Fleet Monitoring System

Design Document

Team #12
Lofti Ben Othmane
Joe Herrera - Mobile Developer
Marco Yepez - Embedded System and Mobile Developer
Nicolas De la Cruz - Back-end Developer
Lorenzo Chavarria - Web Application Developer
sddec20-12@iastate.edu
http://sddec20-12.sd.ece.iastate.edu/

Executive Summary

Development Standards & Practices Used

- Agile Development Standards
- JavaScript Coding Standards
- Google Java Style Standards
- Oracle PLSQL Coding Guidelines

Summary of Requirements

- Develop Android and iOS mobile applications for Fleet Monitoring System
- Develop a web application for Fleet Monitoring System
- Provide GPS location of each vehicle in a fleet
- Provide data analytics of vehicle data
- Provide client chat
- Provide SMS notifications

Applicable Courses from Iowa State University Curriculum

List all Iowa State University courses whose contents were applicable to your project.

- CPR E 288
- COM S 309
- COM S 363
- COM S 352
- S E 319
- S E 329
- S E 339

New Skills/Knowledge acquired that was not taught in courses

- Server configuration
- Rest API creation
- Hardware equipment knowledge, e.g. resistors

Table of Contents

List of figures/tables/symbols/definitions (This should be the similar to the project plan)

1 Introduction

1.1 Acknowledgment

Acknowledgment to team SDMay18-2018, the team with the first version of this project.

1.2 Problem and Project Statement

With UPS, FedEx, and numerous fleet companies out there. We need something to monitor and manage these fleets. With our fleet monitoring system, a company would be able to use our user-friendly solution. Our solution revolves around web and mobile applications to view GPS locations of each vehicle, SMS notifications, client chat, and data analytics. Our solution would be based on MongoDB for our database and a server running Node.js Rest API for microservices. The fleet would need a PiCan for each vehicle to be able to send data from the vehicle to the server. Through the connectivity of each component, our solution would provide clients a fleet monitoring system.

1.3 Operational Environment

The conditions for this product are regular conditions inside the car for the PiCan device. Regular conditions would be restricted to the device being dry, avoiding all liquids.

1.4 Requirements

- Develop Android and iOS mobile applications for Fleet Monitoring System
- Develop a web application for Fleet Monitoring System
- Provide GPS location of each vehicle in fleet
- Provide data analytics of vehicle data
- Provide client chat
- Provide SMS notifications

1.5 Intended Users and Uses

- Managers
    - Add users to groups
    - Accept new driver account request
    - Analyze data from vehicles
- Drivers
    - Submit driver account request
    - Indirectly produce usage data while driving

1.6 Assumptions and Limitations

Assumptions:

-There will be one manager and many groups that will consist of drivers. The manager will be able to control the different groups and see different data analytics.

Limitations:

-The idea of hardware is that it has to be in a small workspace.

-PiCan device can only be tested and used on cars of the year 2008 and newer.

-Limited knowledge of hardware and server.

1.7 Expected End Product and Deliverables

Deliverable 1: Web Application - This web page will allow a user (manager or driver) to log in. As a manager, they would be able to view real-time tracking of all the vehicles currently being driven. A chat system will be implemented so that the manager can communicate with drivers.

Deliverable 2: Mobile Application - The mobile application, in practice will be similar to the web application. There are performance bottlenecks with mobile devices as they don't have the computing power of a computer. Therefore, the mobile may contain only a smaller amount of data analytics and may not have the performance to run real-time tracking.

Deliverable 3: PiCan - The PiCan is the hardware that will be installed on the vehicles to extract data and send it to the server. Data that is extracted from the vehicle includes mileage, fuel, speed, etc.

Deliverable 4: Server - The server will be a Node.js Rest API to provide Microservices for PiCan data posting and the client can request and post data to the database.

Deliverable 5: Database - Mongo Database to store user information, registered vehicles, registered drivers, and vehicle data.

The end product would include the web application, mobile application, PiCan, server, and database that all work together to create a fleet monitoring system. The fleet monitoring system includes hardware that must be installed on a vehicle. The hardware will communicate data to the server so that the manager and drivers can use the web/mobile application to view data analytics. The Fleet Monitoring System is set to be completed by December 2020.

2. Specifications and Analysis

2.1 Proposed Approach

Approach:

- Use MongoDB for scalability, as we need to collect extreme amount of data from vehicles
- Rest API for Microservices in Node.js using Microsoft Azure Web API Design
- Dynamic web page creation using Node.js and Express.js
- Google Java Style Standards

2.2 Design Analysis

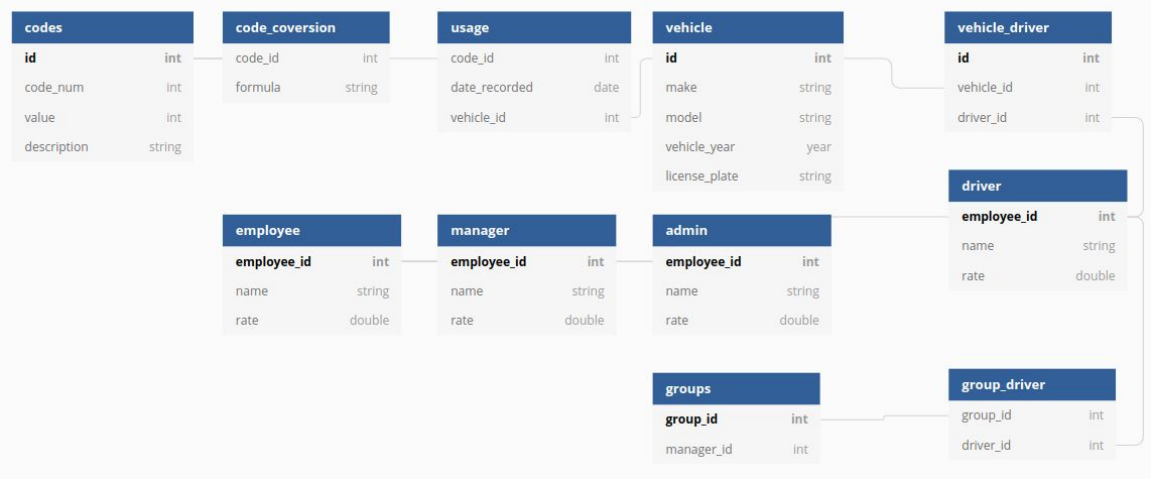
In these weeks, we have met with our adviser and client to discuss what the previous team from a past semester completed. We discussed what and how we should implement for a successful project. Our team has been able to produce design prototypes for an Android and website app. We have also been able to successfully replicate what the previous team completed for the raspberry pi device and vehicle. In the end, we want to implement these front-end designs, refractor the embedded system program, and create a successful, sustainable backend. Our strengths will be that we are most familiar with how our product and the software will work. Our weakness is mostly that we lack some knowledge of hardware and server.

2.3 Development Process

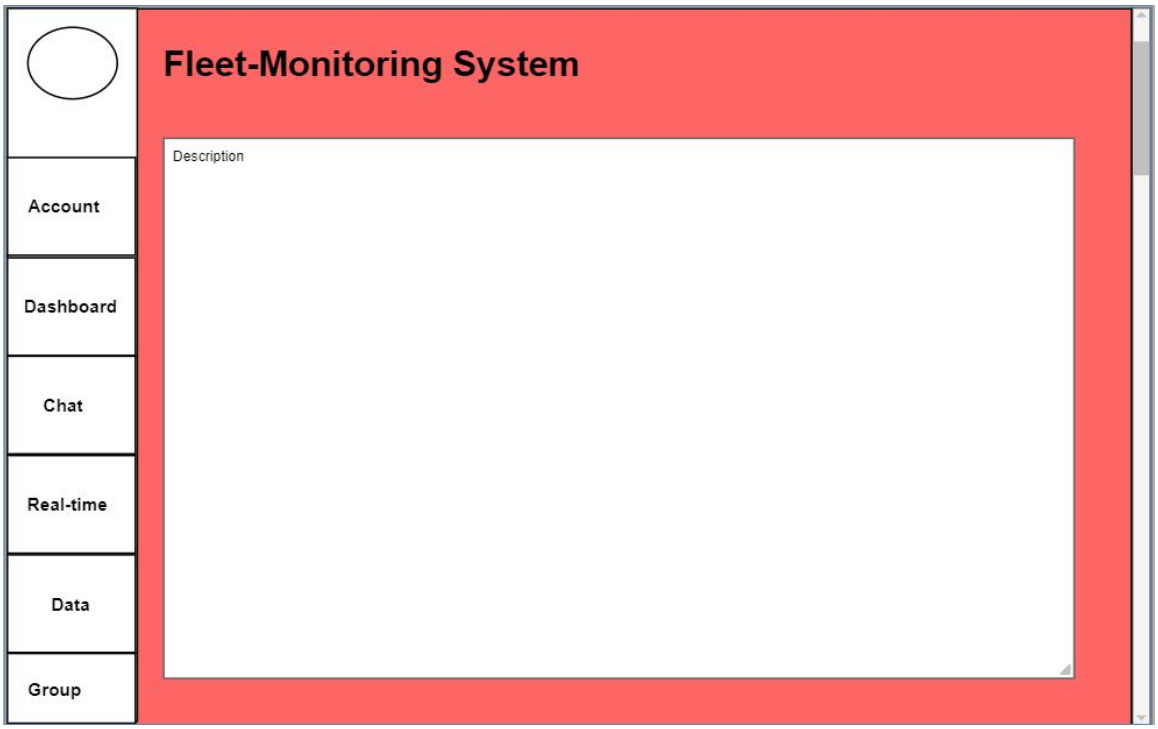
Our team will follow the Agile software development process, our development sprints will be three-weeks long. We meet with our advisor and team weekly for updates, questions asked, and any other relevant topics for our project.

2.4 Conceptual Sketch

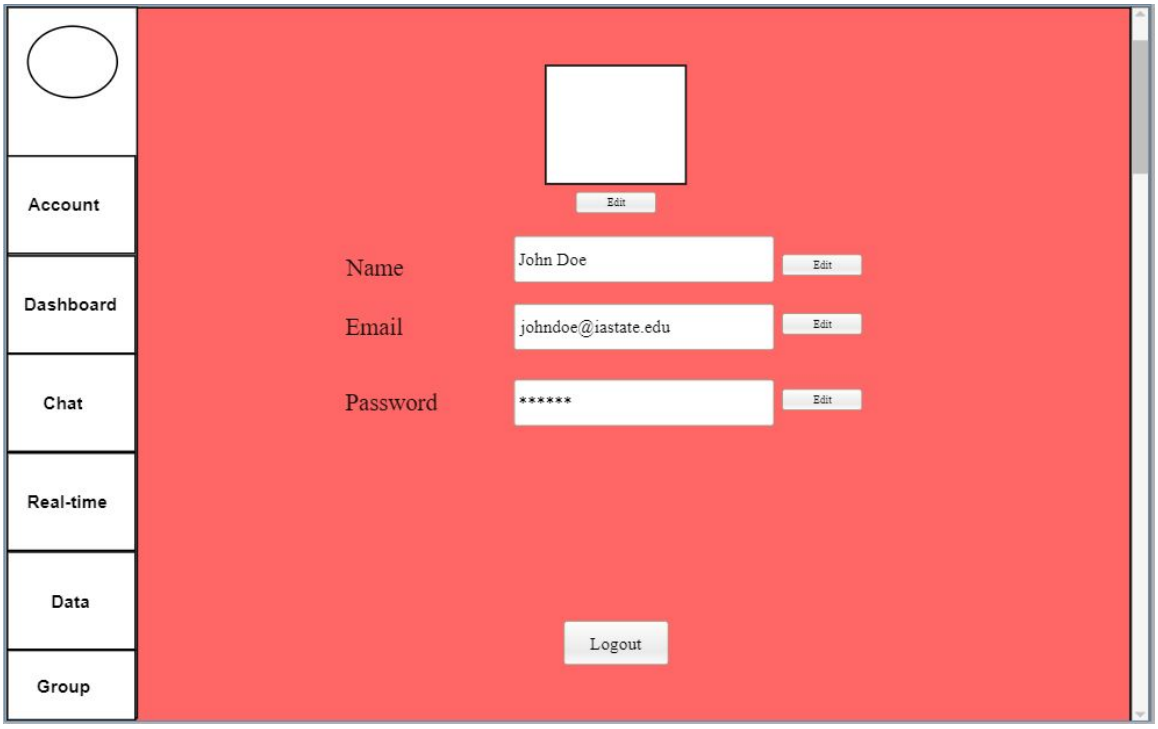
Database Architectural Design:



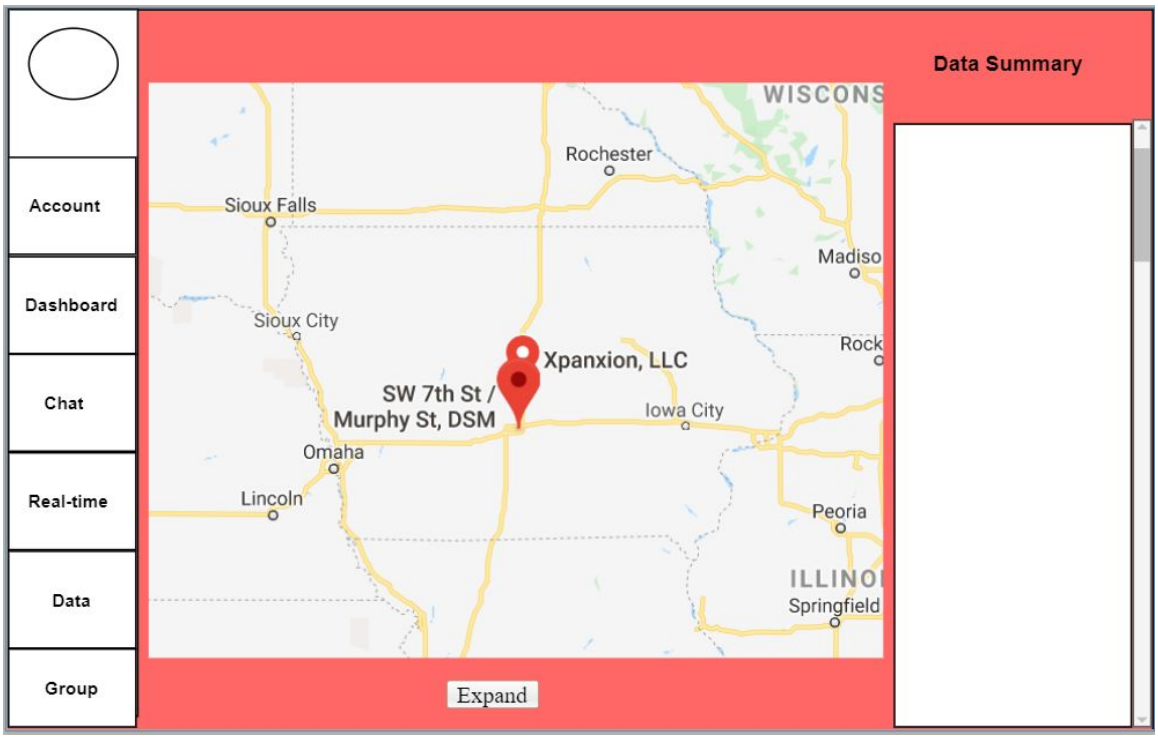
Web Application Design:



