



IN SHOE FORCE SENSOR

Sponsor: Dr. Wolfgang Bauer

ECE 480 CAPSTONE TEAM 2

Goksu Adanali

Alexander Chambery

Steven Criner

Mohamadou Diatta

Alexandra Mackenzie

Facilitator: Dr. Virginia Ayres



DEPARTMENT OF ECE
ELECTRICAL AND COMPUTER ENGINEERING

Table of Contents

Chapter 1: Introduction and Background

- I. Motivation
- II. Commercial Potential
- III. Summary of Benefits

Chapter 2: Solution Space and Selection of Approach

- I. Overview
- II. FAST
- III. House of Quality
- IV. Gantt Chart
- V. Budget

Chapter 3: In Shoe Weight Sensor: Device Design and Implementation

- I. Overview
- II. Sensor
- III. Power
- IV. Bluetooth Communication
- V. Microcontroller Interface
- VI. Application Development

Chapter 4: In Shoe Weight Sensor: Tests and Performance

- I. Overview
- II. Weight Response
 - a. Fixed Weights
 - b. Subject Weights
- III. Power
 - a. Operational Power Requirements
 - b. End of Lifetime Performance
 - c. Maximum Lifetime
- IV. User Friendliness Factors

Chapter 5: Design Issues

- I. Intellectual Property
- II. Product Liability
 - a. Electrical Shock Hazard
 - b. Portal to Phone Hacking Concerns
 - c. Psychological Impact

Chapter 6: Final Costs, Schedule, Summary and Conclusion

- I. Budget
 - a. Current Budget
 - b. Projected Production Budget
 - c. GANTT Chart Variation
- II. Summary and Conclusion

Appendices

- I. Appendix 1: Technical Roles, Responsibilities, and Work Accomplished
- II. Appendix 2: References
- III. Appendix 3: Technical Documents, Code, etc.

CHAPTER 1: INTRODUCTION AND BACKGROUND

MOTIVATION

People have begun to tap into technology in their efforts to become healthier,. With the rise and success of many devices such as FitBit and Nike ID, the future of this market looks bright. Exciting new devices are constantly being introduced. The reason for the success for many of these devices is that they allow each user to individually track their progress towards becoming a healthier, more active person, becoming, in effect, their “personal trainer”.

Here at Michigan State University, Dr. Wolfgang Bauer from the Department of Physics and Astronomy, takes a scientific interest in the features of these many health oriented devices. While researching their capabilities, he realized that one of the most important factors in determining someone's fitness, their weight, was missing from the current offerings. Dr. Bauer began to consider the possible solutions that would allow an individual to monitor their weight with on-demand convenience. As his ideas become a design, he sought collaboration with the College of Engineering by becoming the sponsor for development of the In-Shoe Weight Sensor project as a Senior Capstone Design project in the Department of Electrical and Computer Engineering.

Until now, the only viable option to determine weight is to step on a scale. While useful at the end or beginning of the day, this leaves a large gap when a person's weight fluctuations are unknown. Before and after exercise, in between meals, during common versus strenuous activities – these time points all produce data that adds up to fitness trends that are important to recognize. For this reason, Dr. Bauer determined that having an on-demand device that gives the user an instantaneous measure of their weight throughout the day would be the most effective tool in gaining insight as to how one's daily routine reflects upon weight and overall fitness.

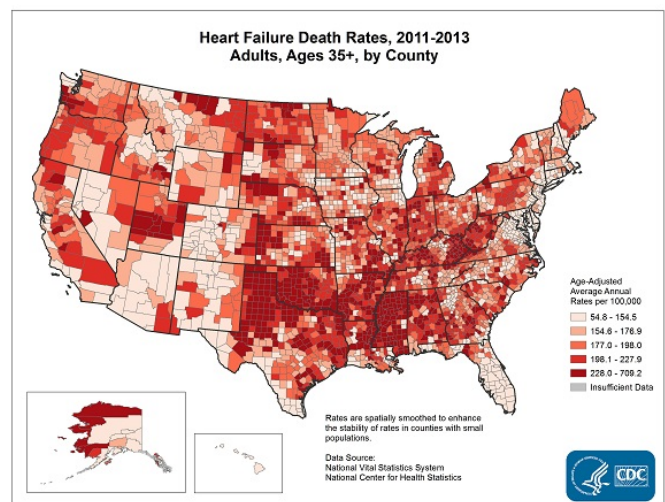
Capstone Team 2 joined the project and accepted the challenge. The key issue to resolve, and why no well-known device currently exists on the market, is that the sensing must be both physically small and electronically exact. The concept of an in-shoe force sensor was ultimately chosen by Team 2 as the best option. By installing the Team 2 In-Shoe Force Sensor in an ordinary pair of shoes, the user is able to view their weight throughout the day so long as they have their shoes on. In order to make the In-Shoe Force Sensor more user-friendly and commercially strategic, the In-Shoe Force Sensor is connects to a smartphone via a wireless connection and a specially Team 2 developed application (app). This enables users to navigate a simple application i to view the read out of their In-Shoe Force Sensor pair. Additionally, the application enables users to track their progress from day to day and gain the satisfaction of seeing progress made. By creating this novel device, Team 2 has successfully bridged the large gap that currently exists in health tracking devices.

Commercial Potential

One of the appealing aspects of this project is its commercial potential. According to an initial study, the mobile health market is expected to grow from \$5.1 billion in 2013 to \$41.8 billion in 2023. This remarkable growth is noticeable even today as devices such as Apple Watch, NikeiD and FitBit continue to develop hundreds of millions and even billions of dollars in revenue for their parent companies. The interactive nature of these devices offers motivation to increase activity and overall health, in addition to data. There are currently no devices that can effectively monitor weight, a noticeable hole in the market that could be filled by the successful design for this project.. While it is true that estimates of calories burned, steps taken and distance traveled can help you get an idea of how fit you are, tracking the weight of a user throughout their fitness journey is invaluable.

Throughout the project, our design team has conversed often with our sponsor regarding the potential of this device to become a commercial success and feel that it is viable. One of the keys to achieving this success involves patenting our device so that we may then eventually sell or license it to a company for commercial production and Team 2 has submitted Invention Disclosure Forms through MSU Technologies. The resources of the large company would allow a drop in cost of manufacturing, refinement of the design, and marketing of the product. Our initial design that was built for the ECE 480 Capstone can become commercially viable and a real contribution to the fitness landscape.

In addition to the personal fitness market, there are also many opportunities for this product to succeed in the medical field. In certain cases, such as with eating disorders and other illnesses/diseases, monitoring a patient's weight can be critical to ensuring that they are making improvements with their condition. One such condition is congestive heart failure, which affects nearly 5.1 million people in the United States alone and costs the nation an estimated \$32 billion annually. One of the symptoms of heart failure is a sudden increase in weight over a short period of time. After receiving treatment for this disease, patients are instructed to monitor their weight rigorously to help identify if they may be at risk again. Due to the failure of many patients to monitor their weight, a large majority of those who have already been treated for the disease will end up back in the hospital for further treatment. With the implementation of this device, users could simply give access to their doctors so that they may monitor their weight. By doing so, users who would normally be oblivious to the symptoms could be alerted in time for preventative measures to be taken. A further refinement can be the development of a specific application that analyzes the weight trends of a user to determine when a doctor should be alerted. This method would save a lot of time for the doctor by only alerting him/her to the weight logs of individuals with a high risk of further complications.



SUMMARY OF BENEFITS

With the development of the In-Shoe Force Sensor, Team 2 has enabled a user to monitor a defining characteristic of their health: their weight. Since no current device on the market presents a solution to the on-demand weight measurement problem, the combination of this new capability with available health tracking devices would allow unprecedented insight into the fitness progress being made by individuals.

A ready market for the device exists in fitness-savvy people considered to be of 'average' health. In addition to this market the In-Shoe Force Sensor could prove invaluable to those battling extremities of weight management. Obese users would be able to reliably track their progress towards becoming a lower weight by determining which meals offered a lower impact on their weight overall. For example, a large meal of healthier foods such as broccoli and carrots may initially depict a large rise in weight but if followed by a recognizable pattern of non-weight retention over a relatively short time period, the positive cause-and-effect becomes clear and motivating to the user. Similarly, users with low weight complications such as anorexia could use this device to motivate them to eat more. In all cases, the progress of users towards their optimum weight would mark not only an increase in physical well-being, but also in mood and overall quality of life.

Four key technical areas were identified by Team 2 as critical to a successful design. These are: Sensors, Power, Communication and Software. The technical area goals were critically supported by Planning, Milestones and Tests. The next Chapters detail the planning, investigations and tests that resulted in the creation of our successful prototype and the achievement of our Project goals

Chapter 2: Solution Space and Selection of Approach

Effective planning, with clear identification of specific subsystem goals and milestones, was critical to project success. Analytical tools including Functional Analysis System Technique, House of Quality Selection Matrix development and Gantt Chart timeline/milestone development were all used to provide quantitative planning metrics. The Budget for prototype development documents that the In Shoe Force Sensor project was implemented by Team 2 on time and within budget.

FAST

Our first step in the design process consisted in getting a clear understanding of the problem statement and knowing exactly what our design would have to accomplish. We developed a Functional Analysis System Technique (FAST) Diagram in order to define our product's functionality and overall scope.

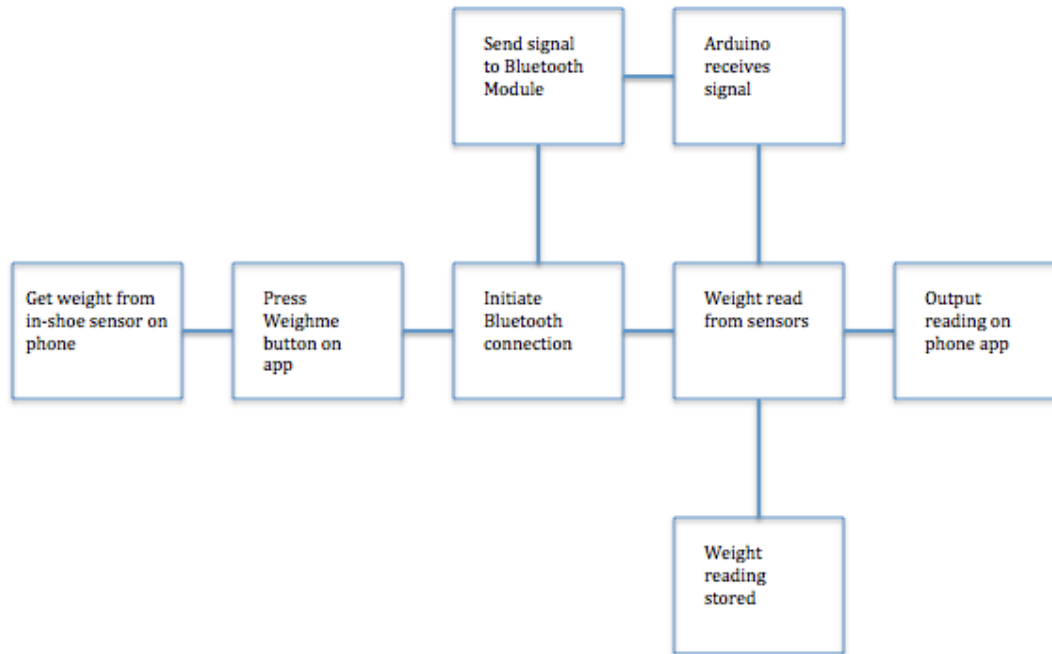


Figure 1. Team 2 FAST Diagram

The overall objective of our design is to measure the weight information of the person wearing the device and display that information in interactive app format. The user mainly interacts with the device through the smartphone app, which in turn communicates with the weight sensors via Bluetooth. The process begins when the user opens the app. Upon pressing the **WeighME** button on the app, a command is then sent to an Arduino microcontroller that triggers it to initiate communication with the attached Bluetooth module. Once the phone and Arduino are paired, the latter then records the readings from the sensors and relays the information back to the phone via Bluetooth. This information is then displayed to the user as weight information.

HOUSE OF QUALITY

A House of Quality connects project criteria with customer requirements. For this project our main customer was our sponsor, Wolfgang Bauer. There were Critical Customer Requirements for the project that he laid out. One original customer design specification was that the weight readout would be accurate to within 10 grams. Team 2 researches determined that, while feasible, the cost of the sensors needed to meet this requirement would exceed the ECE480 budget and could also generate price-point concerns. Our sponsor agreed to a more flexible accuracy to within 90 grams, which our prototype meets. With accuracy comes the issue of battery life.: in order to keep accuracy high, the battery life needs to be sufficient. Therefore, battery life is also a critical customer requirement. Continuing the analysis, the next important requirement was that the device could connect to a smartphone for the weight readouts. The last major customer requirement was that the production version of this project needs to cost less than \$100. The House of Quality selection matrix informed Team 2's brainstorming and resulted in the conclusion that the best way to accomplish the goal was through a Bluetooth connection.

GANTT CHART

A Gantt Chart is an important analysis tool used to set key project milestones within an achievable timeline. Team 2 iterated the Chart given below until self-consistency was reached. The projected Gantt Chart timeline and the actual project sub-goal achievements were close.

