# Design Document

**Player Tracking Camera**

**Team 21**

**TA: Feiyu Zhang**
**Shivang Charan**
**Aksh Gupta**
**Oreoluwa Sunmola**

## Introduction

**Problem Statement**:

In the digital age people love recording themselves and their surroundings as a way of remembering times in their lives. Whether it be on vacation, at a sporting event, or even at home, it has become commonplace for people to have devices to capture all these moments.

Unfortunately, it is very hard for the general public to record themselves while playing sports. The core issue is that individuals are usually not able to record themselves as they play. Therefore, they are required to ask someone to record them while they are playing. This is a major inconvenience and one which can be automated away.

**Solution Overview**:

In order to solve this problem, we intend on creating a moving stand that will keep the intended target in the view of their camera. This stand will be adjustable to fit most existing cameras. By setting up small beacons around the playing area we can calculate the position of the target person running - which would then be used to rotate the camera to capture the subjects' movements in real-time. Functionally, this would eliminate the need to have a cameraman follow the game's movements and would automate the filming process.

**Visual Aid:**

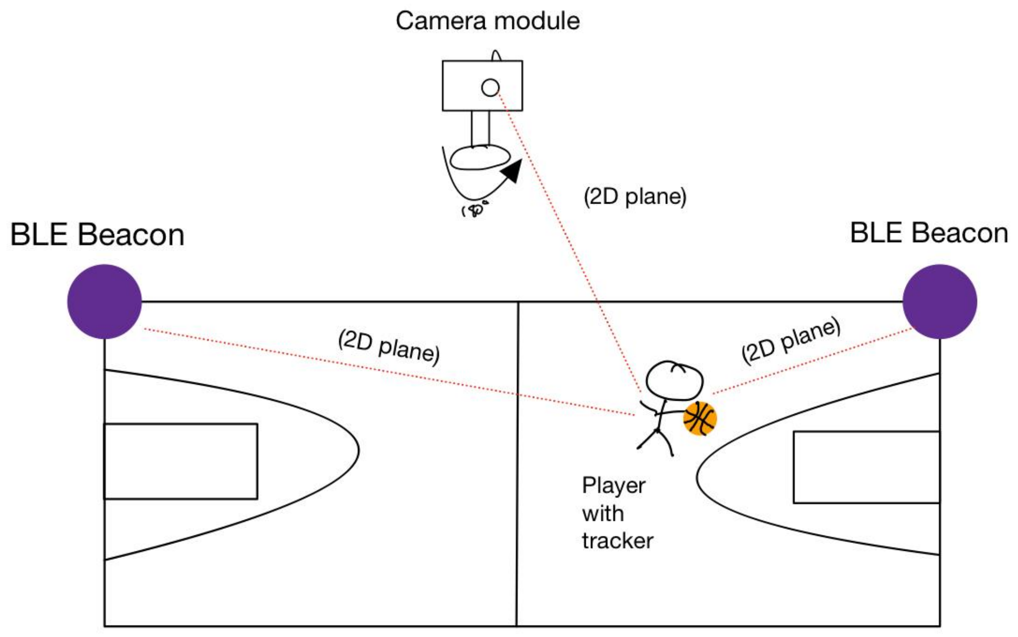

Fig 1: Visual Aid of our project

**High-level requirements list:**

BLE beacons should be able to create a unique 31-byte data packet and have it constantly transmitting, so that the player tracker is able to pick up the data packet.

The location is determined using the iBeacon protocol between the BLE Beacons and the target player's iPhone to determine an approximate position within a 10 meters radius.

The camera system should be able to keep the user within the camera's view for 90 percent of the recording duration and have the person within the frame but not necessarily centered.

