**Test Plan**

This test plan aims to ensure the proper functionality of the eeZPower project and outlines the areas of testing that will take place. This includes tests under active use of both the app and physical device. These tests will be a combination of manual testing, software tests, and physical design tests. Given the scope of this project, testing can be broken down into three sections: Software, Hardware, and Design / Functionality.

Software testing will focus on the Android mobile app and will feature various unit tests to ensure functionality is as expected. The microcontroller will need to be tested as well and this will be carried out with manual tests to ensure proper functionality and communication between the motor controller and mobile app over Bluetooth. Hardware testing will be carried out manually with freely available software tools such as the vesc tool. This testing will cover the functionality of the battery, motor controller and remote control together. Design / Functionality testing will be carried out in an outdoor environment to ensure the device functions as expected in real world environments.

**Test Cases**

SA1.1 **Software Test for Application 1**

SA1.2 Ensure the eeZPower device is recognized as a Bluetooth device in the app

SA1.3 On the main app page, when the get paired devices is pressed after pairing with the ESP32 on the eeZPower device, “eeZPower” shows up

SA1.4 inputs: running ESP32, running eeZPower app, user button press

SA1.5 Esnure eeZPower is listed as a paired device

SA1.6 normal

SA1.7 whitebox

SA1.8 functional

SA1.9 unit

SA2.1 **Software Test for Application 1**

SA2.2 Ensure current stats of eeZPower device is populated within the stats page

SA2.3 Open the app, connect to the eeZPower device, and user moves to stats page while the eeZPower device is on

SA2.4 inputs: running eeZPower device, user on stats page

SA2.5 Current RPM and battery voltage should be populated

SA2.6 normal

SA2.7 whitebox

SA2.8 functional

SA2.9 unit

**MC1.1**  **Motor Controller Test VESC**

MC1.2 Ensure the motor controller works as expected

MC1.3 Plug motor controller into laptop and run vesc open-source software to read parameters

MC1.4 inputs: motor controller

MC1.5 vesc software can recognize, connect, and read and write app and motor configuration of the motor controller

MC1.6 normal

MC1.7 blackbox

MC1.8 functional

MC1.9 integration

**RC1.1**  **Remote Control Test**

RC1.2 To ensure the motor controller is reading input from remote

RC1.3 Leverage the vesc open source software to test the range of input the motor controller receives from the remote

RC1.4 inputs: motor controller (connected to laptop) and remote

RC.15 The range should be -100% when full brake on remote, and 100% when on full throttle, as well as 0% when giving no input on remote

RC.16 normal

RC1.7 blackbox

RC1.8 functional

RC1.9 integration

**BT1.1** **Battery Test**

BT1.2 To ensure the battery can power all components

BT1.3 Ensure the Battery can power the motor controller, motor and ESP32 concurrently

BT1.4 All physical components (motor, motor-controller , ESP32, remote, battery)

BT1.5 ESP32 stays on while motor is fully powered

BT1.6 normal

BT1.7 whitebox

BT1.8 functional

BT1.9 integration

EP1.1 **ESP32 Test UART**

EP1.2 To verify microcontroller is reading information from motor controller

EP1.3 View output from Serial Port with microcontroller attached to computer while operating the motor controller to verify output is as expected

EP1.4 inputs: motor controller , ESP32

EP1.5 Serial monitor should show current battery voltage and rpm

EP1.6 normal

EP1.7 whitebox

EP1.8 functional

EP1.9 integration

EP2.1 **ESP32 Test Bluetooth Connect**

EP2.2 To verify ESP32 is creating a Bluetooth connection

EP2.3 Attempt to connect / pair with the ESP32 using some other device (phone)

EP2.4 inputs: ESP32, mobile phone w/ Bluetooth capability

EP2.5 ESP32 should show up as “eeZPower” and be able to be paired with from phone

EP2.6 normal

EP2.7 whitebox

EP2.8 functional

EP2.9 integration

EP3.1 **ESP32 Test Bluetooth send data**

EP3.2 To verify ESP32 is sending correct data over Bluetooth connection

EP3.3 Using a Bluetooth terminal, read the output from the ESP32 over Bluetooth during operation of the device to view change in rpm & voltage

EP3.4 inputs: motor controller in use while varying rpms, ESP32, Bluetooth terminal app

EP3.5 The string containing rpm, voltage, and tach should be constantly updating in the Bluetooth terminal

EP3.6 normal

EP3.7 whitebox

EP3.8 functional

EP3.9 integration

**MT1.1 Motor Test**

MT1.2 To verify motor is connected correctly and spinning in correct direction

MT1.3 Connect the motor to the motor controller and test operating functionality w/ remote or vesc software

MT1.4 inputs: motor controller, motor, battery, vesc or remote input

MT1.5 The motor should spin in a direction that sends the board forward when receiving forward throttle from remote

MT1.6 normal

MT1.7 blackbox

MT1.8 functional

MT1.9 integration

**PD1.1 Physical Design Test**

PD1.2 To ensure physical design is a proof-of-concept prototype and demonstratable

PD1.3 Does the physical design keep all electrical components secure to the board and allow for the board to move when placed on the ground

PD1.4 inputs: proof of concept prototype attached to board and remote

PD1.5 All components should stay securely on the board when forward throttle is applied on the remote and the board should move forward, with no loose connections

PD1.6 normal

PD1.7 blackbox

PD1.8 functional

PD1.9 integration

