EcoNav

Design document

Antoine Gagnon

December 20, 2019

Contents

1 Overview

1.1 Goal

The main goal of EcoNav is to give tips to drivers based on their driving habits to improve their fuel usage.

1.2 Method

In order to record users driving habits, EcoNav requires a small device such as a Raspberry Pi to act as a server and gather data from the ODB2 port of the vehicule via a scanner (wired or bluetooth) and video footage from a Raspberry Pi camera. The acting server could then stream live data to a smartphone or any other network enabled device as well as run analysis on the data.

2 Features

2.1 EcoNav Score

The EcoNav score is a metric to measure the fuel efficiency of the user based on their driving habits. This score comes as a mark (A+, A, etc.) and is throughly explained to the user.

2.2 Real-time analysis

Live readings from the system and how it affects their fuel consumption are be made available to the user, streaming it to a smartphone attached on the dash of their car.

2.3 Trip overview

Each trip is archived and exports are made available via the EcoNav application highlighting user actions that affected their fuel economy and footage of what happened at the time as well as providing raw data, statistics and an EcoNav score.

2.4 Lifetime overview

Exports are also be available from data over the whole lifetime of the application. Enabling users to see changes over time.

3 Technical design

3.1 Hardware

3.1.1 Data collection

Car data

The car data is read from an ODBII port scanner, more specifically an ELM 327 mini, over bluetooth and stored on the Raspberry Pi SD Card.

Video footage

Using a Camera module on the Raspberry Pi mounted on the dash

3.1.2 Live monitoring and feedback

Live data and feedback is streamed directly to the user's smartphone mounted on the dash.

3.2 Architecture

This is a simplified overview of the different modules in the app as well as the technology used.

3.2.1 Back-end

The backend-end is the software that runs on the acting server (Raspberry Pi).

Its job is to collect car data and footage, run analysis on the data and serve it to clients.

The backend is written in TypeScript, running on NodeJS.

Modules

- Car data reading
- Video reading
- Data storage
  - Raw data
  - Data Analysis
  - Video footage
  - Other metrics and statistics
- Data analysis
- Data serving
  - Over HTTP with GraphQL
  - Streaming via WebSockets for data, MJPEG for video feed
  - Exports

3.2.2 User Interface

There is multiple user interfaces for EcoNav.

Mobile

The mobile interface provides live data reading as well as live driving analysis and tips.

(To be determined) The app will either be a native Android app or a web application.

Desktop

The desktop interface will be provided via a web application, allowing to see more in-depth analysis of the data as well as footage and more. The desktop app is using React with JavaScript.