ADVANCED VEHICLE BLACKBOX

Jose Garza, Prisha Srivastava, Lide Su, Christian Loth

Problem Background

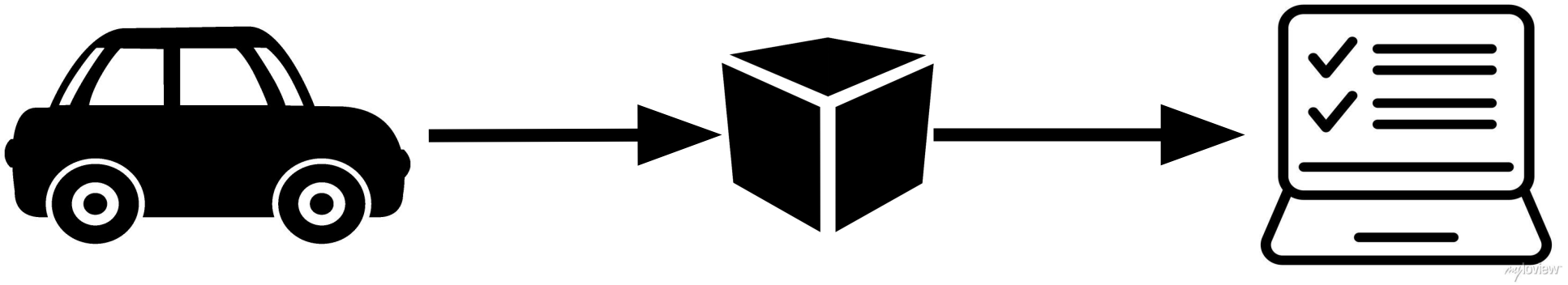
- Chances of collision increase as drivers perform unsafe maneuvers.
- Drivers are unaware of their driving behaviours.
- Insurance companies charge for insurance based on assumptions of age and car model. Creates unfair system.
- Accidents are preventable.

Goals and Objectives

The goal of this project is to encourage healthy and responsible driving practices through automated, real-time monitoring systems. Incorporating a user-specific interface creates an accurate profile such that it increases self-awareness. We hope that by creating a time sensitive profile algorithm, we are able to realistically grasp a driver's current driving behavior and through consistent monitoring, reduce their chances of collisions or accidents.

