Group E1: Drivaid

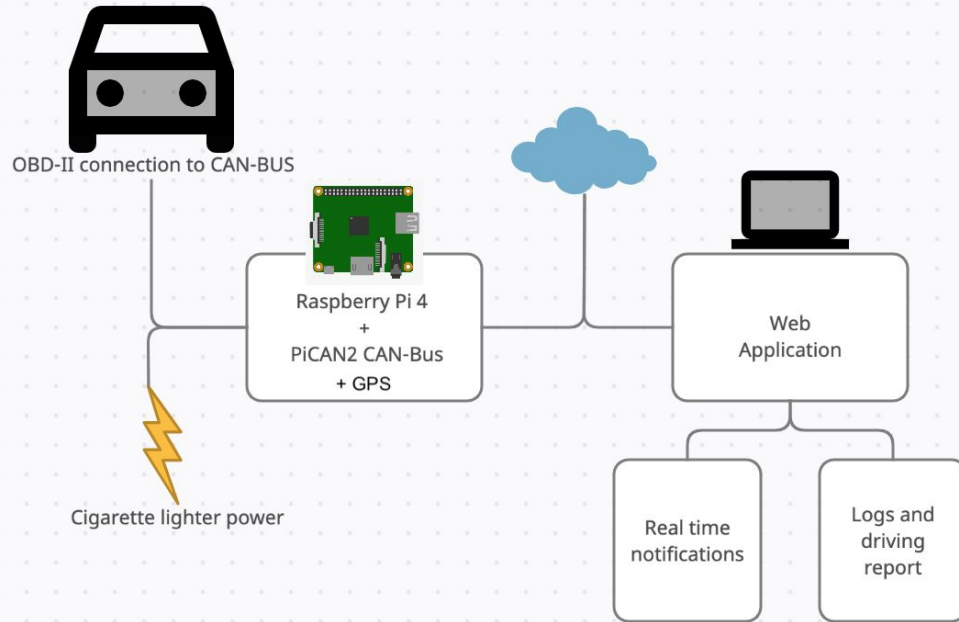Samraj Kalkat, Ryan Vimba, Reid Yesson

Application Area

Why?

- Drivers need feedback to improve driving
- Lack of feedback or constructive criticism

Solution:

