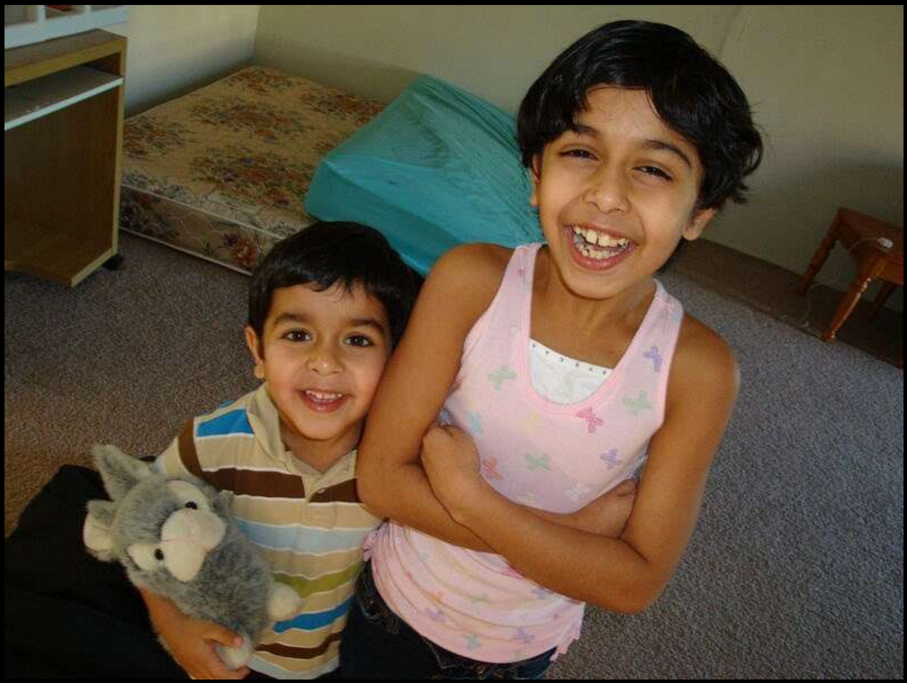


Hi, I’m Aum. When I was 10, my sister Palak was diagnosed with autism and ADHD. Though I didn't understand what that meant at the time, I knew she was the same sister who rescued me from drowning and helped me scoot a cardboard box as I learned to walk. Palak had outbursts of frustration and anger as her schoolwork grew in difficulty, but my parents never scolded her. Instead, they calmly guided her through assignments, and I witnessed the value of patience.

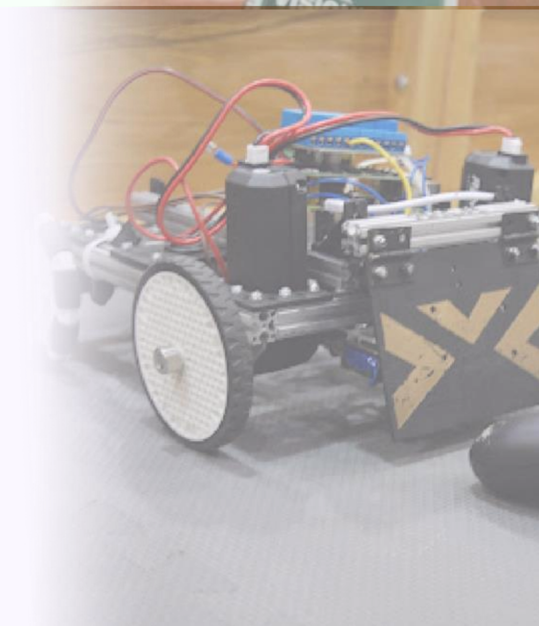
Months later, my parents registered Palak and I for Taekwondo classes. When I noticed she struggled with a roundhouse kick during class, I adopted my parent’s attitude and spent time practicing the form with her at home. The next week, she successfully roundhouse kicked her sparring partner.

Palak is a large part of who I am. She’s shown me that people face underlying struggles and this has redefined my approach to new situations with a more sympathetic perspective. She’s taught me to be proactive rather than reactive—a philosophy I incorporate in all my activities.

To this day, I continue to take care of my sister. Just last week, we sat together and learned how to convert braille to Roman numerals for her General Mathematics class. It’s nice seeing her excitement as we figure out problems side by side. Yes, there are bumps along the path, like debating who gets the last roti, but she’s given me a completely new outlook on life, and I will be forever grateful for her.