vBox Requirements

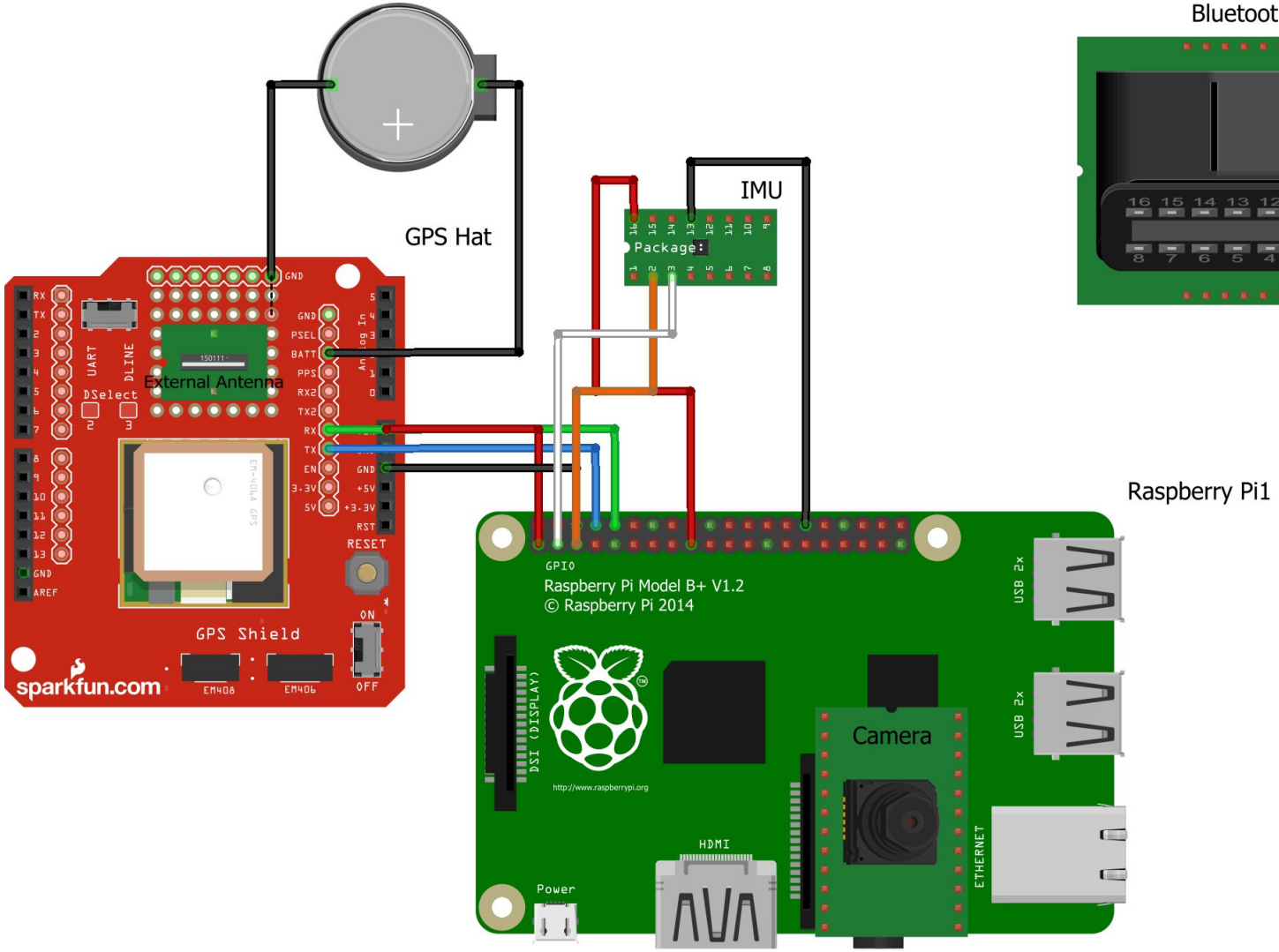
- vBox incorporates a custom algorithm that detects the quality of drive in order to detect safe and unsafe driving practices.
- vBox effectively and constantly gathers data from its sensors in order to have enough information to draw conclusions of the driving behaviors.
- vBox collects real-time data to consistently monitor behavior to improve behaviors with realistic approach.
- vBox creates separate driver profiles to uniquely identify and cater users of different ages, experience driving in different road conditions in order to accurately identify and advise on their drive.



