

2nd Semester Work Statement

Team Name: Shocks In Awe

Team Members: Hector Vazquez, David Leung, Vladimir Nava, Kian Farley

Scope:

The Smart Bus project is designed to assist students, parents, and administrators who utilize the public school bus system. Our app will allow parents and students to track bus locations and receive notifications sent by administrators. The bus tracking will be accomplished by a raspberry pi attached to the physical bus, which will send GPS location data to an online database. This database will allow the app to receive location data, as well as confirm login information and other privileges related to each individual account. Our team will be responsible for the hardware, database management, and the application itself, both the UI and background code.

Period of Performance:

Week 1 - Week 2:

- Access to github using source control with Xcode
 - Each member will be shown how to access the application code using Xcode in collaboration with source control which will allow us to all access the project with github.
 - Team members of the front end will need access to an apple machine with xcode.
- Assign new roles - Authentication, Notifications, Maps , Database
 - The team will have 3 front-end and 1 back-end developers
- Contact Haysville
 - Set up a meeting with Haysville to present the progress we've made and our feature plans with the project.
 - Speak about the legal remedies of a tracker on the bus.
- Tracking
 - Research if there is a better method then using the raspberry pi 3, apple air tag?

Week 3 - week 6

- Apple Developer Licence
 - Speak to the professor about this licence.
- Working on assigned roles
 - Research on how to use firebase to store sensitive user login information
 - Research on how to store notifications in firebase and how to send them out
 - Was there a better method than using a raspberry pi?

- Change the appearance of the map and figure out how to display a different map per user.
- How we should organize the Data base - team
 - Create the Database, we need a collection of users and busses
- GPS testing
 - In vehicle testing
 - Data being sent to firebase?

Week 9 - Week 11

- Start to tie work done by individual team members to the project
 - Tie firebase to the authentication lock screen along
 - Tie firebase to notifications panel
 - Run application on phone for testing

Week 12 - Week 15

- Testing - Testing of the application and hardware would be complete
 - Authentication must work with created accounts in firebase and allow us to login with that information
 - Alerts must be able to be sent to multiple users
 - With authentication working, that certain user will only be allowed to see one bus which is tied to their account
 - Admin user will see multiple busses
- Prototype is complete for showcase

Task and Deliverables:

Software:

ITEM	Description
User Login	Allow Users to Login with specific Username and password
Store Password/Username	Create and encrypt storage for username and password login information.

Password Recognition and Verification	Allow users to change their passwords and use those passwords to login
Protect Password and Username Inputs from SQL Injection/buffer Overflow attacks	Optional: Protect the database and other inputs from common attacks such as SQL Injection and Buffer Overflow
Notifications and Updates	Allow for only authorized individuals to send notifications while only specific users receive those notifications
Bus Information Display	Display only the information on the bus(ses) the current user has access to
Expected Arrival Time of Bus	As part of bus information display, calculate and display an estimated arrival time based off previous travel times and current time
Update Database	Update Firebase account to allow for greater data transfer or find another way to store data for use.
Raspberry Pi	<p>Update code on Pi to start/stop at a specific time or lack of GPS data change. Focus on preventing unnecessary data from being sent.</p> <p>(possible) Store data on pi for backup and testing purposes</p>

Hardware:

ITEM	Description
GPS Module Implementation	The module must be operational and be able to communicate with the microcontroller (raspberry pi)
SIM Card Module Implementation	The SIM card module will be used to send GPS data using the NMEA standard to the Firebase Firestore Database. The module will need to be configured and tested with the microcontroller along with the necessary code.
Microcontroller Implementation (raspberry pi)	The microcontroller will be configured based on the requirements specified. The microcontroller will need to be tested on a moving vehicle along with the associated modules. This should ensure that the GPS tracker will be usable for future use and scalable for multiple buses.
Power Management Implementation	The GPS tracker will be powered via a micro-usb to cigarette outlet (USB). This will be tested ahead of time in order to make adjustments to the final tracker.

Work and Performance Schedule:

- Week 1/16: Research conducted for microcontroller, GPS module, and SIM card module
- Week 1/23: Hardware chosen and ordered for testing and configuration, GitHub source control configured for team usage
- Week 1/30: Database rules researched and tested, Front-end development and research for notifications started, back-end development for GPS tracker started, developers license acquisition
- Week 2/6: Database management read-write rules tested, stationary GPS tracker tested

- Week 2/13: GPS tracker tested on a moving vehicle, results gathered for refinement, power consumption monitored, Design of app refined
- Week 2/20: Database organization finalized, GPS tracker software refined
- Week 2/27: Refine GPS tracker, building authentication system started
- Week 3/6: App analytics researched and tested for further use cases, building notification system started
- Week 3/13: App appearance and general user interface refined, database organization refined
- Week 3/20: Finalize GPS tracker build, continue working on the front-end for notification system and authentication
- Week 3/27: App analytics researched presented to team, app appearance finalized
- Week 4/3: Database management system finalized, read-write rules finalized
- Week 4/10: Notification system finalized, compilation of work from front-end and back-end done
- Week 4/17: Authentication system tested and finalized
- Week 4/24: Alert system tested and finalized
- Week 5/1: Make final adjustments to front-end UI and database management, start preparing for final presentation showcase
- Week 5/8: Present for finals

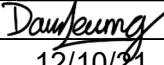
Acceptance Criteria:

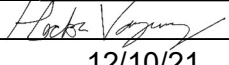
Account Creation	Working button that leads to an account creation screen where user can fill in email and password to create a new account
User Login	There exists a working login screen The login screen allows for different users to authenticate and login
Store Password/Username	Passwords are stored somewhere
Password Recognition and Verification	Users can change password Users are not allowed to login if the password is incorrect
Protect User Inputs from Attacks	Common attacks like buffer overflow or SQL injection do not work

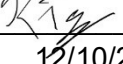
Notifications and Updates	<p>Administrators can send updates and notifications via the app to users</p> <p>Administrators can send updates and notifications via SMS to users</p>
Bus Information Display	<p>The app displays bus information such as:</p> <ul style="list-style-type: none"> - Bus number - Current location - Expected Path
ETA of Bus	<p>The app displays the expected time of arrival of the buses the user can see</p> <p>The database keeps track of expected time of arrival and continually updates the expected time with each new GPS coordinate of the bus</p>
Update Database	<p>The database used can handle more transfers each day. (there is currently a limit)</p>
Raspberry Pi	<p>The pi starts tracking and ends tracking at a specific time/event</p> <p>(possible) the pi can write to its own memory and update a file on recent GPS data transfer</p>


Signatures:

Each member needs to sign and date the Code of Conduct

Print: David Leung
Sign: 
Date: 12/10/21

Print: Hector Vazquez
Sign: 
Date: 12/10/21

Print: Kian Farley
Sign: 
Date: 12/10/21

Print: Vladimir Nava
Sign: 
Date: 12/10/21