

6-12-2013

# Wireless impact detection system

Shawno Auwae  
*Santa Clara University*

Kyle Terriere  
*Santa Clara University*

Follow this and additional works at: [https://scholarcommons.scu.edu/idp\\_senior](https://scholarcommons.scu.edu/idp_senior)



Part of the [Computer Engineering Commons](#), and the [Electrical and Computer Engineering Commons](#)

---

## Recommended Citation

Auwae, Shawno and Terriere, Kyle, "Wireless impact detection system" (2013). *Interdisciplinary Design Senior Theses*. 2.  
[https://scholarcommons.scu.edu/idp\\_senior/2](https://scholarcommons.scu.edu/idp_senior/2)

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Interdisciplinary Design Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact [rscroggin@scu.edu](mailto:rscroggin@scu.edu).

# **WIRELESS IMPACT DETECTION SYSTEM**

by

Shawno Auwae and Kyle Terriere

## **SENIOR DESIGN PROJECT REPORT**

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

Santa Clara, California

June 12, 2013

SANTA CLARA UNIVERSITY  
Department of Electrical Engineering

Date: June 6, 2013

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Shawno Auwae and Kyle Terriere

ENTITLED

Wireless Impact Detection System

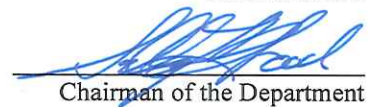
BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN ELECTRICAL ENGINEERING



Thesis Advisor



Chairman of the Department

**Santa Clara University**  
**DEPARTMENT of COMPUTER ENGINEERING**

Date: June 12, 2013

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY  
SUPERVISION BY

**Shawno Auwae and Kyle Terriere**

ENTITLED

**Wireless Impact Detection System**

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

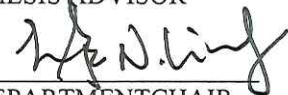
DEGREE OF

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING**



---

THESIS ADVISOR



---

DEPARTMENT CHAIR

## Abstract

Recently, brain trauma caused by multiple concussions has become a pressing issue for contact sports, mainly American football players. The severity of the impacts in these sports have led to discussion concerning the standard safety equipment used, as well as the rules enforcing a certain style of play. Multiple studies have shown that concussions have damaging long-term effects, including dementia and memory loss. The dominant professional football organization, the National Football League (NFL), employs qualified athletic trainers to diagnose and treat players with concussions, but many incidents still go undetected. This problem is magnified at the youth level, where the appropriate personnel and equipment are not readily available, further complicating the detection and treatment of concussions.

Our senior design team worked to create a system that helps athletes and trainers detect severe head impacts that may result in concussions. Our system consists of a small transmitting device comprised of an accelerometer and microcontroller, and a mobile application that will work on the Apple iPad and iPhone, but can later be tailored to any Bluetooth-enabled smartphone or tablet. We were able to design a working system that includes one device connected to a mobile application, while proposing network architecture that would support multiple devices connecting to a single mobile application. In the near future we hope that our product will help create a safer sporting environment by decreasing the odds that a concussion will go undetected and untreated. We expect that data collected from our devices will help researchers prevent concussions and head trauma. Our goal is to encourage participation in contact sports like football without the fear of long-term consequences. We hope that our device can also be modified for use in other areas where brain damage is a major issue, such as the military.

## Acknowledgements

Our group would like to acknowledge the Santa Clara University School of Engineering for providing us with funds and resources to complete this project. We would also like to express our deep gratitude towards Dr. Sarah Kate Wilson and Dr. Silvia Figueira, our project advisors, who consistently provided us with expert guidance in both the technical and practical aspects of our project. We would also like to acknowledge both the Department of Computer Engineering and the Department of Electrical Engineering at Santa Clara University, without whom we would not have been able to this project.

## Table of Contents

<b>Abstract</b> .....	<b>1</b>
<b>Acknowledgements</b> .....	<b>2</b>
<b>1. Introduction</b> .....	<b>5</b>
1.1 Problem.....	5
1.2 Solution.....	5
1.3 Principle.....	6
1.4 Future Application.....	6
1.5 Primary Requirements.....	7
1.6 Secondary Requirements.....	7
<b>2. System Diagram</b> .....	<b>8</b>
<b>3. Device</b> .....	<b>9</b>
3.1 Accelerometer.....	9
3.2 Microcontroller.....	10
3.3 Bluetooth Radio.....	10
<b>4. Mobile Application</b> .....	<b>13</b>
4.1 Design Choices.....	13
4.2 Features.....	14
4.3 Flow Diagram.....	15
4.4 Screenshots.....	15
<b>5. Design Rationale</b> .....	<b>19</b>
<b>6. Testing</b> .....	<b>21</b>
6.1 Test Setup & Procedure.....	21
6.2 Results.....	26
<b>7. Wireless Sensor Network</b> .....	<b>31</b>
7.1 Overview.....	31
7.2 Network Architecture Proposal.....	31
<b>8. Conclusion</b> .....	<b>35</b>
<b>9. List of References</b> .....	<b>36</b>
<b>10. Appendix</b> .....	<b>37</b>
10.1 Source Code (Arduino).....	37
10.2 Source Code (Apple iOS).....	39

## List of Figures

1. System Flow Chart .....	8
2. ADXL377 Accelerometer Evaluation Board .....	9
3. Arduino Uno Microcontroller .....	10
4. Bluetooth Low-Energy Shield for Arduino Uno.....	12
6. Record Screen .....	16
5. Home Screen .....	16
7. Connect Screen .....	17
8. Graph Screen.....	17
9. Save Screen.....	18
10. Test Stage 1.....	22
11. Device Test Setup.....	22
12. Test Stage 2.....	23
13. Test Stage 3.....	24
14. Test Stage 4.....	25
15. Accelerometer Calibration & Verification .....	26
16. Data Calculation & Compartmentalization .....	29
17. Data Transmission.....	30
18. Wireless Sensor Network.....	32
19. Wireless Sensor Network Data Relay.....	33



# 1. Introduction

## 1.1 Problem

Over the past year, brain trauma caused by multiple concussions has become a pressing issue for athletes in contact sports, mainly American football players. The severity of the impacts in these sports has led to discussion concerning the standard safety equipment used, as well as the rules enforcing a certain style of play. Multiple studies have shown that concussions have damaging long-term effects, which include, among other effects, dementia and memory loss. (Radiological Society of North America) These consequences are very real, and have led many athletes, as well as their parents, to question their participation in contact sports. Many are concerned that there is no clear way to detect that a concussion has occurred. (Zhang, Yang and King)

Recently, steps have been taken to combat the rising number of concussions, such as redesigning the football helmet to provide better protection against impacts. Some helmet manufacturers have added more padding to their products. Other proposed solutions even call for the removal of the facemask to discourage players from hard tackles aimed at the head. The dominant professional football organization, the National Football League (NFL), employs qualified athletic trainers to diagnose and treat players with concussions, but many incidents still go undetected. (DeKosky, Ikonovic and Gandy) This problem is magnified at the youth level, where the appropriate personnel and equipment are not readily available, further complicating the detection and treatment of concussions. (Wein)

## 1.2 Solution

Current solutions to this problem include the Riddell Head Impact Telemetry System (HITS) and both the xGuard and xPatch made by X2 Biosystems. The HITS system features a custom helmet outfitted with six accelerometers, recording impact information over the course of a game. This information can only be accessed after a game or practice has completed, since all information must be uploaded to a computer program using a wired connection. In contrast, both solutions from X2 Biosystems

transmit data in real-time using a wireless connection. The xGuard is a specialized mouthguard outfitted with multiple accelerometers and gyroscopes, while the xPatch is a patch that can be applied to the back of an athlete's head, or to the neck. (X2 Biosystems) X2 Biosystems solution accomplishes real-time data transmission by incorporating a sideline base station into the system as a router. This base station relays data from each xGuard or xPatch to a cloud storage unit, from which a mobile application can access all the unique data. (X2 Biosystems) The HITS system is currently available for \$1500 US, while the xGuard and xPatch are not commercially available, but are being implemented by NFL organizations and select college football programs.

We have designed a mobile-application solution utilizing wireless communication technology that will help to detect severe impacts to the head. A device inserted into the helmet will communicate data to the user, describing the forces applied to the helmet. All helmets fitted with the device will send a unique signal to the application. The application will record this data over a set period of time, allowing the user to graphically display information necessary to properly diagnose and treat the athlete.

### **1.3 Principle**

Though much research and planning went into designing a full system, the scope of our project was minimized to include creating a functional prototype of a single device connected to a mobile application.. Consequently, the goal of our project was to successfully implement this design and prove that a functional system consisting of multiple devices would be a feasible and cost-effective solution.

### **1.4 Future Application**

In the near future we hope that our product will help create a safer sporting environment by decreasing the odds that a concussion will go undetected and untreated. We expect that data collected from our devices will help researchers prevent concussions and head trauma. Our goal is to encourage participation in contact sports like football

without the fear of long-term consequences. We hope that our device can also be modified for use in other areas where brain damage is a major issue, such as the military.

### **1.5 Primary Requirements**

To prove that our system could conceivably be implemented, our project required us to accomplish three main goals:

1. Design a physical device component that reliably and accurately measures and records data from impacts to a helmet.
2. Compile and transfer recorded data wirelessly to a mobile application, using the Bluetooth 4.0 (Low-Energy) specification.
3. Design a mobile application that correctly interprets data sent by the device, and allows this data to be graphed and manipulated by a user.

By accomplishing these three main goals our team could safely argue that our design could be relied upon to successfully collect, interpret, and store data relating to head impacts and acceleration. These primary requirements are examined and tested in Test Stages 1 & 2. These Test Stages are explained in depth in Section 6.

### **1.6 Secondary Requirements**

After completing the primary requirements for our project, our team would proceed to implement various secondary requirements. These requirements included:

1. Designing a wireless sensor network that would allow multiple devices to dynamically connect to a single mobile application.
2. Designing a non-intrusive physical device prototype that could be attached to the inside of a helmet and withstand the rigors of sport. Considerations include user safety, invasiveness, temperature and moisture conditions, and aesthetics.

## 2. System Diagram

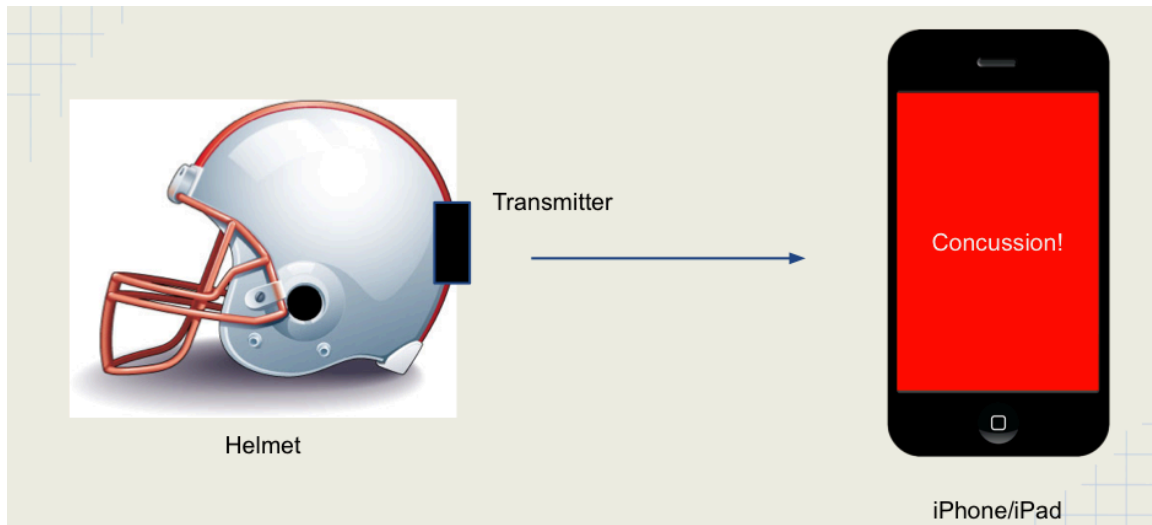


Figure 1 - System Flow Chart

### 3. Device

This section contains information about the technologies incorporated into our proposed physical device. The functions of each of the three main components (accelerometer, microcontroller, and Bluetooth Module) are explained in detail, as well as the reasoning behind our particular design choices.