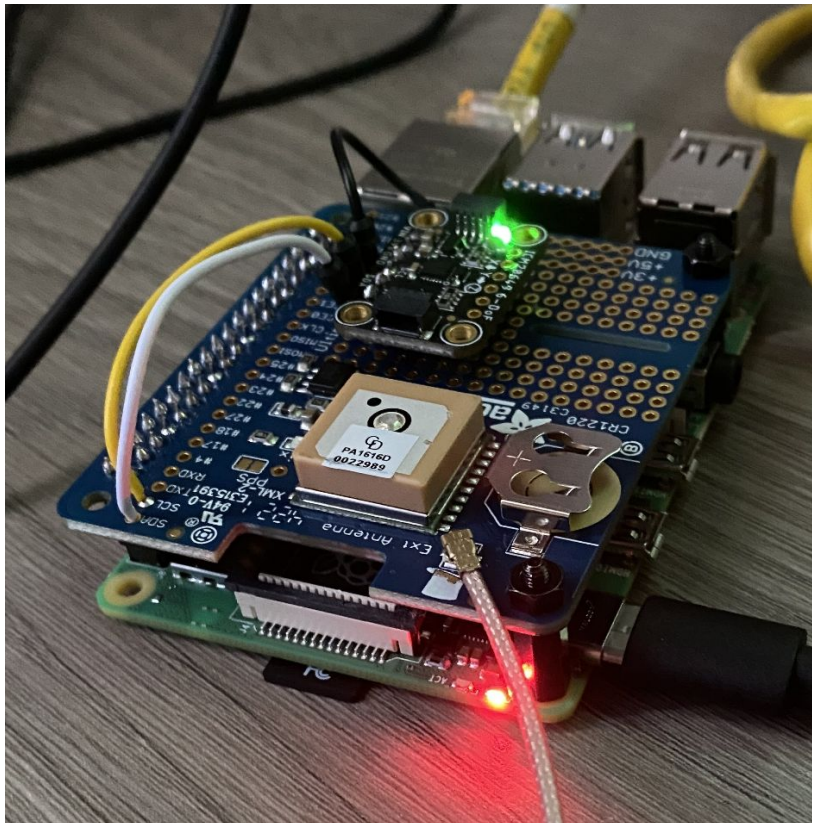
High Level Block Design

Sensor Connections

TEXAS A&M UNIVERSITY

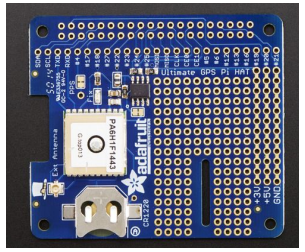


Bluetooth OBB

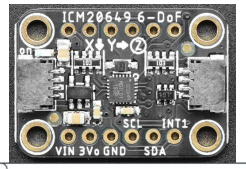


Software Sensor Data Retrieval

TEXAS A&M UNIVERSITY



Main Thread:
1.-Creates Hardware Devices Threads
2.- Obtains data from each of the threads
3.- Writes Data to CSV file
4.- Signals threads to end



Thread 1: OBD Bluetooth Device thread

Thread 1: Initializes Bluetooth connection looking for serial connection establishment

Pulls RPM, Speed, Throttle Position

Closes Bluetooth Serial Connection

Thread 2: GPS Hat thread

Thread 2: Initializes GPS Connection. Proper connection with Satelites is established

