**System Requirements Specification**

**EcoCar UI**

**Senior Design, Fall, 2020**

Team Name: EcoCar UI Team

Team Members:

* Sabrina Yepez
* Nathan Rose
* David Fadini
* Raymond Kwasneski (Sky)
* Teja Tiriveedhi

Contents of this Document

Introduction

* System to be Produced
* Applicable Standard
* Definition, Acronyms, and Abbreviations

Product Overview

* Assumptions
* Stakeholders
* Event Table
* Use Case Diagram
* Use Case Descriptions

Specific Requirements

* Functional Requirements
* Interface Requirements
* Physical Environment Requirements
* Users and Human Factors Requirements
* Documentation Requirements
* Data Requirements
* Resource Requirements
* Security Requirements
* Quality Assurance Requirements

**Section 1: Introduction**

System to be Produced:

EcoCar is a semi-autonomous vehicle that aims to assist the user while driving. EcoCar utilizes Adaptive Cruise Control (ACC) to maintain a certain distance from itself and the car in front of it. To communicate to the user what the car’s internal database is viewing, the EcoCar UI aims to provide a system to allow a user to have live feedback when using the adaptive cruise control feature. This UI will display the car itself as well as any foreign body outside of the car.

Applicable Standards

* There are no current standards designated for the system requirements

Definitions, Acronyms, and Abbreviations

* ROS: Robot Operating System
* UI: User Interface
* ACC: Adaptive Cruise Control
* TPS: Throttle Position Sensor

**Section 2: Product Overview**

Assumptions:

* The ACC System is complete and works adequately
* The UI software is able to access all the environment information through ROS
* The UI software is able to access the state of the adaptive cruise control through ROS
* The UI software is able to run on a functional system
* Drivers understand the fundamentals of driving and operating a car
* The ROS master node has been created
* ROS melodic is being used

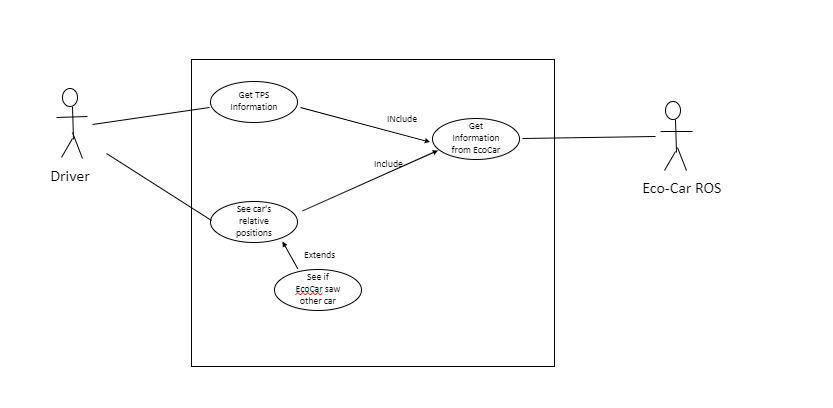
Stakeholders:

* EcoCar Controls Team: This is the team that has developed the hardware and software to make the EcoCar run semi autonomously. This team would like to see their work displayed in a clear manner for the user to view.
* Driver: This person would like to see a fully developed UI system that will display what the car is currently seeing. It is important to this user that the UI is as correct as possible, so that they can make proper driving decisions when needed.
* Department of Transportation: These people want to ensure that the UI will not negatively affect the ability of the driver to see traffic or severely distract or disrupt the driver from being able to manually control this vehicle.

Event Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Event Name | External Stimuli | External Responses | Internal data and state |
| Notify user of object being too close to the car while in ACC | Car that is less than x feet away | The user may choose to turn off ACC and take manual control of the car | The car’s ACC will kick in to slow the car down to get x feet away from car |
| Turn on ACC | User pressed ACC on the steering wheel | The car’s ACC will begin to run | The UI will display the cars position and surroundings |
| Show updated position | An external car’s relative position changes | EcoCar will update the position of the car for the user | The current location of the car is updated |

Use Case Diagram (Nathan Rose)



Use Case Descriptions:

As the Driver I want to be able to look at the UI system to determine the current status of the TPS system

As the Driver I want to be able to look at the UI system to determine a nearby car’s relative position

As the Driver I want to be able to look at the UI system to determine if ECO car can se a nearby car

