6-10-2020

A Low-Cost and Energy-Effcient Wearable Device For Posture Monitoring

George Shappell

Ying Huang

Ali Nazemi

SANTA CLARA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: June 10, 2020

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

George Shappell
Ying Huang
Ali Nazemi

ENTITLED

A Low-Cost and Energy-Efficient Wearable Device For Posture Monitoring

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

B. Dezfouli

Thesis Advisor

N. Ling

Department Chair

A Low-Cost and Energy-Efficient Wearable Device For Posture Monitoring

by

George Shappell
Ying Huang
Ali Nazemi

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 10, 2020

A Low-Cost and Energy-Efficient Wearable Device For Posture Monitoring

George Shappell
Ying Huang
Ali Nazemi

Department of Computer Science and Engineering
Santa Clara University
June 10, 2020

ABSTRACT

Millions of Americans suffer every year from back problems, now imagine if their was a way to help track and prevent back problems. Our solution to this problem is PostureBot a device that will help its users to correct their back posture and maintain good back posture.In doing so this device can help elevate and prevent people from developing minor and serious back problems in their future.

Table of Contents

List of Figures

Chapter 1

Introduction

Back pain is an ailment that will most likely affect every person in their lifetime. The American Chiropractic Association lists back pain as the single leading cause of disability (1). Back pain is detrimental to the health of an individual, limiting mobility and comfort-ability, but can also pose a serious risk to a company's productivity; back pain is estimated to account for 264 million lost days of work in one year(1).

An improper curvature of the spine can be corrected by surgery. However, the procedure is extremely costly and physically taxing with a long recovery time. Spine surgery is costly and the recovery is time-enduring. When combined with surgical costs, medications, magnetic resonance imaging (MRI), rehabilitation and disability, the average spine surgery case approaches $100,000 or more. The direct costs are astronomical and may reach as high as $169,000 for a lumbar fusion and $112,000 for a cervical fusion(2). Furthermore, as the problem progresses, patients sometimes fail to adjust their lifestyle and may result in relapses of improper postures. Therefore, it is important to continuously monitor one's back posture for prevention and treatment.

As society spends more time sitting at desks and typing at computers, a non-invasive and proactive solution to back pain becomes paramount. There are several existing solutions on the market that strive to address this issue.

A back posture corrector acts as a harness around one's back which aims to "pull" the wearer's back. It prevents the wearer from slouching and forces the user to maintain a correct posture. This is a costly equipment which usually costs up to $100 yet fails to satisfy different wearers' body type and lifestyle.(3)

Wearable posture monitoring devices have been designed by a handful of startups. A popular device on the market called Upright provides back postures monitoring and mobile phone connection. Such device only monitors the wearer's back, and they fail to capture accurate data and back monitoring while maintaining a low cost.(4)

In this paper, we propose a low-cost and energy efficient wearble device capable of monitor a wearer's back posture. The device can record and relay that data to a mobile device. Our design is primarily focused on operation in a home environment, where the user is either sitting or standing still.

The purposed device consists of a hardware side and a software side. The hardware side includes an Arduino

Nano 33 board with two inertial measurement units(IMUs), a rechargeable battery and a vibration unit. The IMUs consist of an accelerometer and a groscope. The accelerometer is capable of motion detection and calibration while the gyroscope detects orientation and back angle measurement. The data collected by the hardware is transmitted to a mobile device via a Bluetooth link. The iOS application provides a visual feedback to the user.

Our design has the following features:

Posture Measuring

The device contains an Inertial Measuring Unit(IMU). They measure the back position of the user and his/her posture by calculating its position relative to the even earth.

Posture Monitoring

The device records the user's posture and compare it with the pre-set good posture by calculating its difference. It sends the data via the Bluetooth link to the user's mobile device and alerts the user when a bad posture is observed.

Unrestricted Mobility

The lightweight device is situated between the user's shoulder blades with an adhesive unit. This allows the user to move freely without any restriction to his/her mobility.

Low Price Point

The device utilized an Arduino board on the market. It is positioned to be affordable. The entire system is priced under $30.

Chapter 2

Design Requirements

2.1 Functional

2.1.1 Critical

- The PostureBot hardware unit must measure the orientation of the user's spine and back.
- The PostureBot hardware unit must be able to monitor the user's posture and calculate their postures.
- PostureBot must alert users to poor posture and slouching via notification through the PostureBot hardware unit or a mobile application notification.
- The hardware unit should be able to transmit recorded data to the user's mobile device via Bluetooth.

2.2 Nonfunctional

2.2.1 User Interface

- The PostureBot system should provide a visual interface to display a user's posture, alerts, and reports.
- The PostureBot system should provide the user with a report on their postures. The report shows their postures over time, and how their postures are improving when using the device.
- The mobile application should send the user alerts to stand and perform light stretching and exercises periodically to strengthen back muscles and combat slouching.

2.2.2 Performance

- The device should be able to function for at least 2 hours before needing to recharge its battery.
- The device should comfortably adhere to the user and not limit their daily mobility.

2.3 Design Constraints

- The software should function through an iOS mobile application to provide users with a visual interface.
- The device must stay in contact with the user's body at all times to accurately measure the user's posture.
- The PostureBot hardware unit should utilize the Bluetooth Low Energy (BLE) connection to pair with a mobile device and communicate with the PostureBot application.

Chapter 3

Use Cases

3.1 Configuration Use Case

When a user activates their PostureBot device for the first time, configuration is necessary in order to identify the user's range for acceptable posture. PostureBot asks the user to orient themselves to good posture; shoulders rolled back, chest up, facing forward. PostureBot then measures and stores the user's spinal orientation for use in determining good and bad posture in later use cases.