## Design

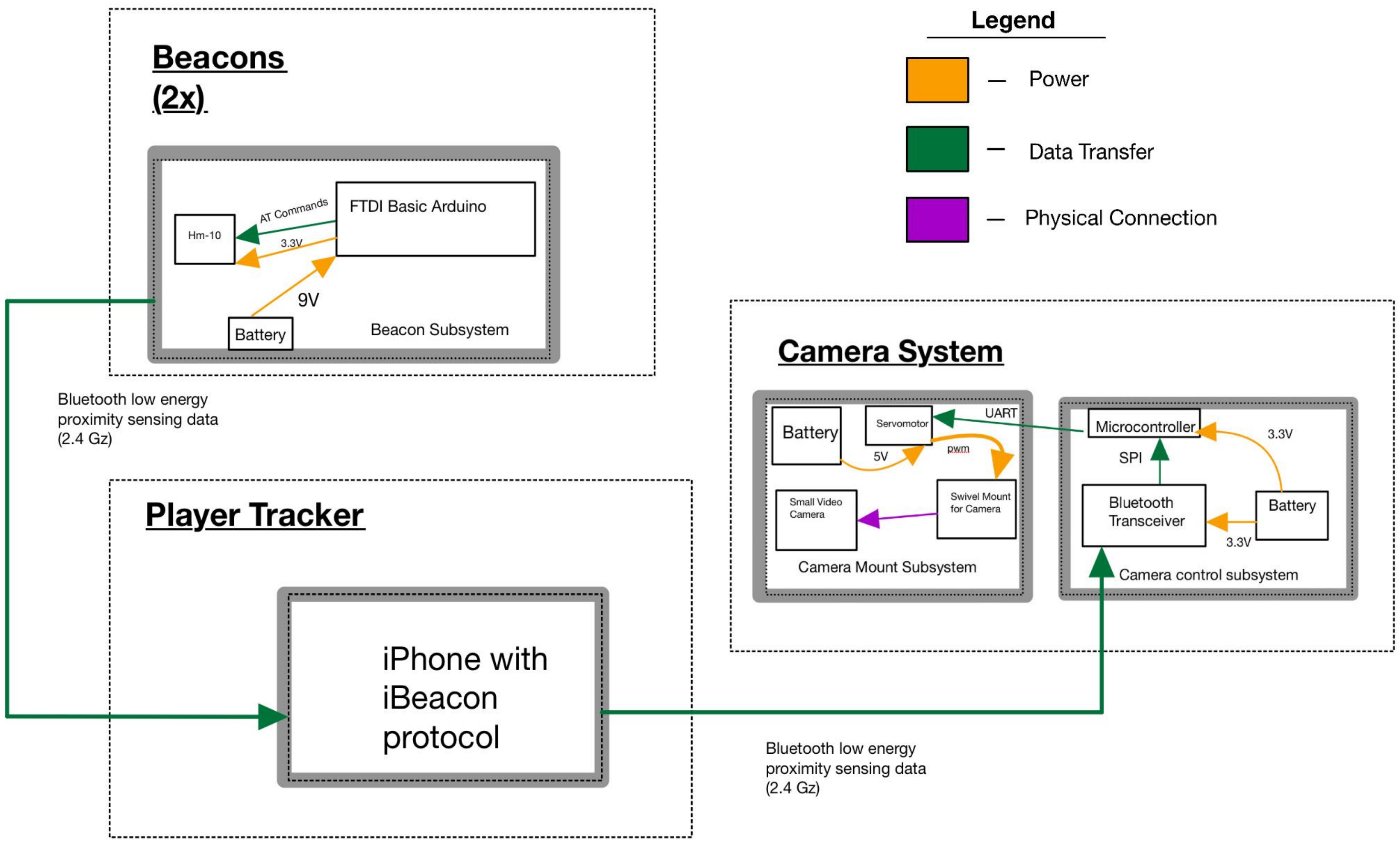

Fig 2: Block Diagram of project subsystems

## Subsystem Overview:

### Beacons:

There will be 2 BLE beacons that will be responsible for sending a packet of 31 bytes of data that can be used in conjunction with the player tracker to determine the relative 2D position of the user. Our BLE beacons are cheaper, faster, and more power consumption efficient than traditional Bluetooth beacons. Beacons send a small amount of data that can be translated in order to calculate position from the receiver. This gives a strong way to get an approximate location of the user so that we can see if that position has left the camera's view or not.