ACTIVITIES

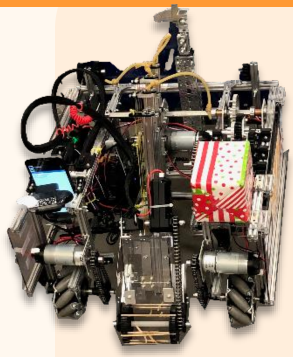
Table of Contents



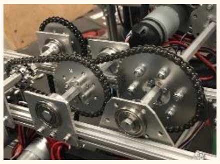
Worked with Nicholas Harty, Cayden Keene, Max Bazan, and Ishaan Ahmad

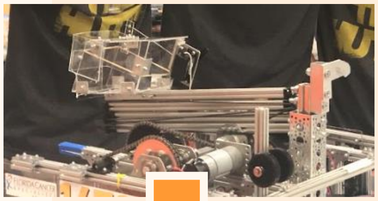
Robotics

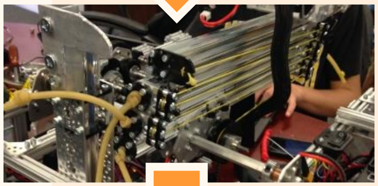
Year One: *"The Rookie / The Grabber"*

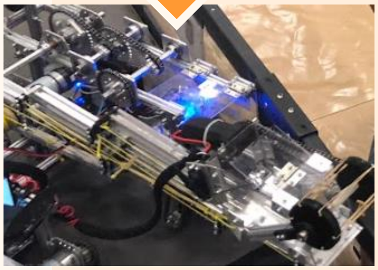


Challenges	Solutions
Building a stable linear actuator	Using direct drive to rotate the collection arm was unstable as the arm would fling from the forward to backward position, nearly breaking itself on every rotation. The linear actuator was held back by surgical tubing that would wear over time and stretched forward by a yellow string on a makeshift wooden spool.
	In the short term, to solve the problem with lifting the arm, I created a geared-down system to increase the precision of arm control on a much larger HEX shafting, shifting from REV parts to GoBilda parts. This switch eventually paid off.
	After a couple of months from watching videos on seasoned teams and meeting with a mentor from BlueSparq, a local engineering firm, Nick and I altered our design by removing the surgical tubing (which had been held inside the GoBilda rods) and jumping into Fusion360 to CAD design a new dual-string spool that had an additional inverse string to provide pulling tension. This new spool, after a few iterations, lasted us the rest of the season as it rarely tangled, and its wear was minimal. There was that one time it snapped after a faulty autonomous period, but it was ultimately solved.









Challenges	Solutions
Steady motion in Autonomous	Originally, we relied on timed movement for the first few competitions as we worked out the quirks of arm motion.
	After talking to other programmers in our league, we explored encoder motion, relying upon the built-in GoBilda encoders for navigation.
	Encoder motion revealed its inaccuracies during precision turning. As a result, we explored gyroscopic options after scavenging online threads on the subject. Eventually settling on using the built-in IMU and a while loop got us through the rest of the season.
	During the offseason, we conducted research on odometry-based robotics which would eventually be the basis for our Odometry-IMU-Vuforia correction navigation system of the third year.

```
public void turn(double powercus,int sleeptime ) {
    front_left.setPower(powercus);
    front_right.setPower(powercus);
    back_left.setPower(powercus);
    back_right.setPower(powercus);
    sleep(sleeptime);
    front_left.setPower(0);
    front_right.setPower(0);
    back_left.setPower(0);
    back_right.setPower(0);
}
```

```
public void turnDeg (  double ang    double speed) {

    // keep looping while we are still active, and not on heading.
    while (opModeIsActive() && !onHeading(speed, angle, P_TURN_COEFF)) {
        // Update telemetry & Allow time for other processes to run.
        telemetry.update();
    }
    lastAngle = getTrueAngle(angle);
    telemetry.addData("Last Angle:", lastAngle);
}
```

Summary & Takeaways

My first year as a founding member of Java the Hutts is a time I will treasure forever. After attending a "Mathletes" camp at FGCU, Cayden, Nick, and I set forth to create a robotics team and truly challenge our perception of engineering. We first established meeting dates, then roamed YouTube and REV for any reasonably priced starter kits, and finally built a working chassis. The progress in the beginning was slow and could've easily fallen apart given that we were the first team within our area, but powering through and challenging the process made the community we cultivated so much more worth it. Personally, I consider this robotics journey as my formal "Homecoming" into the function of an engineer. My first year taught me how to fundraise, presenting in front of rooms full of local engineers and educators. My first year taught me the importance of trusting initiative and the worth in believing in a long-term goal. However, above all, my first year taught me how to work long hours with others and converge multiple ideas with an open mind.