Pulls latitude and longitude coordinates

Closes Serial Connection

Thread 3: IMU Device thread

Thread 3: Initializes I2C communications to IMU

Measures acceleration and rotational axis of the vehicle

Closes Connection

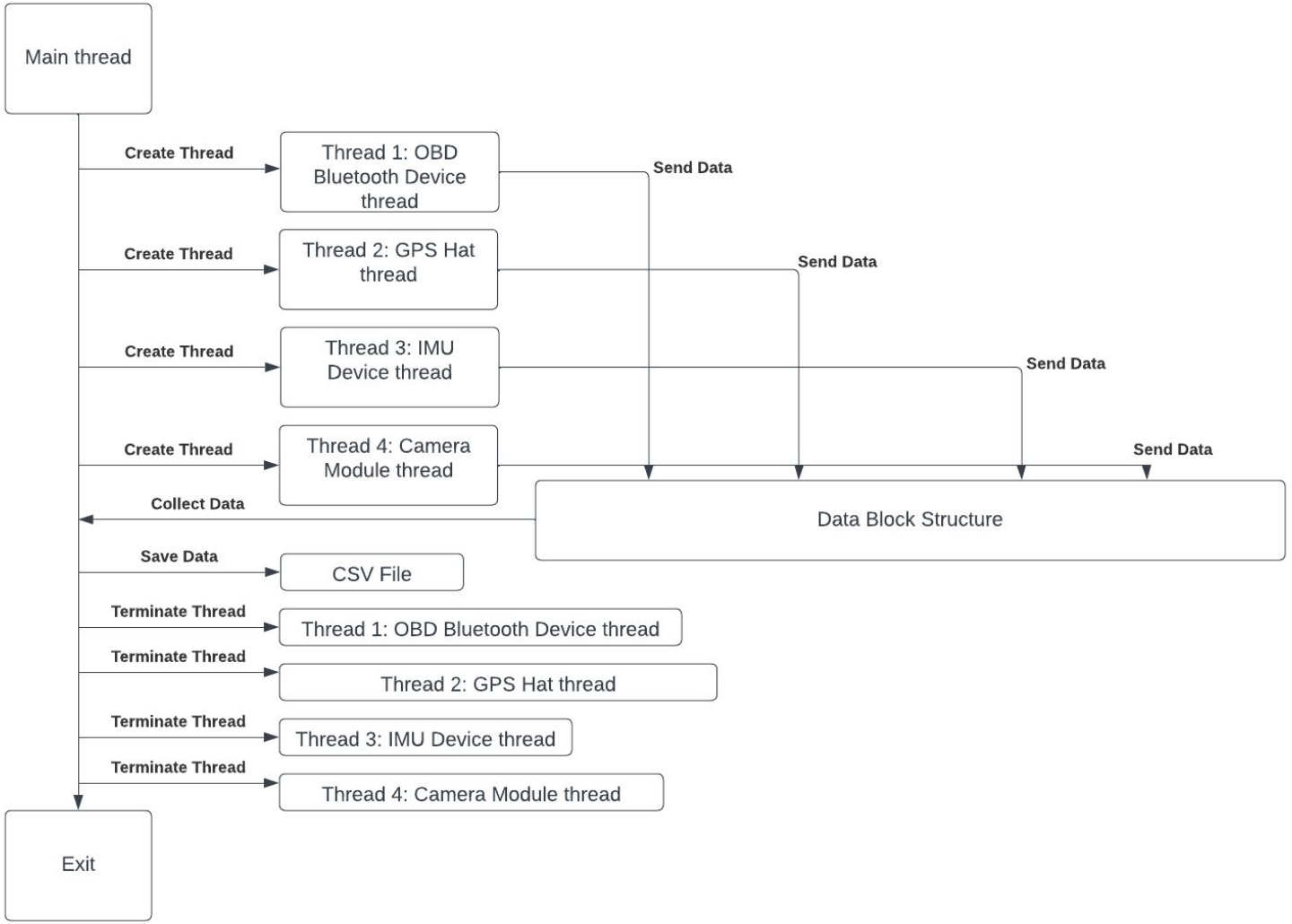
Thread 4: Camera Module thread

Thread 4: Starts Camera Module

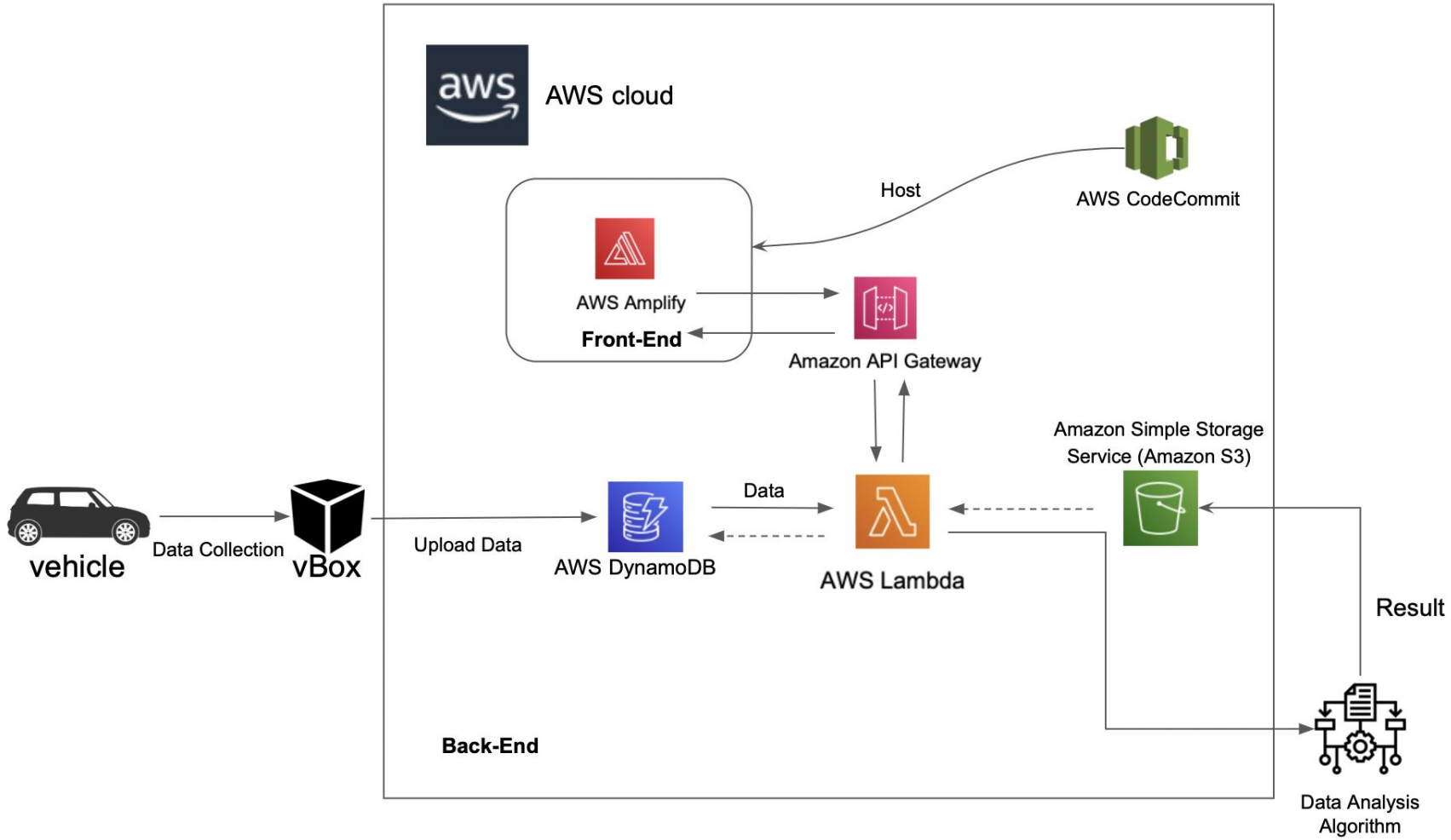
Records video

Turns off Camera

Software Sensor Data Retrieval



Embedded System



Evaluation of alternative solutions

- USB OBD
  - Would require another cable running from the OBD port which is located under the steering wheel to the Raspberry Pi in the windshield.
- Standalone GPS (no external antenna)
  - Would not provide accurate GPS readings. In some cases the “FIX” takes up to 30 minutes in closed spaces due to the poor integrated antenna. This led to the necessity of an external antenna.
- Standard IMU
  - IMU lacks insight from road conditions (i.e bumpy, traffic giving inconsistent speed reading) needed for an advanced driving detection system.
- GPS based Web Platform
  - Location tracking and speed detection through GPS may lack context in situations where driving above or below the speed limit is realistic. Speed is not as accurate as the speed of the car ECU.
- Real Time Data Upload
  - Would increase complexity and costs by adding another module.

Approach for design validation

Our design is intended to transition from prototype to final deployment through a series of layered testing that allows us to thoroughly examine the functionality of hardware and software aspects of our project.

- Embedded Hardware System
  - Run simple python commands on Raspi terminal to check each sensor status.
  - Generate terminal output of sensor readings.
  - Use multimeter to check soldering connection.
- Web App Software Testing
  - Using Agile Methodology for software development.