### Player Tracker:

In order to use the iBeacon proctol we are going to use an iPhone so that we can actively track the approximate current location of the client. Since this device is actively being worn by the user it has to be safe and not cause the user danger so we decided to use an iPhone. Our product would be mostly used for physical activity; therefore, it was important to use an iPhone which is already used as a device while doing physical activity which proves it is safe for the wear. This iPhone will be responsible for transmitting coordinates of the person to the camera subsystem, so that the servo can keep the person in the view of the camera.

### Camera Mount Subsystem:

This subsystem is responsible for securely holding the camera and then based on the feedback received the Camera Control Subsystem will have the servos turn the camera smoothly to the position that will keep the user in the view of the mounted camera.

### Camera Control Subsystem:

The Bluetooth receiver gets coordinates from the player tracker. By using logical computation in the microcontroller it is able to determine the turn angle of the servo. The servo then adjusts the position of the camera to keep the user within its line of sight.

## Subsystem Requirements:

### Beacons:

The beacon will consist of a FTDI Basic Arduino, HM-10m and 9v battery that supplies 225mAh. BLE is better than regular Bluetooth because of its low energy usage; therefore, these beacons can be active for ~3 years on a single 9v battery. [1] It is important to mention the difference between BLE and iBeacon. The iBeacon is just a protocol that is used to standardize the data sent from the BLE beacons. This protocol allows us to use iPhones in order to collect the data and keep track of the user's position. BLE chips are also 60 - 80 percent cheaper than regular Bluetooth chips which can make the cost to produce lower and more desirable to consumers.[6] The range of a BLE is 70 meters so there will be full coverage of the player's movements over the course of a game. [7]

***Data transmitted**:*

These beacons transmit a packet of 31 bytes of a unique code sent. Specifically for iBeacon format is ( "iBeacon prefix (9 bytes), Proximity UUID (16 bytes), Major (2 bytes), Minor (2 bytes), TX Power (1 byte)").[6] The most important part from all of this is the TX power because this is used to calculate the distance from the receiver to the iBeacon. The typical speed for BLE is ~3ms which is much faster than normal Bluetooth which is ~100ms. To send the 31 byte data the FTDI Basic Arduino will be working in union with the HM-10 by programming it with AT commands.[2] The AT commands are most commonly used in modems. For this subsystem however, they are important in order to tell the HM-10 chip when to send data and what specific data to send.

*Requirement 1*: BLE Beacon should be able to create a 31 byte unique code and have to constantly send so that the player tracker can pick it up.

*Requirement 2*: The Arduino and HM-10 should be able to properly communicate with AT commands in order to send the data packet.

***Power Consumption requirements***:

There are two major components that will be drawing power inside the beacon:

HM-10 Bluetooth 4.0 BLE UART module. This module transmits Bluetooth data at 2.4GHz ISM band. This will be an arduino plug-in, it will be tapping into the 3.3V, 50mA voltage supply on the arduino board . It has two working modes (sleeping state and active state). During active state, it has a power rating of 8.5mA while at sleep state, it ramps down to 50~200uA.

Arduino UNO: The board has a recommended input voltage of 7-12V. We will be using commercial 9V batteries as the power supply.

**Player Tracker:**

In order to monitor the players' constantly changing position we will be using an iPhone that can be easily worn on anyone's arm or kept in their pocket, for example. An iPhone is already safe to be used while running around and produces the least amount of risk to the user when they use our product.

***Input Data****:*

BLE Beacons send out data about every 10s or so. The iPhone constantly has an app running that will be picking up the 31 byte data packet that is sent from the BLE Beacons using Apple’s iBeacon protocol.[3] The requirement is for the iPhone to be able to actively pick up the data packets from our BLE Beacons and convert them into 2d coordinates and then send them to the Bluetooth receiver connected to the microcontroller on the camera control subsystem.