- Detect and identify poor driving habits
- Notify driver in real time of infractions
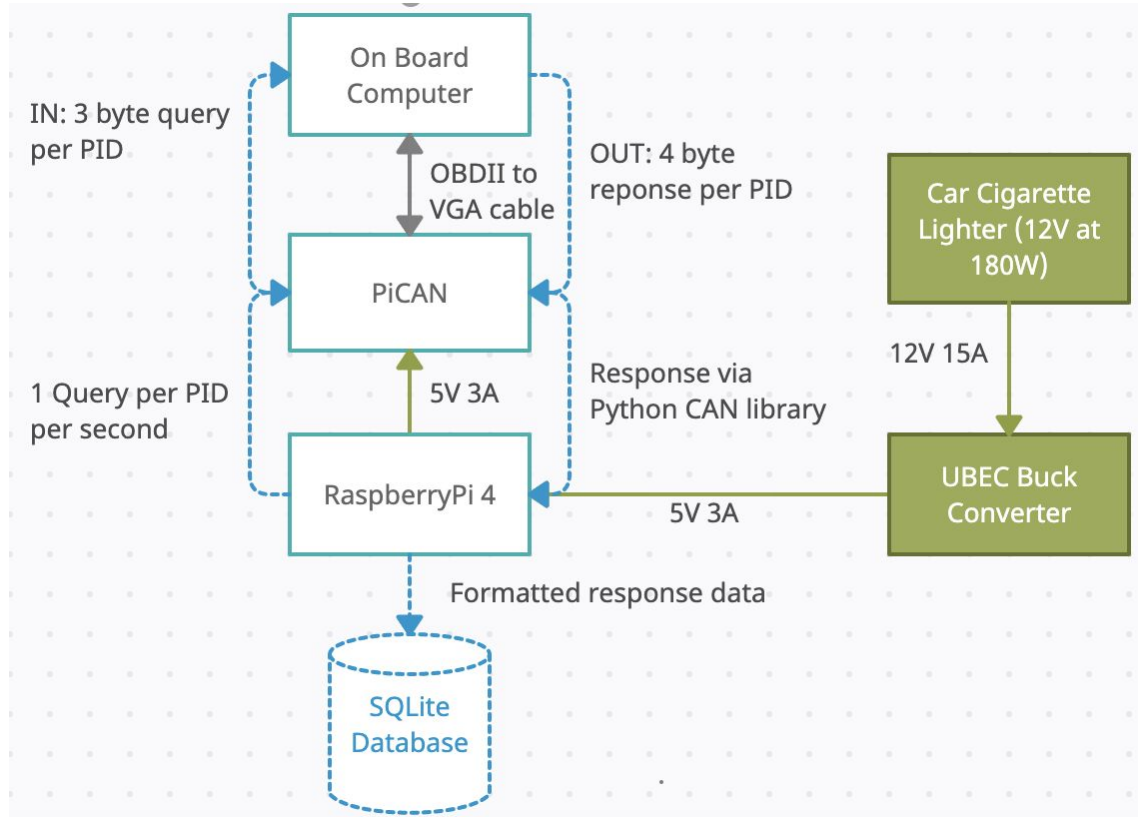- Log and share driving records
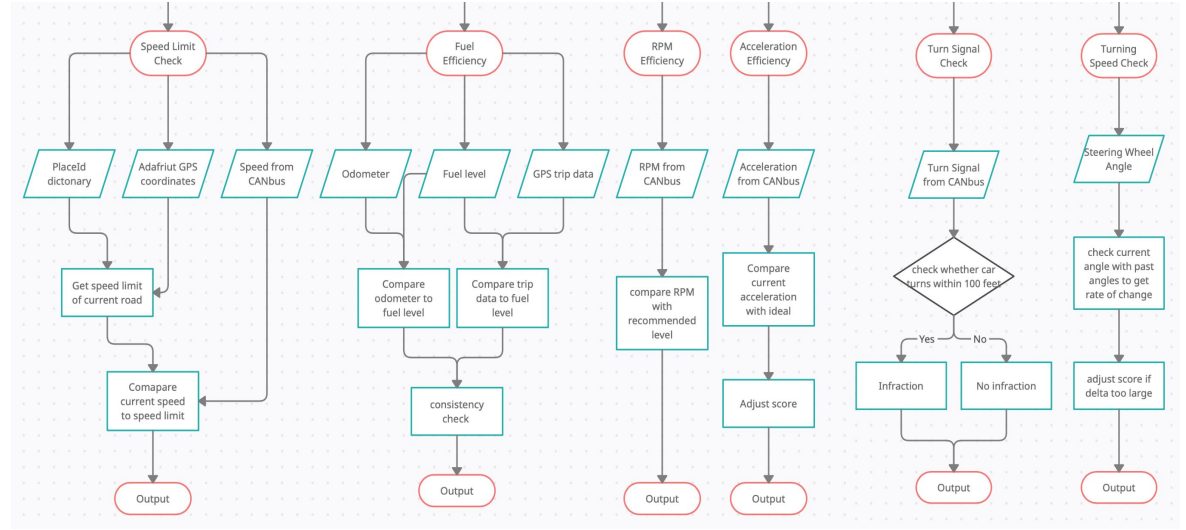
Solution Approach

Main Components:

- PICAN2
- GPS
- Raspberry Pi 4
- Web Application
- Driving Report/Logs
- Power from car cigarette lighter outlet

System Specification / Block Diagram (OBD-II Logger)

System Specification / Block Diagram (Infraction Detection)
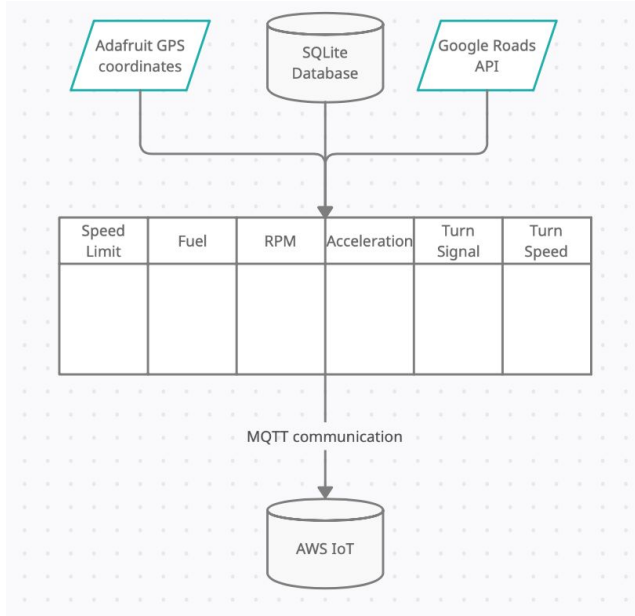
System Specification / Block Diagram (Web Application)
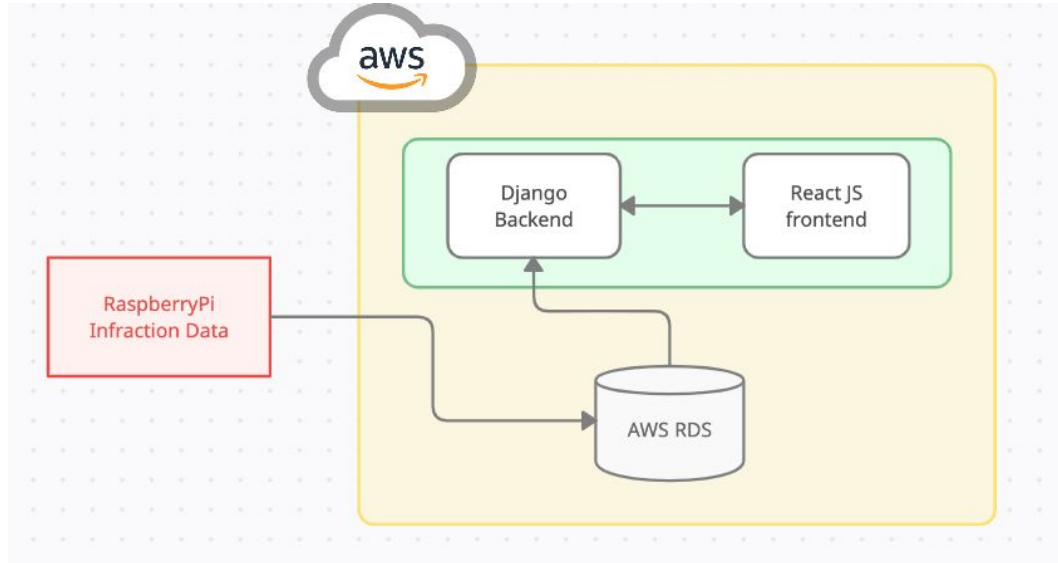
Implementation Plan

