10/13/2020, Treyvion Johnson

Project Proposal: **My Drop List Mobile**

### **Project Summary:**

The goal of this application is to transform the MyDropList web application into a mobile application. React Native will be used to create a front-end UI that will support Andriod and IOS devices. The front-end will connect to a Node.js API server to retrieve and process data. The QuickCharts API will be used to generate graphs that give meaning to the data stored.

### **The Problem:**

In my Capstone part one, I created a web application that solved some of the core problems of the company I work for ([Capstone one project proposal](https://docs.google.com/document/d/1HjXl9XsuWgirSDU3nMV01GXg1k_L0M1MsdNwRffoVvc/edit?usp=sharing)). The issues it did not solve were, workers needed an application they could access quickly, and be notified of changes that occur in the application. I found out a web application is not efficient enough and is too cumbersome for a fast-paced work environment.

The user experience is considerably clunky when trying to access the web application on their phones. It requires too many steps to gain access. A user must load the browser application, type in the URL (with their thumbs), and wait for the page to load. For an environment that is all about speed, there are too many steps. I find myself preferring to write my drop list on paper instead of going through these steps.

The other issue is the web application accessed by phone. Mobile-friendliness was the core design of the web application. There are features like swiping and other gestures that are helpful for mobile UI design but would be time-consuming or challenging to implement in a web application. By not having these mobile features, the user experience is not as friendly as it needs to be for mobile users.

### **The Solution:**

A React Native mobile application will allow me to be able to create a more mobile-friendly and consistent UI. In part one of this project, I had to redesign the UI for adding items to a droplist because I could not implement the swiping gesture on a web application. With a mobile application, I can use that original design that allowed items to be added in a less cumbersome manner.

I am using React Native to build the front end of the application because React Native allows for simultaneous development for both IOS and Andriod devices. The current backend is not flexible enough to support different applications with different UIs. I will replace the backend with a Node.js REST API because I can keep the web application (with some minor tweaks) and have a mobile application talking to the same backend while being in sync.

#### **Stretch Goals**:

* User online status (online, away, offline).
* Request expiration when not accepted or declined.
* Forwarding requests to other drivers.
* Notifications/Vibration when a request status change