#### 3.1 Accelerometer

Our design requires a 3-axis accelerometer that measures acceleration resulting from motion, shock, or vibration, with a range of around  $\pm 150$  g's (impacts in football are typically around 50-100 gs). This instrument allows our device to reliably and accurately measure impacts and provide our system with quantifiable data relating to the severity of these impacts.

After careful research and consideration, our group settled on the ADXL377 accelerometer, which is manufactured by Analog Devices, Inc. This is a 3-axis accelerometer with a scale range of  $\pm 200$  gs, and operates using a low-voltage on par with most microcontrollers, making it perfect for our design. This component is already being used to measure impacts in stock car racing and is being tested by some professional football organizations. (Higgins, Halstead and Snyder-Mackler) For design testing purposes we used the ADXL377 Evaluation Board (Fig. 2), which measures 1" x 1". The actual ADXL377 accelerometer chip is a compact 3mm x 3mm x 1.45mm.

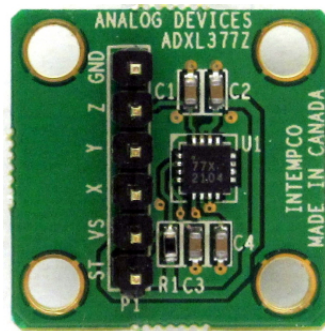


Figure 2 - ADXL377 Accelerometer Evaluation Board

### 3.2 Microcontroller

A microcontroller is a small computer containing a central processing unit (CPU) among other things. It is used in our design to interpret the analog data outputted from the accelerometer and turn this data into digital information that can be transmitted wirelessly. In the context of our design, the most important part of the microcontroller is the analog-to-digital converter (ADC). This component receives analog input from the accelerometer and converts it to digital signals, which the CPU can then manipulate to fit the needs of the system. The microcontroller is the component that allows the physical detection device to operate, bridging the gap between raw data compiled by the accelerometer and the final information we see with the mobile application.

Our project incorporates the Arduino Uno, which is among the most powerful microcontrollers on the market and provides our team with reliability and unparalleled design capabilities. With its fast processing capabilities, it is ideal for our design, which requires high-speed data transfer and precise calculations. The Arduino Uno is shown in Fig 3.

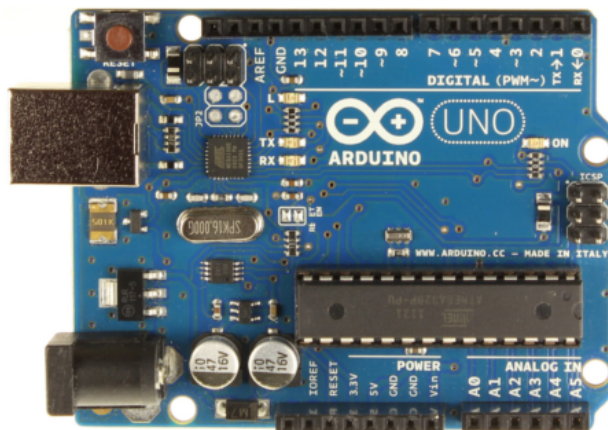


Figure 3 - Arduino Uno Microcontroller

### 3.3 Bluetooth Radio

The Bluetooth radio is the final component of our detection device. The Bluetooth radio allows information compiled by the microcontroller to be transmitted wirelessly to the mobile application utilizing the latest iteration of the Bluetooth specification, Bluetooth 4.0 or Bluetooth Low-Energy. Currently there are several wireless

specifications being included in commercial mobile devices and instruments. The three most popular specifications are Wi-Fi, Bluetooth (classic and low-energy), and ANT. Each of these specifications has their respective strengths and weaknesses, and is applied to products with very different uses. For example, the Wi-Fi specification covers a much larger range than the other two protocols (Wi-Fi networks typically extend hundreds of meters, while a typical ANT or Bluetooth connection is limited to a 10 or 20 meter radius), but requires extra technology to achieve this scale, specifically a wireless network access point, or router. In contrast, both the ANT specification and Bluetooth specifications are peer-to-peer wireless specifications, meaning a device can connect directly to another device without a router. Our team recognized that choosing one of these two specifications would eliminate the need for a stationary access point (such as the base station in the X2 solutions), which would reduce the cost of our system, and simplify the user experience, so we removed Wi-Fi from our potential wireless specification list.

The difference between Bluetooth and ANT, and identifying features that would allow us to choose between them, was a lot subtler. Peer-to-peer wireless specifications are implemented in a few different architectures or topologies, the main three being star, tree, and mesh. A *star* topology is simple: each device in a system can connect to a central hub device, but cannot connect directly to each other. A *tree* topology is an extension of the *star* topology, and allows devices to connect based on a hierarchy model. There is root node device, which can connect to secondary nodes, and these secondary devices connect to tertiary devices, extending the network reach. A *mesh* topology is a true peer-to-peer wireless topology, in that every device in the system can connect directly to every other device in the network. Each device in the *mesh* acts as both a transmitter and receiver with every single device in the network, whereas a device in a *tree* topology can receive data from one set of devices, but can only transmit to a separate set of devices. Bluetooth uses a modified *tree* topology, where one device is a master in a piconet of up to eight devices, and the rest of the devices are slaves. A slave in one piconet can be a slave of another piconet to form a bridge between what is called a scatternet. This topology is flexible and allows for multiple connections, although no device can be a master of multiple piconets at one time. In contrast, ANT is implemented

with a pure *mesh* topology, creating a very secure network, with redundancy to ensure data preservation. While this makes the ANT protocol desirable for our design when considering pure network architecture, commercial availability was a major deciding factor in our decision to ultimately use Bluetooth in our design.

Virtually all mobile phones and tablets on the market have Bluetooth capability, while most of these products require special attachments to enable ANT connectivity. Bluetooth 4.0 also uses less power than ANT and Wi-Fi, prolonging battery life and allowing us to increase storage and memory on our device. (Kamath and Lindh) Implementing a Wireless Sensor Network can also extend the range of Bluetooth. (Khissibi and Idoudi) This network is discussed in depth in Section 9. The wide range of products currently using Bluetooth technology, specifically tablets and mobile phones, as well as the ease of use and low power consumption, make Bluetooth Low-Energy an ideal wireless specification for our design.

For the Bluetooth radio to work with our microcontroller, a shield is required that allows all components to be attached to each other utilizing a breadboard. We examined all the Bluetooth shields on the market, finally deciding on the Bluetooth LE (Low-Energy) Shield (Fig. 4), which is made by Sparkfun Electronics. This device was made with the Arduino microcontroller series in mind, and is specifically synchronous with the Arduino Uno. It features a relatively strong Bluetooth radio that suits our needs, and there is a lot of documentation regarding its usage for different projects, most of which highlights its reliability and ease of use. All of these attributes make it a perfect fit for our design project.

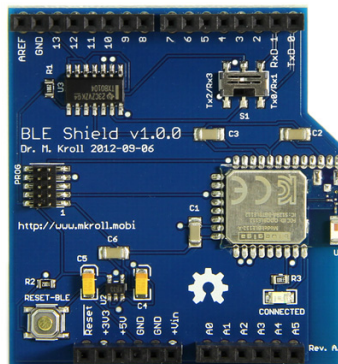


Figure 4 - Bluetooth Low-Energy Shield for Arduino Uno



## 4. Mobile Application

This section contains information on the mobile application, including design choices and application features. Screenshots of the application are provided, as well as designs of proposed application features.

### 4.1 Design Choices

The main goal of our design is to create a system that is affordable for youth athletes and their families while still being easy to use. We made our decisions on parts based on what was popular and what was easy to obtain. We realize that our project would perform better with more expensive parts but we wanted our project to be affordable to our target audience of high school students.

#### **iPad Mini**

The iPad Mini was our top pick for mobile device because of its large screen and its BLE capabilities. We wanted our graphs to be very visible and easy to read and we also wanted the computing power that many smaller mobile phones lack. We knew that if we could get our detection device to communicate with an Apple product we would be able to reach out to large numbers of users.

#### **Xcode**

Although there are a few options for coding an Apple product, we chose Xcode for its seamless integration with the iPad. The graphical user interface allows for the dragging and dropping of buttons and screen transitions on screen. Xcode also gives us the ability to simulate an iPad on the screen for quicker debugging.

#### **Core Plot Framework**

The Core Plot framework was our choice for graphing our results. This framework features simple labeling of axes, a wide variety of colors to use, and live-

updated graphs. We wanted to show our results very clearly on screen for doctors to be able to interpret them.

## 4.2 Features

The purpose of the mobile app is to allow the user to easily connect to devices and view results from the helmets. We noticed that our competitors had a sideline hub or built in memory to store their information, but we wanted to use a very common mobile device that was popular among consumers. Below are some of the features:

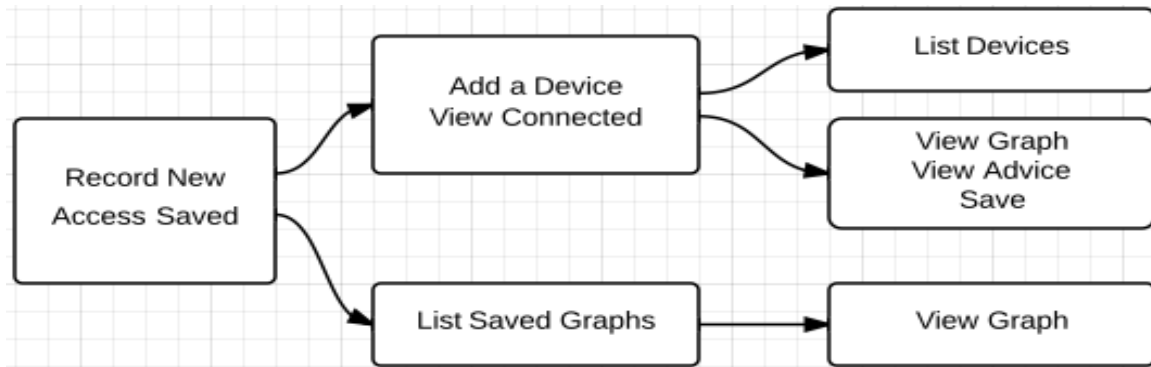
### **Wireless Connectivity**

The goal is to send the data remotely to the user with minimal impact to the helmet wearer. Accordingly, we chose a small, lightweight transmitter, a Bluetooth device, to send the data. Wireless connectivity allows us to connect multiple helmets to one phone or tablet and allows the user to choose which helmets are being recorded.

### **Sideline Monitoring**

Using a mobile app also allows us to have live viewing of the readings from the helmet. Parents and coaches can easily monitor the players on the field to watch out for dangerous collisions. Sideline monitoring also allows the user to save the recordings from the helmet for future discussion and interpretation from a doctor. By using sideline monitoring, we can give the user information on how to treat different head injuries, something that our competitors are unable to provide with their systems.

### 4.3 Flow Diagram



This diagram represents each screen our app will have and the arrows represent which screen the corresponding button will take you to. We use this diagram to show a whole layout of the application interface and how each feature can be reached and used.

### 4.4 Screenshots

The screenshots with descriptions below explain what each screen does for the system and how they are accessed.

## HOME

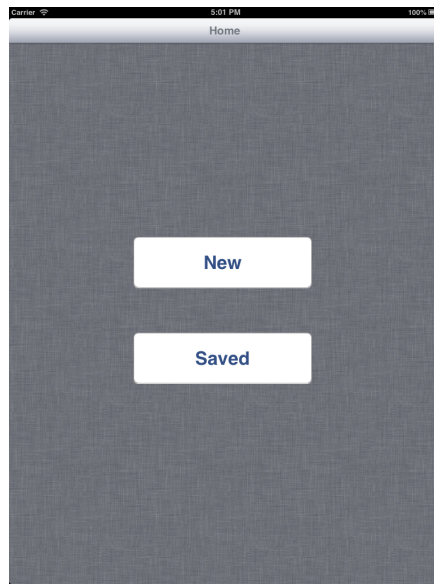


Figure 5 - Home Screen

This screen is the first screen users will see when they open our application. There will be two options: start a new recording or view a saved one. Tapping the “New” button will take you to the RECORD screen. Tapping the “Saved” button will take you to the LIBRARY.