As the Ecocar I want to be able to give the UI system information to display

Section 3: Specific Requirements

|  |
| --- |
| No: 1 |
| Statement: The UI system shall turn on ACC when the ACC button on the steering wheel is pressed and ACC is not currently running. |
| Source: EcoCar Team |
| Dependency: No. 2, No. 3 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The UI will display a symbol to clearly show the user that the ACC is on, and the car’s internal computer is running ACC. |
| Revision History: Sabrina Yepez, September 23rd 2020, Revision 1 |

|  |
| --- |
| No: 2 |
| Statement: The UI system shall turn off ACC when the ACC button on the steering wheel is pressed and ACC is currently running. |
| Source: EcoCar Team |
| Dependency: No. 1 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The car’s display will no longer display that the car is in ACC and the car will run without cruise control. |
| Revision History: Sabrina Yepez, September 23rd 2020, Revision 1 |

|  |
| --- |
| No: 3 |
| Statement: The UI system shall turn off ACC, when ACC is currently running, and the user presses the brake pedal. |
| Source: EcoCar Team |
| Dependency: No. 1 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The car’s display will no longer display that the car is in ACC and the car will run without cruise control. |
| Revision History: Sabrina Yepez, September 23rd 2020, Revision 1 |

|  |
| --- |
| No: 4 |
| Statement: The system shall display the position of the car on the UI when ACC is turned on. |
| Source: EcoCar Team |
| Dependency: No. 1, No. 2, No. 3 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The car’s display will show a visualization of the cars position in respect to its surroundings. |
| Revision History: Sabrina Yepez, September 25th 2020, Revision 1 |

|  |
| --- |
| No: 5 |
| Statement: The car shall display surrounding objects on the UI when an object is detected |
| Source: EcoCar Team |
| Dependency: No. 1, No. 2, No. 3 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The car’s display will show a foreign object in the same position as it is relative to the car on the UI display. |
| Revision History: Sabrina Yepez, September 25th 2020, Revision 1 |

3.1 Functional Requirements (Teja Tiriveedhi, David Fadini)

* Adaptive cruise control shall have the ability to be turned off and on
* The UI System shall turn on when car ignition is on and shall turn off when the car ignition is off
* Throttle position sensor shall be working with the user interface
* The UI System shall be displaying information no more than 10 milliseconds old data

3.2 Interface Requirements (Nathan Rose, David Fadini, Raymond Kwasneski)

* The UI system shall receive relevant ACC information through ROS messages
* The System shall display relative information of the nearby cars to the user
* The System shall display information of the TPS to the user
* The System shall display relevant information of ACC on and off
* The UI System shall be displaying information no more than 10 milliseconds old

3.3 Physical Environment Requirements (David Fadini, Raymond Kwasneski)

* The UI system shall work on the dashboard of a car
* The UI system shall work in 95% humidity
* The UI system shall work at all times of the day

3.4 User and Human Factors Requirements (David Fadini, Teja Tiriveedhi)

* The system shall support users who know how to use ACC
* The system shall support users who do not know how to use ACC
* The system shall display any issues that the on-board computer detects in the ACC system

3.5 Documentation Requirements (Teja Tiriveedhi, Sabrina Yepez)

* The documentation required for this system is online
* The assumed skill level of the audience for documentation is general engineering knowledge
* The documentation shall meet the scrum process standards and meet the engineering standards that highlights the process from start to end
* The documentation shall have the designs and implementations over time to time similar to iterative model

3.6 Data Requirements (Teja Tiriveedhi, Sabrina Yepez)

* There are currently no data calculations that need to be made. The UI system will not be calculating distance from outside objects, but rather gathering already calculated data from the on-board computer.
* Data that needs to be retained is if the ACC system is on. This must be analyzed and displayed all the times when the car ignition is on.
* Data would consist of car features and needs such as how much gas is remaining and the outside weather/humidity

3.7 Resource Requirements (Teja Tiriveedhi, Sabrina Yepez)

* To build the system, soldering and coding skills are needed.
* The use the system a training is required.
* To maintain the system a vast understanding of the systems UI is needed to troubleshoot and solve any maintenance issues.
* The system will require available space on the dashboard of the car to function.
* The system will require x Watts of power, but no more than y Watts of power.
* The system will require funding to buy the necessary equipment, such as the ROS and display.
* The system will require basic python programming skills in order to write scripts that can send and receive messages
* The system will require object detection system for the UI to detect the cars relative positions
* The system will require a throttle position sensor to be integrated with the UI

3.8 Security Requirements (Raymond Kwasneski)

* The ACC shall only have access to the car with the listed methods in this section of requirements
* The ACC shall have access to the car’s acceleration
* The ACC shall have access to the car’s breaks
* The ACC shall have access to the car’s sensor information that contains the distance of objects in proximity of the car
* The ACC system shall turn off with the “ACC Button” if the system is working properly or not
* The ACC system shall turn off by breaking if the system is working properly or not
* The ACC shall not be turned off/on by the car – only the user

3.9 Quality Assurance Requirements (Teja Tiriveedhi, Sabrina Yepez)

* The UI system must be extremely reliable as it is important that the user know what is going on in the cars internal system
* The UI system will have its reliability demonstrated as the user can easily verify if what they are seeing on the display is what is being seen outside of the car
* If the UI system detects a fault, it must notify the user so that they may take manual control of the car until the fault is fixed
* The system must always be available that the car is turned on
* The response time of this system is 100 ms, so that the system can be perceived as instantaneous to the user.
* The UI system shall be able to detect if the car ignition is on or off
* The UI system shall display the correct readings including how much gas is left
* The UI system shall display the seat belt sign that matches correctly with the car built in display
* The UI system is extremely reliable that can help navigate the user the adaptive cruise control system