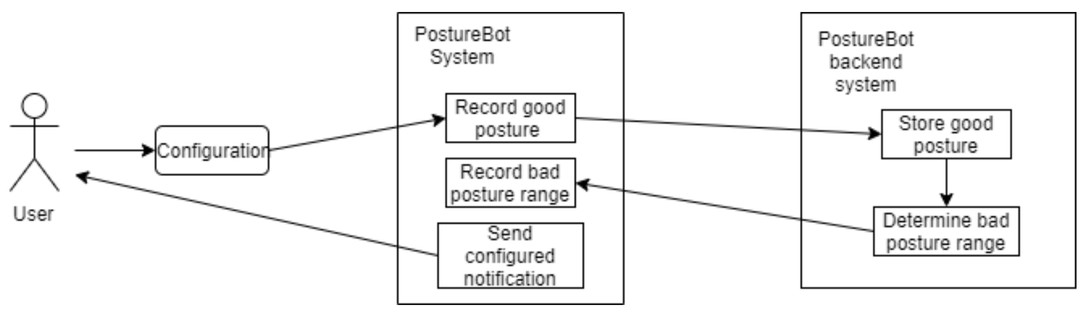

Figure 3.1: Configuration Use Case

3.2 Bad Posture Use Case

When a user is maintaining a bad posture, PostureBot records the posture for use in a posture report and send an alert to the user to adjust their posture. The alert can be sent in two ways; through a notification on the mobile app or a vibration alert on the device. Once the user has corrected their posture, the alert and logging of user posture ceases.

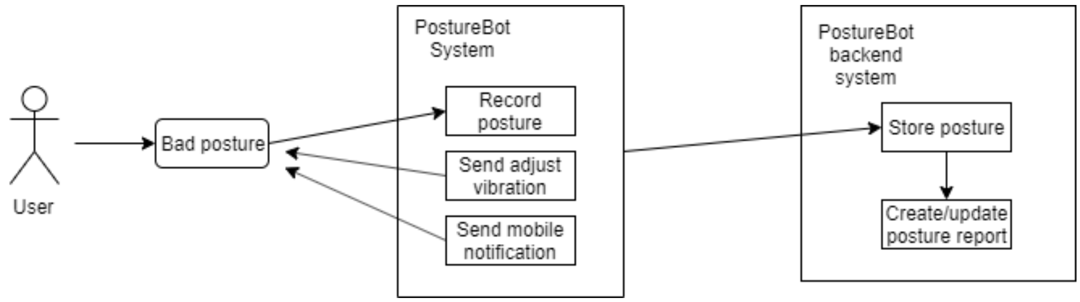

Figure 3.2: Bad Posture Use Case

3.3 Good Posture Use Case

While a user is maintaining acceptable or good posture, PostureBot periodically records the user's posture for use in generation of a posture report. The posture report can show a user how long they held good and bad posture during their day.

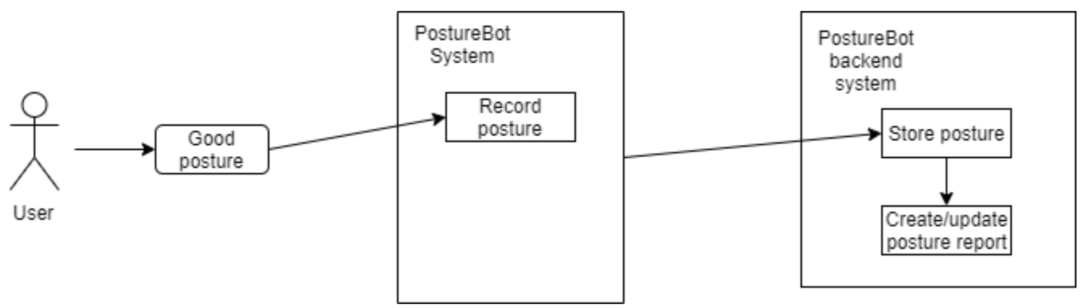

Figure 3.3: Good Posture Use Case

Chapter 4

Activity Diagram

4.1 Mobile App and PostureBot

The following diagram represents the various paths/flow the user will take in order to set postures and monitor his/her postures.