## RECORD

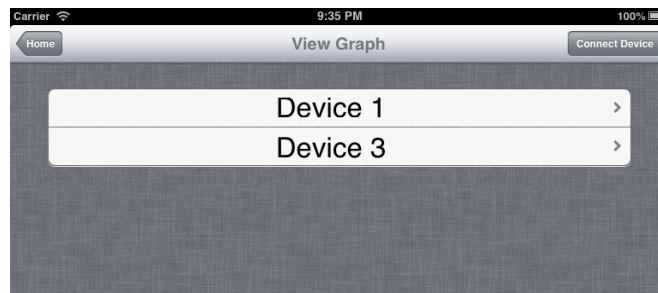


Figure 6 - Record Screen

The RECORD screen will display the connected devices that are sending information to the iPad. Touching a device name in the table will take you to the GRAPH screen. If the user wants to add more devices, he may touch the “Connect Device” button in the top right corner.

## CONNECT

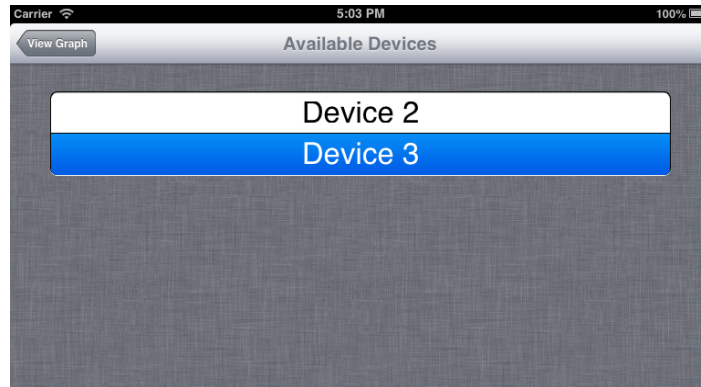


Figure 7 - Connect Screen

A user uses the CONNECT screen to search for devices in the area. If a device is found the user may highlight the device by tapping on its name to add it to the table in the RECORD section. This screen will highlight all connected devices in blue. The user may disconnect from a device by tapping on a name again.

## GRAPH

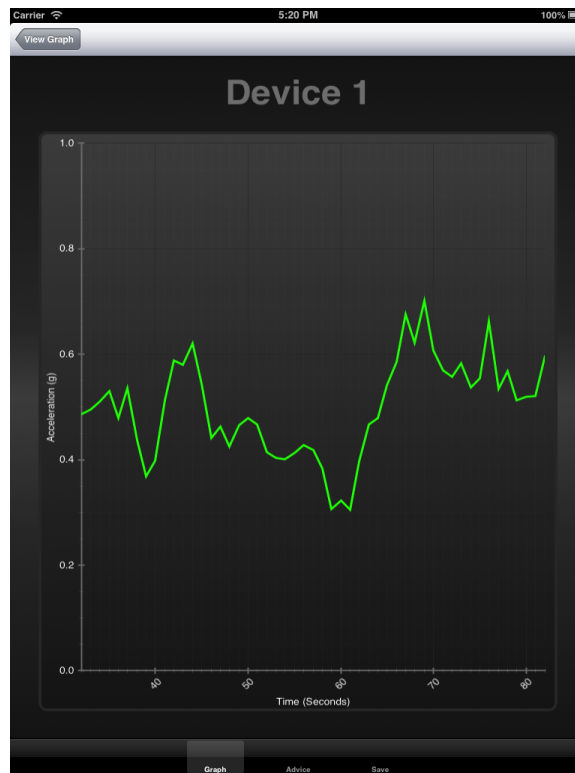


Figure 8 - Graph Screen

After a user chooses a device, the GRAPH screen will show the readings from the chosen device and will display them on the graph with live updates. If the readings go above a designated threshold, the graph will pause 2 seconds after the original peak, giving the user time to interpret the impact and decide what to do next. This screen will also have two other tabs at the bottom: ADVICE and SAVE.

## ADVICE

While we cannot give proper advice for handling a concussion, we leave this tab in our application for future use. We do not have a screenshot at this moment because it is just a placeholder for future development. Some concussion tests implemented today could potentially be placed in this section for the parent or coach to use for detecting concussions. We hope that this section can be used to better diagnose players based on how hard they have been hit.

## SAVE

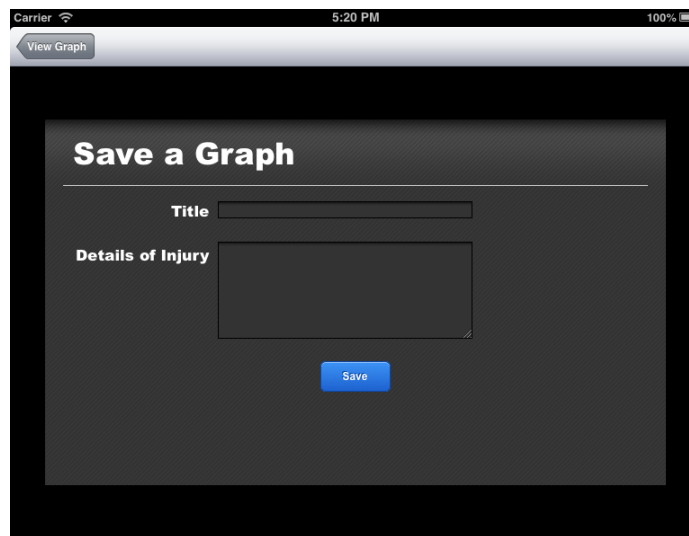


Figure 9 - Save Screen

Once a reading from the device triggers a pause in the graph, the user can click the SAVE tab to write notes on the play and save it in the library. The user will be prompted to give a name for the player and a description of what happened and some notes on how the player was feeling after. We hope that coaches and parents will use this utility to share information on the impact event with a doctor.

## 5. Design Rationale

The main challenge we faced as a design group is that research being done concerning collision-induced head injuries is in an elementary stage, and there isn't conclusive evidence that standardizes concussion diagnosis. With this in mind, our design is intended to help identify potential severe impacts based on current medical research, while providing easy customization that will allow our product to adapt to this changing medical landscape. Doubly, our product is another means for gathering data about head injuries to further medical and safety research. Ethically, we have to understand that our design will likely be obsolete (and potentially dangerous) in months if we construct the system without accounting and providing for new data. The recent increase in head injury studies informed our decision to allow numerous customization options in the application, including, but not limited to, establishing a default "factory" threshold based on our medical research, while allowing the user to define a threshold for what constitutes a 'severe impact' using data from studies conducted post-production of our product.

With this goal for our product, naturally the next challenge is product reliability. Since this data will be used to further improve understanding of head injuries, and ultimately to provide a standard by which impacts to the head can be diagnosed and regulated, it follows that our product must be reliable and trustworthy. Device calibration, electronic failure, poor mobile application design, and other factors could lead to skewed results and data misrepresentation. While pitfalls such as these would render our product useless for medical studies, the real-time consequences are much more severe. Bad data could cause misdiagnosis of a player or wrongly identify who experienced the impact. Even worse, the failure of our system might not identify, or alert the user to, severe impacts. Dysfunctional technology is extremely dangerous in medical situations such as ours, and could potentially put the user in a position of unnecessary harm, from which we are striving to protect the user. These potential dangers mean that we must subject our design to rigorous testing, under a variety of conditions, to better identify and evaluate potential problems in our design. Consequently, we must maintain a transparent testing and design process that will serve as evidence for product reliability.

In addition to the required technological testing, preparing our product for completion (*ready for manufacturing/market*) involves designing housing for the device

(which would be inserted into a helmet) that achieves both form and function. Safety risks involving the physical design of the device include the danger that the design interferes with the construction of the helmet and actually reduces its effectiveness as a concussion-prevention apparatus. Due to budget and time restraints, our group has confined our project scope to include a proposed physical product design, while not creating a manufacturable physical design prototype. Chief among our reasons for this decision is the restraint that our device be non-intrusive to the helmet apparatus, which could potentially require skills and a budget outside of our current capabilities. Thusly, our main goal was to use theory, experimental results, and product constraints to complete a physical design plan that will be ready for prototype manufacturing.

Even with all the precautions we will be taking to ensure that our product functions according to our design, our product must remain a user-oriented system. We are providing a tool that provides the user with information, but our product is intended to aid the user in deciding how to take further action (and if action should be taken at all) or not to take action. Our system is ultimately a detection and alert system, and as far as usability is concerned we have to decide if any restrictions should be placed dictating what persons should be allowed to use our product. Due to the state of research in the field of head injuries, our product will be unsuccessful if we do not allow a significant range of customizability, which unfortunately results in a larger probability for user error. If our system were put into production, we would have to provide instructions on how to use our system, supporting easy usability while minimizing the risk of misuse. After all, our goal is to be able to make this tool available to as many people as possible, encouraging information sharing and allowing the user(s) to utilize statistical data to best decide what action to take. The fact is there is no more risk to using our product than not using it; there are only benefits. We have to stress that our system is not a stand-alone concussion management system, and is instead intended as another tool available to help identify and manage severe head impacts. At worst, our product might signal a severe impact that did no significant damage to the user, resulting in a (recommended) further assessment of that user's physical state. We want the users of our system to become increasingly aware of severe impacts, as heightened visibility of these impacts will help to limit the damage of the impacts and actually create a safer environment for the user.



## **6. Testing**

Section 6 outlines testing strategies for our design, explicitly defining success and failure within the scope of our project. Each test strategy reflects the requirements for our design as specified in Sections 1.5 and 1.6. Test implementation and results are discussed, as well as changes we would have made to both our design and experiments.

### **6.1 Test Setup & Procedure**

Our test strategy consisted of four stages of testing. Moving in a logical procession of increasing complexity allowed us to better assess where potential problems may arise. This strategy will also facilitate troubleshooting by helping us to identify what components need to be fixed, rather than running subsequent tests to pinpoint the problem.

## Stage 1 - Device Optimization

Our goal during this phase of testing was to ensure that the device was working properly, by observing the readings of the accelerometer as outputted by the microcontroller. The device test set up is shown in Figure 11.

	Stage 1
Test Strategy	Our device will be replicated on a breadboard, which will be secured to a helmet. Instead of the microcontroller transmitting wirelessly, we will connect it to a computer. We will then apply different amounts of force to the helmet, and observe the results.
Success	Data from the accelerometer reflects the amount of force applied during testing trials.
Failure	Data from the accelerometer does not accurately reflect amount of force applied during trials and/or is unable to accurately measure above our required threshold
Troubleshooting	Circuit connections will be reviewed and retested. If this results in another failure, different accelerometers and configurations will need to be incorporated into design.

Figure 10 - Test Stage 1

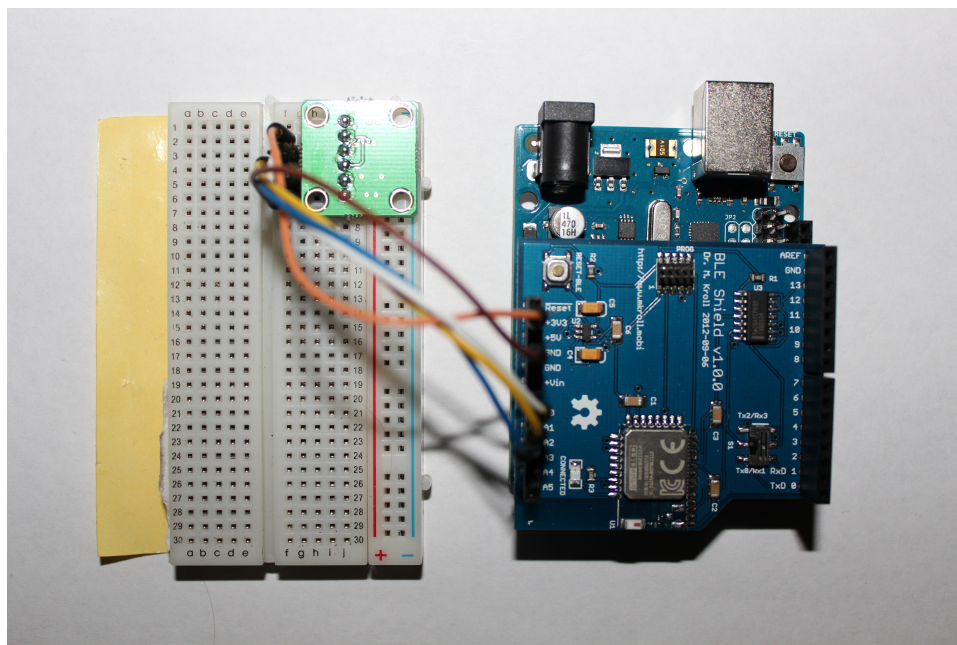


Figure 11 - Device Test Setup

## Stage 2 - Data Transmission Within System

Our goal during this testing phase was to examine the wireless interaction of the device and mobile application. This stage actually had two criteria for success, as the device had to accurately transmit the data using Bluetooth, and the mobile application had to accurately interpret the data it received from the device.

