BU Department of Electrical & Computer Engineering

# Boston University

# EC 464 Senior Design Project II

## Final Prototype Test Report

FLEXTEND

Written By
Team 24: Flextend

Team Members
Thomas Scrivanich
Carmen Hurtado
Sohaib Ansari
Jack Halberian
Murat Sencan

## 1.0 Required Materials

### 1.1 Hardware

1. Thigh strap with circuit
    a. MPU6050 Accelerometer Gyroscope unit
    b. Arduino Nano 33 IoT
    c. 9V Battery
    d. 1 Switch (on/off)
    e. Buzzer
    f. Soldering board and cables
2. Shank strap with circuit
    a. MPU6050 Accelerometer Gyroscope unit

### 1.2 Software

1. Flextend Application
    a. iOS operation (iPhone simulator/iPhone)
    b. Android operation (Android emulator/Android)
2. Google Firebase

## 2.0 Set Up

2.1 The setup for the testing is in three parts: Arduino board with MPU6050 sensor strapped on the thigh. MPU6050 sensor strapped to the shank. React Native mobile application, both Android and iOS, for software. The HW and SW are now connected via Bluetooth Low Energy (BLE), and part of the test will show this functionality. The mobile device is connected to a computer via USB cable and the application will be run through there. We will be testing login and registration authentication with *Google Firebase*. In the login screen we will be testing that as a phone number is entered the login is successful (navigates to home screen) and that the phone number gets stored in Google Firebase. After navigation to the home screen, we will navigate to the previous result screen where any values for the login user that is stored in the database will be displayed on screen (flexion and extension). We will also be testing the User Profile, where the user can track their progress and set calendar events, and so on. Next step will be to test the BLE functionality that it's implemented using *react-native-ble-plx* module. To do this we will show that measurement readings for the MPU6050 units display properly on the "Begin Measuring" page. On the Arduino side, we have the two MPU6050 connected to calibrate and measure the degrees of extension and flexion. This data is sent to the Nano and then to the mobile application, and lastly stored in Firebase.

| **Circuit Pinout Connection** | |
|---|---|
| **MPU6050 Gyroscope 1** | **Arduino Nano 33 IoT** |
| GND | GND |
| Vin | +3.3V |
| SDA | SDA (A4) |
| SCL | SCL (A5) |

| **Circuit Pinout Connection** | |
|---|---|
| **MPU6050 Gyroscope 2** | **Arduino Nano 33 IoT** |
| GND | GND |
| AD0 | +3.3V |
| SDA | SDA (A4) |
| SCL | SCL (A5) |

## 3.0 Pre-testing Procedure

### 3.1 Hardware

1. Upload measurement + BLE code to Arduino (if not done already)
2. Put the device on a flat surface for calibration.
3. Ensure the 9V battery is connected to power in the breadboard and switch is on.
4. Make sure Nano 33 IoT and gyroscope lights are on.
5. Once the mobile application is started, connect to Arduino and press the calibration button on the app (This will take a couple of seconds).
6. Put on the device by strapping the top piece to the thing of the user and the bottom piece to the shank of the user.
7. Press the "measurement" button to begin sending measurements to the app.
8. Press stop measuring when the process is done.

### 3.2 Software

1. Plug in the iOS or Android device to the host computer (if testing iOS this must be a Mac, otherwise we will use a Windows PC)
2. Start the app on the host computer and begin testing on the mobile device.
3. Login to the Firebase project on computer and open the database.

## 4.0 Testing Procedure

### 4.1 Arduino and Electronics

1. Calibrate device on a flat surface through the button signal on the application.
2. Strap the device to the leg.
3. Press the measure button to start measuring.
4. Tilt leg to angle on paper and observe accuracy
5. Confirm that the Arduino is efficiently outputting measured rotation data in degrees to the computer or mobile app.
6. Repeat steps for each angle and calibration modes.