*Requirement 1*: The app on the phone should be able to gather data from multiple BLE Beacons and calculate the cartesian coordinates and transmit them to the Camera Control Subsystem.

*Requirement 2*: Phone must be constantly listening to the BLE Beacons and constantly sending the changing coordinates to the Camera Control Subsystem.

**Camera Mount Subsystem:**

This will be a pivoted support that permits rotation in the x-axis (from left to right). This camera mount system will consist of the following components:

***Servo Motor:***

This will be rotating the phone and it's holder. The motor will be located at the neck (middle) of the whole camera system. The motor is connected to the microcontroller to determine when to activate the servo, hereby turning the camera phone and in which direction the motor should spin. It will have its own dedicated power supply through a battery, whose size will depend on the servo motor we chose later on in the project.

*Requirement 1*: Motor should be able to support weight of phone and it's holder
*Requirement 2*: It should have the capability to spin forward, turning the phone to the right and backward, turning the phone to the left within 180 degrees
*Requirement 3*:  It should be able to make a full 180 degree rotation within a second
*Requirement 4*: During operation, temperature must remain within safety limits of below 50 degree celsius.

***Mount/Stand:***

This will be built using a selfie stick phone holder in order to keep the phone held securely and have the ability to fit most small cameras. In order to attach this to our servo motor we will have a middle level component that would be 3d printed in order to properly connect the motor to the phone holder, which would be secured to an adjustable tripod base to reach the height of play the camera will record.

*Requirement 1*: Have a mount which can fit a camera phone properly and securely.

**Camera Control Subsystem:**

This subsystem is responsible for receiving coordinates of the user from the player tracker and then needs to calculate if the camera mount should spin or stay stationary. If its stationary means that the person is in the view of the camera mounted. It will be a PCB containing a microcontroller, bluetooth receiver, and it's power supply.

***Microcontroller:***

will gather all the data from the BLE beacon, and the player tracker and then calculate the degree of turn needed for the servo motor to position the phone camera to capture the person.

*Requirement 1*: It must be able to perform computation required to determine the x and y coordinates of the user wearing the player tracker within 1~3ms.

*Requirement 2*: It must be programmable using popular programming languages.

***Bluetooth Receiver:***

This Bluetooth receiver device will be different from the HM-10 chips because HM-10 can only send 31 bytes of data whereas from the player tracker we want more bytes of data to be transferred. This is also important because in the future if we want to send more specific data this will let us do that. Bluetooth is slower than BLE; however, the transmit rate for Bluetooth is ~100ms which is still faster than a person running outside of the window view.

*Requirement1*: Receive coordinates from the player tracker within 100ms and then be able to transfer data to the microcontroller.

***Power Supply:***

All of the components in the camera control subsystem will be embedded on a PCB and will be powered by 5V coppertop commercial batteries.

*Requirement 1:* The battery must be small enough in order to downsize the module at the midsection to keep the overall structure of the whole stand structurally stable.

*Requirement 2:* It must be easily replaceable

**Requirements & Verification Tables**

**Beacons**

| **Requirements** | **Verification** |
|---|---|
| 1. BLE Beacon should be able to create a 31 byte unique code and have to constantly send so that the player tracker can pick it up. | 1A. Will set a test program to listen for what the BLE Beacon is sending and confirm that the correct code is being sent. Also, to calculate the position from each beacon, we can use a measuring tape to make sure the distance is approximately the same. |
| 2. The Arduino and HM-10 should be able to properly communicate with AT commands in order to send the data packet. | 2A. Will write output statements to the console while coding in order to see if the correct AT command is being sent from the HM-10 chip to Arduino. |
| 3. The arduino UNO must have a steady power supply of 9V while the HM-10 bluetooth model will be tapping into the 3.3V output ports provided by the arduino | 3A. Measure the output voltage using an oscilloscope at the power input of the Arduino UNO and ensure it maintains a steady voltage of 9V.<br>3B. Measure the output of the 3.3V output port to ensure the HM-10 can properly run off the supply |