	Stage 2
Test Strategy	Our device will be replicated on a breadboard, which will be secured to a helmet. During this stage, the Bluetooth radio will be enabled. Different forces will be applied to the helmet, and the results will be viewed on the mobile application, instead of a computer.
Success	Data is transmitted to and accurately interpreted on the mobile application.
Failure	Device does not pair with the mobile application and/or information sent by the device is not interpreted correctly by the mobile application.
Troubleshooting	Mobile application programming will be reviewed and implemented differently. Bluetooth radio connections will be examined and reviewed.

Figure 12 - Test Stage 2

### Stage 3 - Data Transmission With Multiple Devices

After the device-application connection was optimized, our team tested if the system worked when connecting multiple devices to one application. The goal in this stage was to ensure that each device would be identified by the application, and information from each device would be interpreted uniquely.

	Stage 3
Test Strategy	The system constructed in Stage 2 will be implemented on (3) helmets. Different forces will be applied to each helmet separately, with the results viewed on the mobile application. After success in this test is confirmed, forces will be applied to all the helmets simultaneously, with the results observed on the mobile application.
Success	Data from each device is interpreted separately and accurately on the mobile application
Failure	Multiple devices are not able to pair with the mobile application and/or the application is unable to distinguish or interpret data transmitted from different devices.
Troubleshooting	Application programming will be examined. If need be, a system will be put in place describing how to pair multiple to devices to application. This documentation will be made user-friendly and included with system.

Figure 13 - Test Stage 3

#### Stage 4 - Range of Data Transmission

The goal of this final stage of testing was to determine the range of our system, and if this range could be increased by allowing data to “hop” from one device to another, and finally to the mobile application.

	Stage 4
Test Strategy	The system constructed in Stage 2 will be implemented on (2) helmets. The first test in this stage is to establish maximum range of one device, by applying a defined amount of force to one helmet while moving the mobile application further away from the device. During the second part of testing, we will introduce the second helmet at the distance from the mobile application defined in the first test. This helmet will remain stationary. A defined amount of force will continue to be applied to the first helmet, while this helmet is moved further away from the stationary helmet, to determine if the transmission can reach the mobile application by being transmitted between the two devices.
Success	The introduction of the second device will increase the range of the system.
Failure	The introduction of the second device does not increase the range of the system.
Troubleshooting	Device programming will be examined to identify if transmissions between devices before transmission to the mobile application can be accomplished without compromising unique device signatures.

Figure 14 - Test Stage 4

## 6.2 Results

Testing Stage 1 required two separate design verifications to ensure that our device was working properly. The first verification that needed to be done was calibrating the accelerometer and verifying that normal and accurate values were being recorded and transmitted from the accelerometer to the microcontroller (Fig. 15). The unit of measurement associated with accelerometers is the g-force or g, a unit of acceleration felt as weight. For reference, 1g is the acceleration due to gravity at the earth's surface. Consequently, when our device is at rest it measures exactly 1g. To verify that our accelerometer was working correctly we observed the measurements taken at rest and confirmed that a measurement of 1g was being felt in the direction of the y-axis.

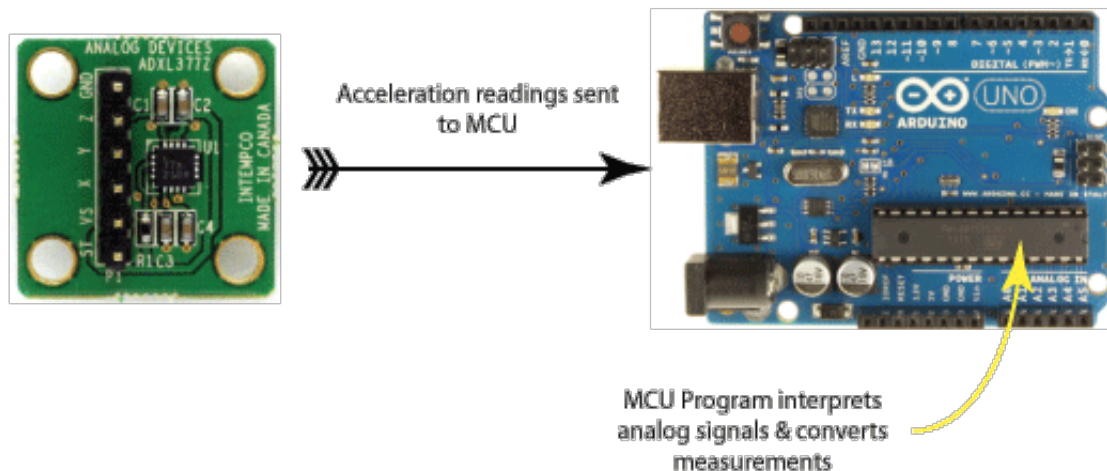


Figure 15 - Accelerometer Calibration & Verification

The second verification that needed to be completed in Testing Stage 1 was to ensure that the microcontroller (specifically the ADC and CPU components) converted the measurements from the accelerometer into magnitude data that could be parsed into data packets. Accelerometer measurements were sent to the microcontroller as analog signals, represented by values from the ADC. The 10-bit ADC of the Arduino Uno allowed for 1024 ( $10 \text{ bits} = 2^{10} = 1024$ ) possible ADC values representing the magnitude of acceleration in a certain direction. A sample reading of the accelerometer over a set period of time could look like this:

350	346	360
351	347	360
347	346	350
346	346	345
450	444	461

Each row corresponds to one acceleration reading, while each column represent the reading from each pin, or the acceleration at the x, y, and z-axes, respectively. These ADC values are what the microcontroller reads, so we needed to manipulate these values to fit our needs. After all the calculations are done we want the information to be presented in *gs*. Before we do this, we need to know some key information about the accelerometer and microcontroller. The data sheet for the ADXL377 accelerometer used for our project specifies the zero-g voltage of the accelerometer (simply, the voltage outputted when the accelerometer reads zero *gs*). This zero-g voltage provides us with a base voltage from which we normalize our voltage readings. The specification changes depending on the source voltage ( $V_s$ ). In our case the source voltage is 3.3 V, and the zero-g voltage is as follows:

$$\text{ADXL377 zero-g Voltage} = 1.65 \text{ V (When } V_s = 3.3 \text{ V)}$$

The next crucial piece of information we find in the data sheet is the sensitivity of the accelerometer at our respective source voltage. This piece of data tells us how much voltage represents 1 *g*.

$$\text{ADXL377 Sensitivity (@ } V_s = 3.3 \text{ V)} = 7.15 \text{ mV/g} = .00715 \text{ V/g}$$

To correctly calculate magnitude, we still need a couple of pieces of information about the microcontroller. The first, which we covered earlier, is the resolution of the ADC. The ADC of the Arduino Uno has 10-bit resolution. Secondly, we need to know the reference voltage of the microcontroller ( $V_{ref}$ ). From the Arduino Uno specification

sheet we know that  $V_{ref} = 5V$ , but after doing physical tests we concluded that our particular Arduino Uno had a reference voltage of 4.9 V. We tested 3 separate Arduinos and they all differed slightly, but 4.9 V was the average voltage we measured consistently.

Now that we have all the relevant specifications, we can proceed to calculate the magnitude of acceleration measured by our device. First we convert the ADC values back to voltage readings:

$$\text{Measured Voltage} = \frac{\text{ADC Value} \times V_{ref}}{\# \text{ of ADC Values}} \text{ Volts}$$

or with our specifications

$$\text{Measured Voltage} = \frac{\text{ADC Value} \times 4.9}{1024} \text{ Volts}$$

After we have the measured voltage we subtract from it the zero-g voltage, and calculate the magnitude in gs by normalizing the resulting voltage with respect to the sensitivity:

$$\text{Magnitude} = \frac{\text{Measured Voltage} - \text{Zero-g Voltage}}{\text{Sensitivity}} \text{ gs}$$

or with our specifications

$$\text{Magnitude} = \frac{\text{Measured Voltage} - 1.65}{.00715} \text{ gs}$$

Now we have the information we want by having the magnitude represented in gs. Our code repeats this process for each axis so we have three magnitude readings for each acceleration event. This gives us four pieces of data that we must send as a single packet to



the mobile application (or computer in Testing Stage 1). Instead of sending these four data pieces, however, we chose to calculate the magnitude of the 3D vector represented by the readings on all three axes. By adding one more equation to our microcontroller program

$$3D \text{ Vector Magnitude} = \sqrt{(\text{magnitude } x)^2 + (\text{magnitude } y)^2 + (\text{magnitude } z)^2} \text{ gs}$$

we are able to minimize our data packets to include only two pieces of information; 3D vector magnitude, and time of acceleration event. This is significant because it reduces the amount of memory usage by 50% and the data packets can be easily parsed into two separate pieces that represent Cartesian coordinates, making it simple to interpret and graph using the mobile application.

Stage 1 of testing was successful, as both the microcontroller and accelerometer were verified to be working properly, and adjustments were made to account for memory and power usage. The last step of this stage was to prepare the data to be transmitted wirelessly. This task consisted of creating data packets, as discussed in the previous paragraph, and placing these data packets in a buffer in the Bluetooth module on the Bluetooth LE shield. This step is illustrated in Figure 16 , and provided our team with a natural transition to Testing Stage 2.

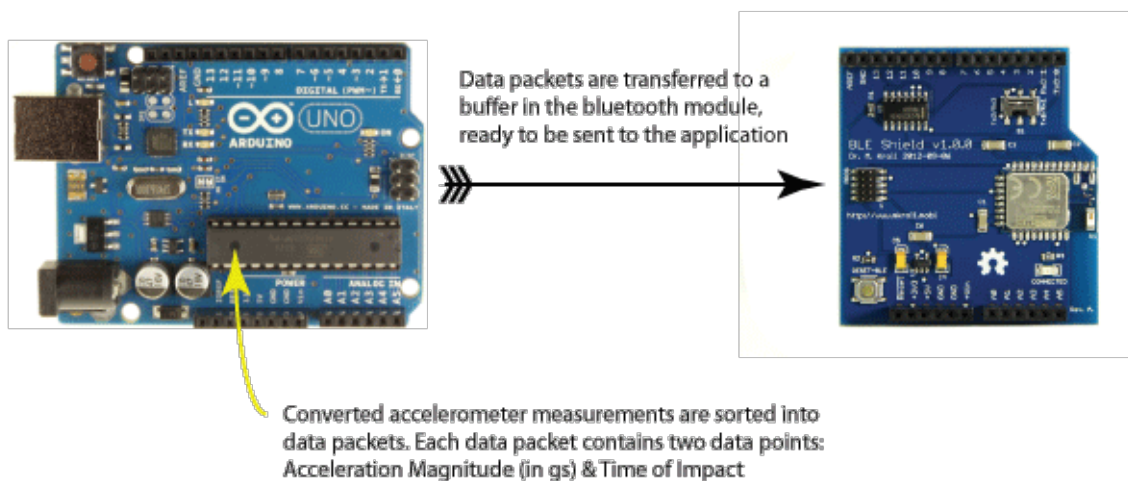


Figure 16 - Data Calculation & Compartmentalization

Stage 2 of testing was designed to test wireless data transmission between a single device and the mobile application. The first step was to get the mobile application to recognize and pair with the device using the Bluetooth 4.0 protocol. After completing this step and making sure that the device itself was discoverable, our team moved on to the task of transmitting the data packets from the Bluetooth module buffer to the mobile application. Lastly, the mobile application was optimized to extract the coordinates from the data packets (time stamp & vector magnitude) and graph these coordinates on a plot that updated in real time.