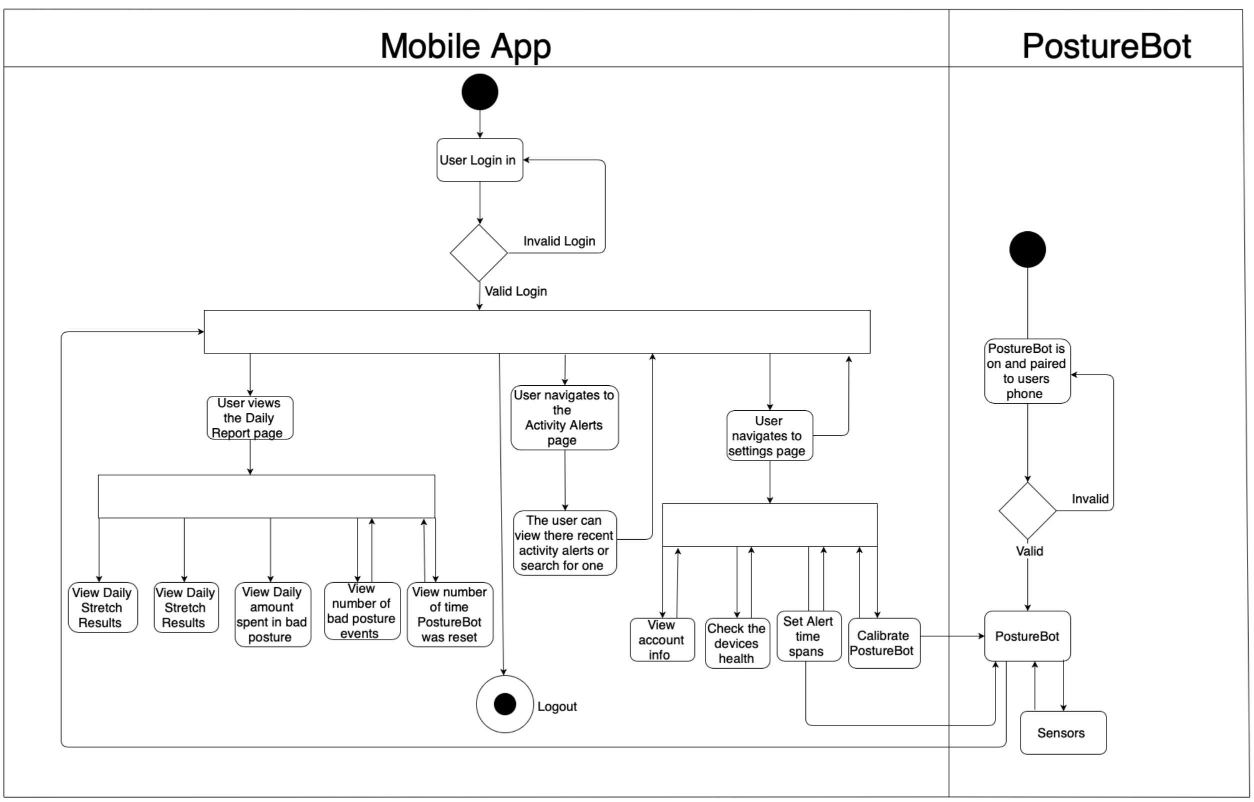

Figure 4.1: Activity Diagram

Chapter 5

Technologies Used

5.1 Hardware

- Arduino Nano 33 board
- Inertia Measuring Unit (IMU) Sensor

5.2 Back-end Software

- Spring MVC framework
  - The Spring framework is a framework for Java that is useful for configuration and management of the different components of an application. The Spring MVC framework allows for configuration and development suitable for a system using the Model-View-Controller architecture.
- Arduino Integrated Development Environment
  - The Arduino Integrated Development Environment (Arduino IDE) is an IDE that allows for editing and running of scripts on Arduino compatible devices. The Arduino IDE allows for the writing of Arduino scripts in the Java programming language.
- Bluetooth 5.0
  - Bluetooth 5.0 is software that allows for pairing and communication between devices. Bluetooth 5.0 is backwards compatible with other Bluetooth software versions and also adds Bluetooth Low Energy (BLE) technology to reduce power output of devices communicating using Bluetooth.

5.3 Front-end Software

- Swift Programming Language

- Swift UI Framework
- IOS app development environment
- Xcode

Chapter 6

Architectural Diagram

6.1 Device Overview

The proposed solution features four main components; the physical unit, web application with accompanying API, and mobile application. The physical unit contains the micro controller, sensors, and actuators of the system. The unit is responsible for collection of sensor data and actuation of vibration alerts. The unit is adhered to the users body in order to read their posture. A rechargeable battery is used for power supply. The web application processes the information received by the physical unit and prepare it for storage in the database, as well as determine the posture range in which bad posture alerts will be raised. The database stores the user's posture information, and use this information to construct a posture report unique to the user. The mobile application is responsible for relaying information to the user such as alerts and reports, as well as configuring the user's posture ranges.

The PostureBot system uses a center processing unit combined with a rechargeable battery and a vibration actuator on the user's back. The data recorded by the hardware is then relayed to the user's mobile device via a Bluetooth Link.Figure 7.2 provides an overview of the system.