**Player Tracker**

| **Requirements** | **Verification** |
|---|---|
| 1. The app on the phone should be able to gather data from multiple BLE Beacons and calculate the cartesian coordinates and transmit them to the Camera Control Subsystem. | 1A. Set the beacons to a specific constant 31 byte and distance so that we can test if the code is correctly calculating the distance. Then change the beacon to give different coordinates and make sure that output is correct based on the formula we have for calculating distance. |
| 2. Phone must be constantly listening to the BLE Beacons and constantly sending the changing coordinates to the Camera Control Subsystem. | 2A. We will walk around everywhere on the court to make sure that it is able to pick up the BLE Beacon constant code from within the BLE Beacon range. We will walk out of range of the BLE Beacon range to make sure that the phone is still searching for the 31-byte data packet. |

**Camera Control Subsystem**

| **Requirements** | **Verification** |
|---|---|
| 1. Provide 5V +/- 10% from a 4V-12V power source | 1A. Measure the output voltage using an oscilloscope and ensure it maintains a steady voltage of 5V allowing a 10% deviation |
| 2. The power source can supply output current of 250mA-400mA for at least an hour of continuous use | 2A. Ideally we would love to use the PCb as the test case for this verification but due to unavailability, we will be using the simple circuit in Figure 3 for this test.<br>2B. Connect probes across the battery to ensure the voltage test in 1A is satisfied.<br>2C. Adjust variable resistor between 12.5 ohms to get the maximum current (400mA) and 20 ohms to get the maximum current (250mA)<br>2D. Record the time to ensure the test satisfies our conditions for at least an hour. |
| 3. Maintain temperature below 75° C | 3A. During verification of requirement 2, use an IR thermometer to measure the temperature of the PCB over 5 minute |

| | |
|---|---|
| | intervals |
| 4. Receive coordinates from the player tracker within 100ms and then be able to transfer data to the microcontroller. | 4A. Check if the bluetooth is able to pick up a single constant data packet and run a time module on the code to see the time elapsed from the iPhone to the bluetooth chip. |
| 5. The battery must be small enough in order to downsize the module at the midsection to keep the overall structure of the whole stand structurally stable and must be easily replaceable | 5A. We will be using 5V coppertop commercial batteries as the power supply and placing the battery holder somewhere on the mount easily accessible to the user. |

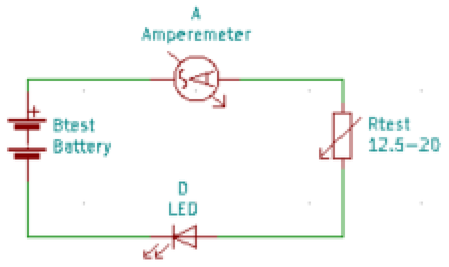

**Fig 3: Test Circuit**

### Camera Mount Subsystem

| Requirements | Verification |
|---|---|
| 1. Servo motor should be able to rotate a minimum of 17 oz load in a clockwise and anticlockwise motion between 0°-180° within a second | 1A. We will be using a function generator to send PWM waves to the signal input of the servo.<br>1B. Every 20ms a high pulse must be sent to the signal input (0.75ms to keep at 0° and 2.25ms to rotate it 180°).<br>1C. We will be alternating sending the signal pulses and measuring the time the servo takes to make the rotation |
| 2. Power supply must provide a stable voltage of 5V +/- 10% (power coming from camera control subsystem PCB) | 2A. We will be using one or two 5V coppertop batteries to send 5VDC to the power input of the servo motors<br>2B. Measure the output voltage using an oscilloscope and ensure it maintains a steady voltage of 5V allowing a 10% deviation |
| 3. The power source can supply output current of 140mA+/-50mA for at least | 2A. While conducting experiment 1 and 2, we will be measuring the current flowing into the |

| | |
|---|---|
| an hour of continuous use | servo from the battery using an oscilloscope 2D. Record the time to ensure the test satisfies our conditions for at least an hour. |
| 4. Maintain temperature of servo motor below 50°C | 3A. While conducting experiments 1,2, and 3, use an IR thermometer to measure the temperature of the Servo over 5 minute intervals |
| 5. Have a mount which can fit any camera phone properly and securely | 5A. Before we purchase the mount, we will make sure it is versatile enough to fit any popular phones |