  - Local testing of web application to test if profile data exists and is valid.
  - Front-end testing to verify if DynamoDB datasets are displayed correctly.
  - Receives analysis output from backend script and presents on the application through visuals and real-time record.
- Back-end Algorithm Testing
  - Using 6 testing scenarios (or “switch cases”), different in observed speed and road type, a prediction model identifies certain points and calculated average score for the trip.
  - Required variables are satisfied by retrieving data from database for the execution of program
  - Score and Analysis is sent over to Webapp through API calls to server
  - Input variables are split between using dummy data points and user-generated values.

Economic Analysis

- The parts originally ordered have been able to achieve the desired results.
- Camera module has not been delivered to the team.
- Total = $345.50

Description	Price	Quantity
Raspberry Pi 4 Camera	$24.59	1
Adafruit GPS Hat	$29.95	1
External GPS antenna	$19.95	1
IMU (Accelerometer & Gyro)	$14.95	1
Standoffs for Pi HATs	$0.75	1
RF Adapter cable	$3.95	1
Bluetooth interface OBD port	$13.99	1
Raspberry Pi 4	$167.95	1
SD card with 512 GB	$29.99	1
Car charger for Raspberry Pi 4	$14.44	1
Suction Cup Mount	$24.99	1
3D Printed Enclosure	$25.00	1

Manufacturability, Sustainability and Economics of vBox

- **Manufacturability:**
  - Product will use SLA resin to create a thermoset case for the Raspberry Pi.
- **Sustainability:**
  - 3-D printing produces more sustainable solutions than traditional manufacturing methods.
    - 70-90% less scrap waste.
    - 41-64% less energy consumed.
- **Economics of vBox:**
  - Cost of Production: $345.50, RasPi 4 is 4.8 times more expensive than usual price.
  - Bulk Purchase of items would significantly reduce cost for larger-scale production.