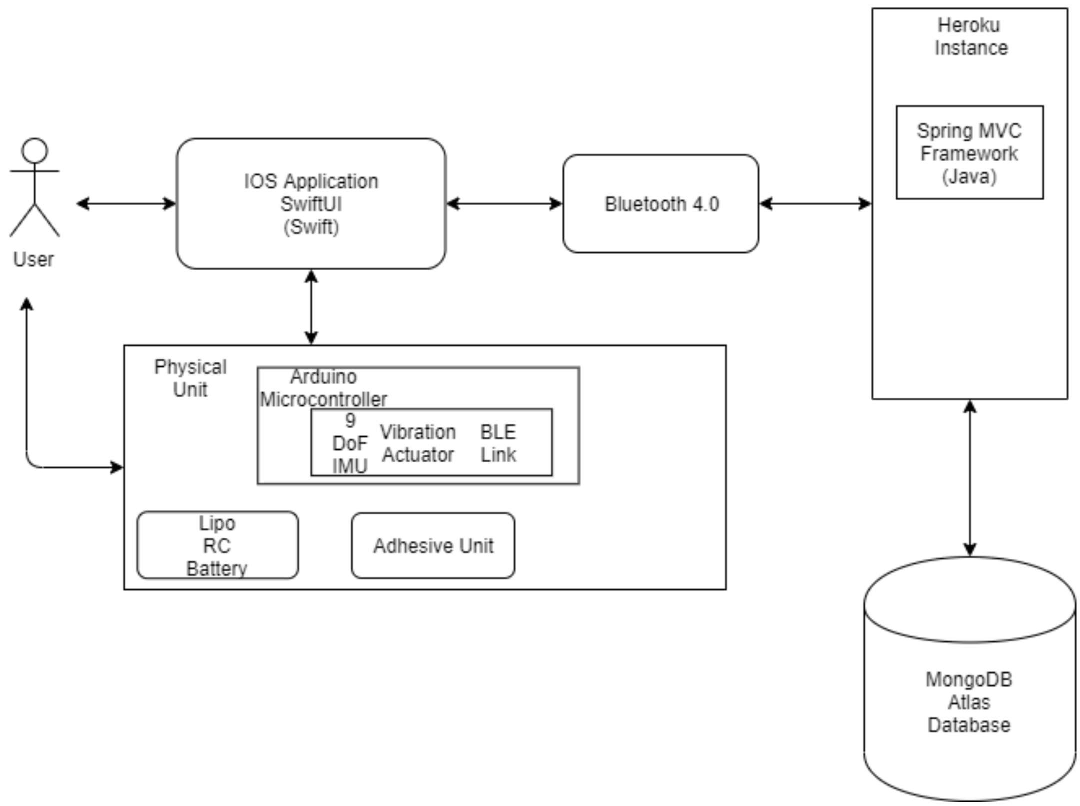

Figure 6.1: Architectural Diagram of PostureBot

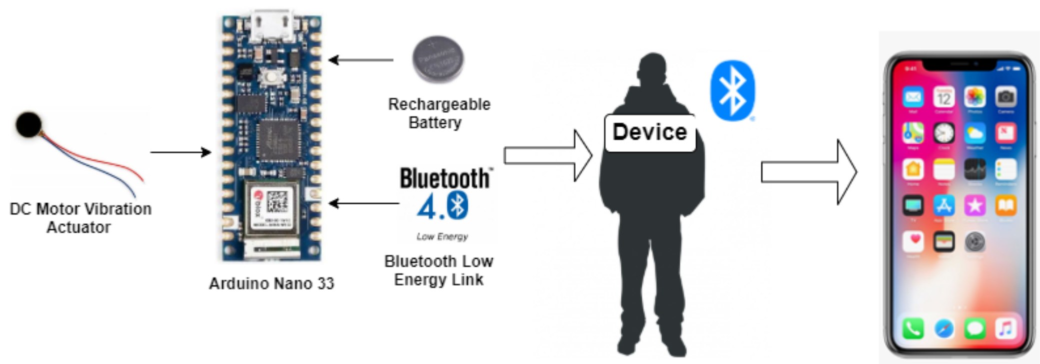

Figure 6.2: Device Overview

Chapter 7

Design Rationale

7.1 Hardware

7.1.1 Bluetooth 5.0/Bluetooth Low Energy

Bluetooth Low Energy (BLE) is a wireless personal area network designed for interconnecting devices on a personal user-space. BLE is an improvement of the traditional Bluetooth technology, and draws less power from the device than the standard Bluetooth. Because of its power efficiency and its ability to reach the same ranges as traditional Bluetooth, we have decided to implement PostureBot with a BLE connection. This connection allows the PostureBot to interconnect with a user's mobile phone and pair to an app in order to exchange data with both the back-end and the user interface.

7.1.2 Arduino

The Bluno Nano board is an Arduino product that offers Bluetooth 5.0 integration into an Arduino Nano module. The Bluno Nano was the best available board for the PostureBot design for being capable of providing BLE over its Bluetooth 5.0 software and also integrated into an Arduino Nano module allowing for rapid development and prototyping.

7.1.3 Inertia Measuring Unit (IMU) Sensor

The PostureBot design comes equipped with three Inertia Measuring Units (IMUs). an IMU offers nine degrees of measurement:

1. 3-axis Accelerometer

2. 3-axis Gyroscope

Using the measurements produced by the IMUs, we are able to calculate the orientation and axis of rotation of a user. These measurements will form the foundation of our posture determining system and allow us to accurately measure our users' postures. The accelerometer is an electromechanical device used to measure acceleration forces. By measuring the amount of static acceleration due to gravity, we can find out the angle the device is tilted at with respect to the earth. By sensing the amount of dynamic acceleration, we can analyze the way the device is moving. This provides orientation detection for our device when trying to measure the user's back posture. The gyroscope is a device that uses Earth's gravity to help determine orientation. Its design consists of a freely-rotating disk called a rotor, mounted onto a spinning axis in the center of a larger and more stable wheel. As the axis turns, the rotor remains stationary to indicate the central gravitational pull, and thus which way is "down." The gyroscope maintains its level of effectiveness by being able to measure the rate of rotation around a particular axis. This allows us to measure how much the user is slouching.

7.2 Software

7.2.1 App Development Environment

Swift is a programming language developed by Apple Inc. We are using Swift to develop our app, due to the high use and more popularity of iOS. We have chosen Swift as our front end language because it is well documented and the most commonly used ios development language.

7.2.2 Web Server and API

Our web server is ran using a web API constructed using the Spring Framework. Spring allows for optimized memory management and other features that make the implementation of the PostureBot a smoother process.

7.3 Non-Technical Components

7.3.1 Adhesive

To ensure that the PostureBot unit is properly positioned on the user, silicon adhesive pads is used to secure the device to the users body. Silicon adhesive pads were the most practical option for their adhesiveness and durability, allowing for users to be mobile and active while wearing the device.

Chapter 8

Implementation

8.1 Hardware implementation

The current protopye of the PostureBot system(Figure 9.1) is implemented with an adhesive unit on the back of the board. The board is then stick to the user's back between the two shoulder blades. The device is able to run for several hours on its battery and is capable of transmitting data to the user's mobile device.

