KeepVSafe

sddec19-22

Client & Advisor: Andrew Guillemette & Daji Qiao
Team Members: Andrew Damon, Sydney Ehlinger, Freya Gaynor, & Skand Gupta

Problem

Our project will help make buses safer by monitoring drivers for behaviors that could potentially lead to a collision. Drivers that have worked the same route repeatedly will eventually pick up unsafe habits, and - left unaddressed - pose a risk to other drivers, their passengers, and themselves. If a collision occurs, fleet managers often must go through the costly process of letting go of the old employee and onboarding a new one, not to mention a rise in insurance premiums. Using ODB2 data collected from the vehicle, we can look to identify risks and alert fleet managers before an incident occurs. By addressing these signs of danger proactively, managers can decrease or even negate these costs altogether.

Solution

To provide an interface for DART's safety managers, dispatchers, and bus drivers to receive alerts when driving performance deviates from safe behavior. This system will allow the managers to address the deviations before they become vehicle collisions by providing training refreshers for the drivers or - in the case of repeated problems - seek an alternative employee. Examples of potential warning signs that demonstrate a need for action on the part of the safety managers include sudden starts/stops, jerky turns, side-swiping mirrors, et cetera. These warning signs and other data collected from the vehicles by the graduate students' project will be analyzed for trends and then presented in a user-friendly interface by our team.

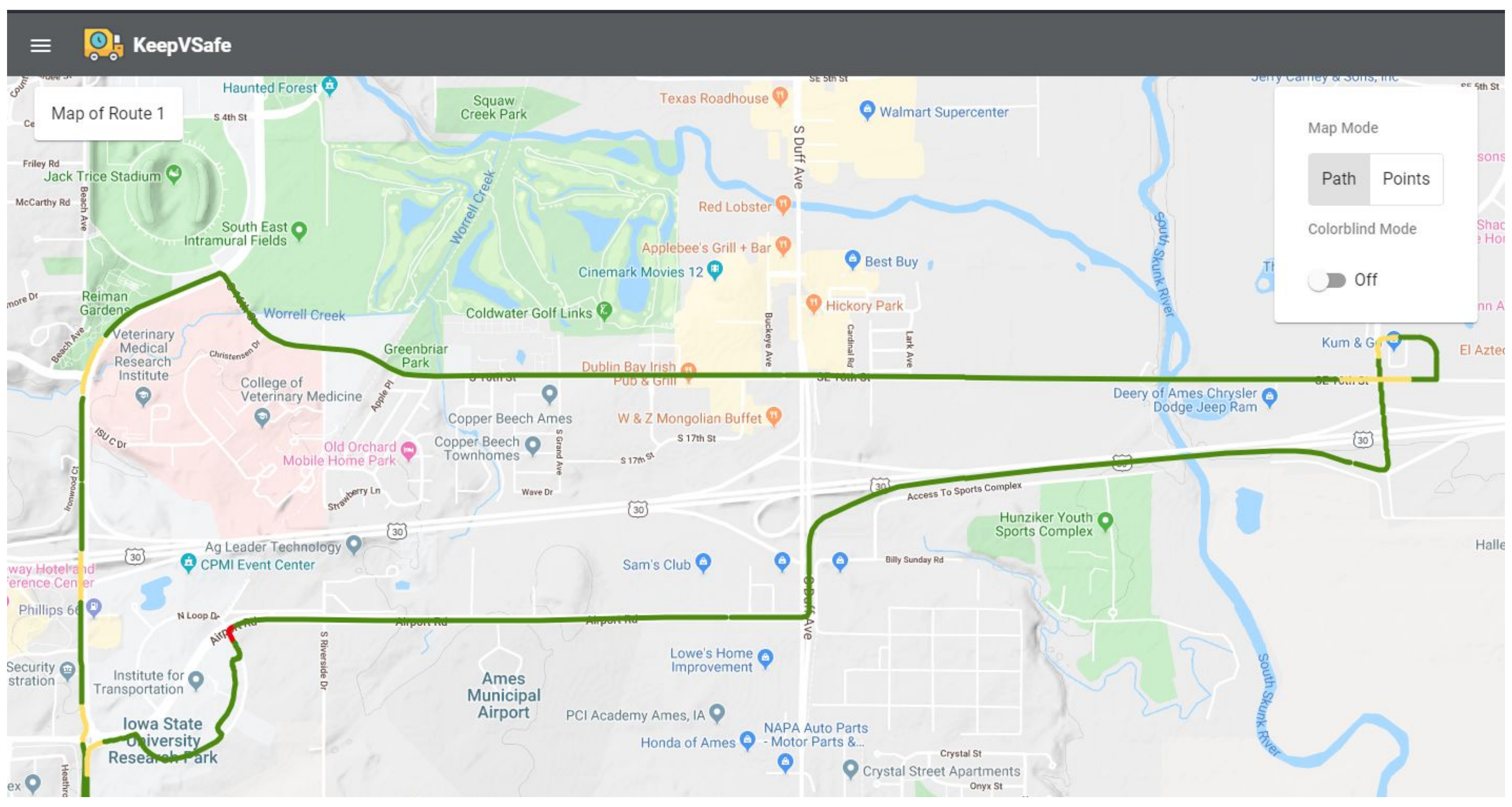

Figure 1.1 Concept Sketch

Functional Requirements

As an Admin, I want to be able to...