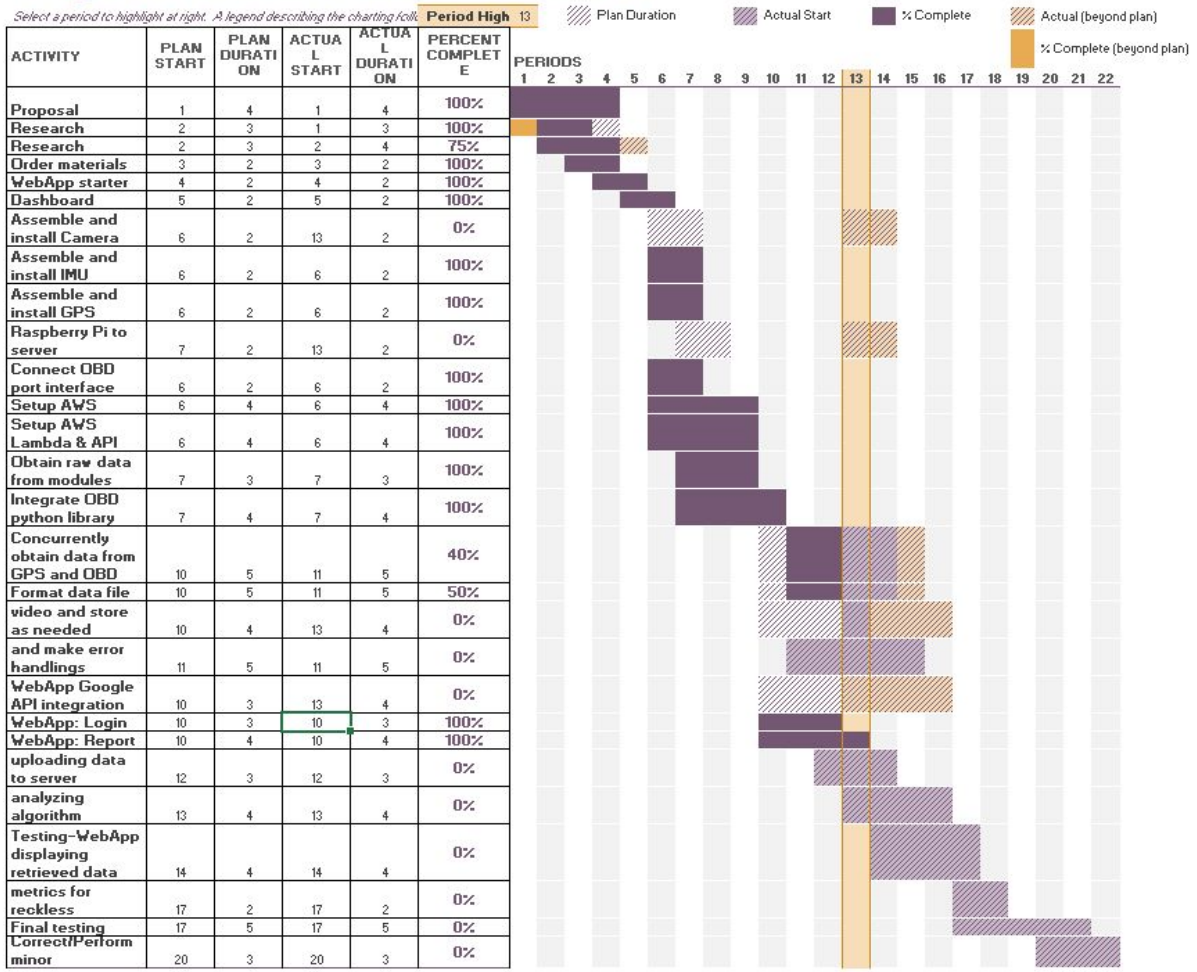
Health/Societal/Political Concerns

- Social
  - Society benefits directly from reduced poor driving practices -- more saved lives.
  - Blackbox will be manufactured in a compact design such that driver's ability to drive is not impacted by the device.
  - Reduce the amount of crashed cars. Reducing the amount of cars that are deemed totaled.