The device was calibrated to be working vertically using Arduino's programming environment. The preset posture requires the user to wear the PostureBot and maintain that posture for an extended period for messing purposes. We than program that posture to be the default posture. Any deviation from that posture (An angle deviation from being vertical) would trigger the device to alert the console in the computer. This deviation is timed so that it does not send constant alerts to users when they are moving. A preset 15 second window is required for the internal processor to record that information and transmit to the mobile device via Bluetooth. The good posture period is then calculated by subtracting the total time the device is on minus tht total time the user is observed in a bad posture.

8.2 Software implementation

The iOS application is set up using Swift programming language. It sets up a communication via Bluetooth with the device to receive and send information. The iOS application is able to process that information and seperate on whether the user is maining a good posture. The application then store that information for bad posture alert via push notification. The iOS application has the following pages:

Figure 8.1: Current prototype of the PostureBot system

- User registration screen (figure 9.3) is set up with several text boxes. These text boxes can record user information and allow the user to register with their own email and password. The username and the password must be 6 or more characters long for security. In order to maintain account security, the password must contain letters, numbers and symbols. The resulting password is salted and hashed.

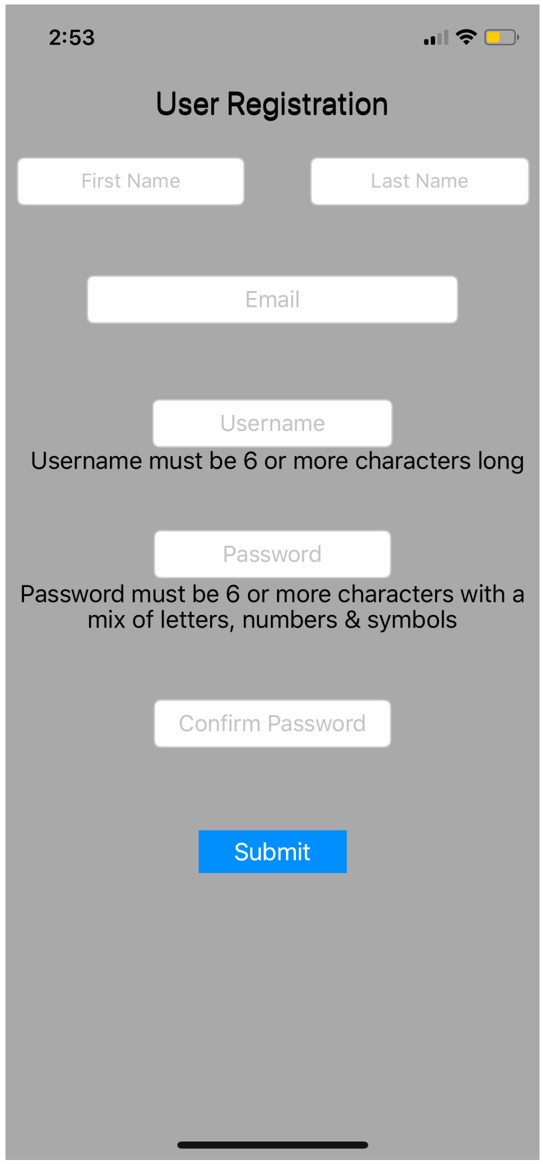

Figure 8.2: Registration Page

- User login page (figure 9.3) is similar to that of the sign up page.It allows the user to login with their credentials. It also has a forgot password option when the user forgets his/her password. The page contains a sign up option which would lead the user to the signup page.

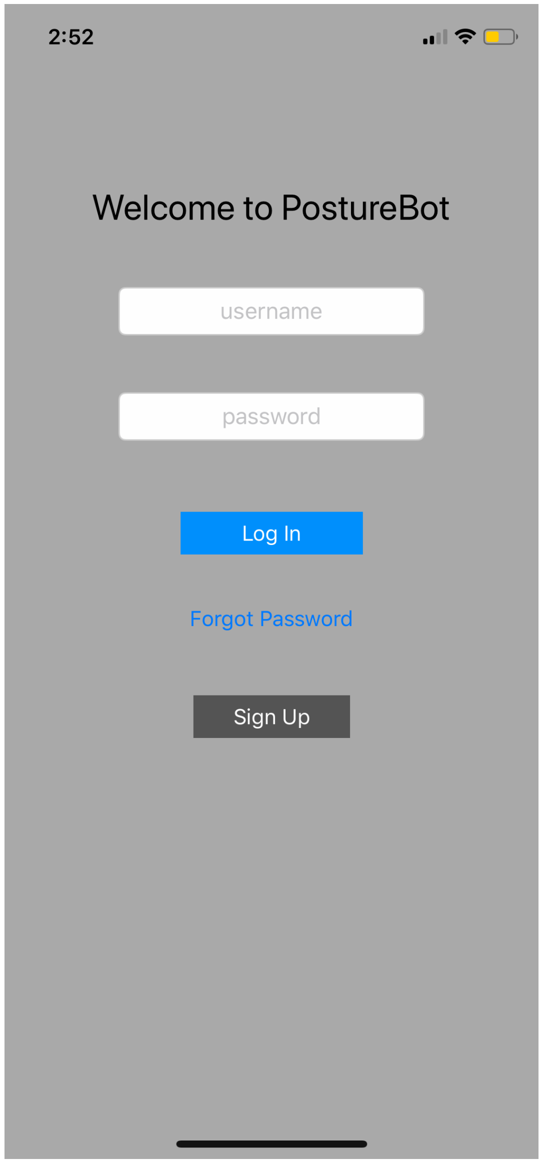

Figure 8.3: Login Page

- Bluetooth device paring page allows the user to look for and pair specific devices. When the hardware is on and not paired, the device id would show up on the paring page. The user can select the Bluetooth device from the list and start the paring process. It alerts the user when a device is successfully connected.