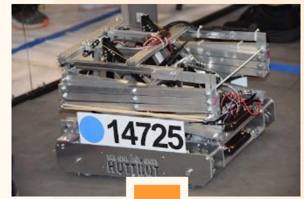
Worked with Nicholas Harty, Cayden Keene, Max Bazan, and Ishaan Ahmad

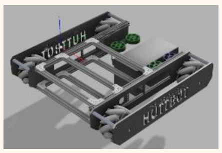
Robotics

Year Two: *“The Tall One”*

Challenges | **Solutions**

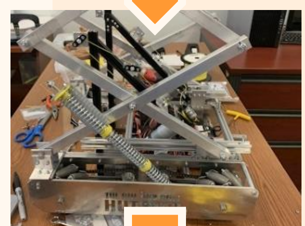
“Climbing the mountain” with scissor lifts





The scissor lift was no doubt the most complex and thought-out physical portion of the bot. Our decision to use a scissor lift over any other mechanism had two main reasons for it. One was due to the supposed speed advantage over other designs; the other was its relative simplicity, though it didn’t turn out to be simple in the end. Our first version of the scissor lift used ¼ inch metal tubes as the main chassis of the lift. This was our first mistake, as it was far too heavy for our motors to handle.

Our second course of action was to order and mount gas springs to try to alleviate some of the pressure being put on the motors. This was nearly perfect, as the dampening on the springs made the lift very smooth, yet quick going up and down. Unfortunately, we found out later that these springs did not meet regulation. Considering this, Nick and I looked for further solutions.



This was a significant drawback and caused us to order very strong metal springs and 3D print connectors for them. The other problem was the connections between scissor lift segments: We used ¼ inch screws, causing tremendous amounts of friction and contributing to the overall problem. This created a sluggish design that would need to be addressed later in the season.

To solve this issue, we changed to a thin C-channel with bearing cutouts and used bearings instead of screws as connections. Our design varied greatly, transforming from a string and pulley system to a screw drive to lift the mechanism up. These changes enabled the lift to rise in under two seconds and improved stability. We also implemented 3D printed “X” blocks to prevent the scissor lift from swaying and worked on the top grabber that was comprised of a plexiglass base with a three-arm extender. This mechanism allowed us to lower the grabber while remaining parallel to the ground.

Learning from new resources

With the new resources of our partnership with the I Will Mentorship Foundation, I set out to incorporate as many self-designed elements as possible into our robot’s design (shown by the bright yellow filament). These included custom pulleys, “X” blocks for the scissor lift, and rounded hanging-bars for our three-arm grabber. Spending so much time working on internal manufacturing made Fusion 360 a close rival of Android Studio for most of our second season’s time.

In addition to 3D printing, one of the first elements I worked on in Fusion 360 were the side panels. After contemplating the final design for days, I sent the design over to be laser cut (a service provided by our sponsor “Bob Dean Supply”) and permanently dubbed our year two robot as “The One and Only Huttbot.”

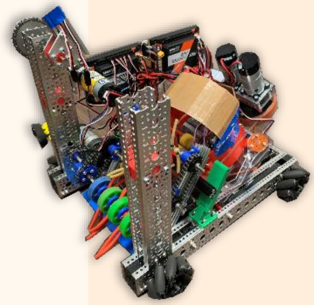
Summary & Takeaways

In my rookie season with Java the Hutts, I was exposed to the variety of tools, designs, and possibilities that grew my aspirations for the second season. After partnering with the I Will Mentorship Foundation, a local nonprofit committed to STEM education, we gained access to their inventory of machinery that would later culminate in the makerspace we constructed that summer. This gave us opportunities to experiment with in-house CNC machines (aluminum/wooden), a lineup of MakerBot 3D printers, and a larger space to host team events. My eyes were opened to wonders of industrial engineering and working on fun cutouts to my heart’s delight. The nostalgia I feel looking back on this second year is mostly surrounded in my improvement with CAD and the iteration on iteration of parts that became crucial to our robot’s final design.

