**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

* df4rSystem 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. When the user first enters the car, they will undergo a quick training that will play a video that describes what the system is doing and what they should be seeing and doing. 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 a reference for the user as well as relevant and important objects outside of the car such as pedestrians and other vehicles. To achieve all these goals, we will install an external display on the dashboard of the car that is connected to a Raspberry Pi. This Raspberry Pi will gather data from EcoCAR’s on-board computer and run it through a python program to display all relevant information in an intuitive manner on the external display.

Applicable Standards

* FMVSS No. 101: Controls and displays

Definitions, Acronyms, and Abbreviations

* ROS: Robot Operating System
* UI: User Interface
* ACC: Adaptive Cruise Control

**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 written in Python and functional on a Raspberry Pi
* The Raspberry Pi is able to transmit data to an external display
* Drivers understand the fundamentals of driving and operating a car
* The ROS master node has been created
* ROS melodic is being used

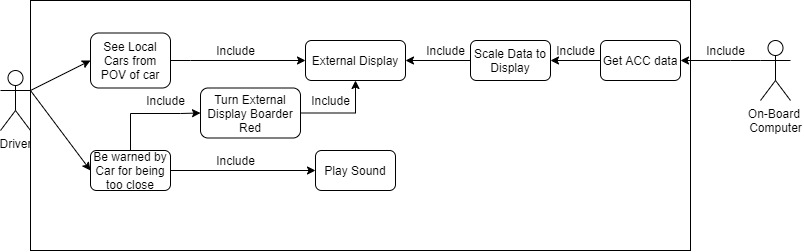
Stakeholders:

* EcoCAR Organizers: The EcoCAR Organizers would like to see their work inside the car visible on the UI System to verify that their data is flowing properly. The EcoCAR Organizers would also like the UI System to be innovative and intuitive enough to be a strong competitor in the EcoCAR competition.
* 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 | The On-Board computer sends data saying a vehicle is less than 6 feet away. | The ACC will take control and being slowing down. The user may also decide to take manual control to avoid collision. | The UI Display will have a red border and the distance between the EcoCAR and the external vehicle will be displayed. |
| 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, and the On-Board Computer | EcoCAR will update the position of the car for the user | The current location of the car is updated |

Use Case Diagram



Use Case Descriptions:

As the Driver I want to be able to look at the UI system and it displays nearby cars’ relative position

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

As the EcoCAR I want to be able to integrate the UI System very easily with the rest of my subsystems.

Section 3: Specific Requirements

|  |
| --- |
| No: 1 |
| Statement: The UI system shall always be on while the car is running |
| Source: EcoCAR Team |
| Dependency: No. 2, No. 3 |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The UI will display the running program on the screen |
| Revision History: David Fadini, October 19th 2020, Revision 2 |

|  |
| --- |
| 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 relevant 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

* The UI System shall turn on when car ignition is on and shall turn off when the car ignition is off
* The UI System shall be displaying information no more than 100 milliseconds old data

3.2 Interface Requirements

* The UI system shall receive relevant ACC information through ROS messages
* The System shall display the relative local position of the nearby obstacles to the user
* The System shall display the lane type to the user
* The System shall display the lanes location to the user
* The UI System shall display information no more than 100 milliseconds old
* The UI System shall display the cruise control set speed
* The UI System shall display a button to start the brief training video
* The UI external display shall not block the view of the driver from seeing outside the windshield

3.3 Physical Environment Requirements

* The UI system shall work on the dashboard of a car
* The UI system shall work from the car’s Always On 12V power line
* The UI system shall work in 30° F to 120° F
* 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

* The system shall support all users that are able to drive the car
* The system shall provide a brief training video for users to learn how the ACC system and UI system works
* The system shall get the users attention by turning the top of the screen red when the car detects a vehicle less than 10 feet away
* The system shall get the users attention by playing a short chime sound when the car detects a vehicle less than 10 feet away
* The system shall display any issues that the on-board computer detects in the ACC system
* The system shall display a top down view of the surrounding lanes, displaying data that the car sees

3.5 Documentation Requirements

* 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

* There will need to be a calculation to scale the position of the cars to the screen that is being utilized
* Data would consist of car information such as the current speed that the ACC is set to

3.7 Resource Requirements

* To build the system, soldering and coding skills are needed.
* For use the system, an optional training video can be played.
* The system will require available space on the dashboard of the car to function.
* The system will require 5V of DC voltage from the EcoCAR
* The system will require funding to buy the necessary equipment, such as the Raspberry pi, step down converter, and display.
* The system will require basic python programming skills in order to write scripts that can send and receive messages

3.8 Security Requirements

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

3.9 Quality Assurance Requirements

* The UI system will demonstrate what is seen 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 will start once the car turns 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 current ACC set speed