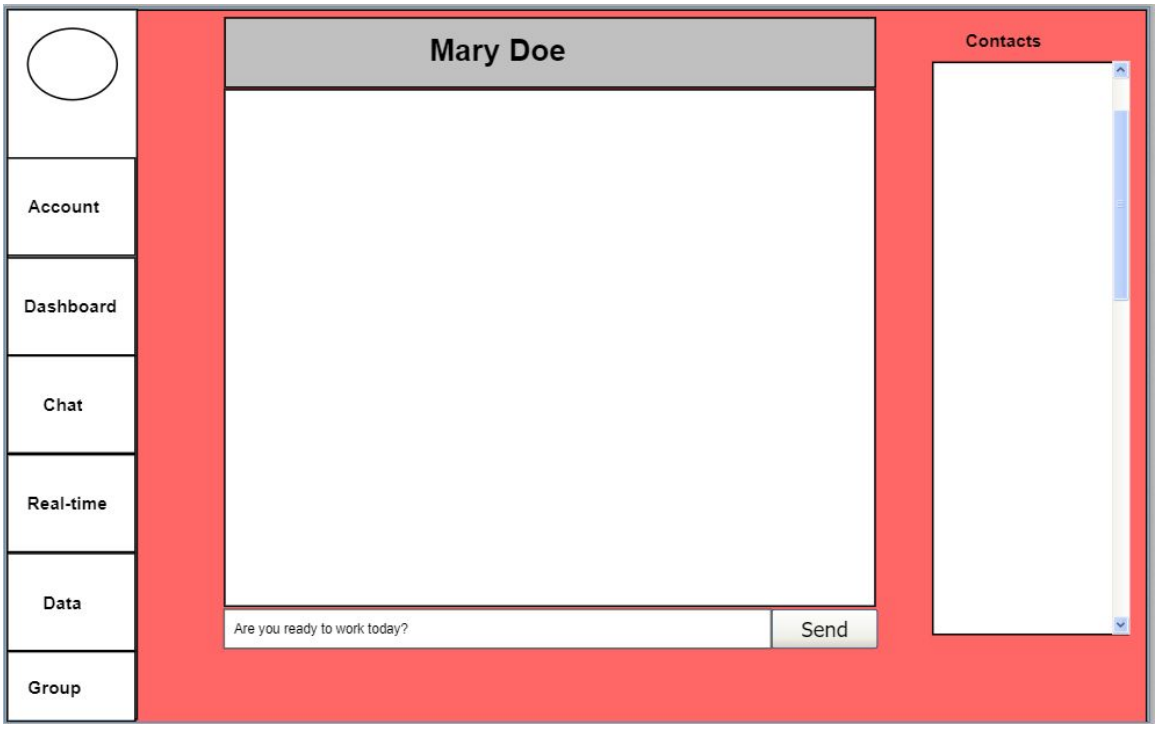
Home Page



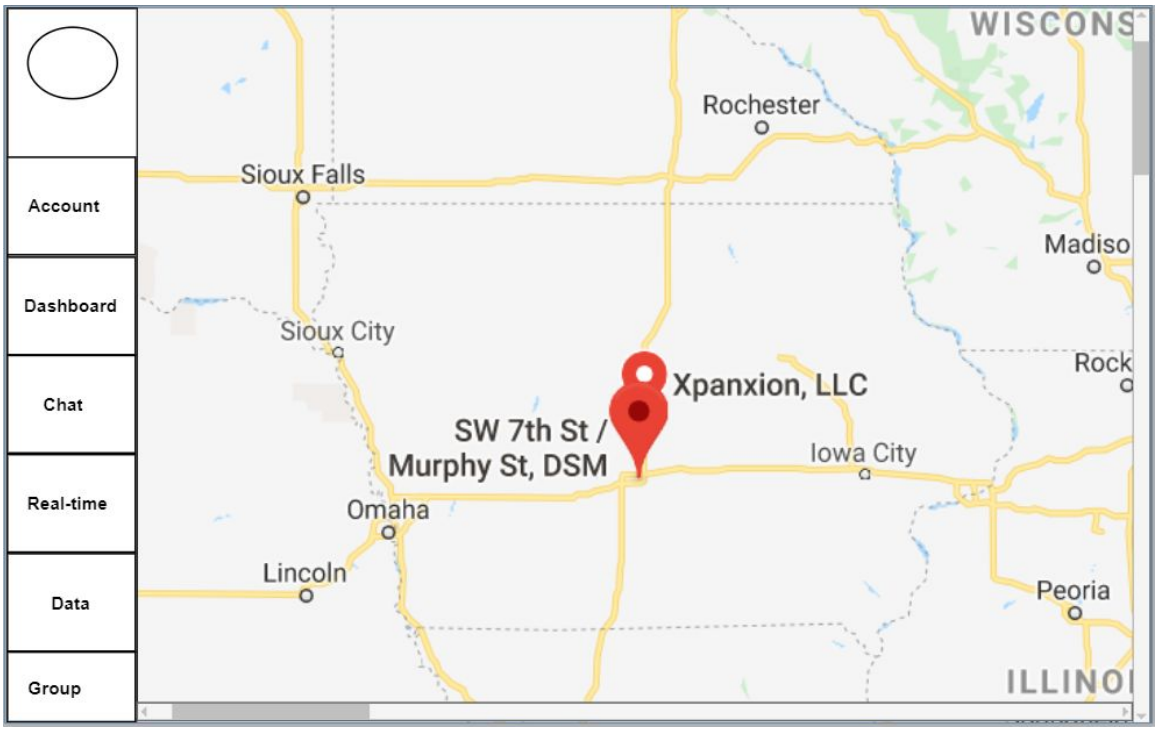
Account Page



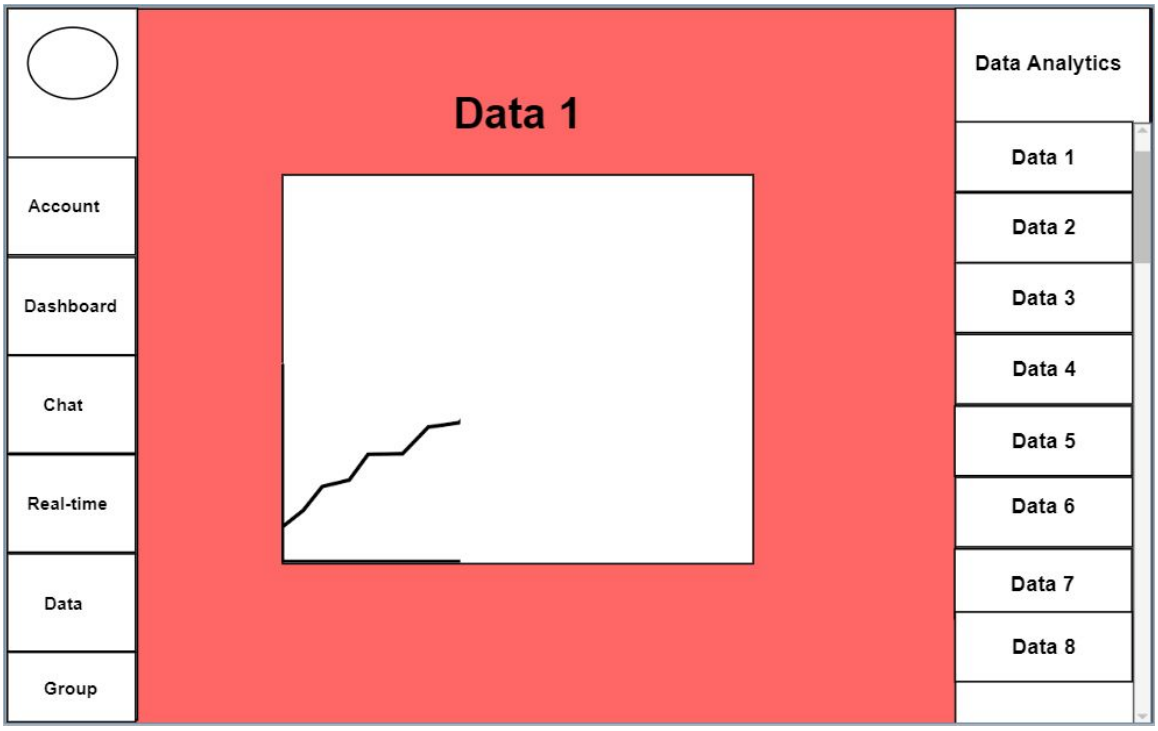