## 4.2 Mobile Application

1. User registration: enter user information and confirm, redirects to Login Screen
2. On the login screen enter a valid phone number of an unregistered user with the correct format (+1 234 567 8910).
3. This action will fail as all users need to register before accessing the application's home screen.
4. Enter a valid phone number of the user we previously registered. This action will succeed after authentication of the confirmation code and the user is redirected to the Home screen.
5. Verify that the registered phone number is stored in Firebase database (Through computer screen inside Firebase project).
6. Start the device.
7. Navigate to the Measuring Screen. Calibrate device. Start measuring and recording data from the device. Having a protractor in front of the device confirms that the values in the screen are accurate. Values should be changing as the leg moves to show live measurement functionality.
8. Navigate to the Previous Result screen and verify that it shows information for the logged user.
9. Confirm that it is the same information stored in the Firebase database.
10. Navigate to Profile Screen. Users can change profile photos. Demonstrate this.
11. Pop up modal for Body Metrics. Users can enter their information and it is saved.
12. Pop up modal for Reminders will have two options: add a new event and edit the event. The edit event will only be available if the user has set up a reminder before.
13. Confirm that these both work by creating an event, then go to the phone's calendar and make sure it is there.
14. Edit this event from the app and check that the changes are reflected in the phone's calendar.
15. Navigate to the Progress screen and see past measurements for the logged in user.
16. Generate a report for the user and verify it has the correct data.

17. Confirm that these values are correct by looking at the Firestore DB.
18. Add a new goal in the Goals pop up modal. Verify that it is added to the Firebase collection. Verify that the new goal is shown in the Home screen.

## 5.0 Measurable Criteria

### 5.1 Hardware

1. The MPU6050s successfully capture movement of the leg in degrees.
2. The Arduino accurately transfers measurements in real time through bluetooth.
3. Calibration of sensors work effectively.

### 5.2 Software

1. The user is able to Register to the application with their phone number and name.
2. The user is able to log into the application by providing their phone number as well as the country code.
3. Google Firebase records any phone number used to sign and register into the application.
4. The user navigates the application successfully from any given screen.
5. The hardware device can successfully connect to the application and send data over this channel.
6. User can calibrate and start measuring from the application screen.
7. The user can see their individual past measurement on the Past Results Screen. Google Firebase Firestore database contains all specific user metrics for knee flexion and extension with timestamps. The application successfully gets data and pulls data from Firestore.
8. Users can successfully select and change their profile photo.
9. Users can successfully add and edit calendar events for reminders.
10. Users can see their past progress in the form of a graph with legible and organized values.
11. Users can add new goals.

## 6.0 Score Sheets

### 6.1 Hardware Power Switch

| Device Power On? (Y/N) |
|---|
| |

### 6.2 Hardware Calibration

| Calibrated Properly? (Y/N) |
|---|
| |

### 6.3 Hardware Accuracy

| Accuracy |
|---|
| |

### 6.4 Software Score Sheet (Registration/Login)

| Registration/Login Successful? (Y/N) |
|---|
| |

### 6.5 Software Score Sheet (Navigation)

| Navigation Errors? (Y/N) |
|---|
| |

6.6 Software Score Sheet (Bluetooth)

| **Successful Connection to App? (Y/N)** | **Data** |
|---|---|
| | |

6.7 Software Score Sheet (Calendar Reminder Event )

| **Event Created (Y/N)** | **Event Shows on Calendar (Y/N)** | **Event Can be Edited (Y/N)** |
|---|---|---|
| | | |

6.8 Software Score Sheet (Goals)

| **Goal Created (Y/N)** | **Goal Shows on Home Screen (Y/N)** |
|---|---|
| | |

6.9 Software Score Sheet (Metrics)

| **Previous Results Correct? (Y/N)** | **Progress Correct? (Y/N)** | **Report Screen w/ Metrics Correct? (Y/N)** |
|---|---|---|
| | | |