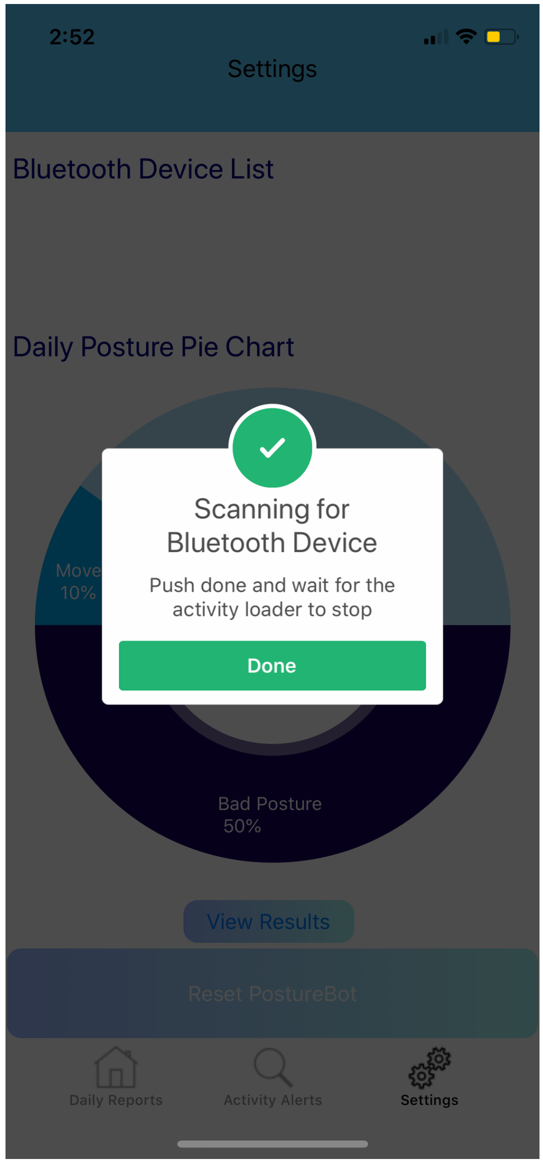

Figure 8.4: Device Paring

- The device records all the posture information transmitted by the hardware. It establishes a list of bad and good posture duration. It notifies the user in case of a bad posture via push notification. It also contains information of different activities alerts such as Stretch alert and, low battery alert.

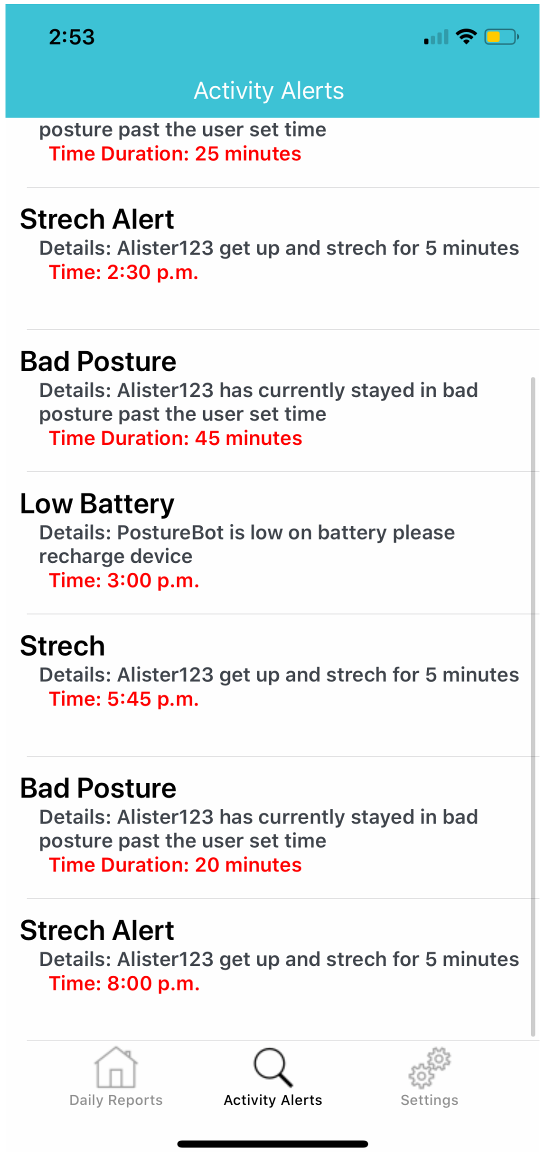

Figure 8.5: Activity Alert

- The daily report page provides the user with an assessment of the wearer's day using the device. It specifies how long the user has been in good posture, bad postures and how many times the user has stretched in one day.