Dashboard Page



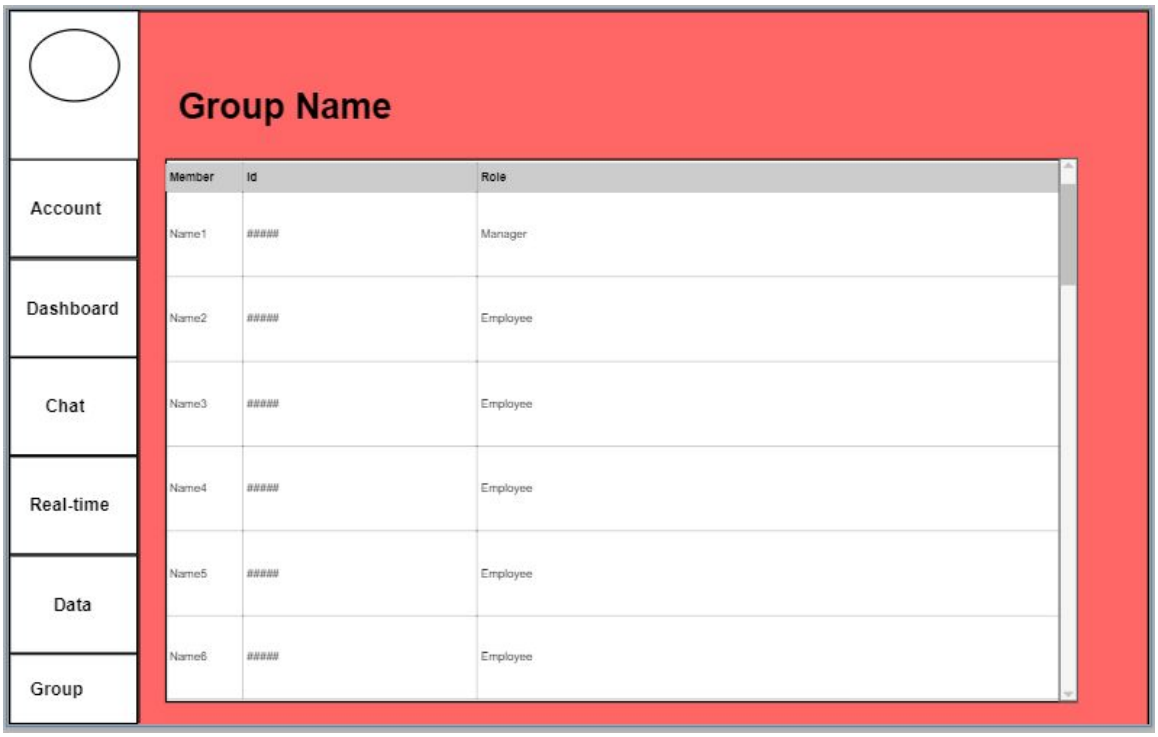
Chat Page



Real-time Data Page

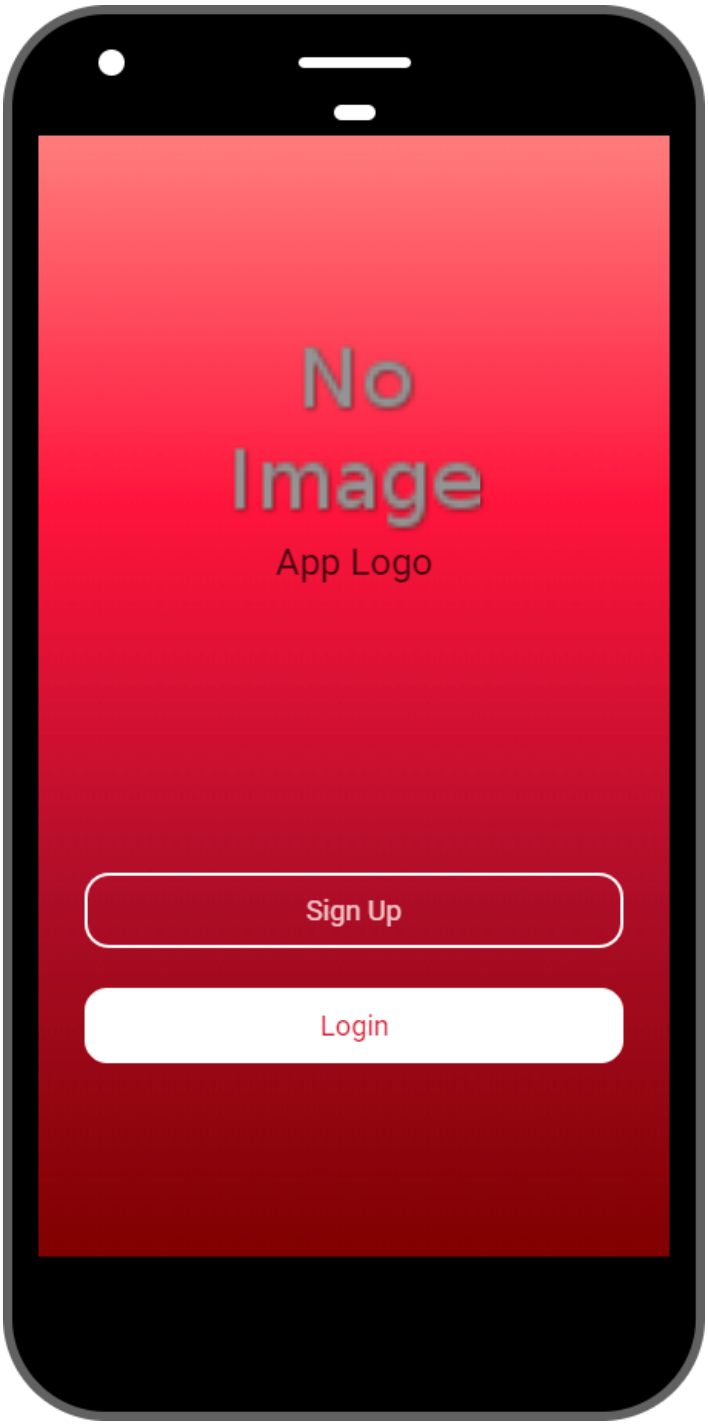


