**UI System Requirements Specification**

**EcoCAR UI**

**Senior Design, Fall, 2020**

Team members:

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

**Contents of this Document**

Introduction

* UI system to be Produced
* Applicable Standard
* Definitions, Acronyms, and Abbreviations

Product Overview

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

Specific Requirements

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

**Section 1: Introduction**

UI System to be Produced:

EcoCAR is a semi-autonomous vehicle whose goal is to help its driver drive better. EcoCAR uses Adaptive Cruise Control (ACC) to maintain a certain distance from itself and an obstacle in front of it. (An obstacle includes external cars, trucks, and pedestrians.) When a driver first enters the EcoCAR, they will get training on how to use the UI system; the training is watching a video. The EcoCAR UI (user interface) team will make a UI that informs the driver about the ACC. The UI will display the EcoCAR itself as a reference for the driver as well as external obstacles. To achieve all these goals, the UI team will install a display on the dashboard of the EcoCAR that is connected to a Raspberry Pi. The Raspberry Pi will gather data from the EcoCAR’s on-board computer and run it through a Python program to 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 working properly.
* The UI software can access all environment information through the ROS.
* The UI software can access the state of the adaptive cruise control through the ROS.
* The UI software is written in Python and works on a Raspberry Pi.
* The Raspberry Pi can transmit data to an external display.
* The ROS master node has been created.
* ROS data is being used.

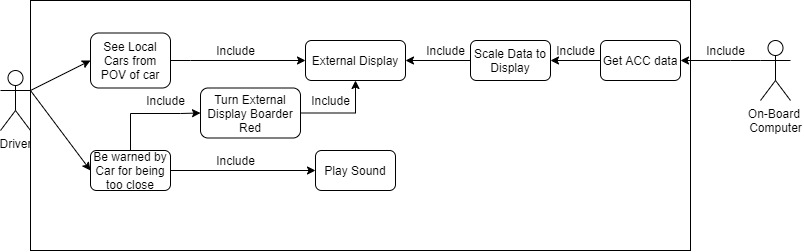
Stakeholders:

* EcoCAR Organizers: The EcoCAR organizers want their work inside the EcoCAR: visible on the UI system to verify that their data is flowing properly. The EcoCAR organizers want the UI system to be innovative and intuitive enough to be a strong competitor in the EcoCAR competition.
* EcoCAR Controls Team: This team developed the hardware and software to make the EcoCAR run semi-autonomously. This team wants their work displayed clearly for the driver to view.
* Driver: This person wants a fully developed UI system that displays what the EcoCAR can see. It is important to this person that the UI is correct so that this person makes proper driving decisions.
* 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 the EcoCAR.

Event Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Event Name | External Stimuli | External Responses | Internal data and state |
| Notify driver of obstacles too close to the front of the EcoCAR while the UI system is active. | The on-board computer sends data saying an obstacle is less than 10 feet in front of the EcoCAR. | The UI display will have a red border with the distance between the EcoCAR and the external obstacle in front of the EcoCAR displayed. | The ACC will provide the internal data and state. |
| Turn activate the UI system. | Driver pressed ACC on the steering wheel. | The UI will display the EcoCAR’s position and surroundings. | The UI will become active. |
| Show updated position. | An external obstacle’s relative position changes. | The UI will update the position of the external obstacle for the driver on the display. | The current location of the external obstacle is updated. |

Use Case Diagram



Use Case Descriptions:

As the driver, I want to look at the UI system and have it display an external obstacle’s relative position.

As the driver, I want to look at the UI system to determine if the EcoCAR can see an external obstacle.

As the EcoCAR, I want to integrate the UI system easily with the rest of my subsystems.

Section 3: Specific Requirements

|  |
| --- |
| No: 1 |
| Statement: The UI system shall always be on (but not always active) while the EcoCAR is running. |
| Source: EcoCAR Team. |
| Dependency: None. |
| 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 display the position of the EcoCAR on the UI while ACC is on. |
| Source: EcoCAR Team. |
| Dependency: No. 1. |
| Conflicts: None. |
| Supporting Materials: None. |
| Evaluation Method: The display will show a visualization of the EcoCAR’s position in respect to its surroundings. |
| Revision History: Sabrina Yepez, September 25th, 2020. Revision 1. |

|  |
| --- |
| No: 3 |
| Statement: The EcoCAR shall display external obstacles on the UI when they are detected. |
| Source: EcoCAR Team. |
| Dependency: No. 1. |
| Conflicts: None. |
| Supporting Materials: None. |
| Evaluation Method: The display will show an external obstacle relative to the EcoCAR. |
| Revision History: Sabrina Yepez, September 25th, 2020. Revision 1. |

3.1 Functional Requirements

* The UI system shall become active/inactive when signaled to do so.
* The UI system shall update display information at an interval of no more than 100 milliseconds.

3.2 Interface Requirements

* The UI system shall get ACC information with ROS messages.
* The UI system shall display the relative position of external obstacles to the driver.
* The UI system shall display the lane type to the driver.
* The UI system shall display the lane’s location to the driver.
* The UI system shall display information at an interval of no more than 100 milliseconds.
* The UI system shall display the set cruise control speed.
* The UI system shall display a button to start the brief training video.
* The UI 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 the EcoCAR.
* The UI system shall work on 5V DC converted from the EcoCAR’s 12V DC power line.
* The UI system shall work in temperatures from 30oF (~ -1oC) to 120oF (~ 48oC).
* The UI system shall work in relative humidity from 0% to 95%.
* The UI system shall work in daytime and nighttime.

3.4 User and Human Factors Requirements

* The UI system shall support all drivers.
* The UI system shall provide a brief training video for drivers on how the UI system works.
* The UI system shall get the driver’s attention by turning the top of the screen red when the EcoCAR detects an obstacle less than 10 feet away in front of the EcoCAR.
* The UI system shall get the driver’s attention by playing a short chime sound when the EcoCAR detects an obstacle less than 10 feet away in front of the EcoCAR.
* The UI system shall display any issues that the on-board computer detects in the ACC system.
* The UI system shall display a top down view of the surrounding lanes.

3.5 Documentation Requirements

* The documentation required for the UI 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 like an iterative model.

3.6 Data Requirements

* There will need to be a calculation to scale the position of obstacles.
* The data the UI gets will be what the ACC gives: set cruise control speed, external obstacles, and whatever else is needed for the UI.

3.7 Resource Requirements

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

3.8 Security Requirements

* The UI system shall be read-only, meaning that the UI system is not able to send information to the EcoCAR.
* The UI system shall turn off with the ACC button.

3.9 Quality Assurance Requirements

* The UI system displays what the EcoCAR sees.
* If the UI system detects a fault, it must notify the driver so that they may take manual control of the EcoCAR until the fault is fixed.
* The UI system turns on (but not active) when it gets power.
* The display updates at an interval of no more than 100 milliseconds.
* The UI system displays the set cruise control speed.