Project Definition Tasks	23 days	Fri 1/15/16	Tue 2/16/16	
Team Introduction	1 day	Fri 1/15/16	Fri 1/15/16	Team
Meet with Facilitator	1 day	Thu 1/28/16	Thu 1/28/16	Team
Meet with Sponsor	1 day	Mon 2/1/16	Mon 2/1/16	Team
Brainstorm early ideas/Develop concepts of first draft for design/application	3 days	Mon 2/1/16	Wed 2/3/16	Team
Definition of Milestones	1 day	Mon 2/1/16	Mon 2/1/16	Team
Draft of Design/Application Proposal	5 days	Wed 2/3/16	Tue 2/9/16	Team
Facilitator feedback meeting	1 day	Tue 2/9/16	Tue 2/9/16	Team
Draft of Oral Design/Application Proposal Presentation	6 days	Mon 2/8/16	Sat 2/13/16	Team
Oral Presentation Rehearsals	3 days	Sat 2/13/16	Tue 2/16/16	Team
Research Prototype Design	10 days	Mon 2/1/16	Fri 2/12/16	
Kinesiology department discussions	5 days	Mon 2/1/16	Fri 2/5/16	Alex Mack
Prior art search	5 days	Mon 2/1/16	Fri 2/5/16	Mohamadou
Research sensor options	10 days	Mon 2/1/16	Fri 2/12/16	Alex Cham
Research power options	5 days	Mon 2/1/16	Fri 2/5/16	Steven
Research hardware-to-software communicatio	10 days	Mon 2/1/16	Fri 2/12/16	Goksu
Craft Selection Matrix(House of Quality)	6 days	Mon 2/1/16	Mon 2/8/16	Team

⚡ Design and Prototyping	52 days	Fri 2/19/16	Mon 5/2/16	
Order Parts and Begin Prototype/software build	6 days	Fri 2/19/16	Fri 2/26/16	Alex Mack
Improve design based on initial results and findings of prototype/software build	11 days	Fri 3/4/16	Fri 3/18/16	Team
Present Demo to sponsors/faculty	1 day	Mon 4/4/16	Mon 4/4/16	Team
Refine EGR Notebook before Photocopy submission	7 days	Mon 4/11/16	Tue 4/19/16	Steven
Prepare Final Report and Poster for Design Day	12 days	Mon 4/11/16	Tue 4/26/16	Team, Alex Cham, Steven
Demo 2 to sponsors/faculty	5 days	Mon 4/18/16	Fri 4/22/16	Team
Submit Final Documentation and return equipment	2 days	Fri 4/29/16	Mon 5/2/16	Alex Mack
⚡ Hardware Design	9 days	Fri 2/19/16	Wed 3/2/16	
Order Pressure Sensors	1 day	Fri 2/19/16	Fri 2/19/16	Alex Mack
Install/Test Sensors	4 days	Wed 2/24/16	Mon 2/29/16	Alex Cham, Alex
Order Batteries	1 day	Fri 2/19/16	Fri 2/19/16	Alex Mack
Install/Test Batteries and Sensors	6 days	Wed 2/24/16	Wed 3/2/16	Alex Cham, Steven
⚡ Software Design	9 days	Fri 2/19/16	Wed 3/2/16	
Smartphone GUI(Android)	8 days	Fri 2/19/16	Tue 3/1/16	Mo and Su
⚡ Interface Design	10 days	Mon 3/14/16	Fri 3/25/16	
Sensor readout to Bluetooth transmitter	4 days	Mon 3/14/16	Thu 3/17/16	Mo and Su
Bluetooth transmitter to smartphone receiver	5 days	Mon 3/14/16	Fri 3/18/16	Mo and Su
Sensor Readout and Data Collection Test	10 days	Mon 3/14/16	Fri 3/25/16	Mo and Su
⚡ Evaluation of Prototype	16 days	Fri 3/18/16	Fri 4/8/16	
Test for force balance	3 days	Mon 3/21/16	Wed 3/23/16	Team
Test for power consumption	3 days	Tue 3/22/16	Thu 3/24/16	Team
Test for component Robustness	2 days	Wed 3/23/16	Thu 3/24/16	Team
Test for Safety hazards	4 days	Thu 3/24/16	Tue 3/29/16	Team
Test for connectivity	4 days	Sat 3/26/16	Wed 3/30/16	Team
Test for Interference with Connectivity	4 days	Fri 3/25/16	Wed 3/30/16	Team
Test for User Friendliness	5 days	Fri 4/1/16	Thu 4/7/16	Team

BUDGET

Budget considerations are always important and doubly so in a project with commercial potential, where price-point is a consideration. The In Shoe Force Sensor prototype was developed for a total cost of XXX, as per the Budget Statement below.

Chapter 3: Device Design and Implementation

Four key areas were identified as critical to ensure a successful design that would accomplish the design goals, as requested by the sponsor. These are: Sensors, Power, Communication, and Software. To create the In-Shoe Force Sensor, research in each engineering subsystem was required to ultimately specify the approach that would satisfy our sponsor's needs as well as fit within our budget. During the prototyping phase, many different approaches were considered for each subsystem before a final design choice was made. Influencing these design choices were factors previously discussed such as cost, device footprint, and customer requirements. Although each subsystem was developed independently, the cohesiveness of the final system integration was essential for this project. For example, if the sensors lacked the ability to communicate with the software application or if the microcontroller failed to convert the voltage output of our sensors to a correct weight, the In Shoe Force Sensor device would fail completely. To combat this, many steps were taken to ensure that, after a subsystem operated correctly as an independent unit, it continued to function when integrated into the device.

Sensor Concepts

Sensors are a core component in devices that are used for weight measurements. They are needed to detect the amount of direct force that a subject exerts in a downward direction, which can be translated into weight. In this project it is desirable for the weight reading to be very accurate so that any slight change in behavior can be tracked through weight change. The final design is accurate within $\pm .2$ lbs, which is the same as most higher-end bathroom scales.

For the first prototype design, a lot of interest was placed in pressure sensors. These are the sensors that are typically used in medical applications to determine improper balance, proper running form, etc. However, after contacting several companies with product offerings, it became apparent that pressure pads and their required software were much more expensive than our budget would allow. In addition, pressure pad sensor accuracy was only $\pm 10\%$. This low accuracy would result in an overall error of 20lbs on a 200lb user and consequently would not meet our sponsor's requirements. The second prototype design was based on the Digi-key FC22 Compression load cell sensor. A load cell is a sensor that converts a force on it into an electronic signal. These sensors output a resistance. They work on the principle of piezo-resistivity and when a force is applied to the sensor, the resistance in the sensor changes. The change in resistance changes the output voltage when an input voltage is applied. The Digi-key sensors are reliable and cost effective; however they are accurate to only $\pm 1.0\%$, yielding a 2 lb error on a 200 lb user. Because load cell sensors were still not accurate enough, they were not a suitable choice for the final design.

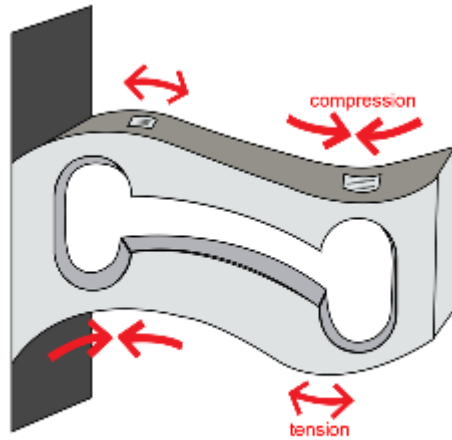


Figure 1. Strain gauge example

Bathroom scales with the desired accuracy are currently available. These higher-end scales also utilize load cells that function as described above. However, the load cell resistors also utilize a Wheatstone bridge to enhance their accuracy. In a Wheatstone bridge configuration there are four separate resistors connected as shown in what is called a Wheatstone network. An excitation voltage is applied to two of the corners and the voltage difference is measured between the other two corners. When there is equilibrium and no applied load the voltage output is zero. When the metallic member to which strain gauges are attached is stressed by applying a force, it leads to a change in resistance in one or more of the resistors. The resistance change also changes the output voltage and this small change in voltage can be measured and digitized. A load cell sensor with a Wheatstone bridge configuration for accuracy enhancement was therefore implemented in the design.

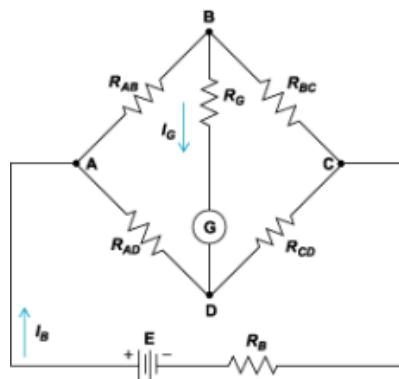


Figure 2. Wheatstone bridge circuit diagram

For the final implementation of the In Shoe Weight Sensor, the decision was made to use the exact same kind of load cells sensors as are used in a popular bathroom scale. This ensured the same accuracy as on a scale, enabling comparison tests. The difficulty with this sensor choice was that they could not be directly connected into a Wheatstone bridge configuration. This problem was resolved Team 2 by purchasing a combinator. The combinator is a printed circuit board (PCB) that combines the 12 wires found in bathroom scale sensors into the standard 4-wire Wheatstone bridge configuration. Team 2 then hooked the combinator up to a load cell amplifier to amplify the read out from the Wheatstone bridge.

Four sensors are attached to the bottom of each shoe sole template. Two are located on the balls of each shoe sole and two on each heel. In total there are eight sensors in one pair of shoe soles. A schematic of the shoe prototype and sensor placement can be found below in Figure 3. Each set of sensors has its own combinator, amplifier, and microcontroller. When the app is prompted, the weight readout is taken from each shoe, added together, and averaged.

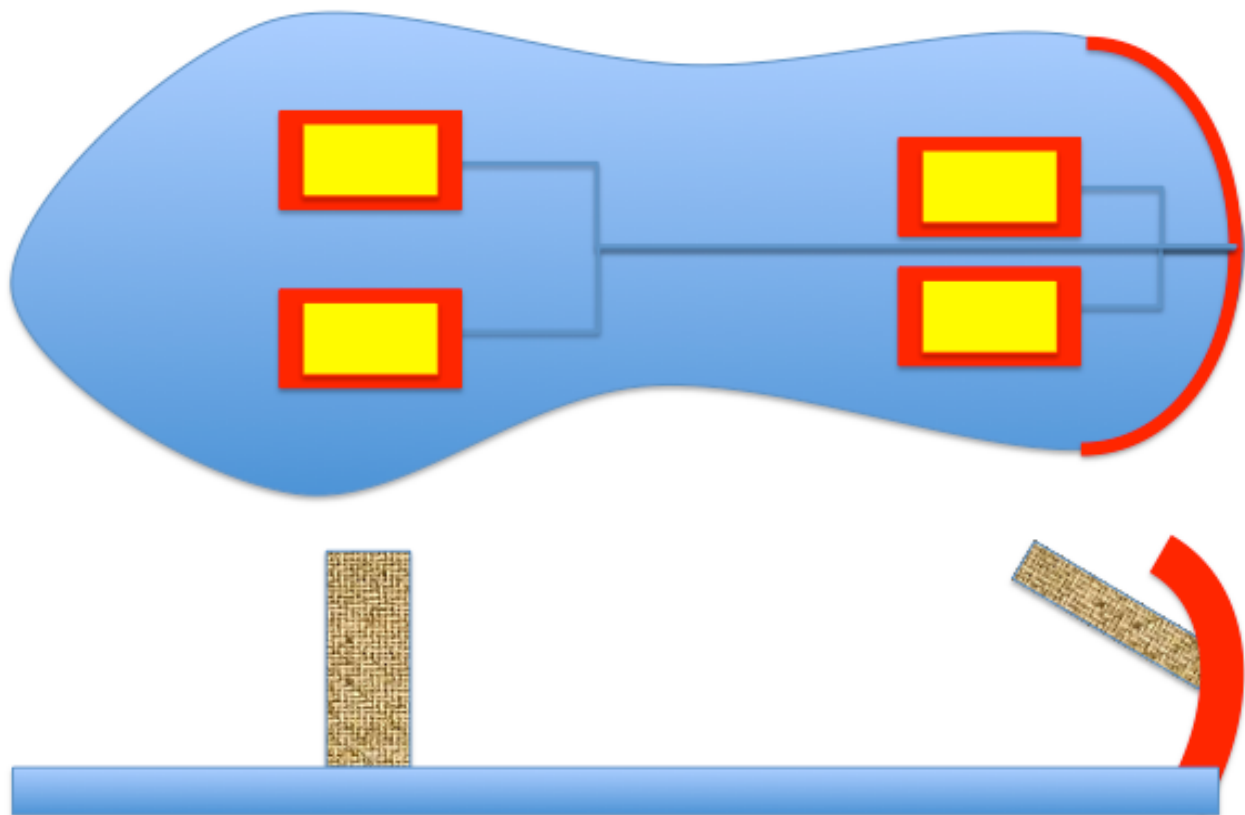


Figure 3. Schematic of shoe prototype viewed from the bottom (above) and side (below)

Power Concepts

An important performance specification of the In-Shoe Force Sensor is its power consumption and battery lifetime. For convenience and practicality, the battery should be able to provide reliable weight readings over the course of a day. In order to ensure this design criterion was accomplished in the final product, it was necessary to first identify how many times the average product user would request their weight in a normal day of operation, then determine if the power source selected could provide the determined power demand. Performance factors that were considered are power draw profile, maximum lifetime, end of lifetime sensor performance, and end of lifetime communication performance.