- Political
  - Road Safety continues to be a high-priority political topic for countries i.e Spain and Italy
  - Previously, many road safety public policies include more resources for police enforcement/traffic management, these policies are hardly effective without required funding.
  - Automating driving assessment and creating custom profiling system provides a cheaper alternative to monitoring road safety.
- Ethical
  - The more lives we save, the more ethical it is. We can save more lives by knowing the problem that caused a crash.
  - It is also ethical to have more safe drivers, and reduce untrained drivers to make our roads much safer for everybody.

Schedule of tasks

Project Planner: Vehicle Blackbox



Select a period to highlight at right. A legend describing the charting follows.

Period High 13

Plan Duration | Actual Start | % Complete | Actual (beyond plan) | % Complete (beyond plan)

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Proposal	1	4	1	4	100%
Research	2	3	1	3	100%
Research	2	3	2	4	75%
Order materials	3	2	3	2	100%
WebApp starter	4	2	4	2	100%
Dashboard	5	2	5	2	100%
Assemble and install Camera	6	2	13	2	0%
Assemble and install IMU	6	2	6	2	100%
Assemble and install GPS	6	2	6	2	100%
Raspberry Pi to server	7	2	13	2	0%
Connect OBD port interface	6	2	6	2	100%
Setup AWS	6	4	6	4	100%
Setup AWS Lambda & API	6	4	6	4	100%
Obtain raw data from modules	7	3	7	3	100%
Integrate OBD python library	7	4	7	4	100%
Concurrently obtain data from GPS and OBD	10	5	11	5	40%
Format data file	10	5	11	5	50%
video and store as needed	10	4	13	4	0%
and make error handlings	11	5	11	5	0%
WebApp Google API integration	10	3	13	4	0%
WebApp: Login	10	3	10	3	100%
WebApp: Report	10	4	10	4	100%
uploading data to server	12	3	12	3	0%
analyzing algorithm	13	4	13	4	0%
Testing-WebApp displaying retrieved data	14	4	14	4	0%
metrics for reckless	17	2	17	2	0%
Final testing	17	5	17	5	0%
Correct/Perform minor	20	3	20	3	0%

PERIODS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

- Project is on schedule
- Major tasks to be completed:
  - Software: Add integration of the Google API. Design improvements /options to WebApp.
  - Hardware: Establish a script to upload CSV file to the cloud. Handle errors in data.
  - Data Analytics: Determine algorithm to score the drivers' behaviors.
  - All: Testing and debugging of the system.

Project Management and teamwork

TEXAS A&M UNIVERSITY

Hardware

- Receiving the parts ordered and soldering the necessary parts.
- Establishing a connection between each one of the components.
- Integrating the module connections together for a more compact device.
- Establish basic python scripts to test the connection between the devices.
- Establishing the CSV file format

Software

- Established AWS DynamoDB.
- Implemented Lambda functions and API gateway.
- Established connection from React app to AWS Amplify to host web app online.
- Made significant progress with designing the dashboard that the users see of the WebApp.
- Added different graphs and different attributes to the webpage.