Worked with Nicholas Harty, Cayden Keene

Robotics

Year Three: *"The Launcher"*



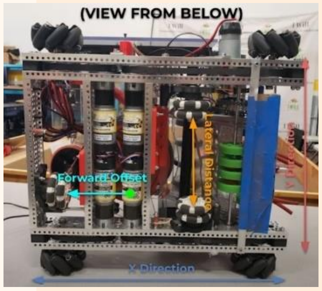
Challenges

Solutions

Improving localization during the Autonomous Period (using three-wheel odometry)

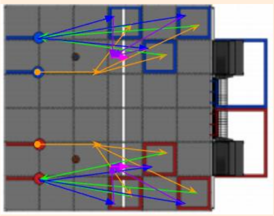
Custom designed and printed odometry modules

After spending the summer learning and implementing a modified pure pursuit algorithm, we discovered its inaccuracy as it relied upon a constant point mass and velocity measures. Neither of these options were viable on the new robot due to the rings' necessary concentration on one side of the robot (as opposed to the empty alternative) and the inconsistency of velocity readings coming from odometry as we made arc movements.

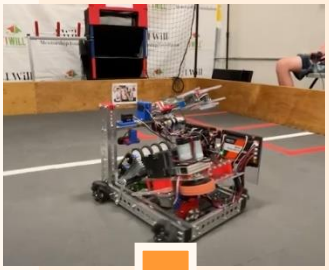


Moving quickly to provide any benefit to localization after postponing our pure pursuit build, we turned to a mature library titled Roadrunner for FTC. After making modifications to support three-wheel odometry systems and appropriate PID curves, we were able to develop necessary arc movements that maintained accuracy (~10cm of error at the end of the period) for the entire autonomous period. These measures would ultimately become less important as the following ring-shooting alignment algorithm sustained mid-autonomous/tele-op correction.

Autonomous plan



Turning to match the angle of the high goal:



Shooting the rings with proficient accuracy during both the autonomous and tele-op (playing) periods

Originally, we relied upon the value of our IMU gyroscope for angle positioning to align with the top goal. However, this process was tedious and required that we begin the automatic motion of turning to a particular angle at a specific position in front of the goal.

We were met with similar failure when implementing constant odometry position. During tele-op, the coordinate positioning would often freeze over time, so the driver would have to restart the robot to orientate the coordinate variables.

Finally, after a late-night meeting, we were able to achieve near-perfect accuracy by relying on a constant: vision. By maintaining a background Vuforia program that would provide angled dimensions from the robot's position to an image played underneath the goalpost, we were able to use a responsive angle that would provide the basis for a delta turn and could adjust for any position (and any distance given the relative size of the image) on the field. It was Pythagorean math from 7^{th} grade geometry that we applied in the real world. Playing with the mechanic for hours, I was in awe at the simplicity of tapping a button and automatically shooting three rings into their netted capture across the field.

Summary & Takeaways

Going into our third year, we wanted to challenge ourselves by going back to square one of robotics: navigation. Up until this point, our autonomous program relied upon rigid/perpendicular movements and lacked vision as a key to motion correction. After working to design odometry modules within the previous season, we set my sights on spline movements that would target specific coordinates on the field as opposed to the delta movements of the past. In addition, the new season lacked significant y-maneuvers (bumps, hills, etc.), and fixed goal targets almost required a refinement of autonomous actions during the playing period to have any consistency in shooting rings into the highest goal. A large problem that plagued the development period of our third season was our limited active pandemic members. Being a team of three (all founding members), we had to be concise in our plans for the season and focused on our set responsibilities. These goals and challenges set the stage for my third season, and looking back, continues to evoke my innate passion for problem-solving and crawling around on foam tiles trying to view the world through the robot's perspective.