Data Analytics

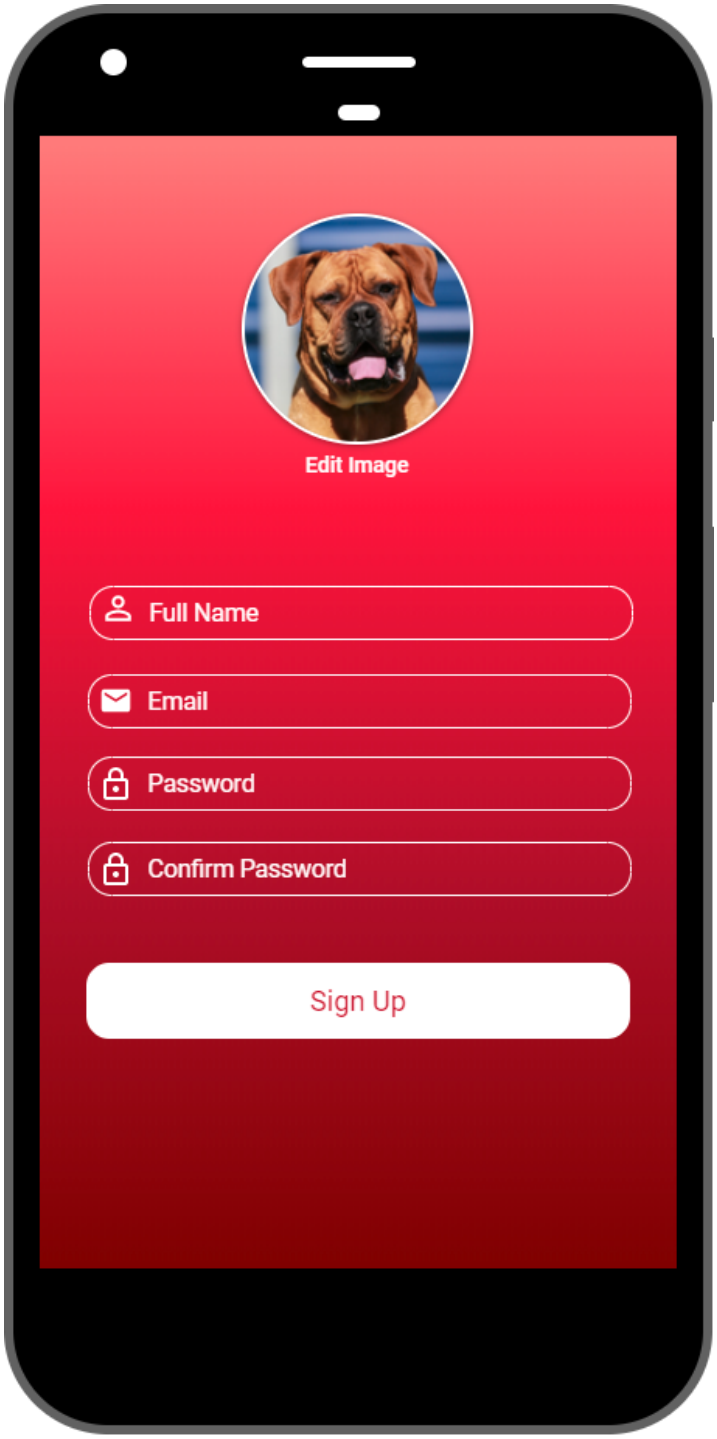


Group Page

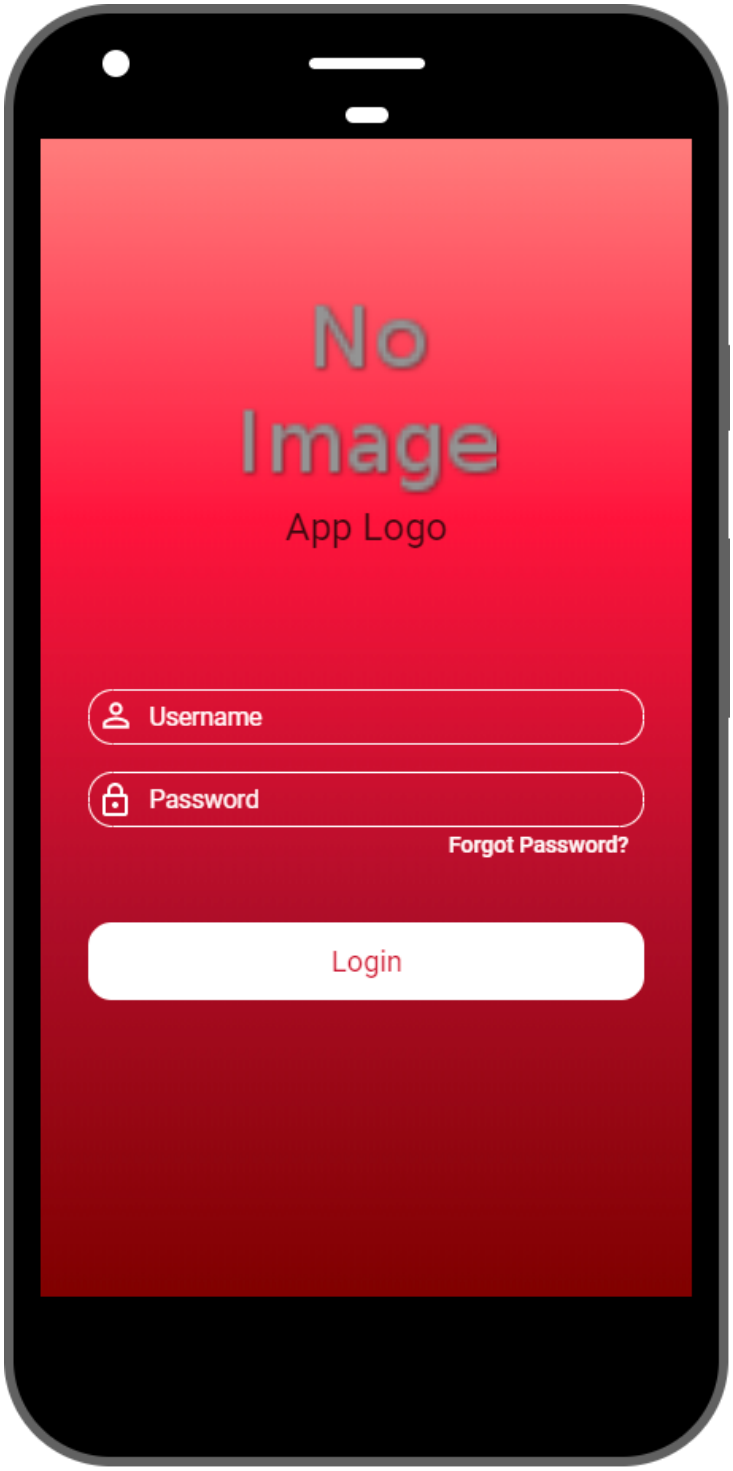
Mobile Application Design:



