**Title Page – GigGauge**

Team Members: Cameron Christensen

Description of project: The project has two parts, a physical device and a mobile app. The physical device is an Arduino uno board with some sensors to measure distance traveled. The mobile app is supposed to connect to the device and show the distance traveled, which is useful for tax deductibles later on.

## Section 1:

Introduction (A paragraph on your reasons for this project. WHY this idea?):

As a college student I would always look for ways to make money on the side and delivering food was one of those ways to make money on the side. I saw a need for something that would simplify tracking my mileage without having to use a whole other app on my phone, and I came up with this idea to create a device that would track my miles and then make it readable to put in tax forms later.

Purpose (A paragraph on WHAT your project/app will accomplish):

The device will track distance traveled while activated, and then the data will be shown in the app on a mobile device when the device is connected to the mobile device.

Scope (A paragraph on what your project will do and what it will not):

Technologies Used:

The project is being developed in the physical space using an Arduino uno board and some components that will help gather the data necessary to track distance traveled. The other part of the project is developed using react native and node.js to create the mobile application. The app will be used on my iPhone to see the final product.

# Section 2a:

Must Have Requirements: “shall”

Usability requirements:

### 2a.1 Device requirements

1. The application shall function on the iOS platform without restrictions.
2. The application shall only function on a smartphone device.

### 2a.2 Usability requirements

1. The GigGauge device shall function by being powered on by the user.
2. The application for GigGauge shall have an easy-to-use visual UI, with everything in one view.
3. The GigGauge device shall work with the mobile app.
4. The mobile app shall show the distance recorded by the GigGauge.

### 2a.3 functionality requirements

1. The main part of the application view shall have the recorded data from the GigGauge displayed.
2. The data recorded shall show the distance traveled.
3. The data recorded shall have a date shown for when the data was recorded.
4. The app shall allow for future updates and patches to add features.

### 2a.4 maintainability requirements

1. The app shall include documentation by the developer (me) outlining the development of the app.
2. The codebase of the app shall be easily modifiable for future updates.

**Section 2b:**

Stretch Requirements: (this is assuming I complete ahead of schedule)

1. The app shall allow a user to create a login with a username and password.
2. The app shall store user data safely in a private database.
3. The app shall have additional views for different features of the app.

**Section 2c:**   
Weekly schedule:

Requirements elicitation/ SRS document: 18hours week 3,4

Research and creating the physical device: 27 hours week 5, 6, 7

Write software for the physical device to track the mileage, test software and physical device: 18 hours week 8

Research and plan the mobile app: 18 hours week 8, 9

Begin writing base of mobile app: 9 hours week 10

Finish writing base and write features into the app: 18 hours: week 11,12

Finish features and final touches, test mobile app and physical device together: 18 hours week 13, 14

Total hours on project in semester: 126 hours planned to spend on project.

**Section 3:** Design Overview of the Product.

**Workflow:**

1. **User Launches the App:**
   * The user opens the GigGauge app on their mobile device.
2. **Menu Interaction:**
   * The left-side menu provides navigation options for the user. Initially, it contains a single item labeled "Menu". Future iterations may include additional items such as "Home", "Profile", "Settings", etc.
3. **Main Content Display:**
   * The main content area displays the title "GigGauge". This area will be used to display dynamic content based on user interactions with the menu.
4. **User Navigation:**
   * The user interacts with the menu to navigate to different sections of the app. Each section will have its own view and data.

**Resources**

**Resources:**

1. **Development Tools:**
   * React Native
   * Node.js
   * Android Studio (for Android development)
   * Xcode (for iOS development)
2. **Human Resources:**
   * Frontend Developer
   * UI/UX Designer
   * Backend Developer (for future iterations)
3. **Design Assets:**
   * App logo
   * Icons for menu items

**Data at Rest**

**Data at Rest:**

* **User Data:**
  + Profiles
  + Preferences
  + Settings
* **Static Content:**
  + App configurations
  + Localized text for different languages
* **Caching:**
  + Temporary storage for faster access to frequently used data.

**Data on the Wire**

**Data on the Wire:**

* **API Requests:**
  + Future implementation may include RESTful API requests to a backend server for data such as user information, app settings, and other dynamic content.
* **Authentication:**
  + Secure transmission of user credentials for login/logout processes (to be implemented in future iterations).

**Data State**

**Data State:**

* **Initial State:**
  + Default view with a blue background, a white left-side menu, and the title "GigGauge".
* **Transient State:**
  + User interactions with the menu that change the displayed content in the main area.
* **Persistent State:**
  + User preferences and settings saved locally on the device.

**HMI/HCI/GUI**

**HMI/HCI/GUI:**

* **Main Interface:**
  + Blue background.
  + White left-side menu with navigation items.
  + Main content area displaying the title "GigGauge".
* **Menu:**
  + Initially contains a single item labeled "Menu".
  + Future iterations may include additional items such as "Home", "Profile", "Settings", etc.
* **Styling:**
  + Consistent use of colors and fonts to ensure a cohesive design.

**Pictures/Diagrams**

**Pictures/Diagrams:**

* **Wireframe of the Initial View (rough low fidelity):**

plaintext

Copy code

-------------------------------

| Menu | |

| | |

| | |

| | GigGauge |

| | |

| | |

-------------------------------

* **Component Breakdown:**
  + **Container:**
    - Holds the entire layout with a blue background.
  + **Menu:**
    - Left-side menu with navigation items.
  + **Content:**
    - Main area displaying dynamic content based on user navigation.

**Section 4:** Verification:

**Demo:** The Demonstration will be recorded and uploaded to my own YouTube channel to be seen publicly.

**Testing:** The testing involved for this project will be manual testing, bothfor the GigGauge device and for the mobile app.

**Sources/Citation/Resources** Links: