parking spot. The driver can choose to continue or stop searching for a spot. If the driver selects no, APA will abort.

6 References

- [1] C. Rooks, C. Mejia, E. Castillo, Y. Wang, and W. McGee. *Active Park Assist*. Michigan State University, 21 October. 2016. Web. "www.cse.msu.edu/~cse435/Projects/F2016/Groups/APA2/web/about.html".

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 cse.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.