Buying
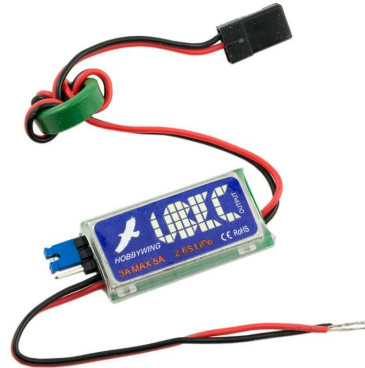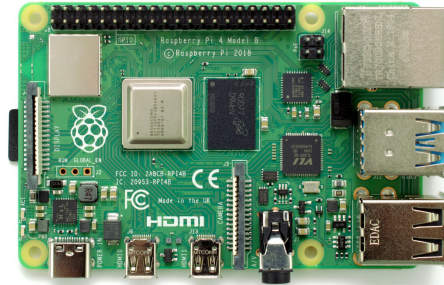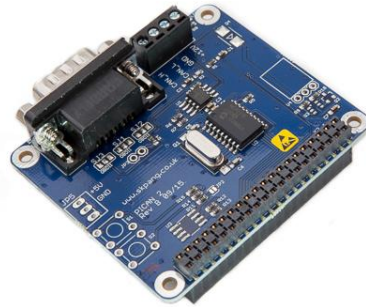
Implementation Plan

Designing

- Program to query the CANbus for specified data once per second
- RaspberryPi SQLite in-memory database to hold messages in specific schema
- Algorithms to detect driving infractions from CANbus data
- Web Application to visualize trip data and notify user of infractions

Metrics and Validation

Requirement	Testing, Verification, and Metrics
Data Logging	- Verify all data is being sent and processed correctly between OBD-II and RPI in real-time - Monitor data log to ensure connection and compare to commercial system - Complete a driving test to make sure velocity, steering wheel angle, turn signal, RPM data, acceleration, fuel levels, and odometer data is stored on Raspberry Pi
Web Application	Unit testing and database testing

Metrics and Validation

Requirement	Testing, Validation, and Metrics
Speed Limit Check	● Simulate exceeding the speed limit with 5 different speed zones
Fuel Efficiency	● Simulate a odometer and fuel level data over a 100 mile trip
RPM Efficiency	● Simulate bringing the RPM above 6500
Acceleration Efficiency	● Simulate accelerating above 1.5 m/s^2 and breaking faster than -1.5m/s^2
Turn Signal Check	● Simulate switching lanes 90 ft. and 110 ft. after using turn signal
Turning Speed Check	● Simulate making a right/left turns from 5 mph to 35 mph

Risk Factors/Unknown

- How to obtain steering wheel and turn signal data?
    - These are not standard PIDs and they vary from car make to car make
- Storing data logs and sending to database
    - Organizing data so that it can be accessed easily
- Testing in a safe controlled environment, especially with speed and hard turning

Project Management

DRIVAID Timeline Samraj = BLUE Ryan = RED Reid = ORANGE

TASK TITLE	2/22	3/1/21	3/8/21	3/15	3/22	3/29	4/5	4/12	4/19	4/26	5/3
Web Application											
Interface design											
Setting up Database											
Creating page layout											
Receive data from Raspberry Pi											
Display Infraction Warnings											
Final Report Generation											
Display Final Report											
OBDII Communication											
Finalize design and order parts											
Understand CAN data formatting											
Connect Raspberry Pi to car with UBEC											
Set up raspberry Pi with PiCAN											
Decipher OBDII data (Speed)											
Decipher OBDII data (RPM)											
Decipher OBDII data (Seat belts)											
Decipher OBDII data (Steering Wheel Angle)											
Decipher OBDII data (Turn Signals)											
Decipher OBDII data (Fuel Level)											
Decipher OBDII data (Odometer)											
Format OBDII data for SQLite database											
Data Analytics											
Setup API for Speed Limit data											
Setup API for Weather data											
Read GPS documentation											
Setup SQLite in-memory database											
Create data format to send AWS IoT											
Create algorithms for canbus data											
Setup communication with WebApp											
Create msgs to send to driver for infractions											
Testing and Integration											
Infractions testing											
Data Log testing											
Web application testing											
Slack time											
Reports and Presentations											
Design Presentation											
Design Review Report Submission											
Final Presentation											