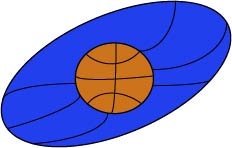
Basketball Vision - Team 3

Senior Design Project



***September 19, 2016***

***Document Authors:***          Jared Becker, Matthew Resch,

   Kendall Weihe, Thomas Underwood, Byron Giles

***Document Editors:***  Jared Becker

***Client:***     Mr. Peter Williams, Teacher, North Oldham High School

Table of Contents

Planning and Estimating …………………………………………………………………….................1

Size Estimate …………………………………………………………………………………….1

Risk List …………………………………………………………………………………………..1

Development Environment Risks ……………………………………………………..1

Product Size Risks ……………………………………………………………………..1

Design Risks …………………………………………………………………………….1

Business Impact Risks …………………………………………………………………2

Schedule & Resource Allocation ………………………………………………………………2

Requirements …………………………………………………………………………………………….3

Introduction ………………………………………………………………………………………3

Project Overview ………………………………………………………………………..3

Project Scope …………………………………………………………………………...3

Document Preview ……………………………………………………………………..4

Project Overview ………………………………………………………………………………...4

Development and Target Environments ………………………………………………………5

System Model ……………………………………………………………………………………5

User Interaction ………………………………………………………………………………….6

Functional Requirements ……………………………………………………………………….7

Nonfunctional Requirements …………………………………………………………………..9

Feasibility ……………………………………………………………………………………….10

Metrics …………………………………………………………………………………………………...11

Web Page and Developer Notebook …………………………………………………………………11

Conclusion ………………………………………………………………………………………………11

Appendices ……………………………………………………………………………………………...12

System Block Diagram ………………………………………………………………………..12

Database Models ………………………………………………………………………………15

Use Case Diagrams …………………………………………………………………………...15

Application Mockup ……………………………………………………………………………21

Table of Figures

Figure 1 ………………………………………………………………………………………………….12

Figure 2 ………………………………………………………………………………………………….13

Figure 3 ………………………………………………………………………………………………….13

Figure 4 ………………………………………………………………………………………………….14

Figure 5 ………………………………………………………………………………………………….15

Figure 6 ………………………………………………………………………………………………….21

# **1 Planning and Estimating**

## **Size Estimate *(Jared)***

We can estimate the size of our project in terms of story points and development hours. We estimate the project is comprised of roughly 40 user stories, which calculates to roughly 200 story points. We also estimate the development phase will take 200 hours, which is about 2 months. Our embedded system code will also require an additional 50 story points and 10 user stories. A more in-depth estimation of the project can be found in the Metrics section of this document.

## **Risk List *(Jared, Matt)***

This project is being built from the ground up. It is not an extension or an improvement to a currently existing product or project. Therefore, there are several potential risks the team could encounter. We have analyzed these potential scenarios and drafted solutions to mitigate these risks from occurring or managing the risk in case it does.

### 1.2.a Development Environment Risks

No one our team has prior experience developing a mobile application for iOS. Therefore, there is significant potential for risk. Possible solutions to mitigating the risk are to follow tutorials online, watch videos, and read articles to answer questions when problems occur.

### 1.2.b Product Size Risks

There is risk that the project ends up larger in size than our initial estimate. Consequently, the effort to implement our product will be underestimated as well. One possible approach to mitigate this risk is to continually voice concerns about the timeline or workload at our weekly team meetings. It is possible we can reallocate resources to meet deadlines and lessen the workload for members feeling pressured.

If the risk continues to grow, one solution to manage the risk is to simplify the design. By eliminating certain features (like bonus features) from the design, we can focus more time on developing the base model and completing the prototype by the semester’s end. Of course, any changes and bonus features can always be added at a later date. We have also made efforts to mitigate this risk by managing client expectations while soliciting requirements to prevent overcommitting on our timeline.

### 1.2.c Design Risks

Byron has worked tirelessly to research the hardware we need to best create the system that will track the player on the court. In doing so, we have discovered a few potential flaws that could occur. One risk is interfering signals if there are multiple players on the court at one time. The solution to mitigate this risk is to assign each player a different channel so that the communication from the system to the player’s phone will be uninterrupted.

A related risk is the novelty of the hardware to our team. While we know that we will be able to assign separate channels to players, we will need to teach ourselves how to operate these channels. We have mitigated this risk by focusing on procuring hardware early to allow time to familiarize ourselves with it through trial-and-error and research.

Another risk in developing our hardware solution is that we have not properly accounted for all of the hardware that we need or that the hardware we are considering will not adequately accomplish its task. We have mitigated this risk through research and by soliciting advice from professors, as well as by prioritizing hardware acquisition to allow plenty of testing and familiarization time.

A third possible risk is the bounce-back of signals when they reach walls and surfaces. Clearly, this can cause unwanted behavior. When signals bounce off walls, their strength is greatly reduced. Therefore, one approach to mitigate this risk is the monitor the strength of the signals. By accepting signals above a certain magnitude, we will capture only the intended signal and nothing that has already been processed or can be considered noise or interference.

A final, major design risk is that we have failed to account for some significant risk due to lack of understanding or oversight. This risk is inherently difficult to mitigate due to its unknown nature, but we mitigate it to some extent by researching thoroughly to understand our requirements and communicating within our team to address concerns and identify potential challenges.