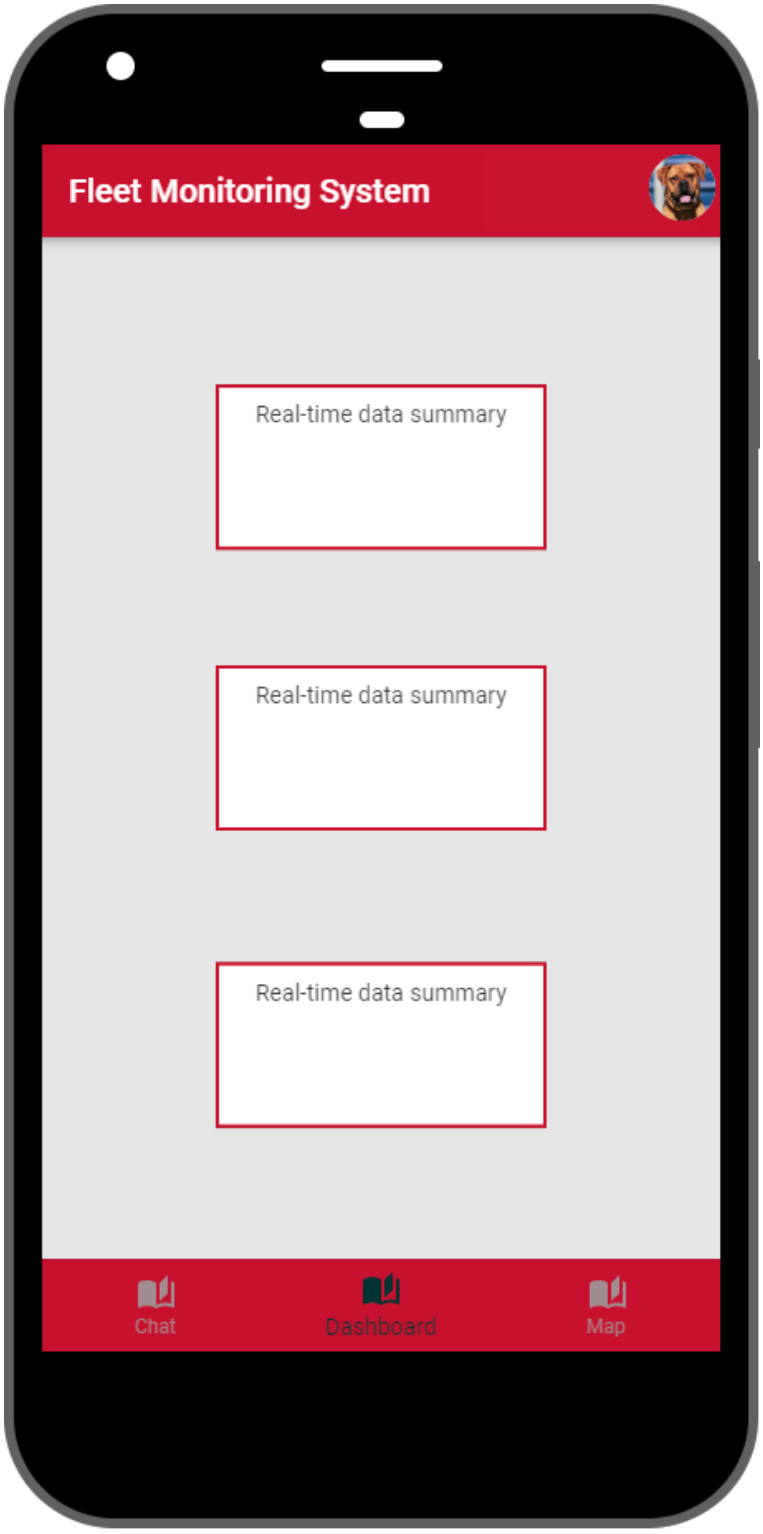
Welcome View


Edit Image
Full Name
Email
Password
Confirm Password
Sign Up

Sign Up View



Login View



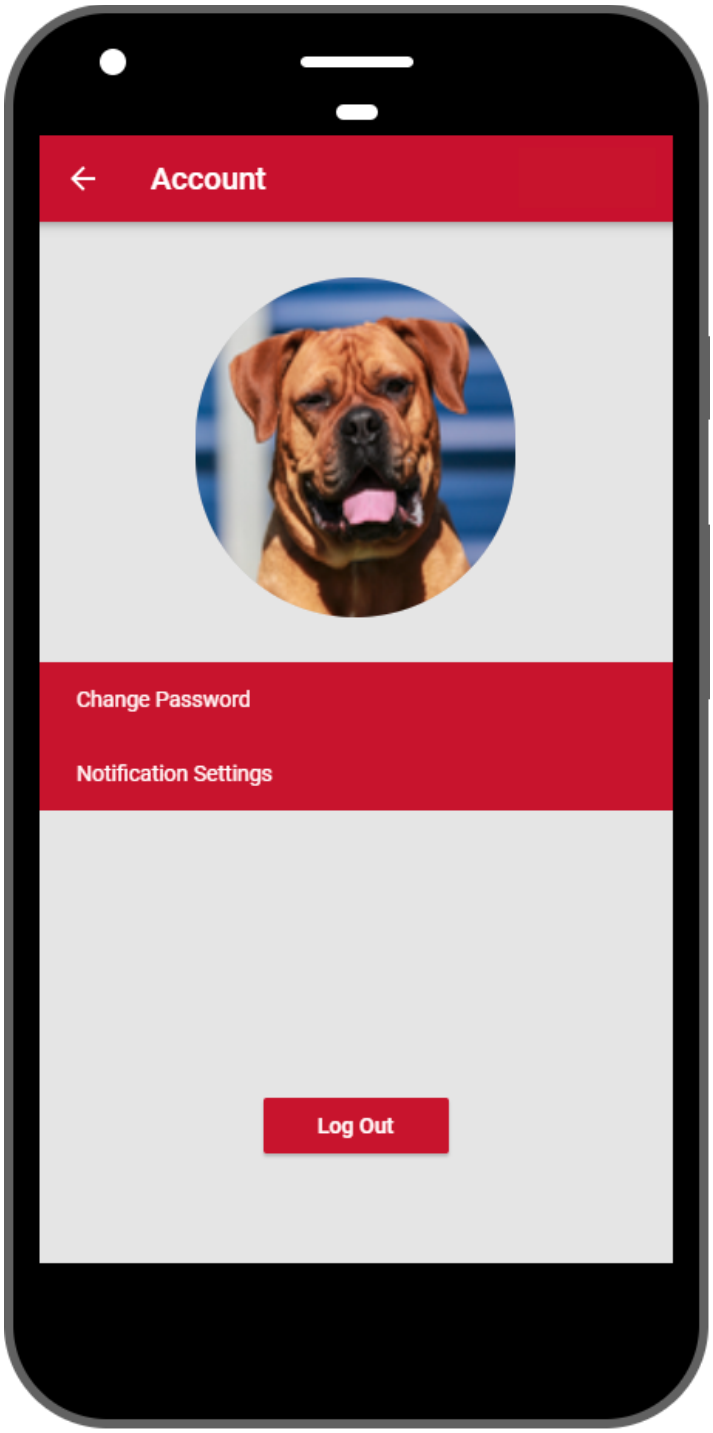
Dashboard View


Fleet Monitoring System
AB
Sinistra regna duas erat aestu
astra poena terris cingebant...
Feb 18, 05:22 pm
Sinistra regna duas erat aestu
astra poena terris cingebant...
Feb 18, 05:25 pm
Chat
Dashboard
Map

Chat View

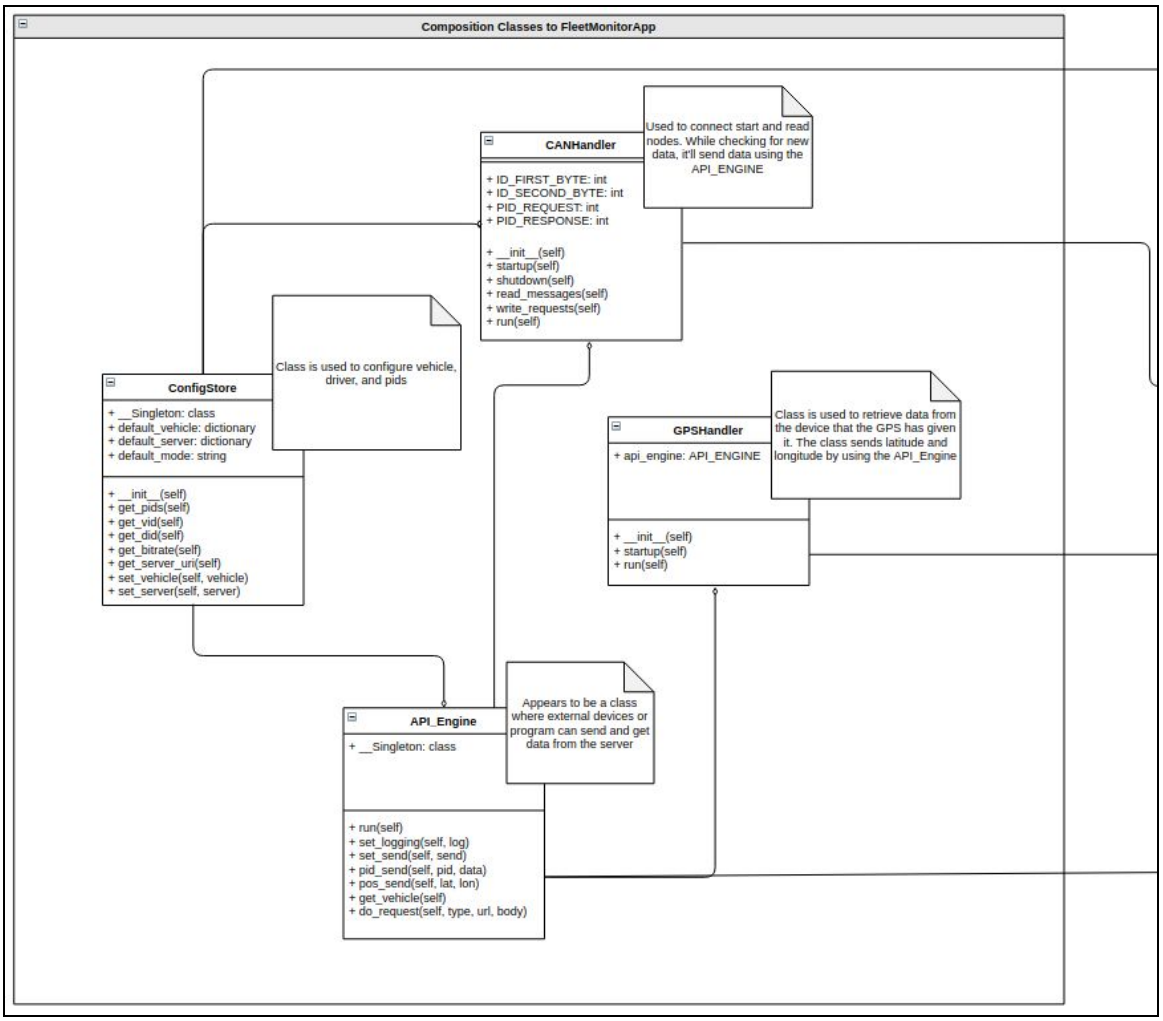

Fleet Monitoring System
Map of all the vehicles in real-time
Chat
Dashboard
Map

Map View

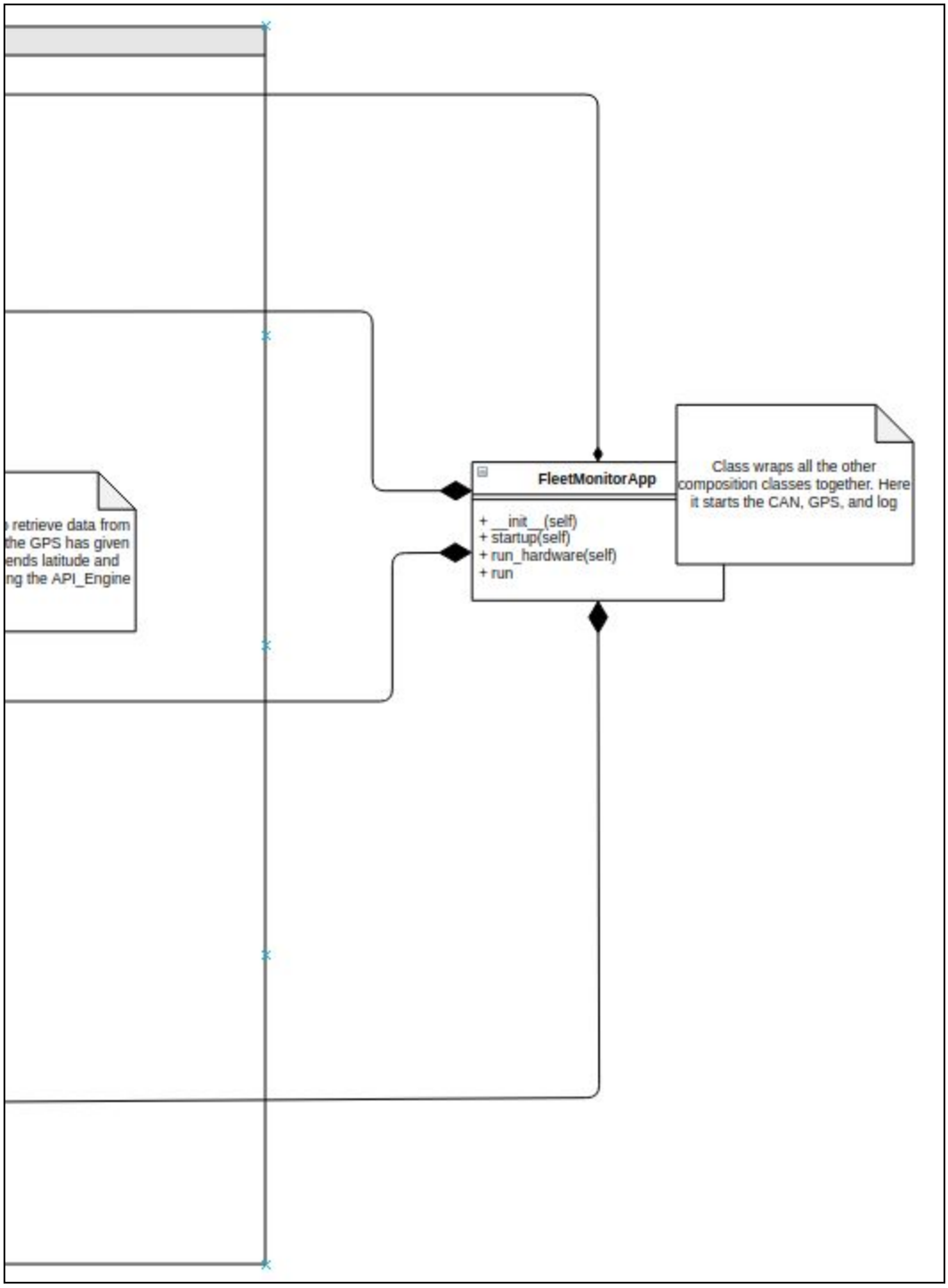


Profile Settings View

Embedded System Class Diagram:



Left-side of diagram (next image is right-side)



Right-side of diagram (previous image is left-side)

3. Statement of Work

3.1 Previous Work And Literature

Teletrac Navman

Features:

- Data analysis of fuel use-per trip and averages, idle time-total events and duration, and more
- Communication between drivers and dispatchers in a two-way messaging system
- Allows for monitoring activity within a user-set perimeter
- Alerts and proactive management

Advantages/shortcomings:

- No real-time tracking
- Data analysis

Disadvantages:

- Cost

WebNMS Fleet Management Solution -- Amazon's:

Features:

- Real-time visibility of vehicle location
- Driver behaviour monitoring
- Fuel/cargo management
- OBD data monitoring

Advantages/shortcomings:

- Made by a well known company, Amazon

Disadvantages:

- Meant for large fleets that do delivery
- Cost

Onfleet features:

- SMS notifications
- Real-time tracking
- Driver chat
- Data visualization

Advantages:

- Drivers have the ability communicated to each other
- Data visualization

Disadvantages:

- Cost

Sources:

https://www.teletracnavman.com/

https://aws.amazon.com/marketplace/pp/ZOHO-Corporation-Private-Limited-WebNMS-Fleet-Mana/B07V9V4CTF

https://onfleet.com/

Note: For the server, android app, and webpage we started from scratch. The embedded system inherianted code from the previous team's project. Although, refracting was needed for the embedded system.

3.2 Technology Considerations

Technology includes:

- RaspberryPi
  - Strength: Adaptable to other technology
  - Weakness: High cost, bulky, small computational power, and inconvenient
  - Tradeoff: Inexpensive
- Android Studio
  - Strength: Very friendly community, tons of features, constantly being updates
  - Weakness: Large learning curve, some components may be hard to research about to learn how to implement, only android applications can be created, Java is used
  - Tradeoff: Free to use, open source, android studio can be used on any operating system
- ExpressJS
  - Strength: Large community, matured framework, Fast app development because you can use the same language for frontend and backend.
  - Weakness: No error handling, tedious tasks, Code organization
  - Tradeoff: Free to use, open source
- Digital Ocean
  - Strength: Easy to use, and deploy for starters
  - Weakness: Smaller company runs it, less features, a lot of components have to be implemented on your own.
  - Tradeoff: As a student I get a lot of credit towards server costs
- MongoDB
  - Strength: Easy to use, and scalable
  - Weakness: High data consumption; no default transaction support
  - Tradeoff: Free to use for a single database

3.3 Task Decomposition

Develop a server

- Migrate to AWS
- Develop API
- Implement Microservices
- Implement Socket Broadcasting for real time tracking

Develop an android application

- Login screen
- Sign up screen
- Real-time tracking of drivers
- Chat system
- Data summary

Develop a web application

- Login screen
- Sign up screen
- Real-time tracking of drivers
- Chat system
- Data summary
- Data analytics

Develop the communication between embedded system and vehicle

- Obtain vehicle real-time diagnostic data
- Retrieve location of current vehicle
- Implement user authentication

3.4 Possible Risks And Risk Management

Some concerns that may affect our plan include knowing which frameworks and resources are best for our project. An example of this knowing which resource is best to use on our server and knowing which framework best fits for creating our web application. A possible risk for our project includes the PiCan, no one in our group has a lot of knowledge with embedded systems. So there are some risks with learning about as we are working.

3.5 Project Proposed Milestones and Evaluation Criteria

Most of our milestones include being able to read data with Pican. being able to run our server, and having different parts of both of our UIs working. An example of one of these milestones includes having a use-case for the mobile application and web application. This includes having both of these UIs read information from a server that is received from the PiCan. A lot of milestones will include having components working correctly with each other. To test these we will meet with our groups and run test cases and see if our components will communicate pass data correctly.