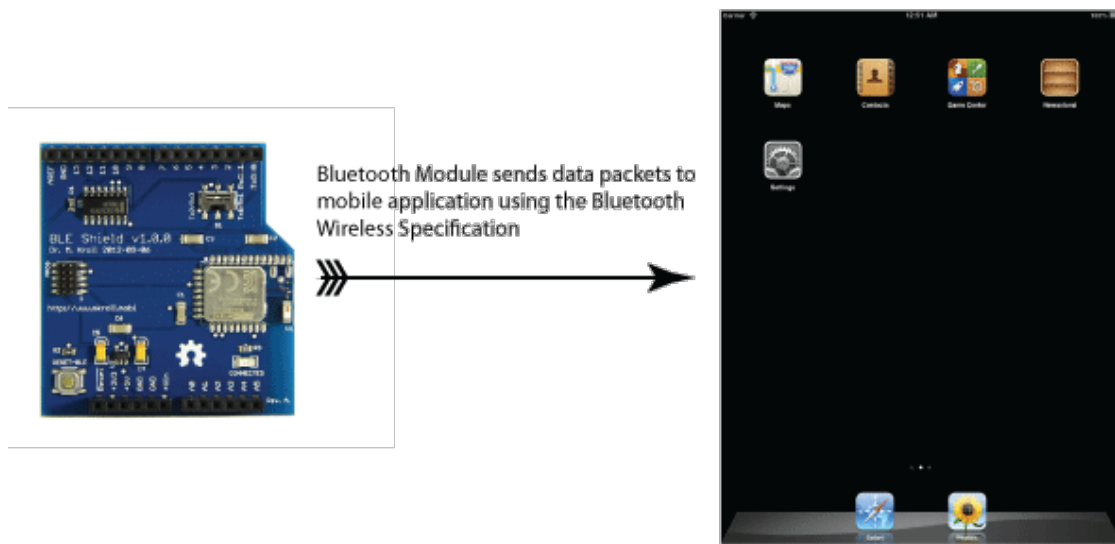


Figure 17 - Data Transmission

## 7. Wireless Sensor Network

### 7.1 Overview

A wireless sensor network allows multiple devices to connect to each other to send information to a database or collection center (in our case, the mobile application). By allowing each device to communicate to each other, a wireless sensor network extends the range of a network while making sure that data from each device is kept unique, and relayed to the database by routing through multiple devices.

### 7.2 Network Architecture Proposal

Our proposed network works by assigning priorities to devices based on their proximity to the mobile phone or tablet. The mobile phone has ***Priority Level 0*** and any device that is able to connect directly to the mobile phone gains ***Priority Level 1***. If another device is out of range of the mobile phone, but within range of a device with ***Priority Level 1***, then the device connects to the ***Priority Level 1*** device, transfers data to that device, and gains ***Priority Level 2***. Likewise, a device unable to connect to a ***Priority Level 1*** device connects with a ***Priority Level 2*** device and so on. Priority levels are based on how many data transfers need to happen before data arrives at the mobile application, and can extend dynamically to fit the needs of the network. Priority levels will change depending on device movement in relation to the mobile application. For example, if a device with ***Priority Level 2*** moves into range of the mobile application, it gains ***Priority Level 1***, and can send data packets directly to the mobile application. The opposite is also true, as the device moves out of range of the mobile application it will lose its ***Priority Level 1*** status and have to reconnect to the highest priority device within range. This process of connecting to devices with a higher priority extends the range of the network, especially for low energy Bluetooth-enabled devices that typically have a range of 50m - 100m.

Connection Sequence:

1. Device attempts to connect to mobile phone/tablet
2. If device **IS ABLE TO CONNECT**: Device gains priority Level 1 & sends stored information to mobile application.
3. If device **ISN'T ABLE TO CONNECT**: Device connects to the highest priority device within range.
4. Device transfers data to higher priority device & gains priority level one lower than that device.

Fig. 18 is an example of implementing this sensor network. Two devices in the figure are in connection range of the mobile application, so they both send data to the mobile application and gain Priority Level 1. A third device is out of range of the mobile application, but in range of a Priority 1 device. The third device connects to the Priority 1 device, and gains Priority Level 2. It then transfers all of its data packets to the Priority 1 device, which then transfers these data packets to the mobile application.

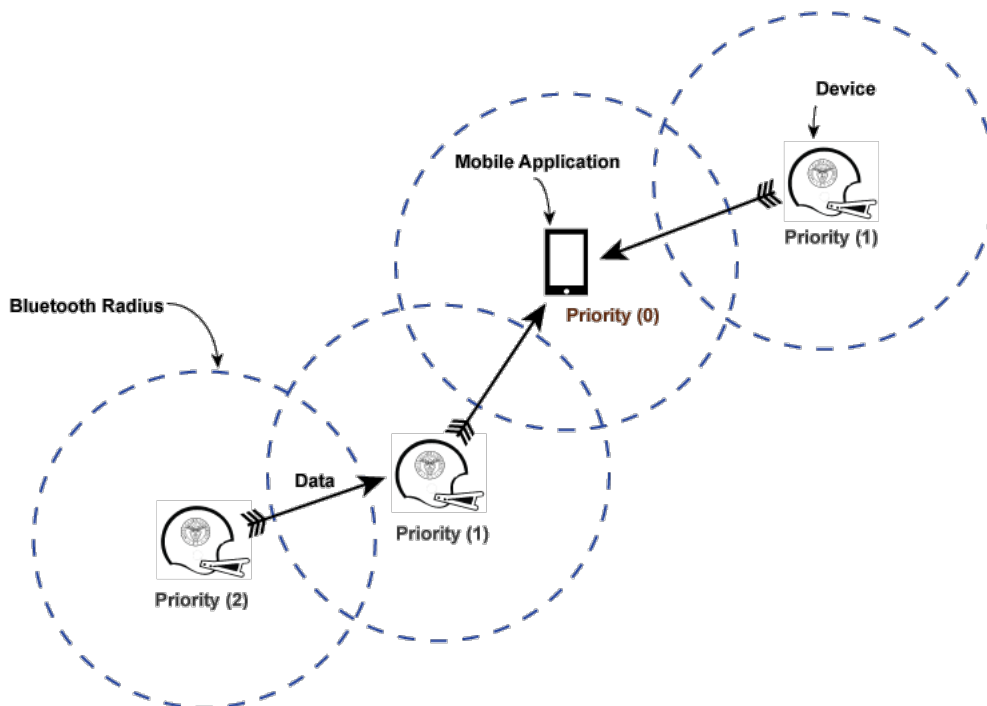


Figure 18 - Wireless Sensor Network

To better illustrate this idea of priority levels and data transfer, we introduce a fourth helmet (outfitted with our device) to the system (Fig. 19). Since this device resides outside of the wireless range of our mobile application, it needs to connect to another device to send data. Our new device establishes a connection with a Priority 1 device, to which it transfers all its stored data packets. Finally, the Priority 1 device relays all this stored data to the mobile application.

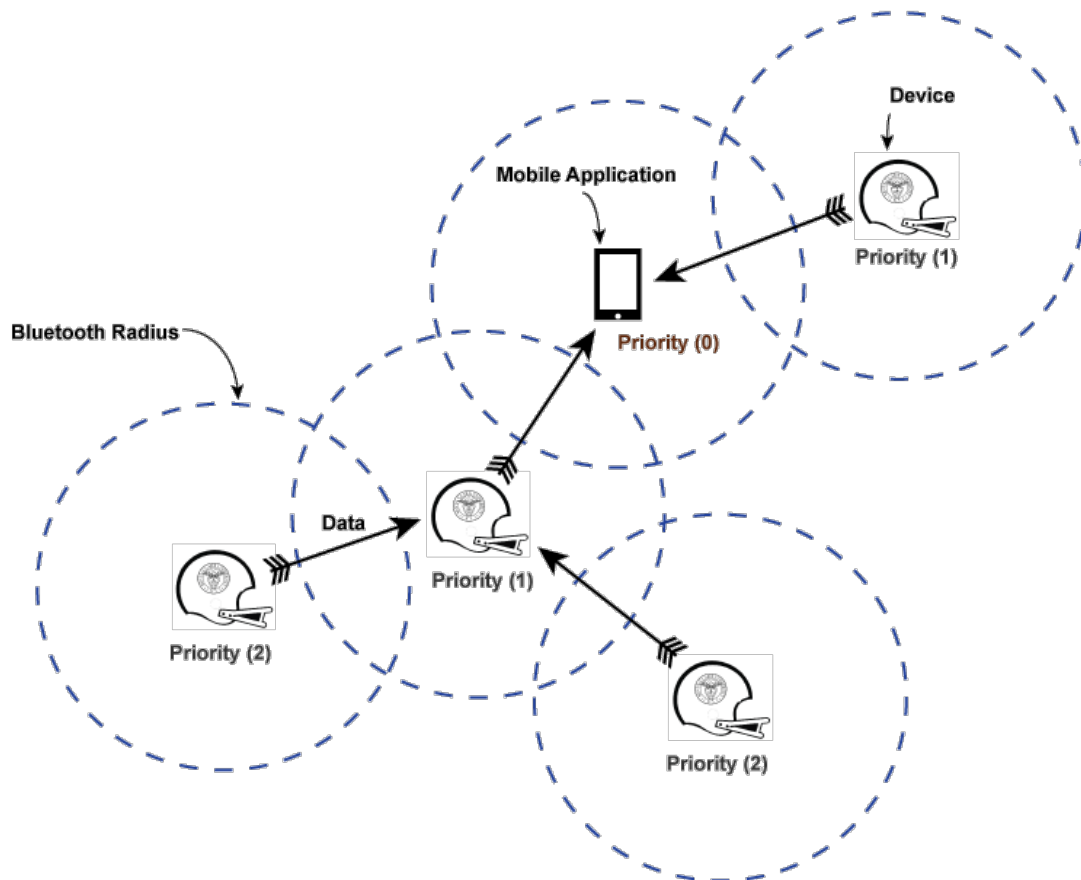


Figure 19 - Wireless Sensor Network Data Relay

By implementing a Wireless Sensor Network like this one, we can extend the range of the system without implementing a different wireless protocol such as ANT, and compromising the low-energy usage of Bluetooth 4.0 that is critical to our device functions. (Blum) The dynamic range of the network also helps to reduce unnecessary noise caused by mobile device interference. Since the network will only extend as far as a single device, the network will usually encompass only the playing field and will cease to

extend into spectator areas, such as bleachers. Possible complications may arise if one device is consistently the only device within range of the mobile application, as this may lead to a data bottleneck and heightened power usage. However, this is a worst-case scenario that should very rarely occur due to the large amount of devices in the system, and constant motion inherent to sports such as football and hockey.

## 8. Conclusion

In completing the first two test stages of our project we were able to prove that the principle of having a device connecting and transmitting data wirelessly to a mobile application can be accomplished using Bluetooth 4.0 and iOS devices. While we didn't test multiple devices connecting to a single mobile application, we were able to compile a large amount of research on wireless sensor networks, and propose a wireless sensor network architecture of our own. Improvements and modifications of existing wireless protocols, such as Wi-Fi Direct or ANT, could encourage us to replace Bluetooth 4.0 in our system, but currently this technology provides all the key features and benefits that our system needs.

As knowledge about concussions, head injuries, and their relation to impacts improves, so will our system. Future iterations of our project should look to implement the wireless sensor network, while also aiming to incorporate current medical knowledge, such as sideline concussion strategies used by governing sports bodies, such as the NFL. Tests such as the Standard Concussion Assessment Tool (SCAT) are used by the NFL, but are subjective and require the user to make many decisions more suited to a medical professional than an amateur coach or athletic trainer. (McLeod, Bay and Lam) Instead, we would like to see the King-Devick test incorporated into our mobile application. This test is used by Boxing and Mixed Martial Arts (MMA) governing bodies, and is an objective test that only requires the user to administer a simple test to an athlete that takes less than a minute. While this test does require baseline testing, this information could be stored alongside an athlete's impact graph in our application, together with other relevant medical information. (Galetta, Barrett and Allen)

Lastly, a physical prototype of the impact device could be designed to be small, flat, and flexible enough to fit many types and styles of helmet, from football to hockey and even bicycle helmets. We hope that the data collected from a system like ours could eventually help to define acceleration levels that would denote an athlete having experienced a concussion, allowing our product to evolve from simple detection to prevention.