As discussed in Chapter 3, the selected power supply for the In-Shoe Force-Sensing device is a 9V Duracell battery. This selection was made based on the current rating of the battery, packaging purposes, and compatibility with the Arduino Uno microcontroller. The battery is connected to the Arduino which has an on-board 5V pin that acted as the voltage supplied to the Bluetooth transceiver and the load cell circuit. With this hardware configuration, the power consumption of the device can be determined by the number of pins in use on the microcontroller.

The Bluetooth transceiver required the transmit and receive pins and the 5V power supply pin of the Arduino. The load cell amplifier and Wheatstone bridge required two pins for communication with the Arduino in addition to the 5V power supply pin. When using this pin configuration, the microcontroller will draw approximately 210mA when the pins are active. Given the 550mAh rating of the microcontroller, it is calculated that there is 2.61 hours of available use of the In-Shoe Force Sensor from one battery. Although this may at first seem like a design flaw, it must be noted that the application of this device does not entail constant power usage throughout a standard 16 hour day. The device takes only a brief moment to collect data from the strain gauges and compute a weight reading for transmission to the smartphone. This translates to only brief moments of brief peak power draw, allowing the user to take frequent weight readings without depleting the battery life. This translates in to several days of usage before battery replacement is necessary. In a commercial application, a battery that is rechargeable would be applied to the system. The possibility of harvesting the kinetic energy of the user for battery charging without need of an outlet can also be investigated.

Selecting a method of powering the in-shoe electronics was dictated by the following criteria: safety, mobility, and longevity. First and foremost it was important that the user would be safe in a worst-case scenario when wearing the device. As none of the components required high operational voltages, a low voltage power supply could be selected to minimize shock hazard. In the design of this product, several power related safety hazards were considered. The In Shoe Force Sensor prototype is an external shoe accessory. For this reason, the hazard considerations were focused on external conditions to which one's shoe may be subjected. Footwear of all types are exposed to a variety of environments, including wet or conductive surfaces. Safety considerations were implemented when choosing a power supply for the prototype. To minimize shock hazard, the power supply was kept low, at a value of 9V. Furthermore, electrical hardware components were selected such that no component demanded more than 5V of electricity when in use. Additionally, the high resistance of the Wheatstone bridge circuit minimizes current, adding to the safety of the device.

The In Shoe Force Sensor device is wireless for the convenience of the user and therefore requires a mobile power supply. The device circuitry of the device is centered around an Arduino Uno microcontroller, to which all other components are peripheral. The Uno is built to supply power to all other circuitry components. The key adaptation required from Team 2

was to make the microcontroller compatible with with a battery instead of a wired power connection. A battery interface to the Arduino exists via the power jack located on the microcontroller board. Given the battery compatibility and 9V voltage requirement, it was intuitive to implement a standard 9V alkaline battery serve as the power supply of the hardware. Its nominal weight of 1.6 oz and size of 1.4 cubic inches makes it unnoticeable to the user. This battery selection met the design criteria and requirements of the project. For commercial production, an even smaller flat cell battery internalized within the hardware of a shoe sole is certainly feasible

Finally, the power supply needs to provide sufficient usage for the convenience of the user. Considering real-world applications of the device, Team 2 decided that it was essential that the battery would provide at least a day's worth of necessary power. A day was first approximated as 16 hours of activity for the average user. Estimations of the battery lifespan can be made with the help of the Arduino Uno datasheet, which states each I/O pin in use demands 40mA of current and the 5V supply demands 50mA. For the weight sensing circuitry and Bluetooth transceiver, four pins and the 5V supply are needed. With the above information Team 2 determined that the stated configuration will demand **210mA** of current assuming constant usage of the device. Given the 550mAh rating of the selected Duracell 9V Duralock battery, it is expected that the power supply would last through **2.61 hours**, and not 16 hours, of continuous use. However, continuous usage for 16 hours is not expected either. Under realistic use conditions, the user of the in-shoe weighing device would request their weight intermittently throughout the day. These brief, sporadic uses of the product would not demand significant amounts of power usage. With this in mind, it was deemed that the practical battery longevity would be considerably longer than 2.61 hours, and that a more realistic estimate would be approximately **a week's** worth of normal use of the device

Considering all of the desired design criteria and requirements stated above, it was determined that an appropriate power supply for the in-shoe weight sensing hardware was a 9V Duracell Duralock battery. Its safe low voltage supply would be a safe option for the user, its portability and small packaging offers high mobility, and its potency offers the user convenience and spares the need for daily recharging or battery changes. Plug-in rechargeable batteries, or better yet, a system that maintains charge by the user's kinetic energy would be a next logical refinement.

Bluetooth Communication

The smartphone application needs to be able to communicate with the sensors installed in the shoe soles and vice versa. As noted in the high level description of our FAST diagram, the phone to sensor communication will be handled via Bluetooth.

The choice of implementing this communication using the Bluetooth standard was motivated by the facts that, compared to other standards of wireless communication like Wi-Fi, a Bluetooth connection uses much less power and due to its lower distance range, Bluetooth communication reduces the risks of data transmission errors.

HC-06 Module

For the prototype, Team 2 uses the HC-06 Bluetooth Module to handle the Bluetooth connection between the sensors and the Android application. The HC06 is a high-performance wireless transceiver system that is affordable and has very low power consumption.

The module is equipped with a Transmit (Tx) and Receive (Rx) pin, which allows it to be able to send and receive data. The module will be connected to an Arduino microcontroller that will act as an interface to handle the sensor-to-Bluetooth module communication but also the Android application-to-Bluetooth module communication.



Figure 4. The HC-06 Bluetooth Module

Microcontroller

The microcontroller selected by Team 2 for this project is the Arduino Uno. The reason for this choice is that it allows us to quickly implement and troubleshoot our design. The Arduino was programmed using the Arduino integrated development environment (IDE). The Arduino also gives us access to a wide range of open source projects and codes that we can leverage in order to accomplish our own project goals.

For the implementation of this project, we used **sparkfun** open source library code from the Github Repository that allowed us to get the correct and accurate weight reading from the force sensors used for our measurements. This open source library enabled us to interact directly with the HX711 chip used to convert the voltage readings from the sensors into a weight value.

In our design, the Arduino is used mainly as an interface for the sensors and the Bluetooth module. The sensor data is collected by the Arduino, which then, upon receiving a signal sent to it via the Bluetooth module, sends that data to the phone application via the Bluetooth Module.

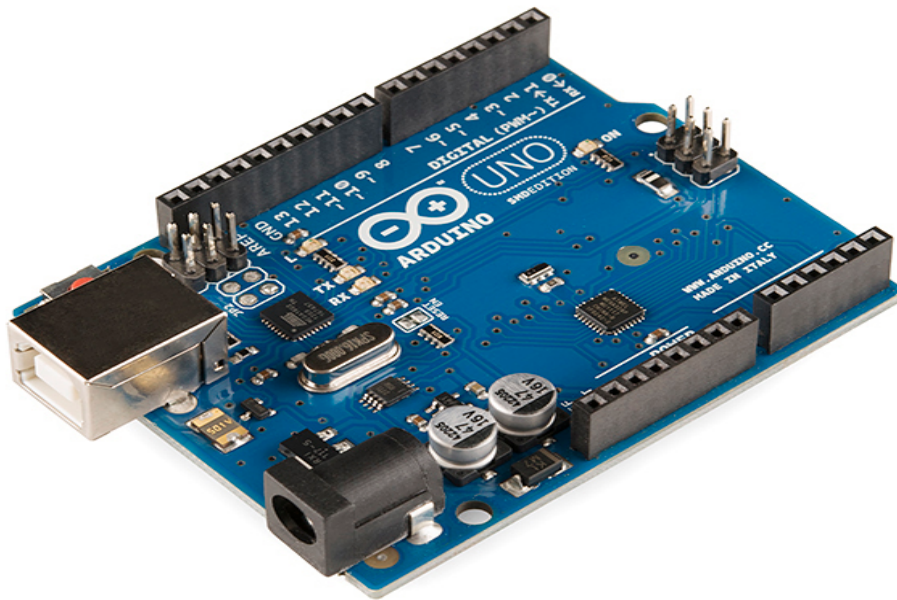


Figure 5. Arduino Uno microcontroller

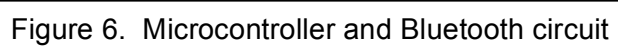


Figure 6. Microcontroller and Bluetooth circuit

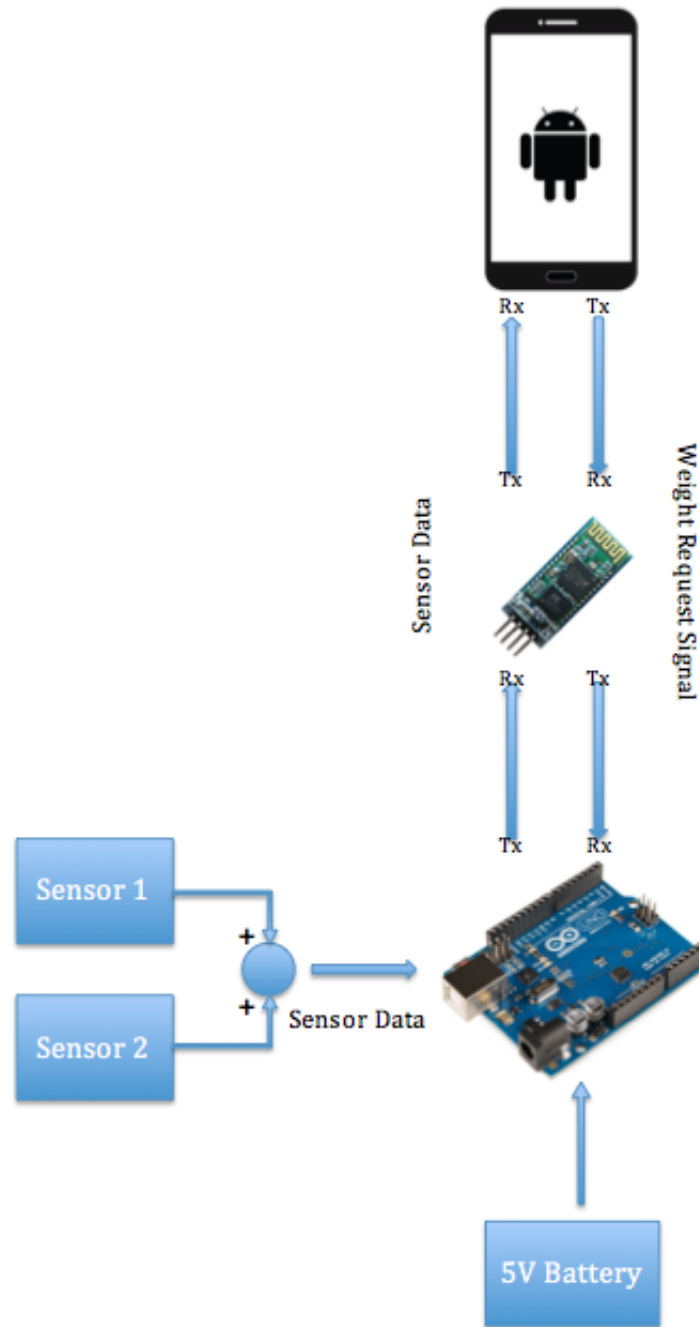


Figure 7. Initial high-level conceptual design

The diagram for our high level conceptual design works as follows:

- User presses button on app to request their weight
- A signal is then sent through the Bluetooth module to the Arduino microcontroller
- The Arduino reads the sensor values
- The Arduino then transmits sensor data to the Bluetooth Module
- The Bluetooth Module relays data to the phone

Application Development Design

One of the most important subsystems of the In Shoe Weight Sensor project involved developing a mobile application that is suitable for an Android or iOS platform. The specs for this portion of the project were to design an application where users are to be able to request a weight reading from the hardware which will then be interpreted by the Arduino and the collected information is presented to the user in a simple but efficient manner.

The software specialists had initial tasks to accomplish first before building the application could start. The first step was design of the architecture of the app that would identify the required features and workflow. The software specialist focused on not only building the app itself within the limits of the requirements but also structuring it in such a way so that the application is flexible enough for possible future improvements after the semester's prototype phase.

After careful considerations, the developers agreed on a simple single-page application where an interface with a button to prompt for a click would result on a weight reading from the shoe, and an environment where the collected information can be presented. Once the features, scope, structure and architecture of the application was established the next step was to decide on the optimal development environment which means deciding on the coding language, development platform, possibly compatibility opportunities and additional frameworks that would enable the application to have broader functionalities.