- Assign information about Drivers to a trip
- View statistics and reports about Drivers.
- Review performance of a Driver across a route at potential problem points.
- Receive a notification when a Driver's performance indicates that there are risks to address.

As a User, I want to be able to...

- Visualize the data collected about my performance.
- Determine what areas of my driving I need to improve upon.
- Filter and search information about my performance
- Authenticate myself to the website to view my driving information securely.

Non-Functional Requirements

- At least 100 users can be logged in concurrently.
- Database must be designed to scale to hundreds of drivers and thousands of trips.
- Driver data must be kept secure so only authenticated users can view it.
- Admins should be notified at most 20 min after a driver receives a bad grade.

Engineering Constraints

- UI should be intuitive for first-time users.
- Application should be designed with DART in mind, but leave room for the potential use by other companies in the future.
- Application should be designed primary as a proof of concept and leave room to be improved upon by client in the future.

Operating Environment

- Hosted on Google Firebase.
- Raspberry Pi receives data from ODB2 connections.

Technical Details

Driving statistics will be collected during drives via the Raspberry Pi connected to the vehicle. An algorithm designed by the graduate students will then analyze the data and provide scores based on performance; these scores and the raw data will be stored in Google Firestore. A web application hosted by Google Firebase using TypeScript framework Angular by Google will retrieve data from Firebase and present it in a user-friendly format using the Google Maps API and Charts.js to visualize performance.

Project Resources

Due to careful use of trial periods, free developer credits, and open source software with limited licensing requirements, our project has currently had cost **$0** to our client. The graduate students' project did have some costs, but given their integration and development was entirely separate to us, we do not know the actual cost they posed. Data is currently stored free-of-charge due to the developer credits that Google Firestore provides.

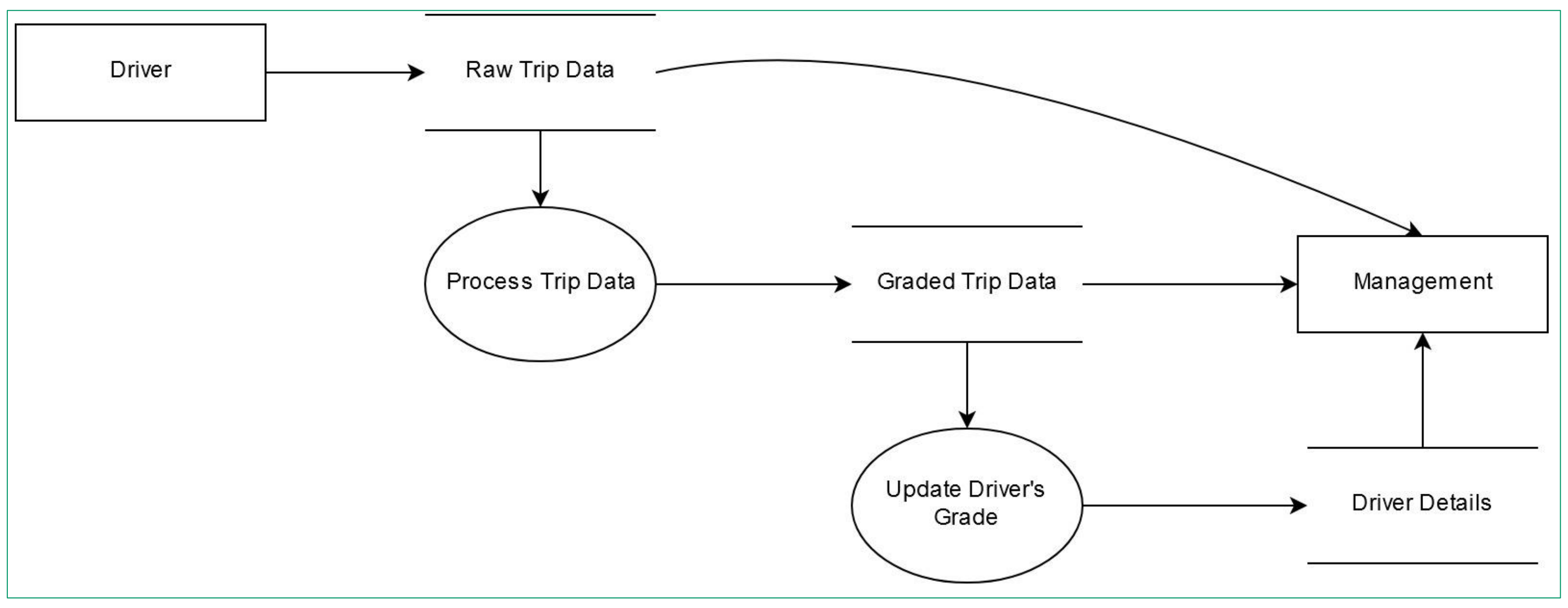

Figure 1.3 Functional Model

Intended Users and Use Cases

- The intended users are DART's safety managers, dispatchers, and bus drivers.
- Users should be able to view data about driving performance and how to improve.
- Managers should be able to easily view what drivers are currently posing a risk so they may address the issues.