## 9. List of References

- Blum, Brian. "Bluetooth low energy, A very Low Power Solution." IAR Systems. 6 June 2013 <iar.com>.
- DeKosky, Steven, Milos Ikonovic and Sam Gandy. "Traumatic Brain Injury — Football, Warfare, and Long-Term Effects." 2010. The New England Journal of Medicine. 6 June` 2013 <<http://www.nejm.org/doi/full/10.1056/NEJMp1007051>>.
- Galetta, K, J Barrett and M Allen. "The King-Devick test as a determinant of head trauma and concussion in boxers and MMA fighters." 76.17 (2011): 1456-1462.
- Higgins, Michael, et al. "Measurement of Impact Acceleration: Mouthpiece Accelerometer Versus Helmet Accelerometer." Journal of Athletic Training 42.1 (2007): 5-10.
- Kamath, Sandeep and Joakim Lindh. "Measuring Bluetooth Low Energy Power Consumption." Texas Instruments. 6 June 2013 <ti.com>.
- Khissibi, Sabri and Hanen Idoudi. "Presentation and analysis of a new technology for low - power wireless sensor network ." 2013. The Society of Digital Information and Wireless Communications. June 2013 <[sdiwc.net/digital-library/web-admin/upload-pdf/00000591.pdf](http://sdiwc.net/digital-library/web-admin/upload-pdf/00000591.pdf)>.
- McLeod, Tamara, et al. "Representative Baseline Values on the Sport Concussion Assessment Tool 2 (SCAT2) in Adolescent Athletes Vary by Gender, Grade, and Concussion History ." The American Journal of Sports Medicine (2012).
- Radiological Society of North America. "Single concussion may cause lasting brain damage." 12 March 2013. ScienceDaily. 6 June 2013 <<http://www.sciencedaily.com/releases/2013/03/130312092642.htm>>.
- Wein, Harrison. "A Bang to the Brain." May 2013. NIH New in Health. 6 June 2013 <<http://newsinhealth.nih.gov/issue/may2013/feature1>>.
- X2 Biosystems. Detection of Head Accelerations with an Instrumented Mouthguard and Virtual Projection Sensors.  
— . X2BIOSYSTEMS Head Impact Monitoring System.  
— . X2's end-to-end system provides real-time data for head impacts. 6 June 2013 <[http://www.x2biosystems.com/x2\\_technology/](http://www.x2biosystems.com/x2_technology/)>.
- Zhang, Liying, King Yang and Albert King. "A Proposed Injury Threshold for Mild Traumatic Brain Injury." J Biomech Eng 162.2 (2004): 226-236.



## 10. Appendix

### 10.1 Source Code (Arduino)

```
/*  
  
ADXL377 Specs  
  
zero-g voltage = 1.65 V (When Vs = 3.3V)  
sensitivity (@ Vs=3v) = 6.5 mV/g = .0065 V/g  
sensitivity (@ Vs=3.3v) = 7.15 mV/g = .00715 V/g  
  
Arduino Specs  
  
ADC: 10-bit resolution (1024 values)  
Vref == 5V (readings may differ slightly)  
  
*/  
/*  
ADXL377  
  
Reads an Analog Devices ADXL377 accelerometer and communicates the  
acceleration to a computer.  
  
Programmer: Shawno Auwae  
  
Components: Arduino Uno & ADXL377 Accelerometer Evaluation Board  
The circuit:  
GND pin: Ground  
3V3 pin: Vs (3.3 v)  
analog 0: x-axis  
analog 1: y-axis  
analog 2: z-axis  
*/  
// these constants describe the pins.  
  
const int xpin = A0;           // x-axis of the accelerometer  
const int ypin = A1;           // y-axis  
const int zpin = A2;           // z-axis  
const float zerog = 1.65;      // ADXL377 zero-g voltage  
const float sensitivity = .00715; // ADXL377 sensitivity  
const float Vref = 4.9;        // Arduino ADC Reference Voltage  
  
char readings[10];             // Array to hold accelerometer  
string readings  
  
void setup()  
{  
  // initialize the serial communications:  
  Serial.begin(9600);  
}
```

```

}

void loop()
{
    float adc_x, adc_y, adc_z;
    float volts_x, volts_y, volts_z;
    float x, y, z, magnitude;

    adc_x=analogRead(xpin); // read acceleration value from x pin
    adc_y=analogRead(ypin); // read acceleration value from y pin
    adc_z=analogRead(zpin); // read acceleration value from z pin

    Serial.print(adc_x); // print raw analog values
    Serial.print("\t");
    Serial.print(adc_y);
    Serial.print("\t");
    Serial.print(adc_z);
    Serial.print("\t");

    int test = 5*5;
    Serial.print(test);
    Serial.print("\t");
    Serial.print("end adc");
    Serial.print("\t");

    volts_x=(adc_x*Vref)/1024; // normalize voltage reading
    volts_y=(adc_y*Vref)/1024;
    volts_z=(adc_z*Vref)/1024;

    Serial.print(volts_x);
    Serial.print("\t");
    Serial.print(volts_y);
    Serial.print("\t");
    Serial.print(volts_z);
    Serial.print("\t");
    Serial.print("end volts");
    Serial.print("\t");

    x=(volts_x-zerog)/sensitivity; // convert readings to g's
    y=(volts_y-zerog)/sensitivity;
    z=(volts_z-zerog)/sensitivity;

    Serial.print(x);
    Serial.print("\t");
    Serial.print(y);
    Serial.print("\t");
    Serial.print(z);
    Serial.print("\t");
    Serial.print("end x y z");
    Serial.print("\t");

    // calculate magnitude of 3D vector

    magnitude = sqrt(sq(x)+sq(y)+sq(z));

```

```

Serial.print(magnitude);    // print 3D Vector magnitude
Serial.print("\t");
Serial.println();

delay(100);

}

```

## 10.2 Source Code (Apple iOS)

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.Storyboard.XIB"
version="2.0" toolsVersion="3084" systemVersion="12D78"
targetRuntime="iOS.CocoaTouch.iPad" propertyAccessControl="none"
useAutolayout="YES" initialViewController="au5-Kt-z3R">
  <dependencies>
    <plugIn
identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin"
version="2083"/>
  </dependencies>
  <scenes>
    <!--View Controller - Home-->
    <scene sceneID="4">
      <objects>
        <viewController id="2" customClass="ViewController"
sceneMemberID="viewController">
          <view key="view" contentMode="scaleToFill" id="5">
            <rect key="frame" x="0.0" y="64" width="768"
height="960"/>
            <autoresizingMask key="autoresizingMask"
widthSizable="YES" heightSizable="YES"/>
            <subviews>
              <button opaque="NO" clipsSubviews="YES"
contentMode="center" contentHorizontalAlignment="center"
contentVerticalAlignment="center" buttonType="roundedRect"
adjustsImageWhenDisabled="NO" lineBreakMode="middleTruncation"
translatesAutoresizingMaskIntoConstraints="NO" id="cQj-Mk-UbI">
                <color key="backgroundColor"
cocoaTouchSystemColor="scrollViewTexturedBackgroundColor"/>
                <constraints>
                  <constraint firstAttribute="width"
constant="316" id="dYd-X9-dLY"/>
                  <constraint firstAttribute="height"
constant="90" id="rTg-np-8QA"/>
                </constraints>
                <fontDescription key="fontDescription"
name="Helvetica-Bold" family="Helvetica" pointSize="33"/>
                <state key="normal" title="New">
                  <color key="titleColor"
red="0.19607843459999999" green="0.30980393290000002"
blue="0.52156865600000002" alpha="1" colorSpace="calibratedRGB"/>
                  <color key="titleShadowColor"
white="0.5" alpha="1" colorSpace="calibratedWhite"/>
                </state>

```

```

        <connections>
            <segue destination="WaZ-1L-pwz"
kind="push" id="OsK-ur-Awu"/>
        </connections>
    </button>
    <button opaque="NO"
contentMode="scaleToFill" contentHorizontalAlignment="center"
contentVerticalAlignment="center" buttonType="roundedRect"
lineBreakMode="middleTruncation"
translatesAutoresizingMaskIntoConstraints="NO" id="BpR-gd-vJ7">
        <constraints>
            <constraint firstAttribute="height"
constant="90" id="AIR-Cv-FcS"/>
        </constraints>
        <fontDescription key="fontDescription"
type="boldSystem" pointSize="33"/>
        <state key="normal" title="Saved">
            <color key="titleColor"
red="0.19607843459999999" green="0.30980393290000002"
blue="0.52156865600000002" alpha="1" colorSpace="calibratedRGB"/>
            <color key="titleShadowColor"
white="0.5" alpha="1" colorSpace="calibratedWhite"/>
        </state>
        <state key="highlighted">
            <color key="titleColor" white="1"
alpha="1" colorSpace="calibratedWhite"/>
        </state>
    </button>
</subviews>
<color key="backgroundColor"
cocoaTouchSystemColor="scrollViewTexturedBackgroundColor"/>
<constraints>
    <constraint firstItem="cQj-Mk-UbI"
firstAttribute="trailing" secondItem="BpR-gd-vJ7"
secondAttribute="trailing" type="default" id="CNP-Pq-j9f"/>
    <constraint firstItem="cQj-Mk-UbI"
firstAttribute="leading" secondItem="5" secondAttribute="leading"
constant="221" id="Ctb-9k-aW0"/>
    <constraint firstAttribute="bottom"
secondItem="BpR-gd-vJ7" secondAttribute="bottom" constant="356"
id="mgS-p0-hXy"/>
    <constraint firstItem="cQj-Mk-UbI"
firstAttribute="leading" secondItem="BpR-gd-vJ7"
secondAttribute="leading" type="default" id="q8c-lb-4j3"/>
    <constraint firstItem="cQj-Mk-UbI"
firstAttribute="top" secondItem="5" secondAttribute="top"
constant="344" id="tNs-X9-0r5"/>
</constraints>
</view>
<navigationItem key="navigationItem" title="Home"
id="e1T-17-MqD"/>
</viewController>
<placeholder placeholderIdentifier="IBFirstResponder"
id="3" sceneMemberID="firstResponder"/>
</objects>
<point key="canvasLocation" x="1023" y="-90"/>
</scene>
<!--Table View Controller - View Graph-->

```

```

<scene sceneID="5h2-iA-YuK">
  <objects>
    <tableViewController id="WaZ-1L-pwz"
sceneMemberID="viewController">
      <tableView key="view" clipsSubviews="YES"
contentMode="scaleToFill" alwaysBounceVertical="YES" dataMode="static"
style="grouped" separatorStyle="singleLineEtched" rowHeight="44"
sectionHeaderHeight="10" sectionFooterHeight="10" id="2NN-zM-tPw">
        <rect key="frame" x="0.0" y="64" width="768"
height="960"/>
          <autoresizingMask key="autoresizingMask"
widthSizable="YES" heightSizable="YES"/>
          <color key="backgroundColor"
cocoaTouchSystemColor="scrollViewTexturedBackgroundColor"/>
          <color key="separatorColor"
cocoaTouchSystemColor="scrollViewTexturedBackgroundColor"/>
          <tableView key="tableViewFooterView"
clipsSubviews="YES" contentMode="scaleToFill"
alwaysBounceVertical="YES" dataMode="prototypes" style="plain"
rowHeight="44" sectionHeaderHeight="22" sectionFooterHeight="22"
id="mtB-Ie-jPB">
            <rect key="frame" x="0.0" y="170"
width="768" height="0.0"/>
              <autoresizingMask key="autoresizingMask"
widthSizable="YES" heightSizable="YES"/>
              <color key="backgroundColor" white="1"
alpha="1" colorSpace="calibratedWhite"/>
            </tableView>
          <sections>
            <tableViewSection id="fGg-52-j5G">
              <cells>
                <tableViewCell
contentMode="scaleToFill" selectionStyle="blue"
accessoryType="disclosureIndicator" hidesAccessoryWhenEditing="NO"
indentationLevel="1" indentationWidth="0.0" id="19Q-7J-fZq"
userLabel="Device 1">
                  <rect key="frame" x="0.0"
y="30" width="768" height="45"/>
                    <autoresizingMask
key="autoresizingMask"/>
                    <view key="contentView"
opaque="NO" clipsSubviews="YES" multipleTouchEnabled="YES"
contentMode="center">
                      <rect key="frame" x="45"
y="1" width="658" height="43"/>
                        <autoresizingMask
key="autoresizingMask"/>
                        <subviews>
                          <label opaque="NO"
clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
text="Device 1" textAlignment="center" lineBreakMode="tailTruncation"
baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="yUj-FX-rZG">
                            <constraints>
                              <constraint
firstAttribute="height" constant="23" id="YC4-4g-Cpz"/>
                              <constraint
firstAttribute="width" constant="203" id="q9X-d2-eb5"/>

```