### 1.2.d Business Impact Risks

The last possible risk is time. Time is a risk because of unforeseen circumstances. In addition, the “twist” we can expect later in the design phase will also test time. There is always risk the project will not be complete by the deadline. The only way to both manage and mitigate this risk is to maintain constant communication with the team and address any and all concerns at the weekly team meetings. By communicating issues, we can best determine how to approach the problem without deviating too much from the timeline. The risk of not completing our task within the timeframe is influenced by each listed risk, as any failure to mitigate risk will impact our timeline. Therefore, each proposed mitigation can also be considered a strategy to mitigate the risk associated with time.

## **1.3 Schedule & Resource Allocation *(Jared)***

The team has met several times to plan our project so that we can meet our projected deadlines. We expect to begin implementing the embedded system by component starting on September 25. Development is projected to begin this week, starting September 19th. Below is an outline of the rest of semester, indicating our intended deadlines:

9/19/2016 - iOS and Android Development Begins

9/25/2016 - Positioning System & Central Hub Implementation Begins

10/2/2016 - Positing System Implementation Ends

10/2/2016 - Distance System Implementation Begins

10/7/2016 - Distance System Implementation Ends

10/7/2016 - User Microcontroller Implementation Begins

10/21/2016 - User Microcontroller Implementation Ends

10/21/2016 - Embedded System Testing Begins

11/21/2016 - iOS and Android Development Ends

11/21/2016 - Production Testing Begins

11/28/2016 - Product Presentation Preparation Begins

12/5/2016 - Production Testing Ends

12/5/2016 - Product Presentation Begins

12/9/2016 - Product Presentation Ends

In order to best achieve and meet these deadlines, each team member has been assigned specific roles. Kendall will take the lead on iOS development. Thomas is leading Android development. Byron is taking the lead on the embedded system. Jared is heading all documentation and keeping the team on schedule and will assist where additional help is needed. Matt is a flexible team member. Like Jared, he will also assist when additional help is requested by team leads.

# **2 Requirements**

## **2.1 Introduction *(Jared)***

### 2.1.a Project Overview

It goes without saying that the visually-impaired experience constant challenges in their everyday lives. They work hard to defy the odds and show the world that their *inability* is rather an *ability*. But there are some things that are not so easy to overcome, like playing one of America’s favorite pastimes: basketball.

Basketball is a visual sport. It is imperative to know the distance to the basket, and the position of the player. And that’s where we come in. It is our goal to build an application that provides the player with all this information so that the player can shoot, and hopefully make, a basket.

Our team is motivated and eager to forever change the sport of basketball.

### 2.1.b Project Scope

Our idea sounds simple, but the implementation is much more complex. Our team has identified the major features and constraints of the project. We intend to use these ideas as our foundation when development begins.

As mentioned before, the most important feature to the project is providing the player with accurate information about his/her distance and location relative to the basket. We have designed a system that will process this data, send it to the user’s smartphone, and finally, read aloud the calculations. The communication of the system to the smartphone is the most important feature because without it, the player will not have the data they need to make a decision about shooting the ball.

We have a lofty goal of making this system compatible to host multiple players at one time. However, each additional player increases the order of complexity. With each player, the system needs to calculate data and communicate the results to the respective player. There is risk of signals interfering with one another, as well as signals bouncing off walls if the game is played indoors. These constraints require additional design to mitigate the potential of errors in the system.

### 2.1.c Document Preview

The purpose of this document is to provide a detailed outline of our project’s basic requirements. We intend for it to be reviewed by the client, Mr. Peter Williams, the Senior Design instructors, and any advisor or professor we deem worthy to both critique the current requirements outlined in this document and provide feedback for improvement before the design process begins.

In the pages to follow, this document will explain our development strategies and outline the proposed system model. Most importantly, the document will identify the project’s essential functional and nonfunctional requirements. Before concluding, we will provide a realistic estimate on the project’s potential for success by proposing two versions of our system: one confident we can complete by the semester’s end and a second version we hope to implement long down the road.

## **2.2 Project Overview *(Matt)***

Basketball is an impactful sport in the state of Kentucky. It is a shared pastime that unites and excites people across the country. Many of the nation’s blind and visually-impaired share in this excitement, but are unable to participate in the sport due to physical limitations. Our goal is to address visual concerns associated with playing basketball to extend this opportunity to the blind.

Potential customers for our system include private citizens who are blind and want to play basketball; someone who wants to facilitate a blind friend or family member play basketball; or schools for the blind or schools with blind students that wish to incorporate basketball as an activity for their students. Stakeholders in our project include us as the design team, previously mentioned potential customers, our client at North Oldham High School, and the blind who wish to play basketball. Our intended users are the blind or visually-impaired who play or wish to play basketball.

We are currently unaware of any existing system that facilitates participation in basketball by the blind or visually-impaired. There are many sports and activities that are accessible to the blind, and there are constant efforts to improve accessibility of sports and activities to people with various disabilities. Currently, there is no available system for purchase that assists the blind or visually-impaired with basketball. We seek to develop one.