## Circuit Schematics

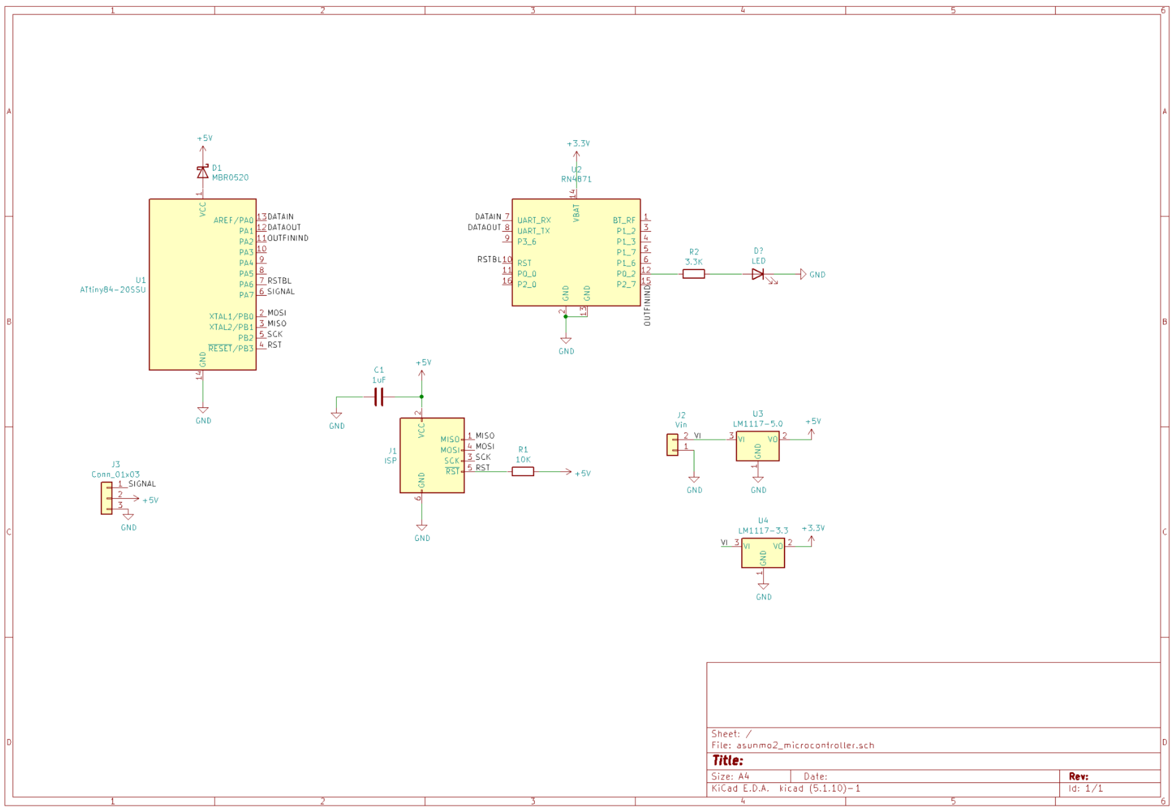

**Fig 4: Schematic of camera control subsystem PCB**

**Tolerance Analysis:**

An aspect of our design that poses a significant risk to the successful implementation of our project is the speed of the camera mount. More specifically, the issue of making sure that the camera is oriented towards the player over the course of a fast paced game is one which presents us with a unique challenge. If the camera does not move fast enough it will make the endeavor ultimately fruitless as the player will not be in frame, as desired. There are two different sub challenges that we are presented with when trying to address this. First there is the issue of input delay. As you may imagine, the player location data takes time to be sent and processed by the microcontroller. The second problem associated with this is that the camera will have to move fast enough to ensure that given the current location data, it is able to swivel at an adequate rate such that the player is actually captured, and hasn't already moved to a completely different part of the playing area.

Through our, quite literal, field work we have developed a strategy to mitigate this issue. First, we intend on operating the camera with a wide angle display. This allows us to optimize the playing surface coverage per frame. Furthermore, BLE technology operates even faster than the already fast Bluetooth technology. By taking roughly 103ms to send and receive data (3ms for the BLE beacons and 100ms for the Bluetooth receiver) we functionally eliminate the issue of input lag as the system will operate at a much faster rate than the player. What this means is that no matter how fast the athlete is moving, our design will operate faster and have a large enough buffer such that they are always within the recording.

An average person runs across the court in 14-15 seconds and that is when a person is making a fast break fully rested. For the purpose of this calculation assume the camera is currently on 0.5x zoom that would cover half of the court. The communication between the BLE beacon and the iPhone takes approximately 10ms.

At this point the iPhone will get passed a distance in meters from the 31-byte data packet. The distance passed is how far the user is from the specific BLE Beacon within a .5 meters. The iPhone will have two distances from both beacons which it will send back to the bluetooth device. The communication between the iPhone and the bluetooth device on the microcontroller has a communication rate of ~100ms.

Then the microcontroller receives distance_x1 and distance_x2. The size of the basketball court is 28 meters. So distance_x1 = poistion_x1 but 28 - distance_x2 = position_x2 since there is some variability in the accuracy you take (position_x1 + position_x2) / 2 = position_final. This position is used to check if it is in frame of the x_1_frame and x_2_frame which represents the view frame of the and check if the position_final is within it. The total time taken to calculate the x-position of the user is approximately ~110ms which is way less than the 7   secs that a person takes to run from one side to halfcourt.

## Cost and Schedule

**Cost Analysis:**

**Labor:**

We will be using the average salary an ECE graduate from the University of Illinois makes to determine our hourly wages of $42.5/hour. We will each be investing at least 8 hours a week into our project, which brings the total cost of labor to:

$42.5/hour x 8hours x 3 partners x 16 weeks = $16,320

We have not received a quote for the machine shop labor hours yet but we estimate that the work should not take more than 8 hours x $42.5/hour = $340

**Parts:**

| **Item** | **Description** | **Quantity** | **Cost** |
|---|---|---|---|
| 1. Servo Motor | Parallax Standard Servo (Parallax #900-00005) | 1 | $30.01 |
| 2. Arduino Uno Rev3 SMD | ATmega328P microcontroller (Arduino A000073) | 2 | $21.90 x 2 |
| 3. HM-10 Bluetooth 4.0 ble module | DSD TECH ML-HM-10 | 2 | $11.99 x 2 |
| 4. Polycase (mount) | A polycarbonate cubic box of size 6.30in x 3.15in x 3.35in | 1 | $20.00 |
| 5. Polaroid 72 inches Premium Tripod | Polaroid PLTRI72 | 1 | $29.99 |
| 6. Car phone mount | Any generic mount | 1 | $15.00 |
| Total | | | $162.78 |

After considering all the individual costs, we have a grand total of:

$16,320 + $340 + $162.78 = $16,822.. by 78

**Schedule:**

| **Week** | **Work** |
|---|---|
| **10/4** | **Shivang -** Design review, Research BLE data transfer with arduino and HM-10<br>**Aksh -** Design review, Start building physical phone holder<br>**Ore -** Design review, Order parts |
| **10/11** | **Shivang -** Finalize PCB<br>**Aksh -** Finalize PCB<br>**Ore -** Finalize PCB, Oder PCB |
| **10/18** | **Shivang -** start writing code to program the BLE Beacons<br>**Aksh -** start writing/researching AT commands to program the BLE Beacons<br>**Ore -** Work on wiring the BLE Beacon |