Worked with Nicholas Harty and Cayden Keene

Software & Product Development

Pathways: A White Cane Navigation Attachment

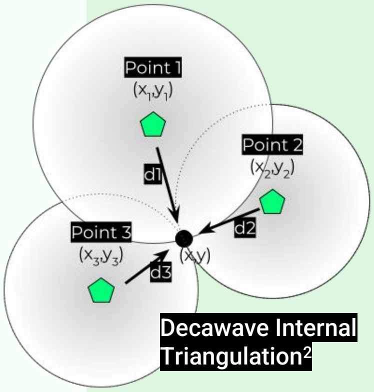
Why?

Although there have been many innovations in assitive technology, most modern exercise utilities, either physical or mobile, don't provide the proper platform to suit people with vision disabilities. With the introduction of mobile positioning and UWB (Ultra-Wideband) technology[1] in the early 21st century, existing and emerging fitness brands invested in technology that would track the well-being of users and display performance data. However, one target market mostly untouched by this overflow of innovation has specifically been visually impaired individuals who often experience the bulk of issues when it comes to maintaining their health. This became ever clear upon meeting with the SWFL Council of the Blind and refining Pathways, our product, through first-hand accounts that put the experience of blind individuals into perspective. Through our innovation, an attachment to the original white cane design (titled the "GEOCane"), we believed we could alter this status quo and truly change the confidence of all partially-sighted individuals in their ease of navigation.

MDEK1001 Beacon[1]

How? & Goals

The design of our product/solution works on existing technology called UWB (Ultra-Wideband). UWB is a radio wave-based technology that's purpose is to transfer data and determine location through radio waves. In our case, we are using UWB technology in combination with triangulation, similar to the triangulation used in GPS satellites. Pathways is a system using UWB radio beacons and receivers to track the exact positioning of a visually impaired user. Beacons consist of the MDEK1001 circuit board (UWB beacon/receiver) connected to a constant power source like crosswalks, streetlights, and traffic lights. The beacons have a radius range of 150 Meters allowing the beacons to be placed at least 300 meters apart.



Decawave Internal Triangulation[2]

$$(x_1 - x)^2 + (y_1 - y)^2 = \boldsymbol{d_1^2}$$

$$(x_2 - x)^2 + (y_2 - y)^2 = \boldsymbol{d_2^2}$$

$$(x_3 - x)^2 + (y_3 - y)^2 = \boldsymbol{d_3^2}$$

$$\boxed{\{y\}} = \frac{(x_2-x_3)\,[\,(x_2^2-x_1^2) + (y_2^2-y_1^2) + (d_1^2-d_2^2)\,] - (x_1-x_2)\,[\,(x_3^2-x_2^2) + (y_3^2-y_2^2) + (d_2^2-d_3^2)\,]}{2[\,(y_1-y_2)(x_2-x_3) - (y_2-y_3)(x_1-x_2)\,]}$$

$$\boxed{\{x\}} = \frac{(y_2-y_3)\,[\,(y_2^2-y_1^2) + (x_2^2-x_1^2) + (d_1^2-d_2^2)\,] - (y_1-y_2)\,[\,(y_3^2-y_2^2) + (x_3^2-x_2^2) + (d_2^2-d_3^2)\,]}{2[\,(x_1-x_2)(y_2-y_3) - (x_2-x_3)(y_1-y_2)\,]}$$

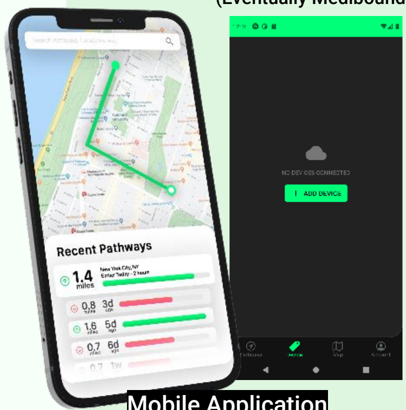
User are tracked with the Geocane. For the triangulation system[2] to work properly, a user would have to be within the range of 3 beacons. By taking the distance from each beacon to the receiver, we are able to calculate the exact positioning of the user by triangulation. The receiver (MDEK1001) for the Geocane is placed inside of a lightweight housing that simply clamps onto a white cane. After meeting with Will Alexander, a representative of the SWFL Council of the Blind, we got feedback on the design and functionality of our cane. The cane needed to be lightweight so it wouldn't strain the user's wrist, especially since they rely on tapping the cane back and forth to feel the environment around them. When designing the cane, we took into consideration the weight of our module to ensure that our device would maintain the physical feedback of a normal white cane. Within our Geocane module, we include integrated features like haptic and audio feedback to guide the user. The haptic feedback is provided to the user through a vibration motor, and the audio feedback is provided through a Bluetooth module.