Future Tasks Distribution

TEXAS A&M UNIVERSITY

Hardware

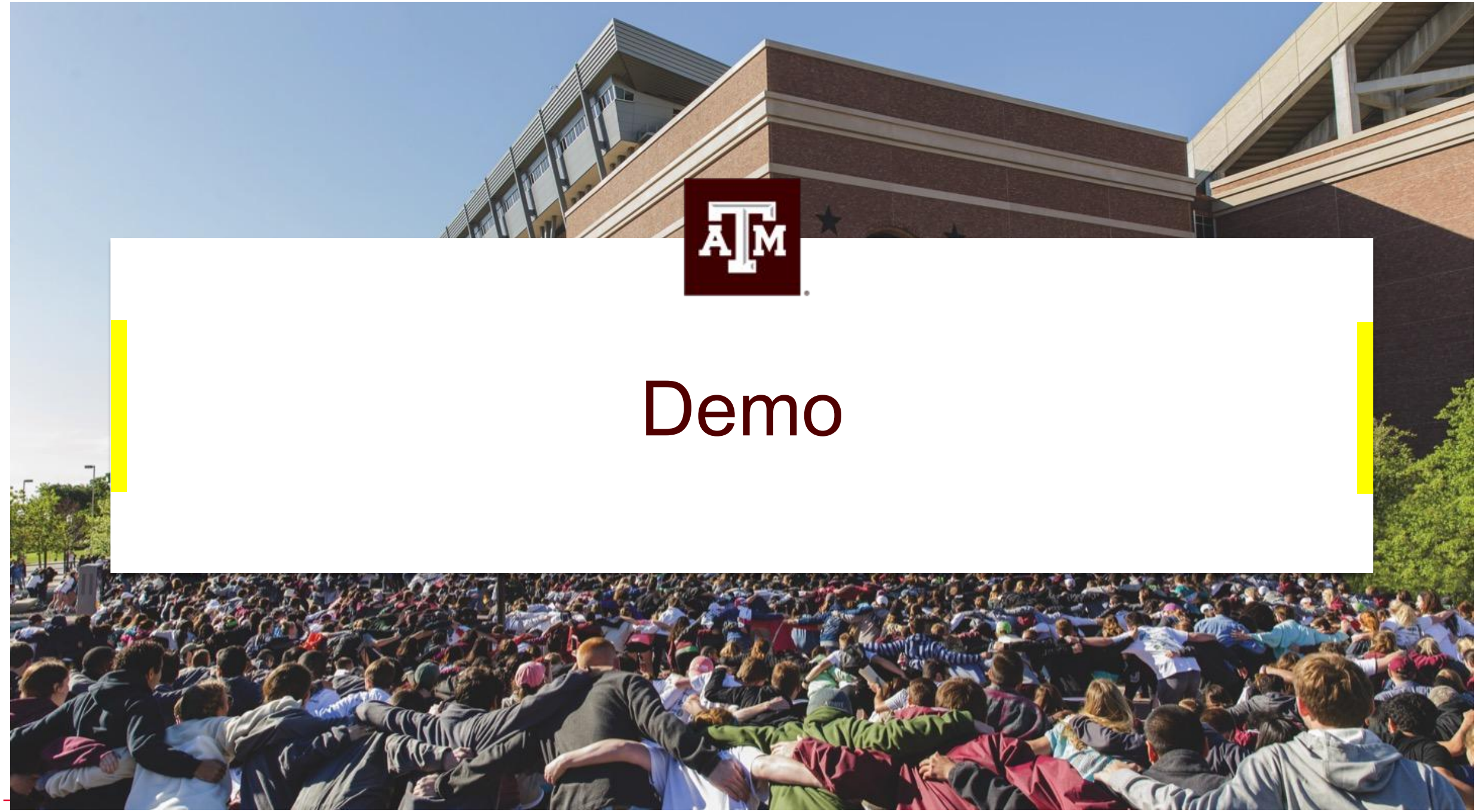
- Upload data to the server and ensure proper formatting and error handling.

Data Analytics

- Develop an algorithm to score the drivers' behavior

Software

- Upload data from Raspberry Pi to AWS DynamoDB.
- Integrate Google API
- Design Improvements and settings.



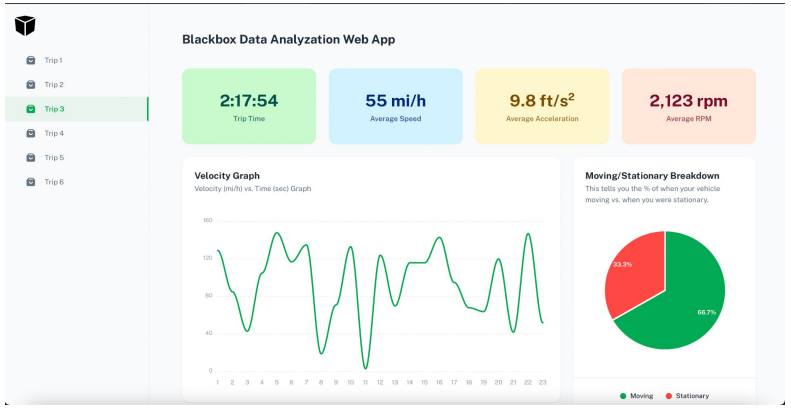
Demo

Future Final Demonstration

Recorded Demo

- Record the experience of a normal user and upload the data to the webpage which can be displayed
- Would demonstrate a physical use scenario

Live Demo



- Using OBD emulator
  - ELM327-emulator - Python based. Used for the development of OBD reading applications.
- Would permit a live demonstration through a series of recorded input to the devices.

We appreciate your attention
Questions?