The primary problem to be solved in making basketball accessible to the blind is that it is a very visual sport in its nature. Playing basketball requires orientation on the court and to the basket, as well as the ability to identify shot positioning. Since we are unable to purchase one, our goal is to develop a system that reports player position, orientation to the basket, and shot data to the user.

The main features of our system will include sensors on the court and backboard that identify the position of the user and the results of attempted shots; a database of user data and shooting history; and iOS and Android applications that allow users to interact with the database data and have their position and shooting reported to the user aurally.

Critical constraints to our system include accessibility to the blind and compatibility with existing basketball equipment. This means that our system should be deployable on existing basketball courts, using standard backboards and baskets. It should also have user friendly mobile application design that is highly compatible with existing accessibility modes in iOS and Android. Our system does not need to interface with other systems, but it does involve the interfacing of hardware, microcontrollers, and software, as well as communication to the user.

## **2.3 Development and Target Environments *(Thomas)***

The user application will be developed for the iOS and Android environment. Android studio will be used to write the Android version of the application and Xcode will be used for the development of the iOS version of the application. Bitbucket will be used as our SVC system as it supports private free private repositories. The project database will be hosted by the Google Cloud Platform (GCP) and will be accessible through the Android and IOS application. The raspberry pi (Debian Linux) will have a database to store the calculations and statistics generated during run time. The raspberry pi will pass data back and forth between the user application (Android and iOS) at which point the data will be displayed on the app, and then stored in the GCP. The GCP has its own format for creating, updating, deleting tables and entries (CRUD).

## **2.4 System Model *(Kendall)***

The system model includes embedded systems and smartphone applications. The user interacts with the smartphone application while the embedded systems collect necessary data. The user’s smartphone is connected to a single central processing node. This way setting up the system is simplified as the user only needs to connect to the embedded systems through a single node. All connections between nodes are wireless connections. The system block diagram can be found at Figure 4 in the Appendices.

There are seven nodes in the embedded system: the central processing node (Figure 3 in the Appendices), the user’s node, four positioning *anchor nodes* (Figure 1 in the Appendices), and a node located on the backboard (Figure 2 in the Appendices). All nodes will be battery powered for portability. All nodes connect to the central processing node. The user’s node is a microcontroller located on the user (worn as an accessory). This node collects user requests through hardware buttons, and then outputs signals to both the central processing node and the four anchor nodes. If the user requests to know their distance from the basketball goal, then they will click the button specific to that feature. The node will then send output signals to the four anchor nodes so that position coordinates can be collected. It then sends an output signal to the central processing node informing the system that distance to the goal should be computed. The four anchor nodes collect signals from the user’s microcontroller, and then output signals to the central processing node where position coordinates are computed. The node located on the backboard includes a sensor placed under the basketball rim that is activated when a shot is made. Given the shot was made, the node will output a signal to the central processing node.

The user smartphone application is a single node in the diagram. The smartphone application is responsible for inputting data from the central processing node and then conveying the information to the user via audio. For example, if the user requests their position on the basketball court the following sequence is executed. The central processing node will collect position coordinates, then send the data to the smartphone, where subsequently the smartphone will convey the information to the user via audio.

## **2.5 User Interaction *(Byron, Matt)***

In the case that a user makes a shot, the system executes the following sequence. First, the user takes the shot. If the shot is made, the break beam sensor on the goal reports to the attached Arduino. That signal is then sent to the central processing node. The node then acquires the relative position that the shot was taken from by the beacons. That position is then reported to the user and recorded to the database. See Use Case #1 in Appendices.

In the case that a user wants to know their distance from the basketball goal, the system executes the following sequence. The user presses a button on the application. A request signal is then sent to the central processing node. Distance signals are also sent to the anchor microcontrollers. The central processing node accepts the requests, and then receives positioning coordinates from the anchor nodes. The central processing node then computes a distance based on [x,y] coordinates. The central node then returns the distance measurement to the user's smartphone where it is the read aloud to the user. See Use Case #2 in Appendices.

In the case that the user wants to request their grid coordinate, the following sequence is executed. The user presses a button on the application. A signal is then sent to the central processing node. The node then acquires the relative position of the player’s module. This position is then reported back to the user through the application’s speaker. See Use Case #3 in Appendices.

In the case that the user wants to login into their player account, the following sequence is executed. When the user opens up the application, they are then prompted to either login or continue. The credentials are sent to the central processing node, where the player’s profile is acquired. The user is then prompted to confirm. If confirmed, player account becomes active account. See Use Case #4 in Appendices.

In the case that the user wants to make changes to their settings, the following sequence is executed. The user presses a button on the application. The application then reports a list of options to the player, and waits for a response. Once the player has selected, a signal is then sent to the central processing node to handle the request. The options will include elements such as, change profile, change game type, and recalibrate.  See Use Case #5 in Appendices.

In the case that a user makes a shot in a game of horse, the shot made sensor will trigger. This signal will be received by the system, triggering an update. The user will have their shooting statistics updated, and the current status of the game of horse updated. See Use Case #6 in Appendices.

In the case that one user makes a shot in horse, but the second user misses, the shot made sensor will not be triggered. Instead, the system will recognize the shot was missed. Then, the user’s shooting statistics will be updated. The user will then have a letter added to the score, or will lose the game, depending on the current state of the game. See Use Case #7 in Appendices.

In the case that a user wants to know their overall shooting statistics, they will request the percentage from the statistics section of the app. The system will then query the database for the user’s shooting statistics. The all-time shooting percentage will be reported to the user. See Use Case #8 in Appendices.

In the case that a user wants to know their shooting percentage in the current game session, they will request the percentage from the statistics section of the app. The system will then query the database for the user’s shooting statistics. The current session shooting percentage will be reported to the user. See Use Case #9 in Appendices.

In the case that a new user wants to register an account in the app, they will select the register new user option. The system will display a registration page. The user will input a desired username and password. If the username is available and the username and password are valid, the system will update the database with the new user account and log the user in. See Use Case #10 in Appendices.

## **2.6 Functional Requirements *(Kendall)***

The goal of this project is to communicate enough information to a visually-impaired basketball player so that they can effectively shoot a basketball. This includes many functional requirements. There are two main subsystems that will be developed: a hardware system that collects information and both Android and iOS applications that convey the information to the basketball player via audio. The hardware subsystem is functionally required to input sensor data and output computation to the user's smartphone. The hardware subsystem is crucial in that the applications rely on hardware to collect accurate information. There are five subsystems for the hardware component: a central hub, a system located on the user, a system to compute user location, a system to compute distance from the user to the basketball goal, and a system that will activate when a shot was made. The applications are functionally required to input data from the hardware, and output information to the basketball player. There are three main subsystems for the application component: a system to provide user authentication, a system for the user to play a collection of game types, and a system to collect and outputs statistical information.

In order to simplify user setup, a central hub microcontroller (or single-board computer) will serve as a *central hub.* Since the central hub will be used by all other embedded systems, it has the highest priority among the hardware systems. The central hub is functionally required to input data from the four hardware subsystems and output data to the user's smartphone. The communication shall be connected via a Bluetooth protocol. An example use case that utilizes the central hub is when a user requests to know the distance to the goal. In this case, the central hub shall input data from the positioning microcontrollers, compute the distance, and output to the user’s microcontroller.

The user will be wearing a microcontroller while shooting basketball, henceforth referred to as the *user’s microcontroller*. Since the user’s microcontroller will serve as the device that the user interacts with, it has the second highest priority among the hardware systems. The user’s microcontroller is functionally required to input requests from the user, and output signals to both the positioning system and the central hub. An example use case that utilizes the user’s microcontroller is when the user requests to know the distance to the basketball goal. The user’s microcontroller shall input the type of request (in this case the distance to the goal) via a button. The user’s microcontroller shall output two signals: a signal to the positioning location and a signal to the central hub.

The user shall be able to make a request to know their position on the basketball court. The positioning system is composed of four *anchor microcontrollers*. There is no standard for indoor positioning systems on the market. For this reason, the positioning system is the cornerstone of this project and has the third highest priority among the hardware systems. Each anchor microcontroller is functionally required to input requests from the user’s microcontroller alongside of a perpendicular distance to the user, and output distance measurements to the central hub. An example use case that utilizes the anchor microcontrollers is when the user requests to know their location on the basketball court. The anchor microcontrollers shall input a signal from the user’s microcontroller that will activate a positioning request. The anchor microcontrollers shall then input a radio signal from the user’s microcontroller, and compute a distance. The distance shall be computed by multiplying the time from which the signal was emitted to when the signal was received by the speed of an electromagnetic wave. The anchor microcontrollers shall then output the distance measurement to the central hub.

The user shall be able to request the distance to the basketball goal. The distance system is composed of the positioning system and the central hub. The distance system is functionally required to input a user request, and output a signal to the user’s smartphone. An example use case that utilizes the distance system is when the user requests to know the distance to the goal. The anchor microcontrollers shall measure user position, the central hub shall then use the user position to compute a direct distance to the goal. The distance system shall then output the distance to the user’s smartphone.

A *made-shot* microcontroller shall activate when a shot is made. The made-shot microcontroller is functionally required to input data when a shot was made, and output data to the central hub. An example use case that utilizes the made-shot microcontroller is when the user makes a shot. The made-shot microcontroller shall input an activation signal (via a sensor), and output that the shot was made to the central hub.

Through their smartphone application, the user shall be able to create an account and login. User authentication has the highest ranked priority among the application systems because each player must have a unique identifier in order to use the applications features. The user authentication is functionally required to input user login credentials and output access to their account. An example use case that utilizes user authentication is when the user wants to create an account. The user authentication shall input a username and password, and output access to their newly generated account. The user authentication shall utilize secure encryption techniques in order to secure user information.

The user shall be able to select, and play a game type. Playing games is what makes shooting basketball fun, and for that reason selecting and playing a game has the second highest priority among the application systems. The game system is functionally required to input data received from the central hub, and output the current status of the game. The game system is also functionally required to communicate with all users playing the game. An example use case that utilizes the game system is when two users request to play the game of HORSE. The game system shall input whether or not user-1’s shot was made or missed, if the shot was made, the game system shall input the user’s location on the basketball court. The game system shall then output to user-1 that his/her shot was made, and output to user-2 that user-1’s shot was made, and the position it was made from.

The user shall be able to play the game of HORSE with multiple other players. The game of HORSE is functionally required to input a made shot and where the shot was made. The game of HORSE is functionally required to output to all players that a shot was made by user-1 and where on the basketball court the shot was made from. An example use case that utilizes the game of HORSE is when user-1 makes a 3-point shot. The system shall input that the shot was made, and that the shot was made outside the 3-point line at position [x,y]. The system shall then notify all users that user-1 made a 3-point shot at position [x,y].

The application shall keep track of statistics while users are playing basketball. The statistics system is functionally required to input made shots and from where the shot was made. The statistics system is also functionally required to output the statistics to a cloud based database. An example use case that utilizes the statistics system is when the user makes a 3-point shot. The statistics system shall input a made-shot signal from the central hub, and input where the shot was made from the central hub. The statistics system shall then output the data to the cloud based database.

The user shall be able to request statistics on both their current session and their entire history. The statistics system is functionally required to input user request for the type of statistic and output the statistical information. An example use case that utilizes the statistics system is when the user requests to know their shooting percentage for the current session. The statistical system shall input the type of request (in this case their shooting percentage for the current session), and output the value of the statistic through audio.

## **2.7 Nonfunctional Requirements *(Kendall)***

The system laid out in this document has various nonfunctional compatibility, efficiency, reliability, and portability requirements. In order for the users to play basketball with our system, they must be using a compatible smartphone. The system shall utilize communication protocols that are compatible with all hardware devices, and any communication signals shall be at legal frequencies. The communication signals shall also be unique in that they do not interfere with existing signals. Response time from a user request to an audible response shall be efficient in that it will take no more than three seconds. The central hub shall be able to scale to multiple players at once without downgrading performance. The mobile batteries shall be reliable in that they last at least two hours, and/or be easily replaceable. Cloud based database queries shall always be up to date. The entire system shall be portable from one basketball court to another. The user shall be able to move around the court as if the system did not exist.

The system has various nonfunctional compatibility requirements. The system assumes that the user has access to a smartphone with internet connection. The smart shall be either an Android or iPhone device. The smartphones shall communicate with the central hub through a compatible wireless communication protocol. The central hub and other microcontrollers shall also communicate through a compatible wireless protocol. All wireless signals used by the system shall be within a legal FCC frequency range. If any of the compatibility requirements above are not followed, then the system may fail to work as a whole. Another nonfunctional requirement is efficiency.

The system has various nonfunctional efficiency requirements. The time from which a user requests information to the time their smartphone’s audio port speaks to them shall be no more than three seconds. An example use case for response time is when a user request to know their location on the basketball court. The user will press a button on their microcontroller, and hear the position coordinates through their smartphone within three seconds. The central hub shall be able to scale to multiple players without downgrading user response time. An example use case of the central hub scaling would be when a second player joins a game. If the second player joins a game, then the user response time shall stay consistent. Another functional requirement is reliability.

The system has various nonfunctional reliability requirements. The batteries that power the microcontrollers shall either last 2 hours, and/or be easily replaceable. All wireless communication signals shall be unique and verifiable. If a signal is not unique, then a receiver sensor may pick up incorrect signals from other wireless devices. If the signal is not verified, then the receiver sensor may pick up a signal that has been bounced off of a surface. An example use case of this is when the user requests to know their position on the basketball court. If the user’s microcontroller emits a signal in the same frequency as another wireless signal (from a different device) then the anchor microcontrollers could potentially compute distance from the wrong source. Cloud based database queries shall always be up to date. An example use case of this is when a user requests to know their shooting percentage for the entire history. If the database has not been updated with their current sessions statistics, then it will not be the correct statistical value. Portability is another nonfunctional requirement.

The system has various nonfunctional portability requirements. The system as a whole shall be portable from one basketball court to another. An example use case of this is when a user, or group of users, want to migrate the system from the local basketball court to another basketball court. If the system is not portable, then they will have to buy a second system. Each user shall be able to freely move around the basketball court as if the system weren’t there.

## **2.8 Feasibility *(Kendall)***

It is likely that our team will not complete all above requirements before the end of the semester. Mr. Williams, our client, and his students have brainstormed a very involved project. In fact, there are other systems that we have set aside to either have completed by the students of North Oldham High School or by us at a later time. This project is entirely novel and has the potential to revolutionize the ways in which visually-impaired persons play basketball. For that reason, our team has set the bar high in hopes of achieving the maximum amount before the end of the semester. Some team members have agreed to continue work with Mr. Williams until the entire project is completed. There are two end products that we have designed.

The first design is a raw prototype: something that we hope to have completed by the end of the semester. The raw prototype includes all the functional and nonfunctional requirements in the system model explained in this document. The prototype design includes five main systems: a system to locate the user's position on the basketball court, a system to compute the distance from the user to the basketball goal, a system to inform the user as to whether or not their shot went in, and a smartphone application for the user to interact with.

The second design is a product that would be ready to bring into industry. The second design includes additional systems. There will be a system that tells the user where their shot landed on the backboard. There will be a system that will throw the ball back to the user (for single player games only). There will be a system that informs the user the location of the ball so that they can retrieve their shot. As development continues, we expect more systems to emerge. After developing all the systems into a prototype, then we will have to fine tune, simplify, and prepare the product for market. That means we must look into different hardware that is cheaper or more precise, whatever tradeoff is deemed more valuable. In order for the product to be ready for market, the entire system must be simple to setup and use.

# **3 Metrics *(Thomas)***

There are approximately 40 user stories. Our story point scale is based on one through five. So if we assume each story is a 5 (highest difficulty) we can assume that the maximum estimate for story points would be 200 story points. Our embedded system code will also require an additional 50 story points and 10 user stories. For a project total of 250 story points.

For each user story there will be tests implemented to do boundary testing. There will be a minimum of 5 test cases per each user story. Resulting in about 200 tests for the application side of development. The embedded system testing will require greater and more rigorous testing than the application due to the number of possible risks (see *Design Risks* in the *Risk List*). Through testing, we hope to develop a robust system. This system will have to go through several full cycles of system, component, and unit testing.

We have estimated that the man power to create all of the code necessary to run our project will be between 200 - 300 hours, or about 2 months. There will be 3 - 4 people working on the code at a time.

# **4 Web Page and Developer Notebook *(Jared)***

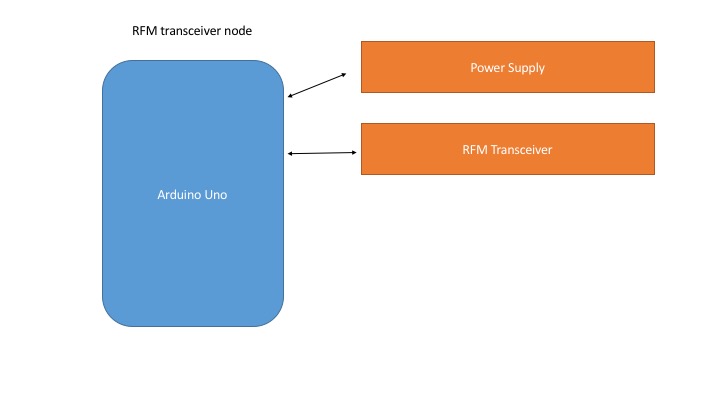
Our team’s website can be viewed at <https://prodigiousmelon.github.io/>. Included on the site is each team member’s individual developer notebook, which is found under the specific team member’s name.

# **5 Conclusion *(Jared)***

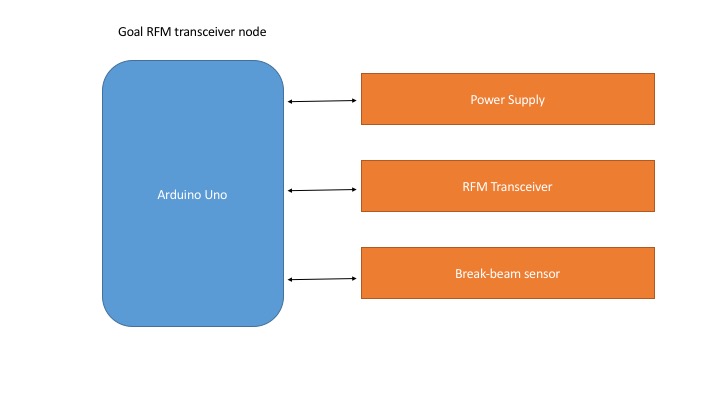
The team has spent many hours planning our project and designing what we envision our product to be able to accomplish at the semester’s end. This Project Plan Document will be a vital reference for us as we begin development and implementation. Our success as a team this semester will dictate the future of this product and whether or not we intend to continue pursuing the project and bring the product to market. We are excited to collaborate as a team and put our studies to good use.

# **6 Appendices**

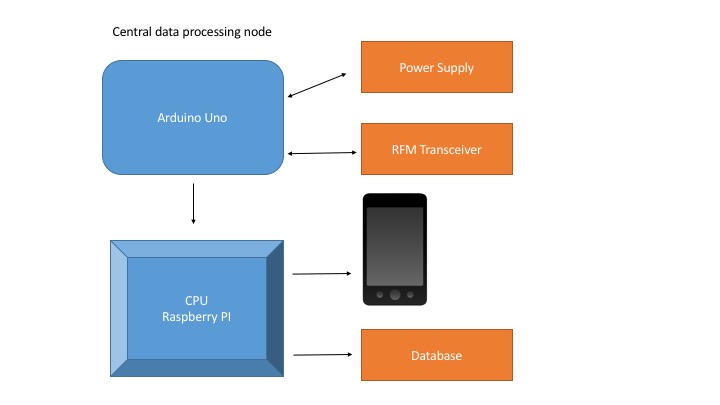
## **6.1 System Block Diagrams**



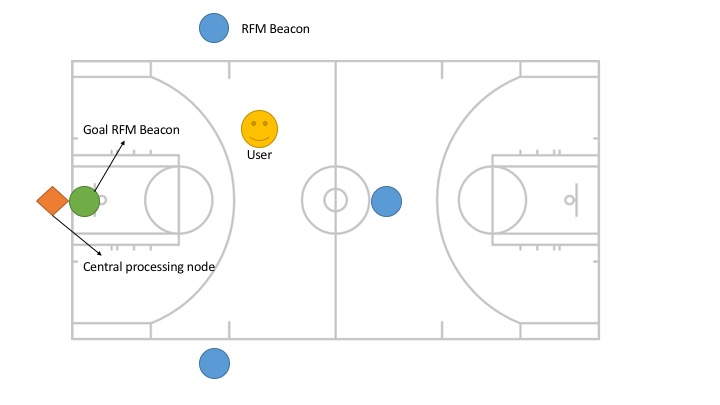
**Figure 1.** This diagram illustrates the setup of the RFM transceiver nodes. There are four in total: one on each side of the court, a third at the top of the key, and a fourth located on the backboard. These nodes are used to calculate the user’s position on the court.



**Figure 2.** This diagram illustrates the RFM transceiver that is at the backboard. It has the same setup as the RFM transceivers along the court, in addition to a break-beam sensor which detects if the ball goes in the hoop.



**Figure 3.** This diagram shows the setup of the central hub. The main Arduino Uno communicates with the RFM transceivers and the CPU Raspberry PI. The PI then communicates with the database and the application on the player’s smartphone.



**Figure 4.** This diagram illustrates where all the hardware will be positioned relative to a standard basketball court.

## **6.2 Database Models**

S499_Basketball_Vision_DB.png

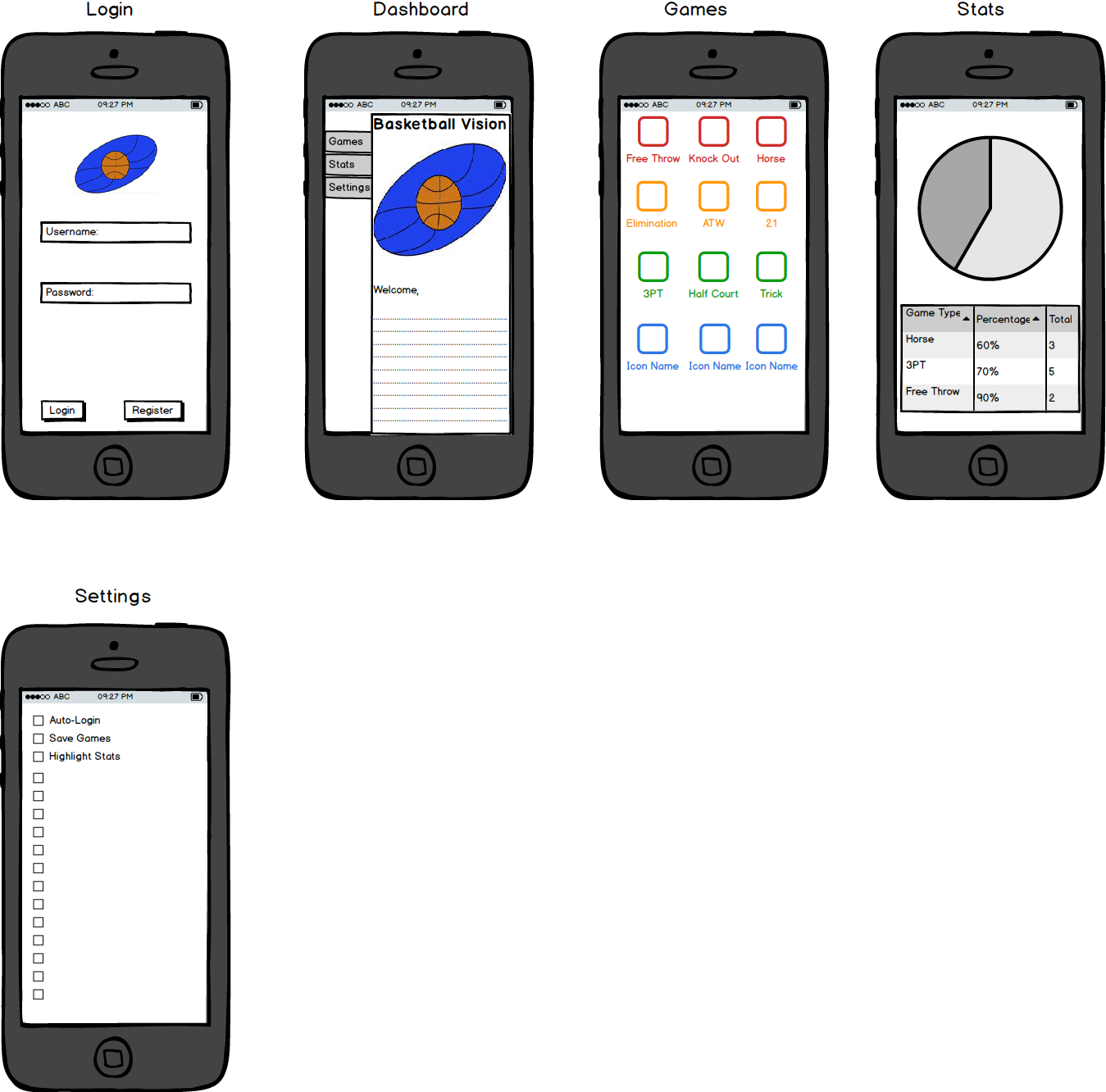
**Figure 5.** User information will be stored in a database. We are using Google Cloud Platform as host for the database.