At this point, literature research and possible development tools were researched. Initially opportunities seemed limited to use of Xcode for Apple OS development or Android Studio with Java for Android development. Picking one of these platforms would limit the cross platform opportunities, meaning should the future developers on this application would desire to have a cross platform functionality, they would have to allocate more resources, time and would have to invest more funds to develop for additional platform. Fortunately, a new development platform was discovered at this stage of the project. Adobe has developed an intermediate environment where developers can write web applications which with certain platform configurations can easily be deployed in both Android and OS environments with a few adjustments to the project. Due to the lightness of the web applications, the desire to have a very fast and efficient codebase and having both of the developers coming in with strong web development backgrounds in the end, the developers settled on developing with Adobe's PhoneGap development tool.

Another important aspect and one of the critical components of the project that needed to be considered before the actual development process was whether or not PhoneGap would enable wireless communication and if so, of what sort(s). As the core of the project is the communication between hardware and software, this was a major deciding point. Luckily, it was discovered that multiple Bluetooth plugins for PhoneGap were documented in recent web resources, that would enable the application to communicate with the hardware implemented in the shoe. Therefore the final design included a single page web application that would be implemented in the native development environment with the aid of Adobe's PhoneGap.

Once the desired features and the scope of the communication subsystem were established, a few mockups of the imagined user interface were presented to the sponsor to see whether or not the developers' plans were fit to the spec of the project. The mockups also included possible future improvements such as a dashboard that will collect the data read by the shoe and delivered to the app, store this information in a database and present the user with progress information when desired, or some type of means of communication where the users

can contact the developers with feedback. The mockups were very well received by the sponsor and the developments continued with the subsystem.

The mockups also took into account usability, ease of access and user friendliness. In development of mobile apps most app success is achieved by correctness in addition to the user interface. Therefore a very minimal simple design, that used white-gray tones with an accent color of purple was implemented. Another important aspect that was taken into account when designing the application is ensuring the least amount of clicks possible. Research has proven that users' tend to abandon an app if it proves to be too much of an inconvenience to get started and use the application's main functionality. Therefore all of the pages are implemented in a way that most of the functionality is up to two clicks, and the user's do not have to input any personal information. Even the weight unit is set to pounds at initial start-up (since it is an application developed in United States), although it is made configurable in the settings window.

Development

The initial goal was to establish a simple interface that would enable users to request and observe a weight reading. To accomplish this a simple window with the text "weighME!" on it was presented to the user. The primary goal was to ensure the maximum accuracy and precision of weight readings, and therefore the main concern in the development process became to ensure the application can communicate back and forth with the Arduino correctly at all times.

Throughout the process of development the core principles of development, Correctness, Maintainability, Rapid Development and Speed of Execution were always a primary concern. Correctness was established by ensuring the application can connect to the Arduino every time the users allowed permissions. The maintainability was achieved by modularizing the app and compartmentalizing most functionality for further development and reusability purposes. Rapid Development was ensured by eliminating all the unnecessary components of the application and preventing "scope creeping" which means deciding to implement additional features mid development process. Even though some ideas came up throughout the development process, any new development work that was outside of the scope was not prioritized due to the time constraints and desire to establish the precise weight readings above all. Speed of Execution was achieved by choosing to implement a single page web application that can be interpreted in native environments.

In the functioning prototype a welcoming HTML page with a weighMe button is presented to the users. The functionality was as followed, when the button is clicked an event handler implemented in a linked Javascript file was evoked. This evoked event handler then called the Bluetooth connection function within the BluetoothSerial plugin provided by open source developers. Then the BluetoothSerial module communicated to Arduino and a callback which is a function that is executed after a function call that awaits a response, either received a read data or an error message. The application then presented this read data or error to the users as their weight reading. At the bottom of the page, a navigation menu that includes links to the dashboard page, the settings page, and a contact us page were included for users.

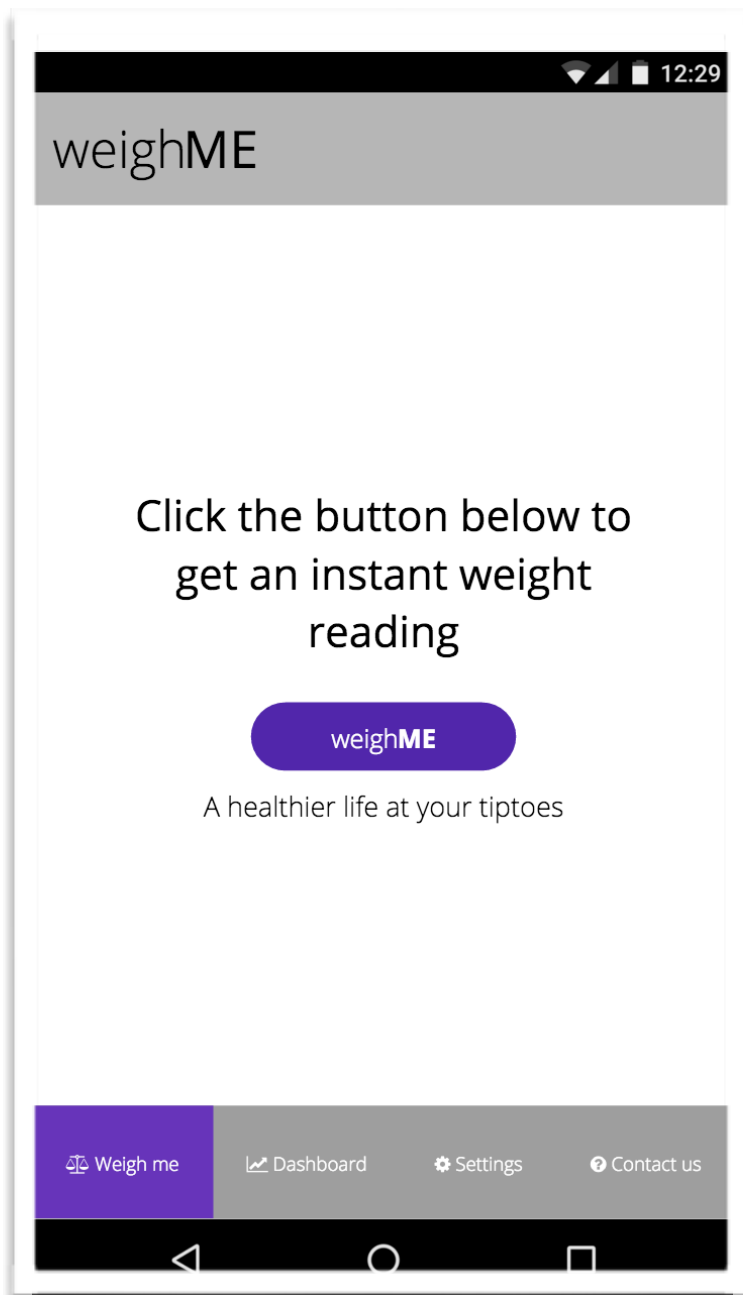


Figure 7. The welcome page of the application

The second link on the application is the dashboard window, which is included as a part of the envisioned future of the app. This dashboard feature is implemented with the hopes of in the future development of the app users will be presented data from a database that will contain collected weight information throughout a certain period of time, and can visualize their progress.

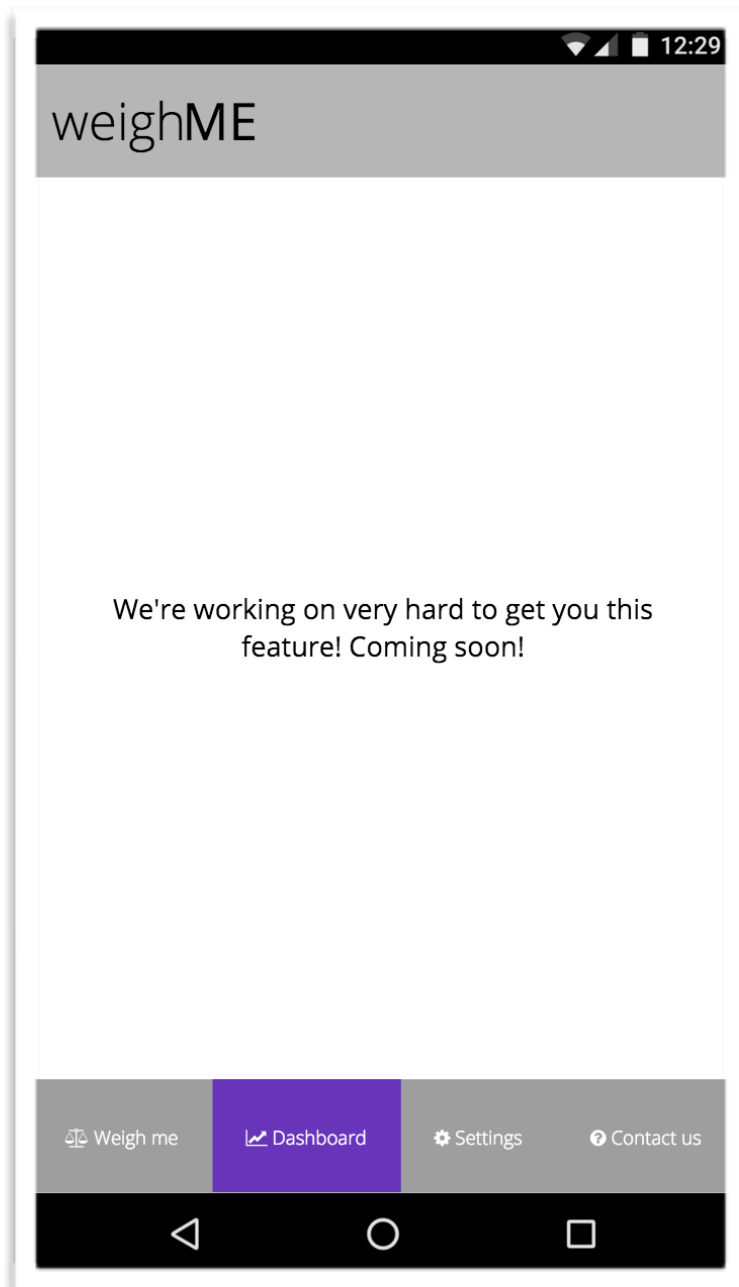


Figure 8. Dashboard view

The next window is the settings window, which enables the users to change the weight units they receive the application in. This window enabled the application to be user friendly and improved its usability stats. By allowing the users to have an option to change and customize their interaction with the application, this feature opened a wide window for improvements. The way the application keeps track of the weight unit is using JavaScripts' built in localStorage window. Proceeding past the prototype phase, this feature can be cached should the application become a cache based application. For future purposes, this window is built in a way that it can implement many other configurable settings. Some of these future developments can be exemplified such as keeping track of user's name, age, desired weight, water in take, daily calorie intake, user's email for news and updates etc...

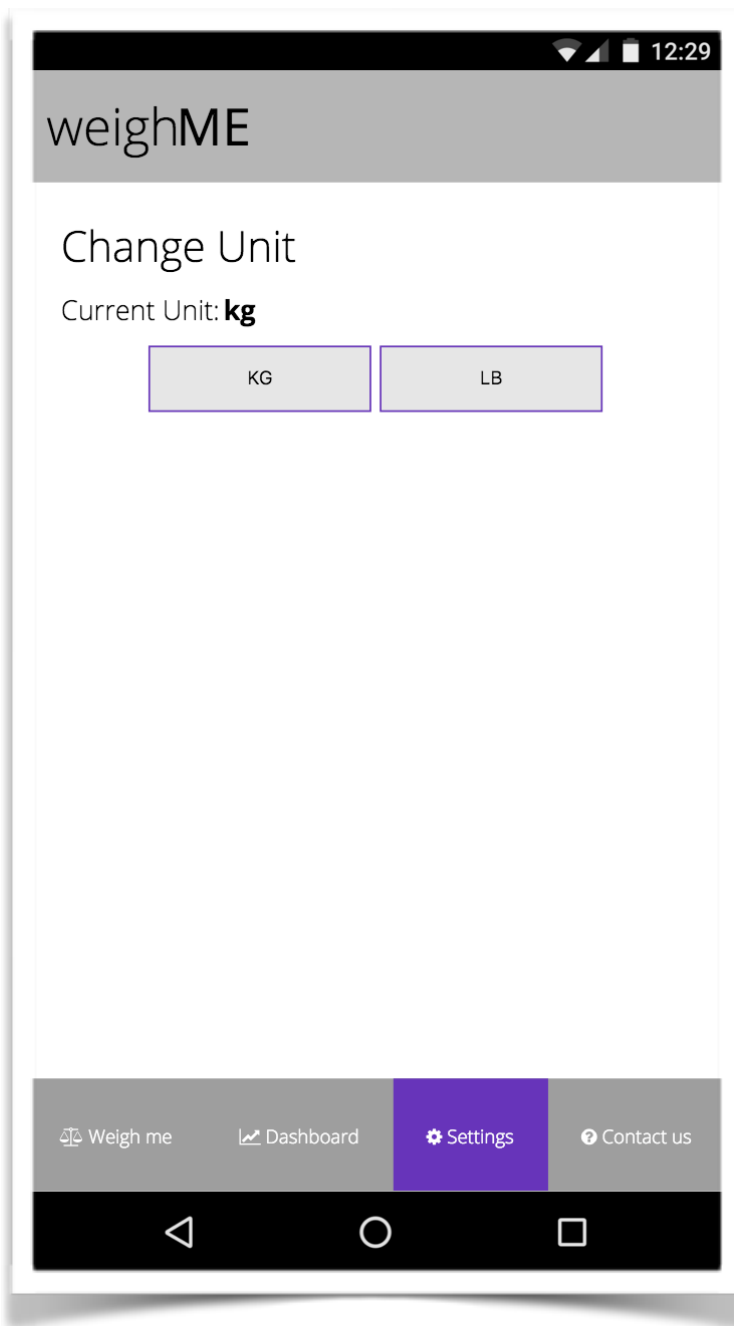


Figure 9. Settings view

Last view is the contact us view, that allows the users to communicate with the developers of the application and raise their concerns. A built in mail server was imagined for this window, however JavaScript does not enable free mail servers or have limited functionality's enabled. Hence, instead of providing a built in emailing service, instead this window gives an information about the roots of the app and allows users to contact them on their choosing.



We would love to hear from you!



weigh**ME!** is a weight tracking application that is designed to help individuals pursue a healthier lifestyle built by Michigan State Spartans. Please do not hesitate to contact us at DT2_WeighMe@gmail.com

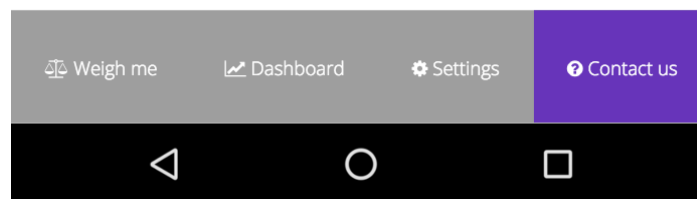


Figure 10. Contact Us

Chapter 4: Tests and Performance of In-Shoe Force Sensor

Following completion of the design and prototype implementation steps, tests to validate the In Shoe Force Sensor device performance were critically important. Team 2 developed and implemented tests for weight response accuracy, and power consumption and how this affects weight accuracy and communication. Information on key User friendliness factors in the app design were also considered.

Weight Response

Fixed Weights

After we developed a final design and achieved operation, there were many tests done to refine our calibrations and ensure that we were meeting the requirements of the device. One of the first and most obvious tests done was regarding the weight accuracy that was achieved from our sensors. In order to do this test, we attached our sensors in a Wheatstone bridge formation to the slots of a bathroom scale and read the output while using the HX711 ADC and the Arduino Uno. The weights used were 2 35lb Dumbbells that were acquired from a group member's home. We ran a basic code that would continually write the weight read out of the scale to a serial monitor that we could then monitor from our computer. The Arduino would output multiple readings per second, so the overall range of these numbers was used to determine the average and then compared to the actual weight. In order to determine the actual weight, an in-tact EatSmart Scale was used. The test was repeated multiple times without zeroing the scale to ensure that the changing stress of the strain gauges would not alter the weight reading. In addition, the dumbbells were placed in different orientations and positions on the scale each time to see how weight distribution affected the reading. The results can be seen in Table 1 for the 35lb dumbbell and used a calibration factor of 10300.

		Our Reading		
Nominal Weight	Commercial Scale	Low Weight (lbs)	High Weight (lbs)	Avg. Weight (lbs)
35	35	35.1	35.1	35.1
35	35	35.1	35.1	35.1
35	35	35	35.2	35.1
35	35	35	35.2	35.1
35	35	35	35.2	35.1
35	35	35	35.2	35.1
35	35	35	35.2	35.1
35	35	35	35.2	35.1

Table 1. Scale showing comparison between our scale's reading *and the*

As can be seen in this table, the weight readout between the EatSmart Scale and our own scale agreed across many of the different tests. In order to get a more accurate reading, the average was taken between the highest weight output from the HX711 and the lowest weight. The average error from the actual weight was no more than .1 lbs for any of the readings and ultimately is equivalent to an error of 45.36 grams. Although this number does not reach the requested 10g accuracy, it was extremely precise and we were pleased with the results.

For the next testing, we combined the two 35lb dumbbells to see how the scale would react to an increased weight. The results from this testing can be seen in Table 2.

Cal Factor	Commercial Scale (lbs)	Our Measurement		
		Low Weight (lbs)	High Weight (lbs)	Average Weight (lbs)
10300	70.8	70.7	71	70.85
10300	70.8	70.8	71	70.9
10300	70.8	70.6	70.9	70.75
10300	70.8	70.7	70.9	70.8
10300	70.8	70.6	70.9	70.75
10300	70.8	70.6	70.8	70.7
10300	70.8	70.6	70.8	70.7
10300	70.8	70.6	70.9	70.75
10300	70.8	70.6	70.8	70.7
10300	70.8	70.7	70.9	70.8

Table 2. Comparison of readings between our created scale and a commercial scale with 70lb dumbbells

As can be seen by the results, the average weight given by our scale was within a .1lb error for all of the readings. This gave us confidence that our scale's calibration factor would not need to be altered to much for future measurements.

As well as the accuracy that was shown by our set-up, it is also very important that the numbers being read from the scale were consistent. This gave us confidence that our weight sensors would not drift from their original reading and allow a user to truly track their weight as the day progressed.

SUBJECT WEIGHTS

After verifying that our scale was giving an accurate reading using known weights, the next test that was conducted used a real person to see how the accuracy would vary. The set up for this test consisted of our scale placed in the holders that they were originally in for the EatSmart scale. The mounts for the sensors were then placed under an aluminum frame and a group member stepped on and off the scale at different points and with different orientations to simulate varying situations. The results from this testing can be seen in Table 3.

Cal Factor	Commercial Scale (lbs)	Our Measurement			Error (lbs)
		Low Weight (lbs)	High Weight (lbs)	Average Weight (lbs)	
10300	199.4	198.2	199	198.6	0.8
10300	199.4	198.4	199.1	198.75	0.65
10300	199.4	198.4	199	198.7	0.7
10300	199.4	198.4	199	198.7	0.7
10300	199.4	198.6	199.3	198.95	0.45
10300	199.4	198.6	199.3	198.95	0.45
10300	199.4	198.6	199.3	198.95	0.45
10300	199.4	198.6	199.4	199	0.4
10300	199.4	198.6	199.3	198.95	0.45
10300	199.4	198.5	199.2	198.85	0.55

Table 3. Comparison of created scale and commercial scale using human subject

Once again, the results of this test showed that our readings stayed within a range of .1lbs from the accepted weight received from the scale. The average weight was used to give a better idea of the overall range of the weight readings. The overall results of this test further cemented our confidence in the accuracy of the scale.

Power Consumption

Operational Power Requirements

The power requirement of the product was dictated by the circuit components that were incorporated into the weight sensing hardware. The components this hardware consisted of the Arduino Uno microcontroller, Bluetooth module, and load cell components. The power demand of each component can be found in Table 4. The highest power demand was from the microcontroller as can be seen in the table.

Power Requirements by Component		
Component	Operational Voltage	Operational Current
Bluetooth transceiver	5V	
Sparkfun Load Cell Amplifier	2.7V-5V	< 1.5mA
Wheatstone bridge circuit	3V	3mA
Arduino Uno microcontroller	5V	170mA (estimated)

Table 4. Power listed for each hardware component.

End of Lifetime Performance

Even at low percentages of battery life, it is important that the weight readings remain accurate, and the Bluetooth communication with the smartphone not be severed. The 9V Duracell battery was chosen to allow sustained performance of the In-Shoe Force Sensor during low power operation. As the Arduino Uno requires an operating power of only 5V, it is not affected by potentially lower voltage output of the battery at the end of lifetime operation. This sustained end of lifetime operation was verified during testing conducted on the In-Shoe Force Sensor; the device provided consistent Bluetooth communication and accurate readings based on the 35 lb weight test until a point where the power supply was depleted. This performance is reflected in Figure 5, where a graphical analysis of the power consumption is given. At this point, the phone lost communication with the Bluetooth transceiver but weight readings showed no fluctuation up to the point where communication was lost. This is the desired low-power outcome of the device, as it would be undesired for the user to receive decreased accuracy in weight readings as the battery is consumed.

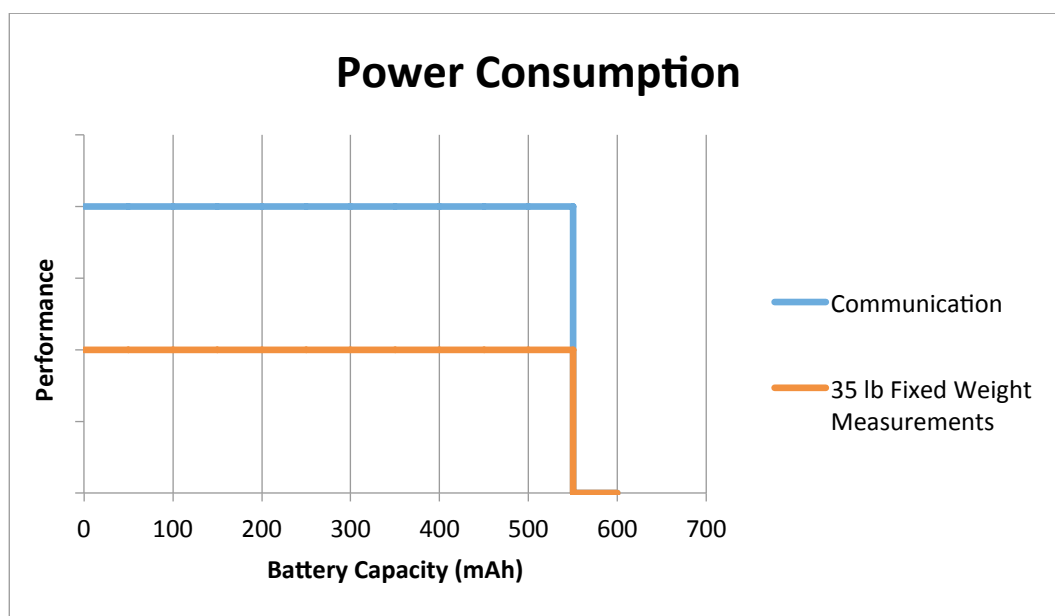


Figure 5. Power consumption graphical representation

Maximum Lifetime Considerations

The power consumption of the In-Shoe Force Sensor primarily occurs during brief periods when a weight reading is requested; otherwise there is little power demand during long spans of inactivity. For this reason, maximum lifetime of the power supply depends largely on the frequency with which the user requests weight readings. If only taking a few weight readings throughout each day, the battery may last as long as 5 days; for more regular weight requests, the battery may only last 1-2 days. For the purpose of prototyping the In-Shoe Force Sensor, 1-2 days of use per battery were typical and was considered acceptable. A week's worth of use per battery would be needed for commercial application and would not be difficult to achieve with the selection of a potent rechargeable battery and a smaller, more efficient, and more specialized microcontroller could be implemented, translating into significant power savings.

User Friendliness Factors

As mobile devices becoming a primary means of convenience in people's life mobile applications are becoming an essential part of one's daily routine. In Shoe Weight Sensor project aims to become a part of this daily routine by even providing even the weight difference before and after a glass of water. That being said, some major aspects need to be completed in a mobile app for it to be user friendly and eventually be successful. There are multiple aspects' consider to accomplish this when building a mobile app. Regardless of how complex the functionality is, the user facing front-end of the application is the most important part of building a good mobile application.

The first factor is the cleanliness of design. A user will not be leaning towards using an application if it is not aesthetically pleasing. Not only this means the compositions of colors, images, fonts and styles, but also it means the application needs to be easy on the eyes. In addition, the design of the application must convey the purpose of it as well. By using contrast of colors, and other UI design principles such as placement, font weight and other components, must complement the major functionalities of the application. weighME! chose a white/gray/purple minimalistic color scheme to achieve this goal. Having the background mostly white with black text on it made the app easy to read. Keeping a header on top of the page and labeling it accordingly conveyed the purpose of the view the user exploring very well. Having the navigation on the bottom to allow users to navigate to any page whenever they want, enabled the user's to have flexibility and rapid execution within the application. In addition, the action button weighME! is placed in the middle of the main window, with very little content surrounding it and emphasizing it as the main button on the page. This way the purpose of the application was communicated to the users very clearly.

The second factor is responsiveness. Responsiveness means that on different window sizes the application adjusts it's dimensions so it doesn't look distorted. If the application looks distorted, it looks unprofessional, inconvenient and it gives the impression that the application is not legitimate. To avoid these possible outcomes, the application was built by using view height and widths, therefore it is flexible and adaptable to use on many different platforms. Most dimensions were set as window percentages, and alignment properties (center, left) and the concept of a flex box which is a styling element that acts as a layout mode that helps for the arrangement of elements on a page such that the elements behave as desired and when encountering different devices and window sizes, the elements are resized and rearranged to fit the windows.

The third factor is the ease of installation. If an application requires a very long download time, and more than a few additional steps after the initial installation period, it becomes a daunting task for users. As the weighME! application was built in JavaScript there were many opportunities to use additional packages and code managers to implement a lot of extra features. The considered packages were AngularJS (a JavaScript framework for single page web applications), JQuery (a JavaScript framework for consistent HTML interactivity), Grunt(a JavaScript task manager that provides functionalities such as concatenating files, uglifying code, translating styling libraries). However only JQuery and small portion of Grunt were elected to limit the application size, and ease of installation. In addition, even though the application initially intended to have a welcome setup that would ask user's demographics questions such as name, age, gender, email this feature was also eliminated to limit the number of clicks users would have to complete as the ease of usage was a primary goal for the sponsor. However, to allow for further development, the application was built flexible enough by compartmentalizing and modularizing different views so that if in the future these features were to be implemented the code page would already be ready and it could be accomplished very easily.

The fourth and last factor is the intuitiveness of the application. Intuitiveness mean where the user's would expect certain functionalities to be, and how would they go about finding certain features. For example, the application doesn't immediately redirect after the unit is changed in the settings. Instead the view is updated with the new weight unit information, so that user's can see the new settings and then they would get redirected. The words are chosen carefully for navigation, and a lot of consideration was put into those labels since they identify the skeleton of the application. The first few seconds a user observes an application is key to engaging them therefore it is important that the app is built in a structure that it doesn't require a lot of effort to figure out how to use it.

Chapter 5: Design Issues

With every newly developed product, it is necessary to consider preexisting and potential design issues. Potential design issues include the infringement on existing prior art, as well as

product liability and mitigation of perceived health risks. This chapter describes the measures Team 2 has taken to secure intellectual property rights and avoid patent infringement, and considerations by the Team for the delivery of a safe, low risk product.

Intellectual Property

Part of the reason that this project was chosen is because of the high marketability of it. The popularity of exercise aids are at an all-time high, especially with the incorporation of technology. Our team has been in contact with the Michigan State university Technology department to look into a patent for our project. Our proposal for this patent is a novel idea that involves integrating a weight sensing device into a shoe. By doing so, a user would be able to wirelessly connect with these sensors in order to see their current weight. This wireless connection would ultimately allow the sensors to communicate with a smartphone application that would act as the user interface between the sensors and the end user. After collecting the weight read out of the sensors, the application will store the information so that users may track their progress and weight fluctuations over time. In the current implementation, four strain gauges have been wired together on each shoe to form a Wheatstone bridge. This Wheatstone bridge receives a 5V DC supply that is provided from an attached Arduino Uno microcontroller. By applying a force to the strain gauges, the resistance of the Wheatstone bridge will fluctuate and therefore alter the output voltage located in the middle of the Wheatstone bridge. This output voltage is then connected to an HX711 ADC that will translate the analog output of the strain gauge circuit to corresponding binary number before transmitting to the Arduino Uno. After the Arduino Uno has received the binary level corresponding to the analog output, it will correlate this number to a weight. The weight is then transmitted to an Android smartphone by using a HC-06 Bluetooth module whenever it is requested by the user through the application. The application consists of a user friendly GUI that was written using the Phone Gap development tool. It includes a startup page, settings tab and a weight log tab that shows the weight fluctuations of user over time. The main page of the application features a large button, that when pressed will connect with the Arduino Uno and request the current weight read out. The overall system design shows proof of concept consistent with the underlying patent proposal.

After this proposal was submitted, a prior art search was conducted. A possible competitive patent was found with a similar design from 2001. There are a few key differences that set us apart from this invention and make our design patentable. The largest difference is our use of Bluetooth and a smartphone to read out and store the weight information. The product from 2001 generates an audible signal to convey any change in weight or to prod the wearing into moving. This product also measures the distance that the user has walked or run and stores the information in a microprocessor. In our design, the weight measurement is taken at the users request through our app. This weight readout method is a huge difference between the two. Our design also utilizes a microcontroller to transmit information, and any data recorded is stored in the phone of the user. The purpose of our design is only to measure weight, unlike the 2001 product. Another difference is the placement of sensors throughout the shoe. The placement of sensors also impacts the accuracy of the device. The 2001 product only utilizes one sensor in the heel of the shoe. Therefore, if the user does not stand with their weight distributed towards their heel, the read out will not be accurate. This is why our product utilizes four sensors. It ensures that no matter how the user is standing, the weight read out will always be consistent and accurate.

Product Liability

Shock Hazard

As mentioned in the power design discussion, a production model of the In-Shoe Force Sensor would require preventative safety measures to protect the user. Placing electronics on footwear presents risk of shock hazard and circuit shorting. When considering this topic, it should first be noted that the final product built for this project was a prototype. As proof of concept, the prototype is designed as a limited usage model to demonstrate that accurate weight readings can be taken via footwear and can be wirelessly communicated to a smartphone device. For this reason, an external hardware device was developed. It is anticipated that a mass produced realization of this product would be almost entirely internal. Whether embedded in the sole or a shoe insert, the hardware would require protection from exposure to a variety of environments. The primary anticipated hazard would come from moisture or submersion in conductive liquid such as water. There are several approaches that could be taken to prevent submersion from harming the user or device. It is proposed that the circuitry of the device could be covered in an insulating gel. This would allow the electronics to be fully submerged in liquid without affecting performance or posing a health hazard. This would also prevent perspiration from the user's foot from interfering with performance or shock hazard as well. Insulation would also prevent the electronics from shorting to conductive surfaces. Of course, insulation can be compromised by continuous flexing of the device during use, or wear and tear from daily usage. In the event the insulation is rendered ineffective, it is important that the safety of the consumer is ensured. This could be accomplished with a fuse or circuit-breaking device, severing the power source from the electronics. When excessive current is drawn from the power source, the fuse or circuit breaker would safely break the circuit, and could then be easily replaced or reset for future use. These measures would guarantee user safety and product durability.

Portal to Phone Hacking Concerns

As mobile devices a big part of daily life, the safety of the applications have become a major concern in society. Not only most users are hesitant to give permissions to an application without trusting it but also it is even often questioned whether or not an application should be installed unless it is fully trusted.

The only permission the weighME application is asking from the users is via Bluetooth. While Bluetooth hacking is possible, the weighME developers tried to keep their application as safe as possible. One of the main considerations was to avoid asking any personal information from the users. As this application is a prototype and can only be launched in development mode on a phone, at the moment the application pairs with a phone if developer tools are enabled. However, the phone will ask for permission to use bluetooth in release mode, and the application will not be able to access any port information should these permissions are rejected.

As Bluetooth hacking would be done by allowing the a harmful application to connect to Bluetooth at a certain port instead of the Arduino, the application is built with a certain address. This address means, only an application that knows the address can connect to the bluetooth port of the phone, therefore as long as the address is protected, it is highly unlikely to have a breach of security on the phone.

Another concern with Bluetooth connection is as the phone is exposed to the outside world (which would be the Bluetooth connection) the phone would be vulnerable as there is an open port. However, in addition to previously mentioned methods, the code works in a way where it follows a connect-read-disconnect cycle, where it connects to the phone on demand and immediately disconnect after the reading. Therefore as the minimum amount of exposition is achieved the developers aimed to keep the app as secure as possible.

To be even safer, weighME! asks for no personal information from the user's at all. Therefore as no personal information is collected by the app, if a possible threat ever comes up there is no danger of retrieval of users' personal information. However, should this application eventually have personal information and the database will be the most vulnerable component. The database there are a few steps can be taken to avoid precautions. The database can encrypt any data that is collected from the user, by salting (combining the password with a random code called a 'hash' that will turn any given string into a completely different form and making it impossible to decrypt unless knowing the hash).

In addition, the codebase is uglified which means combining all of the code together and replacing any variable names with random letters and characters and replacing all equations with different characters as well making the code unreadable from the outside. Therefore the codebase is protected by making it appear completely different to users.

Psychological Impact

Although many users of this device would undoubtedly use it as a means of bettering themselves and becoming healthier, there is certain to be some who use it with negative effects. As with many health centered devices, the possibility of psychological impacts affecting users must be dealt with in a very serious manner. Many people in the world suffer from eating disorders, whether it is anorexia, obesity, or one of countless eating disorders. In addition to their eating disorders, it is possible that those with one of these disorders may also suffer psychologically as a result. According to one study, nearly 50% of those with an eating disorder also meet the criteria for being clinically depressed. The combination of depression that is linked to your weight with a device that offers you instant insight to your current weight could be extremely detrimental to certain individuals. Due to this high risk, it is important to ensure that those with eating disorders do not use our product to perpetuate their illness. As far as steps that can be taken to ensure that this does not happen, it is possible to gather information relating to a person's height within the application. By doing so, the software application could be expanded to determine if a user meets the criterion of being either obese or anorexic and then take further steps. Possible steps that could be taken after collecting this data include providing health provider contact information, online resources, or simply friendly reminders to eat less/more depending on the status of the user. Overall, although there are steps that can be made to offer help and support to users with eating disorders that use this device, there is no way to stop them from buying it in the first place. However, the offer of support and encouraging words may help certain individuals realize the problems they can potentially be causing to themselves and motivate them to alter their lifestyle.

Chapter 6: Final Costs, Schedule, Summary and Conclusion

The completion of the In-Shoe Force Sensor final prototype allows for reflection on the work of Team 2. This final chapter assesses the budget, final expenditures, schedule variations from the original timeline, and a final summary and conclusion.

Budget

Current Budget:

Part	Price	Quantity	Total
Bathroom Scale Sensors	\$35	3	\$105
Shoes Soles	\$12.58	2	\$25.16
Microcontroller	\$27.96	2	\$55.92
Bluetooth Emitter	\$7.99	3	\$23.97
Combinator	\$1.95	2	\$3.90
Load Cell Amplifier	\$9.95	2	\$19.90
Final Poster	\$20	1	\$20
Total			\$253.85

Predicted Production Budget:

Part	Price Per Part	Quantity	Total
Bathroom Scale Sensors	\$0.0175	4	\$0.07
Shoes Soles	\$0.012	2	\$0.024
Microcontroller	\$1.219	2	\$2.43
Bluetooth Emitter	\$1.00	2	\$2.00
Combinator	\$0.04	2	\$0.08
Load Cell Amplifier	\$0.06	2	\$0.12
Total			\$4.724

GANTT Variations

There were few variations from the original GANTT chart. The biggest difference was that there was more overlapping between the set tasks than expected. The hardware and software design also went longer than expected. Much of this could be contributed to delays from ordering parts, and design changes. The research and prototype design stayed relatively on schedule.

Summary and Conclusion

Technological advancement has offered many opportunities to improve daily life. The increasingly wireless world of electronics and growth of “the internet of things” has brought on new applications of technology in areas before not considered. One such area is personal health and fitness tracking. Devices such as Fitbit have allowed users to track hourly activity, distance walked, and calories burned on a daily basis. The rise of such devices bring us to a more health conscious world where one’s personal health and health goals can be tracked and measured for motivation and lifestyle improvement. However, there is one key bio-metric that this sector of technology has not yet been tracking: one’s weight. The application of an in-shoe weight sensor not only eliminates the inconvenience of having to step on a scale for weight readings, but it also provides the opportunity for real-time and on-demand weight information. This would not only help the user stay motivated as they work towards their personal weight goals, it also has potential medical applications. Cases such as eating disorders and other weight related illness can be monitored by a doctor with the patient’s approval. This would help the doctor stay informed about their patient’s health status, make diagnoses, and determine potential treatments. These applications inspired the need for such an in-shoe device, and it was the responsibility of Team 2 to provide a potential solution.

Team 2 set out to create a device capable of taking accurate weight measurements, accurate to within 10 grams, via an in-shoe device that could then wirelessly communicate its weight reading to a smartphone app to be read and tracked to for the user. The final product needed to be compact, mobile, and user friendly. After initial research regarding weight-sensing circuit design and potential weight sensing devices for use in this circuitry, it was determined that the 10 gram weight accuracy was not attainable with the given project budget. In fact, finding load-sensing devices that were both compact and accurate proved difficult. As a solution, a commercial home scale was purchased, from which four strain gauges were harvested for use in our weight-sensing circuit. These strain gauges were configured in a Wheatstone bridge; as the gauges are flexed by the weight load, their resistance changes in a measurable and predictable way that can be used to produce a weight reading. The output of the Wheatstone bridge was then sent to a load cell amplifier and ADC, providing the data needed to output a weight reading via a microcontroller.

An Arduino Uno microcontroller was chosen to conduct the weight reading, calculation, and communication with the smartphone app. The Arduino would receive data from the load cell amplifier for conversion into a weight reading. The Arduino would run a program that would calculate the weight of the user from the data from the load cell circuit. Within this program, several weight samples are averaged to create a final and accurate weight reading. This weight value is then sent from the Arduino Uno to a smartphone app via a Bluetooth transceiver; within the smartphone app, the weight reading is then displayed to the user.