```

        </constraints>
        <fontDescription
key="fontDescription" type="system" pointSize="33"/>
        <color
key="textColor" cocoaTouchSystemColor="darkTextColor"/>
        <color
key="highlightedColor" white="1" alpha="1"
colorSpace="calibratedWhite"/>
        </label>
    </subviews>
    <color
key="backgroundColor" white="0.0" alpha="0.0"
colorSpace="calibratedWhite"/>
    </view>
    <constraints>
        <constraint firstItem="yUj-
FX-rZG" firstAttribute="centerY" secondItem="19Q-7J-fZq"
secondAttribute="centerY" type="default" id="ecn-bT-Trn"/>
        <constraint firstItem="yUj-
FX-rZG" firstAttribute="leading" secondItem="19Q-7J-fZq"
secondAttribute="leading" constant="273" id="pIW-J7-7L3"/>
    </constraints>
    <connections>
        <segue destination="esi-SO-
1wL" kind="push" id="Zpf-jb-aes"/>
    </connections>
</tableViewCell>
<tableViewCell
contentMode="scaleToFill" selectionStyle="blue"
accessoryType="disclosureIndicator" hidesAccessoryWhenEditing="NO"
indentationLevel="1" indentationWidth="0.0" id="ZFr-Sl-vhJ"
userLabel="Device 1">
    <rect key="frame" x="0.0"
y="75" width="768" height="45"/>
    <autoresizingMask
key="autoresizingMask"/>
    <view key="contentView"
opaque="NO" clipsSubviews="YES" multipleTouchEnabled="YES"
contentMode="center">
        <rect key="frame" x="45"
y="0.0" width="658" height="43"/>
        <autoresizingMask
key="autoresizingMask"/>
        <subviews>
            <label opaque="NO"
clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
text="Device 3".textAlignment="center" lineBreakMode="tailTruncation"
baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="56E-2T-FPI">
                <constraints>
                    <constraint
firstAttribute="width" constant="203" id="r5v-q2-Z7C"/>
                </constraints>
            <fontDescription
key="fontDescription" type="system" pointSize="33"/>
            <color
key="textColor" cocoaTouchSystemColor="darkTextColor"/>
            <color

```

```

key="highlightedColor" white="1" alpha="1"
colorSpace="calibratedWhite"/>
        </label>
    </subviews>
    <color
key="backgroundColor" white="0.0" alpha="0.0"
colorSpace="calibratedWhite"/>
        </view>
    <constraints>
        <constraint firstItem="56E-
2T-FPI" firstAttribute="leading" secondItem="ZFr-Sl-vhJ"
secondAttribute="leading" constant="273" id="Ijt-BL-yBv"/>
        <constraint firstItem="56E-
2T-FPI" firstAttribute="top" secondItem="ZFr-Sl-vhJ"
secondAttribute="top" constant="4" id="zxs-0a-3AI"/>
    </constraints>
</tableViewCell>
</cells>
</tableViewSection>
</sections>
<connections>
    <outlet property="dataSource"
destination="WaZ-1L-pwz" id="KGu-zv-ERb"/>
    <outlet property="delegate"
destination="WaZ-1L-pwz" id="AGF-uM-h9H"/>
</connections>
</tableView>
<navigationItem key="navigationItem" title="View
Graph" id="d5j-lz-Jok">
    <barButtonItem key="rightBarButtonItem"
title="Connect Device" id="TG3-Qi-egE">
        <connections>
            <segue destination="YDf-W8-Skz"
kind="push" id="7PH-r5-npt"/>
        </connections>
    </barButtonItem>
</navigationItem>
</tableViewController>
<placeholder placeholderIdentifier="IBFirstResponder"
id="0as-WI-eBU" userLabel="First Responder"
sceneMemberID="firstResponder"/>
    </objects>
    <point key="canvasLocation" x="2589" y="-1127"/>
</scene>
<!--View Controller - Advice-->
<scene sceneID="ja5-go-w9m">
    <objects>
        <viewController id="g8M-9h-7fz"
sceneMemberID="viewController">
            <view key="view" contentMode="scaleToFill" id="ftI-
nu-Lwc">
                <rect key="frame" x="0.0" y="64" width="768"
height="911"/>
                <autoresizingMask key="autoresizingMask"
flexibleMaxX="YES" flexibleMaxY="YES"/>
                <subviews>
                    <view contentMode="scaleToFill"
translatesAutoresizingMaskIntoConstraints="NO" id="8Uz-6D-yvn">

```

```

        <subviews>
            <label opaque="NO"
clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
text="1. Check for conciousness" lineBreakMode="tailTruncation"
baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="2KA-BK-dC7">
                <constraints>
                    <constraint
firstAttribute="width" constant="494" id="LwY-pF-m0D"/>
                    <constraint
firstAttribute="height" constant="46" id="rjP-29-A3U"/>
                </constraints>
                <fontDescription
key="fontDescription" type="system" pointSize="33"/>
                <color key="textColor"
cocoaTouchSystemColor="darkTextColor"/>
                <nil key="highlightedColor"/>
            </label>
            <label opaque="NO"
clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
text="3. Check the victim's eyes" lineBreakMode="tailTruncation"
baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="NXa-0l-elK">
                <constraints>
                    <constraint
firstAttribute="height" constant="46" id="BZI-RF-Lfv"/>
                </constraints>
                <fontDescription
key="fontDescription" type="system" pointSize="33"/>
                <color key="textColor"
cocoaTouchSystemColor="darkTextColor"/>
                <nil key="highlightedColor"/>
            </label>
            <label opaque="NO"
clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
text="2. Check mental awareness" lineBreakMode="tailTruncation"
baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="s5d-W9-bJH">
                <constraints>
                    <constraint
firstAttribute="height" constant="46" id="tx6-QH-LwL"/>
                </constraints>
                <fontDescription
key="fontDescription" type="system" pointSize="33"/>
                <color key="textColor"
cocoaTouchSystemColor="darkTextColor"/>
                <nil key="highlightedColor"/>
            </label>
            <label opaque="NO"
clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
text="Steps for Detecting a Concussion" textAlignment="center"
lineBreakMode="tailTruncation" baselineAdjustment="alignBaselines"
highlighted="YES" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="t0x-5x-L5v">
                <constraints>
                    <constraint
firstAttribute="height" constant="44" id="XV3-UM-ccZ"/>
                </constraint

```



```

firstAttribute="width" constant="635" id="Y0n-AN-nsW"/>
    </constraints>
    <fontDescription
key="fontDescription" type="system" pointSize="39"/>
    <color key="textColor"
red="0.13333334028720856" green="0.47058826684951782"
blue="0.67843139171600342" alpha="1" colorSpace="deviceRGB"/>
    <nil key="highlightedColor"/>
    <color key="shadowColor"
cocoaTouchSystemColor="lightTextColor"/>
    </label>
</subviews>
<color key="backgroundColor" white="1"
alpha="1" colorSpace="custom" customColorSpace="calibratedWhite"/>
<constraints>
    <constraint firstItem="2KA-BK-dC7"
firstAttribute="leading" secondItem="8Uz-6D-yvn"
secondAttribute="leading" constant="156" id="6sF-Av-R9Y"/>
    <constraint firstItem="2KA-BK-dC7"
firstAttribute="top" secondItem="t0x-5x-L5v" secondAttribute="bottom"
constant="8" symbolic="YES" type="default" id="HTG-zC-R0J"/>
    <constraint firstItem="s5d-W9-bJH"
firstAttribute="top" secondItem="2KA-BK-dC7" secondAttribute="bottom"
constant="8" symbolic="YES" type="default" id="Hus-6D-IGr"/>
    <constraint firstItem="t0x-5x-L5v"
firstAttribute="top" secondItem="8Uz-6D-yvn" secondAttribute="top"
constant="20" symbolic="YES" type="default" id="NGs-M8-RDf"/>
    <constraint firstItem="s5d-W9-bJH"
firstAttribute="trailing" secondItem="NXa-0l-elK"
secondAttribute="trailing" type="default" id="czp-9G-IL8"/>
    <constraint firstItem="s5d-W9-bJH"
firstAttribute="leading" secondItem="2KA-BK-dC7"
secondAttribute="leading" type="default" id="dbn-FQ-7Bs"/>
    <constraint firstItem="s5d-W9-bJH"
firstAttribute="leading" secondItem="NXa-0l-elK"
secondAttribute="leading" type="default" id="mWP-vC-ir9"/>
    <constraint firstItem="NXa-0l-elK"
firstAttribute="top" secondItem="s5d-W9-bJH" secondAttribute="bottom"
constant="8" symbolic="YES" type="default" id="p9c-SJ-hp9"/>
    <constraint firstItem="t0x-5x-L5v"
firstAttribute="centerX" secondItem="8Uz-6D-yvn"
secondAttribute="centerX" type="default" id="xMG-jQ-CqB"/>
    <constraint firstItem="s5d-W9-bJH"
firstAttribute="trailing" secondItem="2KA-BK-dC7"
secondAttribute="trailing" type="default" id="z0l-Sm-hZm"/>
</constraints>
</view>
</subviews>
<color key="backgroundColor" white="1"
alpha="1" colorSpace="custom" customColorSpace="calibratedWhite"/>
<constraints>
    <constraint firstItem="8Uz-6D-yvn"
firstAttribute="bottom" secondItem="ftI-nu-Lwc"
secondAttribute="bottom" type="default" id="SE3-dt-F7Z"/>
    <constraint firstItem="8Uz-6D-yvn"
firstAttribute="top" secondItem="ftI-nu-Lwc" secondAttribute="top"
type="default" id="TdH-aj-yBV"/>
    <constraint firstItem="8Uz-6D-yvn"

```

```

firstAttribute="leading" secondItem="ftI-nu-Lwc"
secondAttribute="leading" type="default" id="cS6-Lt-5lX"/>
    <constraint firstItem="8Uz-6D-yvn"
firstAttribute="trailing" secondItem="ftI-nu-Lwc"
secondAttribute="trailing" type="default" id="ckh-f3-gfB"/>
    </constraints>
</view>
    <tabBarItem key="tabBarItem" title="Advice"
id="tAp-Qz-yBH"/>
    </viewController>
    <placeholder placeholderIdentifier="IBFirstResponder"
id="TyR-CX-wU2" userLabel="First Responder"
sceneMemberID="firstResponder"/>
    </objects>
    <point key="canvasLocation" x="4666" y="33"/>
</scene>
<!--View Controller - Save-->
<scene sceneID="RQm-9L-FYD">
    <objects>
        <viewController id="RyD-mJ-eXT"
sceneMemberID="viewController">
            <view key="view" contentMode="scaleToFill" id="5e5-
6Y-mh9">
                <rect key="frame" x="0.0" y="64" width="768"
height="911"/>
                <autoresizingMask key="autoresizingMask"
flexibleMaxX="YES" flexibleMaxY="YES"/>
                <subviews>
                    <imageView userInteractionEnabled="NO"
contentMode="center" image="Screen Shot 2013-05-08 at 4.40.46 PM.png"
translatesAutoresizingMaskIntoConstraints="NO" id="Gm5-3D-NxN">
                        <constraints>
                            <constraint firstAttribute="height"
constant="525" id="DLn-5R-bxV"/>
                        </constraints>
                    </imageView>
                </subviews>
                <color key="backgroundColor" white="0.0"
alpha="1" colorSpace="calibratedWhite"/>
                <constraints>
                    <constraint firstItem="Gm5-3D-NxN"
firstAttribute="trailing" secondItem="5e5-6Y-mh9"
secondAttribute="trailing" type="default" id="Xjo-Qt-tyl"/>
                    <constraint firstItem="Gm5-3D-NxN"
firstAttribute="top" secondItem="5e5-6Y-mh9" secondAttribute="top"
type="default" id="lD8-b2-e6D"/>
                    <constraint firstItem="Gm5-3D-NxN"
firstAttribute="leading" secondItem="5e5-6Y-mh9"
secondAttribute="leading" type="default" id="vDB-EC-d4q"/>
                </constraints>
            </view>
            <tabBarItem key="tabBarItem" title="Save" id="77e-
RQ-wUt"/>
        </viewController>
        <placeholder placeholderIdentifier="IBFirstResponder"
id="7vj-Q7-M9V" userLabel="First Responder"
sceneMemberID="firstResponder"/>
    </objects>

```

```

        <point key="canvasLocation" x="4656" y="1255"/>
    </scene>
    <!--View Controller - Graph-->
    <scene sceneID="SyJ-2I-uEt">
        <objects>
            <viewController id="iCC-ap-rYi"
sceneMemberID="viewController">
                <view key="view" contentMode="scaleToFill" id="8HH-
eK-JYO">
                    <rect key="frame" x="0.0" y="64" width="768"
height="911"/>
                    <autoresizingMask key="autoresizingMask"
flexibleMaxX="YES" flexibleMaxY="YES"/>
                    <subviews>
                        <imageView userInteractionEnabled="NO"
contentMode="scaleToFill" image="iOS Simulator Screen shot May 8, 2013
5.07.09 PM.png" translatesAutoresizingMaskIntoConstraints="NO" id="43o-
mL-a9p"/>
                        </subviews>
                        <color key="backgroundColor" white="1"
alpha="1" colorSpace="custom" customColorSpace="calibratedWhite"/>
                        <constraints>
                            <constraint firstItem="43o-mL-a9p"
firstAttribute="top" secondItem="8HH-eK-JYO" secondAttribute="top"
type="default" id="NYk-Gh-7iC"/>
                            <constraint firstItem="43o-mL-a9p"
firstAttribute="leading" secondItem="8HH-eK-JYO"
secondAttribute="leading" type="default" id="Q7A-3n-Rk9"/>
                            <constraint firstItem="43o-mL-a9p"
firstAttribute="bottom" secondItem="8HH-eK-JYO"
secondAttribute="bottom" type="default" id="cOp-yn-Fxp"/>
                        </constraints>
                    </view>
                    <tabBarItem key="tabBarItem" title="Graph" id="ycV-
j1-dqt"/>
                </viewController>
                <placeholder placeholderIdentifier="IBFirstResponder"
id="2Gv-p1-9qC" userLabel="First Responder"
sceneMemberID="firstResponder"/>
            </objects>
            <point key="canvasLocation" x="4666" y="-1076"/>
        </scene>
    <!--Tab Bar Controller-->
    <scene sceneID="Xcc-HG-vwc">
        <objects>
            <tabBarController definesPresentationContext="YES"
id="esi-SO-1wL" sceneMemberID="viewController">
                <navigationItem key="navigationItem" id="v1E-u2-
MFs"/>
                <nil key="simulatedBottomBarMetrics"/>
                <tabBar key="tabBar" contentMode="scaleToFill"
id="gwt-2R-AW9">
                    <rect key="frame" x="129" y="330" width="163"
height="49"/>
                    <autoresizingMask key="autoresizingMask"
widthSizable="YES" flexibleMinY="YES"/>
                    <color key="backgroundColor" white="0.0"
alpha="0.0" colorSpace="calibratedWhite"/>

```

```

        </tabBar>
        <connections>
            <segue destination="iCC-ap-rYi"
kind="relationship" relationship="viewControllers" id="RfM-ZU-Utg"/>
            <segue destination="g8M-9h-7fz"
kind="relationship" relationship="viewControllers" id="jP4-Eq-Mu6"/>
            <segue destination="RyD-mJ-eXT"
kind="relationship" relationship="viewControllers" id="UYV-XI-XAV"/>
        </connections>
    </tabBarController>
    <placeholder placeholderIdentifier="IBFirstResponder"
id="O7d-vW-QdX" userLabel="First Responder"
sceneMemberID="firstResponder"/>
</objects>
<point key="canvasLocation" x="3684" y="-521"/>
</scene>
<!--Table View Controller - Available Devices-->
<scene sceneID="He4-1Y-ksN">
    <objects>
        <tableViewController id="YDf-W8-SkZ"
sceneMemberID="viewController">
            <tableView key="view" clipsSubviews="YES"
clearsContextBeforeDrawing="NO" contentMode="scaleToFill"
alwaysBounceVertical="YES" indicatorStyle="white" dataMode="static"
style="grouped" separatorStyle="singleLineEtched"
allowsMultipleSelection="YES" rowHeight="44" sectionHeaderHeight="10"
sectionFooterHeight="10" id="a6C-49-QPx">
                <rect key="frame" x="0.0" y="64" width="768"
height="960"/>
                <autoresizingMask key="autoresizingMask"
widthSizable="YES" heightSizable="YES"/>
                <color key="backgroundColor"
cocoaTouchSystemColor="scrollViewTexturedBackgroundColor"/>
                <color key="separatorColor" red="0.0"
green="0.0" blue="0.0" alpha="1" colorSpace="calibratedRGB"/>
                <sections>
                    <tableViewSection id="BTg-QK-bU0">
                        <cells>
                            <tableViewCell
contentMode="scaleToFill" selectionStyle="blue"
hidesAccessoryWhenEditing="NO" indentationLevel="1"
indentationWidth="0.0" editingAccessoryType="checkmark" id="VsF-6v-
a2D">
                                <rect key="frame" x="0.0"
y="30" width="768" height="45"/>
                                <autoresizingMask
key="autoresizingMask"/>
                                <view key="contentView"
opaque="NO" clipsSubviews="YES" multipleTouchEnabled="YES"
contentMode="center">
                                    <rect key="frame" x="45"
y="1" width="678" height="43"/>
                                    <autoresizingMask
key="autoresizingMask"/>
                                    <subviews>
                                        <label opaque="NO"
clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
text="Device 2" textAlignment="center" lineBreakMode="tailTruncation"

```

```

baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="Jsb-5C-4tW">
    <constraints>
        <constraint
firstAttribute="width" constant="152" id="gwI-HU-huQ"/>
        <constraint
firstAttribute="height" constant="23" id="lfI-3R-8af"/>
    </constraints>
    <fontDescription
key="fontDescription" type="system" pointSize="33"/>
    <color
key="textColor" cocoaTouchSystemColor="darkTextColor"/>
    <color
key="highlightedColor" white="1" alpha="1"
colorSpace="calibratedWhite"/>
    </label>
</subviews>
<color
key="backgroundColor" white="0.0" alpha="0.0"
colorSpace="calibratedWhite"/>
</view>
<color key="backgroundColor"
white="1" alpha="1" colorSpace="calibratedWhite"/>
<constraints>
    <constraint firstItem="Jsb-
5C-4tW" firstAttribute="centerY" secondItem="VsF-6v-a2D"
secondAttribute="centerY" type="default" id="XSR-c1-My9"/>
    <constraint firstItem="Jsb-
5C-4tW" firstAttribute="centerX" secondItem="VsF-6v-a2D"
secondAttribute="centerX" type="default" id="duc-e0-r80"/>
</constraints>
</tableViewCell>
<tableViewCell
contentMode="scaleToFill" selectionStyle="blue"
hidesAccessoryWhenEditing="NO" indentationLevel="1"
indentationWidth="0.0" id="3wK-nu-wOy">
    <rect key="frame" x="0.0"
y="75" width="768" height="45"/>
    <autoresizingMask
key="autoresizingMask"/>
    <view key="contentView"
opaque="NO" clipsSubviews="YES" multipleTouchEnabled="YES"
contentMode="center">
        <rect key="frame" x="45"
y="0.0" width="678" height="43"/>
        <autoresizingMask
key="autoresizingMask"/>
        <subviews>
            <label opaque="NO"
clipsSubviews="YES" userInteractionEnabled="NO" contentMode="left"
text="Device 3" textAlignment="center" lineBreakMode="tailTruncation"
baselineAdjustment="alignBaselines" adjustsFontSizeToFit="NO"
translatesAutoresizingMaskIntoConstraints="NO" id="4eI-Dh-xLr">
                <constraints>
                    <constraint
firstAttribute="width" constant="152" id="E6U-sy-E7j"/>
                    <constraint
firstAttribute="height" constant="31" id="UA5-QU-5sf"/>

```

```

        </constraints>
        <fontDescription
key="fontDescription" type="system" pointSize="33"/>
        <color
key="textColor" cocoaTouchSystemColor="darkTextColor"/>
        <color
key="highlightedColor" white="1" alpha="1"
colorSpace="calibratedWhite"/>
        </label>
    </subviews>
    <color
key="backgroundColor" white="0.0" alpha="0.0"
colorSpace="calibratedWhite"/>
    </view>
    <constraints>
        <constraint firstItem="4eI-
Dh-xLr" firstAttribute="centerX" secondItem="3wK-nu-wOy"
secondAttribute="centerX" type="default" id="4rQ-Np-Sig"/>
        <constraint firstItem="4eI-
Dh-xLr" firstAttribute="top" secondItem="3wK-nu-wOy"
secondAttribute="top" constant="6" id="wup-CE-lUZ"/>
    </constraints>
</tableViewCell>
</cells>
</tableViewSection>
</sections>
<connections>
    <outlet property="dataSource"
destination="YDf-W8-SkZ" id="p6U-13-Y6X"/>
    <outlet property="delegate"
destination="YDf-W8-SkZ" id="7QJ-ri-YY1"/>
</connections>
</tableView>
<navigationItem key="navigationItem"
title="Available Devices" id="g7b-8N-L1c"/>
</tableViewController>
<placeholder placeholderIdentifier="IBFirstResponder"
id="Zz7-ED-feP" userLabel="First Responder"
sceneMemberID="firstResponder"/>
</objects>
<point key="canvasLocation" x="3667" y="-2030"/>
</scene>
<!--Navigation Controller-->
<scene sceneID="LS0-KE-JCF">
    <objects>
        <navigationController id="au5-Kt-z3R"
sceneMemberID="viewController">
            <toolbarItems/>
            <navigationBar key="navigationBar"
contentMode="scaleToFill" id="CDZ-C0-qEf">
                <rect key="frame" x="0.0" y="0.0" width="768"
height="44"/>
                <autoresizingMask key="autoresizingMask"/>
            </navigationBar>
            <nil name="viewControllers"/>
            <connections>
                <segue destination="2" kind="relationship"
relationship="rootViewController" id="BWu-UC-MpA"/>

```

```

        </connections>
    </viewController>
    <placeholder placeholderIdentifier="IBFirstResponder"
id="lhn-A3-gwf" userLabel="First Responder"
sceneMemberID="firstResponder"/>
    </objects>
    <point key="canvasLocation" x="3" y="-99"/>
</scene>
</scenes>
<resources>
    <image name="Screen Shot 2013-05-08 at 4.40.46 PM.png"
width="690" height="407"/>
    <image name="iOS Simulator Screen shot May 8, 2013 5.07.09
PM.png" width="768" height="960"/>
</resources>
<classes>
    <class className="NSLayoutConstraint"
superclassName="NSObject">
        <source key="sourceIdentifier" type="project"
relativePath="./Classes/NSLayoutConstraint.h"/>
    </class>
    <class className="ViewController"
superclassName="UIViewController">
        <source key="sourceIdentifier" type="project"
relativePath="./Classes/ViewController.h"/>
    </class>
</classes>
<simulatedMetricsContainer key="defaultSimulatedMetrics">
    <simulatedStatusBarMetrics key="statusBar"
statusBarStyle="blackTranslucent"/>
    <simulatedOrientationMetrics key="orientation"/>
    <simulatedScreenMetrics key="destination"/>
</simulatedMetricsContainer>
</document>

```