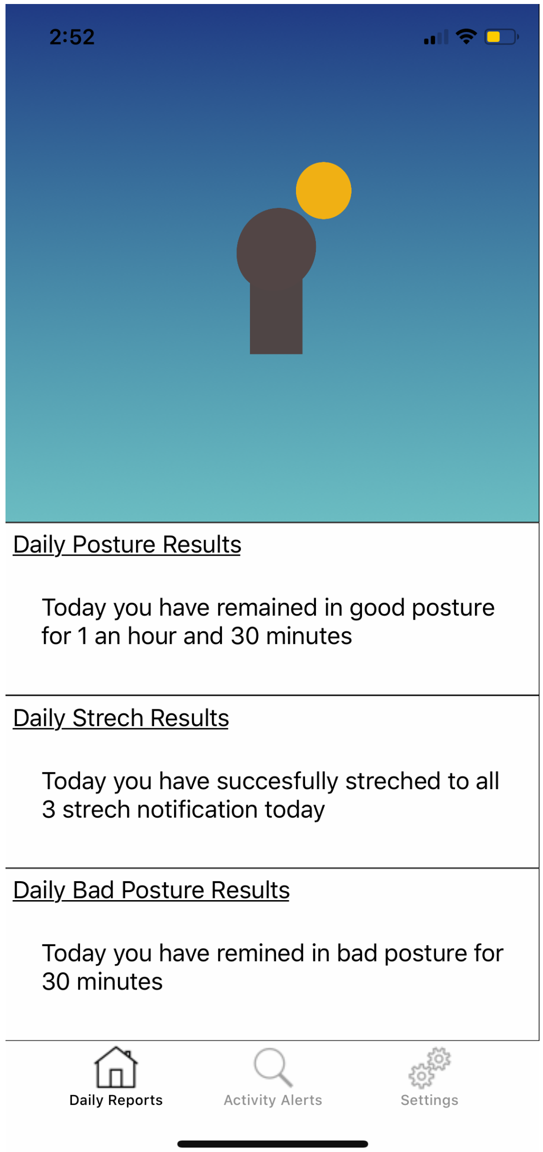

Figure 8.6: Daily Report

- The daily report can also been viewed in a pie chart form which shows a percentage view of how long the person has been in good or bad posture. It also has a reset PostureBot option for new measurement of preset good posture.

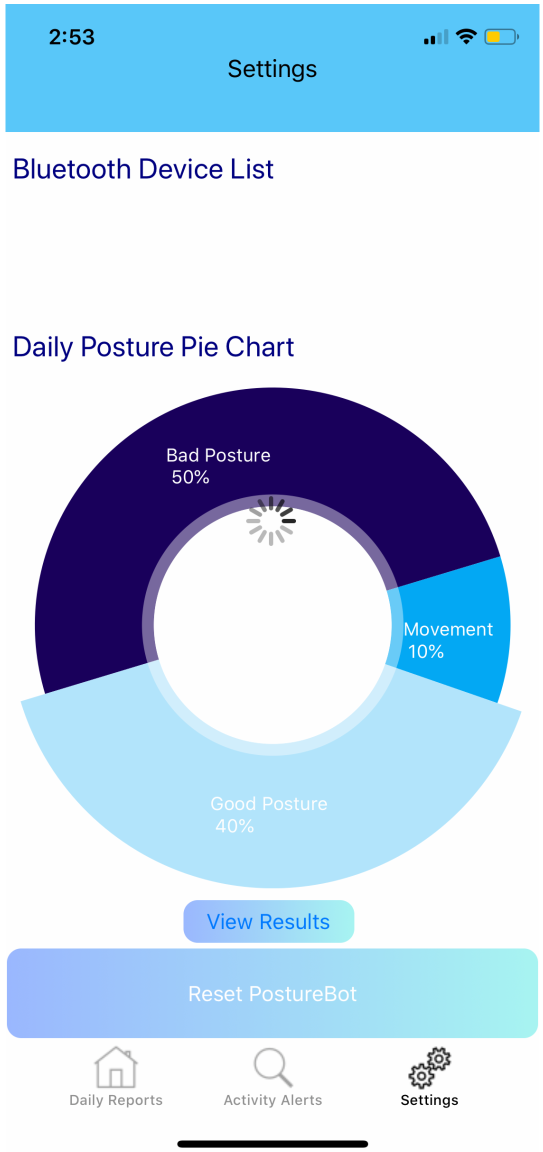

Figure 8.7: Daily Report Pie Chart

- Weekly report chart visually present a week's data of of the device. It uses a chart to show how long the person has been in good posture in the past week. It allows the user to compare his/her progress to different days of a week.

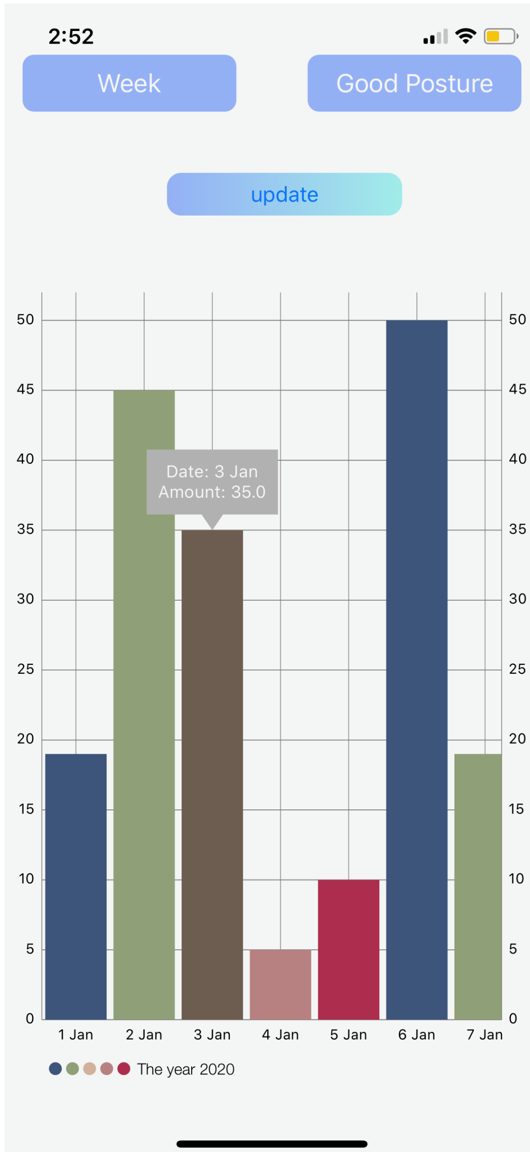

Figure 8.8: Weekly Report

Chapter 9

Testing

Due to difficulties associated with the COVID-19 pandemic, our testing was conducted separately.