The weight-sensing circuit, load cell amplifier and ADC, and Arduino Uno connected to a 9V battery were then packaged into a compact final product that could be worn on the users feet. For the purpose of a proof of concept, the final product took the form of a sandal-like device that could be attached to the bottom of the user's footwear. Once placed on the bottom of the shoe, the user could request a weight reading from their smartphone app and receive an immediate reading. In the realization of a production model of the weight-sensing device, the hardware would be reduced in size, allowing it to fit discretely and comfortably within any desired footwear. This on-demand weight reading capability provides the opportunity for personal fitness and medical application.

Design Team 2 is honored and delighted to have contributed to the initial realization of the in-shoe weight-sensing device. It has been thrilling to be involved in creating a product of such great importance to the advancement of health technology, with excellent potential for commercial production moving forward. The team would like to thank Dr. Bauer for presenting us with such an excellent capstone design opportunity, as well as Dr. Ayres for her key role in the facilitation of the team's success through her expertise and coaching.

APPENDICES

APPENDIX 1: TECHNICAL ROLES, RESPONSIBILITIES, AND WORK ACCOMPLISHED

Goksu Adanali: My part in this project was to develop the application that would communicate with the Arduino. I was responsible for building a user friendly application that is efficient, fast and fit to the specs of the project. To do so, I took part in researching frameworks, development tools and coding languages to see what would be fit for this project. I also had to ensure that the platform picked for the development would be compatible with the Arduino, and would also be able to implement the Bluetooth module that would communicate with the module as well. I made design decisions which meant designing the front end of the application that includes the color scheme, work flow, main architecture and layouts. As one of the software specialists my position held a liaison aspect where I had to ensure the built product is able to communicate with the Arduino near perfect. I was part of coding the majority of the user facing components of the application, building the app with the aid of PhoneGap development tools, styling the presentation application and ensuring the application component is implemented without bugs and built in a manner where it would welcome further growth. To do so, I tried modularizing the majority of the application for maintenance, flexibility and reusability purposes. I also ensured that the application followed the software engineering concepts, such as maintainability and speed of execution. I believe as a software specialist I became an expert in terms of making decisions on the application architecture, prevented “scope creeping” (when a feature is added to the development plan mid development process) and prioritized features depending on their relevance to the project completion. I was involved with the mostly application development process along with helping Bluetooth communication, from scaffolding step to finished process.

Alexander Chamberry: During the course of this project, I was primarily in charge of the coordination of all forms of presentations regarding the project. This includes, but is not limited to, the preparation, review and supervision of all needed visual and oral presentations; where I prepared the skeleton and majority of the documentation. I was also in charge of capturing images and videos of all concepts, ideas, progress and prototypes. Finally, I also assisted the documentation manager in the preparation of all necessary documentation. During the course of the lab, I was assigned as a general support member for both Alexandra and Steve during their work with the Sensor and Power subsystems. As the semester progressed, however, I found myself doing a lot of work with making sure that the individual subsystems were operating correctly when placed together. During many occasions, our team found that while an individual test we performed would give positive results, combining the individual parts would create new problems. For this reason, I did a lot of work in developing the interaction between subsystems and the problems that occurred while prototyping. Specifically, I was able to determine the correct connections that needed to be made on the Load Combinator board to arrange the strain gauges in to a Wheatstone bridge. During initial testing, we had been receiving output voltages from the Wheatstone bridge that would vary wildly and were not representative of the results that we expected. After much troubleshooting, I was able to pinpoint the problem to the PCB board we were using to connect the strain gauges in a Wheatstone bridge and ultimately to our ADC. By rearranging the connections, we eventually got an output that could be very accurately correlated to a force being exerted on the strain gauge. In addition to the large amounts of prototyping that I did, I also was able to design and implement a vector board that was used to connect the Arduino to both the output of the HX711 ADC as well as the HC-06 Bluetooth module. This vector board contained a 5v and Ground strip that was used to power the HC-06 and provide an excitation voltage to the Wheatstone bridge. In addition, this vector

board connected the Tx/Rx pins between the Arduino and HC-06. The Tx pin of the Arduino needed to be stepped down from its original 5v to 3.3v and was done using a basic voltage divider. One of these boards was fabricated for each of the Arduinos and was connected directly over the Arduino using header pins so that the overall footprint was minimal. In addition to these individual contributions that I was able to make to the project, I also assisted other team members with their own subsystems that they were in charge of. Included in this assistance was helping with the calibration of the scale, establishing a Bluetooth connection, transmitting the weight readout of the strain gauges wirelessly, and various other tasks.

Steven Criner: My initial task was to find a power supply that was small, mobile, and potent enough to sustain the power demand of operating the electronics for a desired timespan of no less than one day. Due to its low cost, high current rating, and compatibility with the Arduino Uno microcontroller I selected the Duracell Duralock 9V battery as the power source of the device. To ensure that it met our design criteria, I consulted datasheets of every component we intended to use in the hardware design including the Arduino, strain gauges, and Bluetooth transceiver to calculate the power draw of the device under operation. I then compared this with the current rating of the battery to estimate the lifespan of the battery for anticipated operating conditions of the device. I then worked with Alexander and Alexandra on research, design, and implementation of the force-sensing circuit that would be used to take in-shoe weight measurements. During research of weight-sensing circuitry, I found an article that discussed how strain gauges can measure stress and strain of an object; this article went on to explain how strain gauges can be implemented in a Wheatstone bridge to measure their change in resistance. I proposed that strain gauges could be used in the format of a Wheatstone bridge in full-bridge configuration to provide measurable change in resistance that could then be measured and converted to a weight reading. After the team purchased an electronic commercial scale for reverse engineering purposes and harvesting of its strain gauges, I worked with the team to understand the scale's circuitry and how its strain gauges were configured. Although we were initially unable to identify the configuration the scale's gauges, Alexandra later found another website that further explained how strain gauges could be configured into a full Wheatstone bridge for force measurement and weight readings. We then purchased PCBs and a load cell amplifier designed for use in a home scale to be used in our in-shoe device. In addition to assisting in circuit design, I also collaborated with Alexandra and Alexander to create a prototype for testing and proof of concept. Before implementing the hardware to a footwear prototype, we fabricated the circuitry to ensure that it was capable of accurate weight measurements. Bringing the circuitry to take weight readings and communicate properly with the in-phone app was a collaborative effort of all members of the team. After the hardware and software components were operating as desired, I worked with Alexandra and Alexander to fabricate and assemble a device that could be strapped onto the foot external to a shoe. Together we designed the footwear, packaged our circuitry in a way it could easily be mounted to the footwear, and performed testing and corrective action with the help of Goksu and Mohamadou to compose a fully functioning product.

Mohamadou Diatta:

Alexandra Mackenzie: The technical role that I was originally delegated to was to work on the sensor design as a member of the hardware team. In order to become knowledgeable in the way that the sensors in scales operate, I started to research the patents and technology of bathroom scales. Through the dissection of a scale and other research, one of my group member, Steven, had discovered that the sensors are connected in a Wheatstone bridge configuration. However, with four sensors there are 12 wires to connect in this configuration which did not seem to be possible. In my research on bathroom scales, I stumbled onto a website tutorial about how to make your own scale. The tutorial confirmed our idea of how the sensors were connected and revealed how it was accomplished. A PCB called a combinator is used to configure the 12 wires into a four wire Wheatstone bridge. I immediately ordered these parts and our group incorporated them into our design upon their arrival. With these we were able to start getting weight readings. Once we had calibrated our design, Alexander Chambery and I started testing. We completed a variety of tests at different weights. Alexander placed all of the weights on our prototype device and I recorded all of the readings. I also had a role in trying out different prototype designs. The first thing that I did was cut out a piece of aluminum in the shape of a shoe sole. This was a possible option for the base of the project. However; the aluminum was not easy to work with so our group decided to go with sheet plastic as the sole. Once our shoe soles were returned from being cut out at the ECE shop, I set out on designing the best sensor placement on the soles and how to wire them. After confirming with the group, my design was agreed upon for the final project. The design consists of two sensors on the ball of the foot and two on the heel. The wiring was in a grid like pattern. In tandem with this job, I created the way that the shoes attach to the user. I sewed Velcro to the sole to go over the bridge of the user's foot to keep it in place. I also attached a Velcro strap to a cup on the heel of the shoe to secure around the users ankle and keep it from flopping like a sandal. My final task has been to create a pouch for the microprocessor and battery to go into that can easily attach to the combinator and amplifier on the shoe sole. The design for this pouch is to have it attach around the users ankle. That way the two part of the project are close enough to attach to each other.

APPENDIX 2: REFERENCES

<http://mobihealthnews.com/34601/prediction-mobile-health-device-market-to-have-increased-clinical-focus>

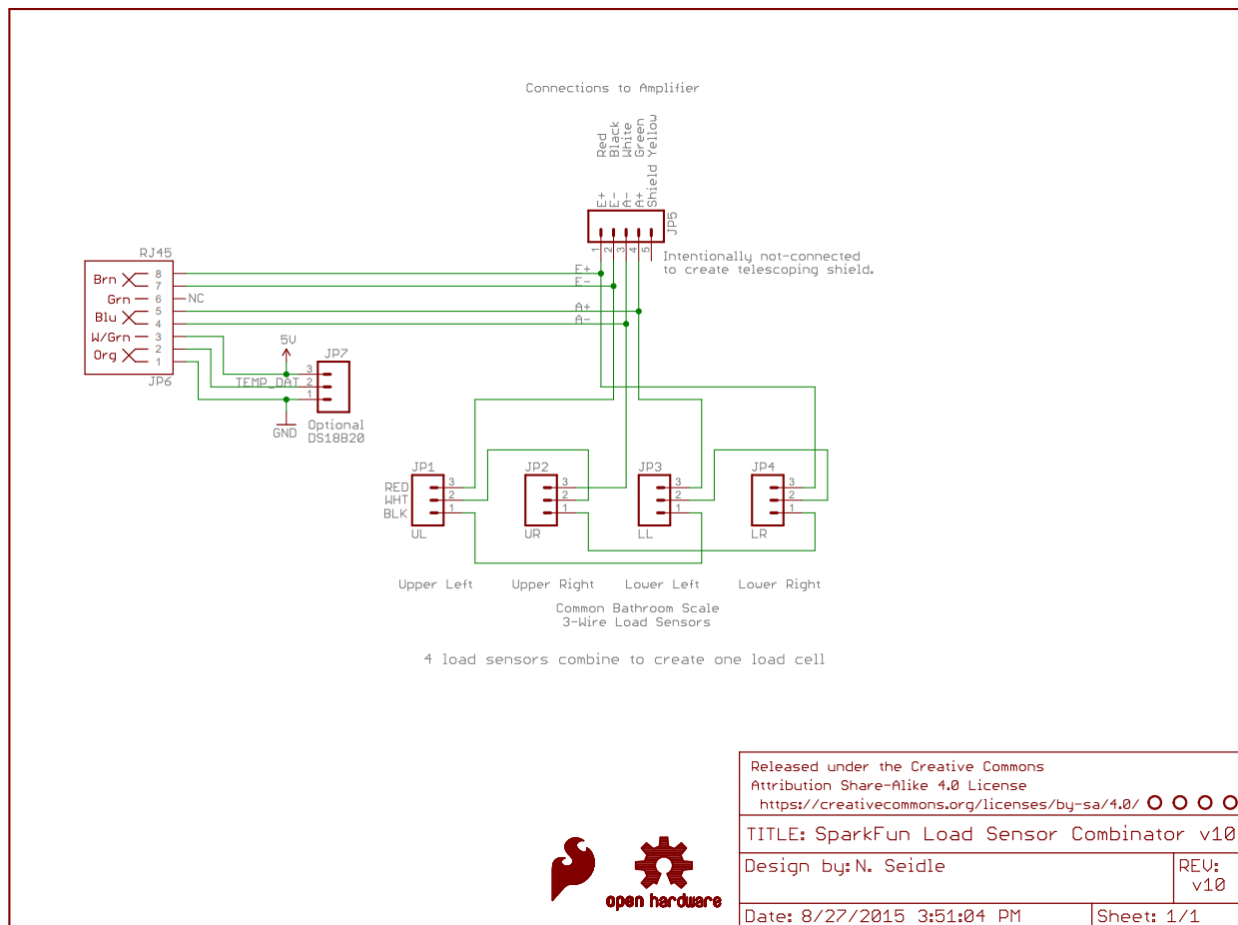
<http://www.anad.org/get-information/about-eating-disorders/eating-disorders-statistics/>