3.6 Project Tracking Procedures

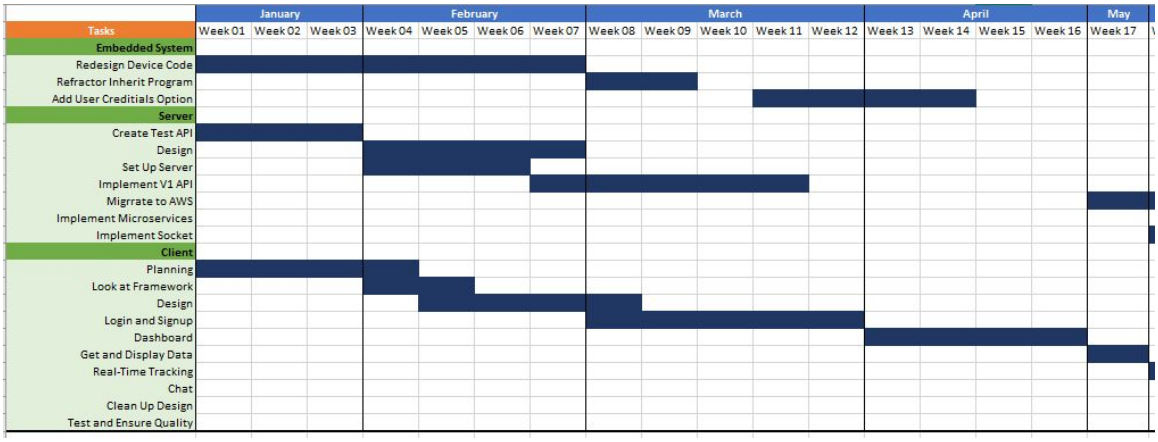
Our group will use Trello to keep track of each task that our members will be doing and we will be using to keep track of tasks that are to be done and tasks that are completed. We will also have weekly meetings to talk about project progress. This involves two meetings, one as a team meeting and with our stakeholder.

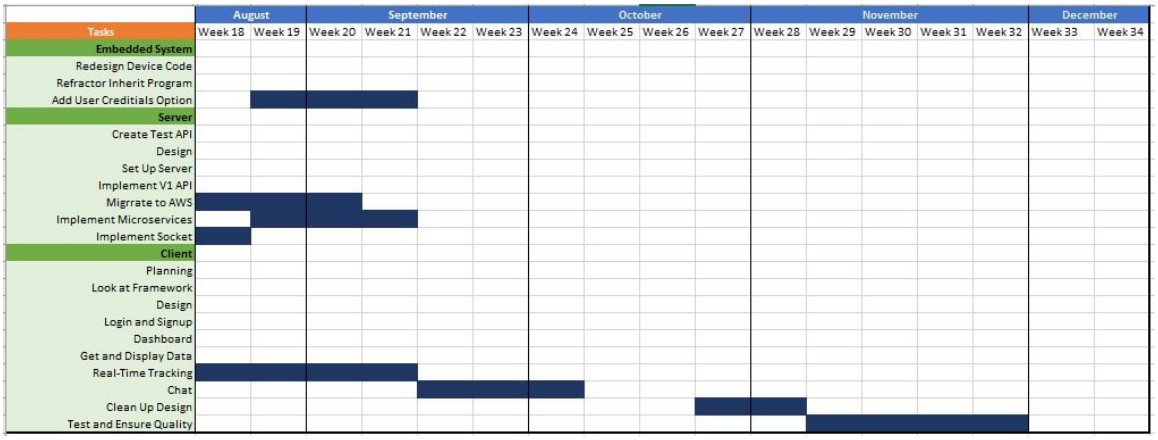
3.7 Expected Results and Validation

The desired outcome is to have a fully functioning Web Application, Mobile Application, PiCan, Server, and Database. These 4 components should be able to communicate with each other and be able to handle data correctly. Our solution will work at High Level if everything will work correctly and the use-cases will be done correctly by our UIs.

4. Project Timeline, Estimated Resources, and Challenges

4.1 Project Timeline





4.2 Feasibility Assessment

Senior Design Group 12's project will be a multi-platform for mobile and web application. The applications will have numerous features like group chat, data visualization, and real time tracking. The challenges from the project is that a lot of the technologies being used are new to our developers, and have to spend numerous hours on learning the new technologies.

4.3 Personnel Effort Requirements

Task	People Worked	Time Spent (hours)
Research current fleet monitoring services	Joe, Lorenzo, and Nicolas	3 (total)
Research database to use for project	Nicolas	1
Run inherited code for RaspberryPi on vehicle	Joe and Marco	10 (total)
Refactor RaspberryPi code	Marco	29
Boot up new server on DigitalOcean	Nicolas	2
Set up database	Nicolas	1
Design database architecture	Nicolas	2
Develop vehicle routes: • POST vehicle OBD status • POST vehicle GPS status	Nicolas	8
Develop vehicle routes: • GET vehicle OBD status • GET vehicle GPS status	Nicolas	8
Develop user routes: • POST login • POST sign up	Nicolas	8

Design mobile app user interfaces	Joe	4
Develop mobile app login user interface	Joe	12
Develop mobile app data display user interface	Joe	12
Design web app user interfaces	Lorenzo	5
Develop web app login user interface	Lorenzo	13
Develop web app data display user interface	Lorenzo	13

4.4 Other Resource Requirements

Materials:

-Hardware: Simulator, HOVA Hologram, GPS MicroSD, Raspberry Pi

-Server: DigitialOcean Web Servers, ISU Web Server

4.5 Financial Requirements

Hardware: $100

TOTAL: $100

5. Testing and Implementation

Testing is an **extremely** important component of most projects, whether it involves a circuit, a process, or a software library

Although the tooling is usually significantly different, the testing process is typically quite similar regardless of CprE, EE, or SE themed project:

1. Define the needed types of tests (unit testing for modules, integrity testing for interfaces, user-study for functional and non-functional requirements)
2. Define the individual items to be tested

3. Define, design, and develop the actual test cases
4. Determine the anticipated test results for each test case 5. Perform the actual tests
6. Evaluate the actual test results
7. Make the necessary changes to the product being tested 8. Perform any necessary retesting
9. Document the entire testing process and its results

Include Functional and Non-Functional Testing, Modeling and Simulations, challenges you've determined.

5.1 Interface Specifications

– Discuss any hardware/software interfacing that you are working on for testing your project

5.2 Hardware and software

– Indicate any hardware and/or software used in the testing phase

– Provide brief, simple introductions for each to explain the usefulness of each

5.3 Functional Testing

Examples include unit, integration, system, acceptance testing

5.4 Non-Functional Testing

Testing for performance, security, usability, compatibility

5.5 Process

– Explain how each method indicated in Section 2 was tested

– Flow diagram of the process if applicable (should be for most projects)

5.6 Results

– List and explain any and all results obtained so far during the testing phase

- – Include failures and successes
- – Explain what you learned and how you are planning to change it as you progress with your project
- – If you are including figures, please include captions and cite it in the text

• This part will likely need to be refined in your 492 semester where the majority of the implementation and testing work will take place

-**Modeling and Simulation**: This could be logic analyzation, waveform outputs, block testing. 3D model renders, modeling graphs.

-List the **implementation Issues and Challenges**.

6. Closing Material

6.1 Conclusion

Summarize the work you have done so far. Briefly re-iterate your goals. Then, re-iterate the best plan of action (or solution) to achieving your goals and indicate why this surpasses all other possible solutions tested.

6.2 References

This will likely be different than in project plan, since these will be technical references versus related work / market survey references. Do professional citation style(ex. IEEE).

6.3 Appendices

Any additional information that would be helpful to the evaluation of your design document.

If you have any large graphs, tables, or similar that does not directly pertain to the problem but helps support it, include that here. This would also be a good area to include hardware/software manuals used. May include CAD files, circuit schematics, layout etc. PCB testing issues etc. Software bugs etc.