9.1 Hardware Testing

We first tested the functionality of the sensors in our device. The two inertial measuring units(IMUs) are Accelerometer and Gyroscope. We tested the posture measuring feature by setting a fixed posture and transmit that data to the front-end application. On the front end we could see that the posture was corrected recorded when the user was wearing the device. The posture was then set to the correct posture. We then adjust the posture and observed in the console that any devication from the correct posture would result in an output of bad posture. The Bluetooth functionality were then tested to see if the device could transmit data to the iOS application. Lastly, We left the device running for a extend of time with its rechargeable battery to test its battery life. We found that the device is able to work up to 5 hours with constant use.

9.2 Software Testing

The iOS application testing is divided into the following components:

- The login screen was tested to determine whether it is the legitimate user
- The daily reports screen was tested to see whether the application is able to provide a report of the user's posture throughout the day.
- The settings Page was tested to see if the application can communication with the hardware and set up a correct posture.
- We tested pushing a notification to the user when he/she is observed to be in a bad posture for an extended of time.

9.3 Acceptance Testing

The last part of our testing ensured that our system fulfilled the requirements that we initially proposed. Here we looked at each requirement, checked our system, and concluded whether we were able to successfully and safely fulfill those requirements. Seeing that we have, we can then conclude that our system works as intended.

Chapter 10

Difficulties Encountered and Lessons Learned

10.1 Difficulties

10.1.1 COVID-19

- Due to the COVID-19 pandemic, our team was unable to meet in person. This had created difficulties for us to work together and test our design. Therefore, we had to work on our own components individually with no means to combine our parts.
- We lacked the access to certain facilities which results in an incomplete hardware component. Our initial design includes a hardware casing customized to our specific device. Due to the shelter in place, we were unable to access 3D printers for this component.
- This product still lacks certain features that would be helpful to the user. out envisioned vibration notification feature was unable to be fulfilled as we were unable to meet in person.

10.1.2 Testing

- Our design at the moment is suitable for people with lifestyles that are similar to ours. It is great for monitoring users who stand still or sit still for a long period of time but is uncertain whether it would work for those with a more active lifestyle. We lack outside testers and have trouble controlling the variables when we cannot conduct mass testing.
- Our system components were tested separately due to the pandemic. It is not clear whether our system would work flawlessly when put together.

10.2 Lessons Learned

- Power consumption of a small portable device is crucial and carefully calculated.

- We learned how to process data sent via a Bluetooth Link
- Remote working requires constant feedback and communications.
- Determining how much information the user wants to see and what to present to the user.

Chapter 11

Suggested Changes

1. A 3D printed case can be utilized for better fitting and protection of the device when worn.

2. Implement a vibration based notification system to alert user of bad posture. Adds new features to the iOS application.

3. Extended testing with outside audience with different lifestyles to make the project more universal.

4. Increase manufacturability of the device by making our own version of the Arduino Board. This would cut down our cost and customize our design to make it smaller and more portable.

Chapter 12

Ethical Considerations

12.1 Privacy

Our main ethical dilemma that comes with most software applications is user privacy. Every decision we made for the application had their privacy in mind. For someone to use our device, an account needs to be created that connects to our database. All we ask for is a username, email, and a password. Then, in order to customize their stimulation, we ask the user to set their idea body posture. This minimal amount of information is important in case of a security breach. In addition, users can be assured that their information is not being shared, since we are not asking for profitable information such as credit card information. There is no way the user can be identified by any of the information they voluntarily give. Another feature to give the user more privacy is an option to clear their account data. Account information is stored in a database, so when the user chooses the option to clear their account, all of their information will be wiped from our database. We use crypto hashing with each users password to ensure no viewing of user passwords in plain text in our database.

12.2 Equipment Safety

The other main ethical dilemma is the problem we face with putting an electrical device on a persons body. The problems that we face with a malfunction with one of the hardware components could unfortunately shock or burn the user. We conducted researches and testings to make sure that our device meets the safety standard and that it is unlikely to harm the wearer.

Bibliography

[1] ”The Hidden Impact of Musculoskeletal Disorders on Americans” American Chiropractic Association.

[2] Cherkin DC, Deyo RA, Loeser JD, Bush T, Waddell G. An international comparison of back surgery rates. Spine (Phila Pa 1976). 1994;19(11):1201-1206. doi:10.1097/00007632-199405310-00001

[3] Obembe AO, Johnson OE, Tanimowo TO, Onigbinde AT, Emechete AA. Musculoskeletal pain among undergraduate laptop users in a Nigerian University. J Back Musculoskelet Rehabil. 2013;26(4):389-395. doi:10.3233/BMR-130397

[4] Wei J. How wearables intersect with the cloud and the internet of things: considerations for the developers of wearables. IEEE Consumer Electronics Magazine. 2014:53–6.