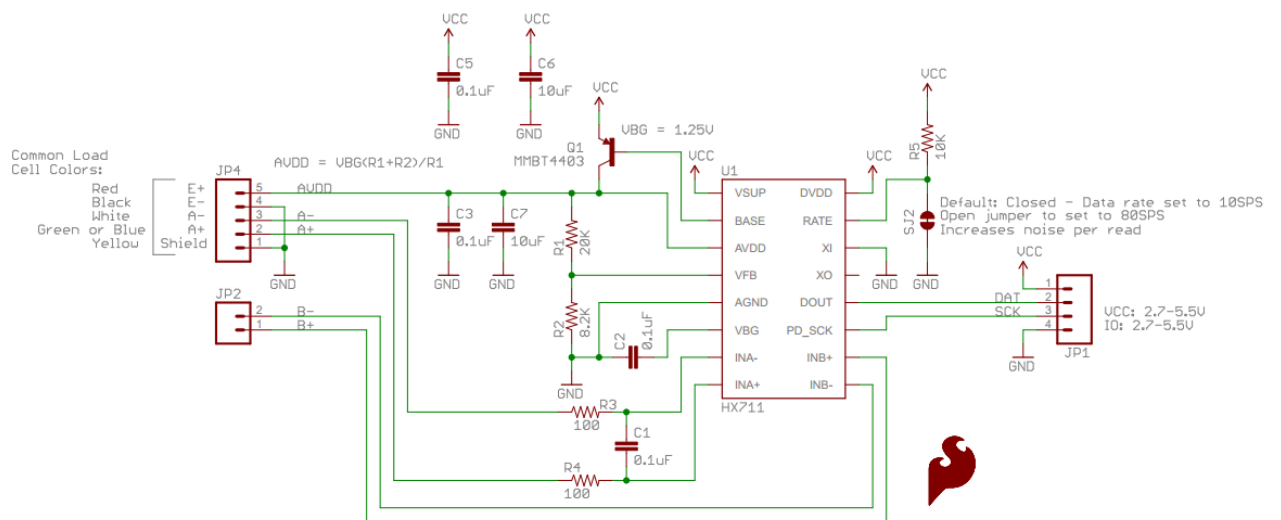
http://www.cdc.gov/dhdsp/data_statistics/fact_sheets/fs_heart_failure.htm

<http://www.cmswire.com/cms/customer-experience/top-5-rules-for-creating-user-friendly-mobile-apps-015841.php>

<https://www.techopedia.com/2/27824/security/common-methods-hackers-are-using-to-crack-your-cellular-phone>

APPENDIX 3: TECHNICAL DOCUMENTS, CODE, ETC.





Released under the Creative Commons Attribution Share-Alike 4.0 License https://creativecommons.org/licenses/by-sa/4.0/	
TITLE: SparkFun_HX711_Load_Cell_v10	
Design by: N. Seidle	REV: v10
Date: 3/3/2015 9:49:54 AM	Sheet: 1/1

Application code

JavaScript files

Index.js -> Main logic

```
1  var macAddress = "98:D3:31:30:76:9E";
2  var macAddress2 = "20:16:02:14:33:51";
3
4  var app = {
5    initialize: function() {
6      this.bindEvents();
7      localStorage.currentUnit = JSON.stringify("kg");
8    },
9    initializeSettings: function() {
10     document.getElementById("current-unit").innerHTML = JSON.parse(localStorage.currentUnit);
11   },
12   bindEvents: function() {
13     document.addEventListener('deviceready', this.onDeviceReady, false);
14     Measure.addEventListener('touchstart', this.connect1, false);
15   },
16   onDeviceReady: function() {
17     console.log("device ready");
18   },
19
20   connect1: function() {
21     console.log("connecting to bluetooth 1");
22
23     var onConnect = function() {
24       // subscribe for incoming data
25       bluetoothSerial.subscribe("\n", app.onMessage1, app.subscribeFailed);
26       bluetoothSerial.write("1");
27     };
28
29     bluetoothSerial.connect(macAddress, onConnect, app.onFailconnect);
30
31   },
32   connect2: function() {
33     console.log("connecting to bluetooth 2");
34
35     var onConnect = function() {
36       // subscribe for incoming data
37       bluetoothSerial.subscribe("\n", app.onMessage2, app.subscribeFailed);
38       bluetoothSerial.write("1");
39     };
40
41     bluetoothSerial.connect(macAddress, onConnect, app.onFailconnect);
42   },
43   onFailconnect: function() {
44     console.log("connection failed");
45   },
```

```

onMessage1: function(data) {
    console.log("reading 1");
    counter.innerHTML = data;
    bluetoothSerial.disconnect(app.connect2, app.failDisc);

},
onMessage2: function(data) {
    console.log("reading 2");
    counter.innerHTML = "data";
    bluetoothSerial.disconnect(app.successDisc, app.failDisc);
},
subscribeFailed: function() {
    alert("subscribe failed");
},
successDisc: function() {
    console.log("Disconnect successful");
},
failDisc: function() {
    console.log("Disconnect Unsuccessful");
},
};

$(document).ready(function(){
    $("#kg-button").on("click",function() {
        localStorage.currentUnit = JSON.stringify("kg");
        document.getElementById("current-unit").innerHTML = "kg";
        return false;
    });

    $("#lb-button").on("click",function() {
        localStorage.currentUnit = JSON.stringify("lb");
        document.getElementById("current-unit").innerHTML = "lb";
        return false;
    });
});

```

HTML Files

Index.html

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="format-detection" content="telephone=no" />
6     <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height" />
7     <link rel="stylesheet" type="text/css" href="css/index.css" />
8     <link rel="stylesheet" href="css/font-awesome.min.css">
9     <link rel="stylesheet" href="css/main.css">
10    <link rel="stylesheet" href="css/header.css">
11    <link rel="stylesheet" href="css/navigation.css">
12    <link rel="stylesheet" href="css/weigh-me.css">
13    <link rel="stylesheet" href="css/settings.css">
14    <link rel="stylesheet" href="css/contact-us.css">
15    <link rel="stylesheet" href="css/dashboard.css">
16    <title>WeighMe App</title>
17  </head>
18  <body>
19    <div class="header-container">
20      <span class="header-text-thin">weigh</span>
21      <span class="header-text-bold">ME</span>
22    </div>
23    <div class="weigh-me-body-container">
24      <div class="weigh-me-prompt">
25        <span>Click the button below to get an instant weight reading</span>
26      </div>
27      <div id="counter"></div>
28      <div id="statusDiv"></div>
29
30      <button class="weigh-me-button" id="Measure">
31        <span class="weigh-me-text-thin">weigh</span>
32        <span class="weigh-me-text-bold">ME</span>
33      </button>
34
35      <div id="tempDiv"></div>
36      <div class="weigh-me-slogan">
37        <span>A healthier life at your tiptoes</span>
38      </div>
39    </div>
40
41    <div class="nav-button text-primary-color">
42      <button class="nav-button text-primary-color">
43        <a href="settings.html"><i class="fa fa-cog"></i>
44        <span>Settings</span></a>
45      </button>
46      <button class="nav-button text-primary-color">
47        <a href="contact.html"><i class="fa fa-question-circle"></i>
48        <span>Contact us</span></a>
49      </button>
50    </div>
51    <script type="text/javascript" src="cordova.js"></script>
52    <script type="text/javascript" src="js/jquery-1.12.3.min.js"></script>
53    <script type="text/javascript" src="js/index.js"></script>
54    <script type="text/javascript">
55      app.initialize();
56    </script>
57  </body>
58 </html>
```

Settings.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height" />
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <link rel="stylesheet" href="css/main.css">
    <link rel="stylesheet" href="css/header.css">
    <link rel="stylesheet" href="css/navigation.css">
    <link rel="stylesheet" href="css/weigh-me.css">
    <link rel="stylesheet" href="css/settings.css">
    <link rel="stylesheet" href="css/contact-us.css">
    <link rel="stylesheet" href="css/dashboard.css">
    <title>WeighMe App</title>
  </head>
  <body>
    <div class="header-container">
      <span class="header-text-thin">weigh</span>
      <span class="header-text-bold">ME</span>
    </div>

    <div class="settings-change-container">
      <form class="change-info-form">
        <div class="input-group-addon header-text">
          <div class="change-unit-header">
            <span class="header-text-thin">Change Unit</span>
          </div>
          <div class="header-text-body">
            <span class="header-text-thin">Current Unit: </span>
            <span class="header-text-bold" id="current-unit"></span>
          </div>
        </div>
        <div class="options-container">
          <button class="weight-button" id="kg-button">
            <span>KG</span>
          </button>
          <button class="weight-button" id="lb-button">
            <span>LB</span>
          </button>
        </div>
      </form>
    </div>
  </body>
</html>
```

```
        </div>
    </form>
</div>
<div class="accent-color navigation-container">
    <button class="nav-button text-primary-color">
        <a href="index.html"><i class="fa fa-balance-scale"></i>
        <span>Weigh me</span></a>
    </button>
    <button class="nav-button text-primary-color">
        <a href="dashboard.html"><i class="fa fa-line-chart"></i>
        <span>Dashboard</span></a>
    </button>
    <button class="nav-button text-primary-color is-active">
        <a href="settings.html"><i class="fa fa-cog"></i>
        <span>Settings</span></a>
    </button>
    <button class="nav-button text-primary-color">
        <a href="contact.html"><i class="fa fa-question-circle"></i>
        <span>Contact us</span></a>
    </button>
</div>
<script type="text/javascript" src="cordova.js"></script>
<script type="text/javascript" src="js/jquery-1.12.3.min.js"></script>
<script type="text/javascript" src="js/index.js"></script>
<script type="text/javascript">
    app.initializeSettings();
</script>
</body>
</html>
```

Dashboard.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height" />
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <link rel="stylesheet" href="css/main.css">
    <link rel="stylesheet" href="css/header.css">
    <link rel="stylesheet" href="css/navigation.css">
    <link rel="stylesheet" href="css/weigh-me.css">
    <link rel="stylesheet" href="css/settings.css">
    <link rel="stylesheet" href="css/contact-us.css">
    <link rel="stylesheet" href="css/dashboard.css">
    <title>WeighMe App</title>
  </head>
  <body>
    <div class="header-container">
      <span class="header-text-thin">weigh</span>
      <span class="header-text-bold">ME</span>
    </div>
    <div class="dashboard-container">
      <div class="dashboard-message-container">
        <span>We're working on very hard to get you this feature!</span>
        <span>Coming soon!</span>
      </div>
    </div>

    <div id="tempDiv"></div>

    <div class="accent-color navigation-container">
      <button class="nav-button text-primary-color">
        <a href="index.html"><i class="fa fa-balance-scale"></i>
        <span>Weigh me</span></a>
      </button>
      <button class="nav-button text-primary-color is-active">
        <a href="dashboard.html"><i class="fa fa-line-chart"></i>
        <span>Dashboard</span></a>
      </button>
      <button class="nav-button text-primary-color">
        <a href="settings.html"><i class="fa fa-cog"></i>
        <span>Settings</span></a>
      </button>
      <button class="nav-button text-primary-color">
        <a href="contact.html"><i class="fa fa-question-circle"></i>
        <span>Contact us</span></a>
      </button>
    </div>
    <script type="text/javascript" src="cordova.js"></script>
    <script type="text/javascript" src="js/index.js"></script>
    <script type="text/javascript">
      app.initialize();
    </script>
  </body>
</html>
```

Contact

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/html">
  <head>
    <meta charset="utf-8" />
    <meta name="format-detection" content="telephone=no" />
    <meta name="viewport" content="user-scalable=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width, height=device-height" />
    <link rel="stylesheet" type="text/css" href="css/index.css" />
    <link rel="stylesheet" href="css/font-awesome.min.css">
    <link rel="stylesheet" href="css/main.css">
    <link rel="stylesheet" href="css/header.css">
    <link rel="stylesheet" href="css/navigation.css">
    <link rel="stylesheet" href="css/weigh-me.css">
    <link rel="stylesheet" href="css/settings.css">
    <link rel="stylesheet" href="css/contact-us.css">
    <link rel="stylesheet" href="css/dashboard.css">
    <title>WeighMe App</title>
  </head>
  <body>
    <div class="header-container">
      <span class="header-text-thin">weigh</span>
      <span class="header-text-bold">ME</span>
    </div>
    <div class="contact-us-container">
      <div class="header-text">
        <span class="header-text-thin">We would love to hear from you!</span>
      </div>
      <div class="spartan-logo">
        
      </div>
      <div class="welcome-body">
        <span class="body-thin">weigh</span>
        <span class="body-bold">ME!</span>
        <span class="body-thin">is a weight tracking application that is designed to help individuals pursue a healthier lifestyle</span>
        <span class="body-thin">Please do not hesitate to contact us at DT2_WeighMe@gmail.com</span>
      </div>
    </div>

    <div class="accent-color navigation-container">
      <button class="nav-button text-primary-color is-active">
        <a href="index.html"><i class="fa fa-balance-scale"></i>
          <span>Weigh me</span></a>
      </button>
      <button class="nav-button text-primary-color">
        <a href="dashboard.html"><i class="fa fa-line-chart"></i>
          <span>Dashboard</span></a>
      </button>

      <button class="nav-button text-primary-color">
        <a href="contact.html"><i class="fa fa-question-circle"></i>
          <span>Contact us</span></a>
      </button>
    </div>
    <script type="text/javascript" src="cordova.js"></script>
    <script type="text/javascript" src="js/jquery-1.12.3.min.js"></script>
    <script type="text/javascript" src="js/index.js"></script>
    <script type="text/javascript">
      app.initialize();
    </script>
  </body>
</html>
```