| | |
|---|---|
| **10/25** | **Shivang -** Finish testing the output of the BLE Beacon<br>**Aksh -** Setup the iBeacon protocol and test to see if the iPhone can pick up a constant signal<br>**Ore -** If PCB arrives start working on soldering the PCB or finish building the second BLE Beacon |
| **11/1** | **Shivang -** Worked on building the stand and the PCB<br>**Aksh -** Test to see if data gathered from the phone can be sent to the bluetooth chip on the microcontroller<br>**Ore -** Worked on building the stand and the PCB |
| **11/8** | **Shivang -** Test the whole system and make changes to fine tune<br>**Aksh -** Test the whole system and make changes to fine tune<br>**Ore -** Test the whole system and make changes to fine tune |
| **11/15** | **Shivang -** Use feedback from mock demo to improve final presentation<br>**Aksh -** Use feedback from mock demo to improve final presentation<br>**Ore -** Use feedback from mock demo to improve final presentation |
| **11/22** | **Shivang -** Fine tune the stand for phone<br>**Aksh -** Fine tune the stand for phone<br>**Ore -** Fine tune the stand for phone |
| **11/29** | **Shivang -** Demo / Write final paper<br>**Aksh -** Demo / Write final paper<br>**Ore -** Demo / Write final paper |
| **12/6** | **Shivang -** Write final paper<br>**Aksh -** Write final paper<br>**Ore -** Write final paper |

## Ethics and Safety

The most relevant ethical concern is related to ACM code of ethics standard 1.6 - about respecting data privacy [5]. This issue is pertinent to people’s personal data being secure and being obtained and used responsibly. In our project we do collect tracking data but we avoid any issues related to data privacy. We do this in two ways. First, we do not store the data for long periods of time. The data input is used instantly and cleared as new data comes in. There is not any sort of memory of location that can be accessed by those with bad intentions. Additionally, if the instantaneous data were to be captured, the data being collected does not disclose any new information as the person turns on the camera and is standing right in front of it so there is no issue of location data being revealed.

As a result of physical devices being used, there is a possibility of them interfering with human movement. According to the IEEE Code of Ethics section I subsection 1. we must “hold paramount the safety, health, and welfare of the public” [4]. We can adapt the usage of our project to not interfere with the game or the players in a way that causes harm. We can suggest to the user to place the BLE beacons behind the hoop/net/playing surface so that they don't come in contact with the players. We also have the camera on a mount so that it can be placed a safe and reasonable distance away from the players such that it does not interfere with the game and cause anyone any bodily harm. Given the extensive range of BLE and Bluetooth users are able to operate the technology without risk of interference and potential injury through collision.

## Citations:

[1] “Bluetooth Low Energy (BLE) Beacon Technology Made Simple: A Complete Guide to Bluetooth Beacons.” *Beaconstac RSS*, blog.beaconstac.com/2018/08/ble-made-simple-a-complete-guide-to-ble-bluetooth-beacons/.

[2] “Determining the Proximity to an IBeacon Device.” *Apple Developer Documentation*, developer.apple.com/documentation/corelocation/determining_the_proximity_to_an_ibeacon_device.

[3] “How to USE HM-10 Ble Module with Arduino to Control an LED Using Android App.” *Circuit Digest*, 4 July 2019, circuitdigest.com/microcontroller-projects/how-to-use-arduino-and-hm-10-ble-module-to-control-led-with-android-app.

[4] “IEEE Code of Ethics.” *IEEE*, www.ieee.org/about/corporate/governance/p7-8.html.

[5] “The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.” *Code of Ethics*, www.acm.org/code-of-ethics.

[6] “What Is Ibeacon? All You Need to Know.” *Bleesk*, bleesk.com/ibeacon.html.

[7] “What Is Ibeacon? A Guide to Ibeacons.” *IBeacon.com Insider*, 13 June 2014, www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/.