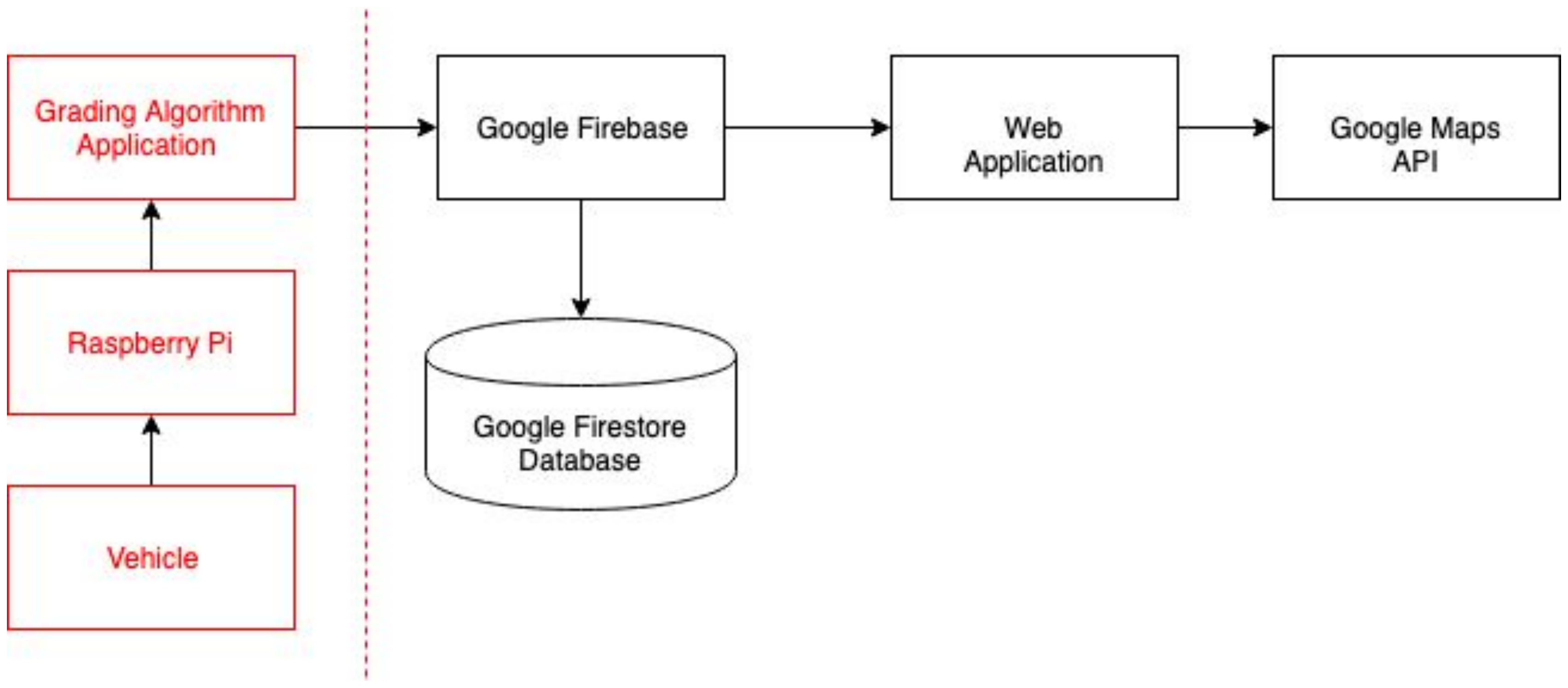

Figure 1.2 Block Diagram

Design Approach

Since our project was broken up into two teams, our design approach was largely reliant on the work the graduate student's were able to complete over the summer. From there we were able to work with our client to come up with a design solution.

We first decided to display all the segment data the graduate students had collected on Google Maps. From there we also displayed the specific data for each of the segments when clicked on.
It was then decided to store all the collected data from the CSV files into a Google Firestore database.
After our main tasks were completed, we were able to work on additional functionality such as charts, driver's list, and drivers details' pages.
Finally, after we completed our priority tasks, we were then able to turn our focus to enhancing the UI of our application.

Standards

Per Google Firebase's best practices documentation; our data is stored in small documents across multiple collections, nested inside of one another to increase speed of access and reliability. In addition, documents can only be accessed by those with the proper authorization explicitly provided by administrators.

Testing

As our project focus was primary proof of concept, testing became a minor focus during the implementation phase with the prioritization of user-friendly UI. As such, minimal unit testing is the only form of testing currently available within the implementation; however, the project has suppose for E2E testing integrated via Protractor such that both unit tests and E2E tests can be added easily when expanding upon this design.