## **6.3 Use Case Diagrams**

|  |  |
| --- | --- |
| **Use Case #1:** | **User Makes Shot** |
|  |  |
| Actor: | Player |
| Goal: | Report made shot to player and record in database |
|  |  |
| User | System |
| 1. User takes shot |  |
|  | 2. Made shot signal sent from goal node to PI |
|  | 3. System acquires position |
| 4. Position is reported to user |  |
|  | 5. Position is recorded to database |
| 2a | Signal fails to reach PI |
| 3a | System acquires inaccurate coordinates |
|  |  |
| **Use Case #2:** | **User Requests Distance to Basket** |
|  |  |
| Actor: | Player |
| Goal: | Smartphone informs the user their distance to the basketball goal |
|  |  |
| User | System |
| 1. User presses button on application |  |
|  | 2. Signal sent from application to PI |
|  | 3. PI collects position from anchor microcontrollers |
|  | 4. PI computes distance based on position coordinates |
|  | 5. PI sends data back to smartphone |
| 6. Smartphone tells user their distance from the basketball goal |  |
|  |  |
| 2a | Signal fails to reach PI |
| 3a | Signals fails to reach module |
|  |  |
| **Use Case #3:** | **User Requests Position on Court** |
|  |  |
| Actor: | Player |
| Goal: | Application reports grid coordinates to user |
|  |  |
| User | System |
| 1. User presses button on application |  |
|  | 2. Signal sent from application to PI |
|  | 3. Position acquired |
| 4. Reports position to user |  |
|  |  |
| 2a | Signal fails to reach PI |
|  |  |
| **Use Case #4:** | **User Logs In** |
|  |  |
| Actor: | Player |
| Goal: | Logs in user to track statistics |
|  |  |
| User | System |
| 1. User presses button on application |  |
| 2. User logs into profile |  |
|  | 3. PI acquires player records |
| 4. Confirm with user |  |
|  | 5. Set player account to active account |
|  |  |
| 2a | Login fails |
|  |  |
| **Use Case #5:** | **User Makes Changes to Settings** |
|  |  |
| Actor: | Player |
| Goal: | Allows user to change account and game |
|  |  |
| Player | System |
| 1. User presses button on application |  |
|  | 2. Reports list of options to player |
|  | 3. Waits for response |
| 4. Player makes selection |  |
|  | 5. Load request |
| 6. Confirm with user |  |
|  | 7. Execute request |
|  |  |
| 4a | Player closes app |
| 7a | Execution fails |

|  |  |
| --- | --- |
| **Use Case #6:** | **User Makes a Shot in Horse** |
|  |  |
| Actor: | Player |
| Goal: | Record Made Shot in Horse |
|  |  |
| User | System |
| 1. Makes shot during a game of horse. |  |
|  | 2. Shot made sensor triggers. |
|  | 3. Updates user statistics and game status |
|  |  |
| 2a Sensor fails to trigger |  |
| 3a Communication from sensor fails |  |
|  |  |
| **Use Case #7:** | **Second User Misses Shot in Horse** |
|  |  |
| Actor: | Player |
| Goal: | Add a Letter to Second Player Score |
|  |  |
| User | System |
| 1. Misses shot following a made shot during a game of horse. |  |
|  | 2. Registers missed shot |
|  | 3. Updates user statistics and adds a letter to score |
|  |  |
| 2a System fails to register that there was a missed shot |  |
|  |  |
| **Use Case #8:** | **User Checks Shooting Percentage Overall** |
|  |  |
| Actor: | User |
| Goal: | Identify Percentage of Shots Made by User All Time |
|  |  |
| User | System |
| 1. Selects overall shooting percentage in statistics |  |
|  | 2. Queries database for all time shooting statistics |
|  | 3. Reports statistics to user |
|  |  |
| 2a User closes app before statistics are reported |  |
| 2a Database query fails |  |
|  |  |
| **Use Case #9:** | **User Checks Shooting Percentage of Current Session** |
|  |  |
| Actor: | User |
| Goal: | Identify Percentage of Shots Made in Current Session |
|  |  |
| User | System |
| 1. Selects current session shooting percentage in statistics |  |
|  | 2. Queries database for current session shooting statistics |
|  | 3. Reports statistics to user |
|  |  |
| 2a User closes app before statistics are reported |  |
| 2a Database query fails |  |
|  |  |
| **Use Case #10:** | **User Creates an Account** |
|  |  |
| Actor: | User |
| Goal: | Create New Player Account |
|  |  |
| User | System |
| 1. Opens the app and selects create new user |  |
|  | 2. Loads registration page |
| 3. Enters username and password |  |
|  | 3. Adds new user profile to database and logs user in |
|  |  |
| 3a User closes app before entering account information |  |
| 3a User enters invalid username or password | s |
| 4a Username is taken |  |

## **6.4 Application Mockup**



**Figure 6.** This project involves development of both an android and iOS application. Development is planned to start this week according to the schedule illustrated in the “Schedule & Resource Allocation” section of this document. This a mockup of the application’s interface.

Word Counts:

Jared - 1668

Byron - 694

Kendall - 2911

Matt - 1287

Thomas - 342