The Beacon-Module System in Action

Over BLE, we planned to allow the user to connect headphones to hear audio instructions. When the user is tapping the cane in the wrong direction, the Geocane will begin to vibrate and the audio will instruct the user to travel either left or right. When the user is traveling in the correct direction, the Geocane will stop vibrating and the audio will instruct the user to continue traveling forward. We also incorporate an interface that scans for UWB beacons using the device and forms a virtual boundary. When nearing the boundary, the device produces a variety of vibrations to alert the user of their distance from the side of their pathway. This is to ensure the safety of users through a padding system within individual pathways. In this sense, the user experience is meant to be intuitive and easy to use as a stand-alone solution. Its longevity is maintained through the use of additional systems to monitor the progress of the user. Such a system is meant to build confidence within the end-user as they move from simply learning to navigate with this device to use it for more advanced forms of exercise. On top of this, our consumer app (early build of Medibound)[3] allowed for family members and other loved ones to monitor and create routes for the visually impaired individual.

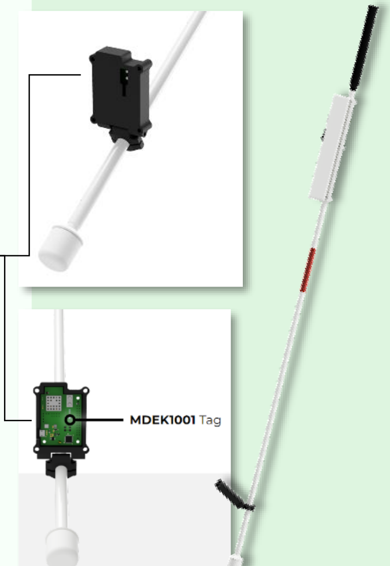
Photoshopped Design

React Native Partially Functional Build (Eventually Medibound)



Mobile Application Designs[3]

Challenges	Solutions
Figuring out localization	**RFID:** Passive RFID systems, within a considerable production cost, often provide limited detection range (<1m in radius). Active RFID modules use up a lot of energy upon constant transmission. **BLE:** Only provides double the accuracy of Wi-Fi based solutions with the localization of individuals being within <8 meters of their actual location. Dependent upon Bluetooth and may vary in success based upon the density of other mobile devices. **UWB:** UWB has a very high accuracy to less than 30 cm when triangulating a location and within in range of 3 beacons/detectors. A beacon/detector has a max range of 150m. With a longer range than BLE , UWB beacons can be spaced further apart, limiting the number of beacons needed triangulate location within in a large area.
Response to feedback and the countermeasures of the lack of initial sample research	When we first were inspired to explore improvements to white cane navigation, we let our ideas run away with us. Before we knew it, we had a solution that was too heavy on the hand, too complex to be taught, and too expensive to implement… a critical truth that emerged upon first meeting with the local Council of the Blind. This reality fostered improvements such as audio-hapic voice assistance, through experimenting with Google Directions API, and modifications to the split layout[4] of the final design as a means to balance the weight of the device[5]
Disconnecting Decaware's software… connecting the device to our application	Other challenges existed primarily in the software. By using Decawave's (the manufacturer of MDEK1001) hardcoded triangulation software, we were able to modify the firmware in SEGGER to enable BLE mode. This detached the board from its connection to Decaware's mobile app and allowed us to receive serial data on the Raspberry Pi. Using that data, we exported the terminal stream to a NodeJS program that utilized the library bleno to reproject that data to the React Native mobile app. Although this solution did present some latency, it wasn't enough to provide significant resistance to border detection and it served as a good opportunity to develop BLE technology.



Post-Meeting Split Design (Beacon Bottom – Control Top)[4]

Outer Encasing Of Final Design[5]

Takeaways & What's Next

- π Although the product wasn't entirely viable for commercial use, it proved to be a powerful instrument to connect us with a pocket of our community and realize the amount of research necessary for successful product development.
- π IoT is quite interesting. The complex data relay system described in the "Challenges" section took a while to construct, but, after it was finished, I was blown away by both the accuracy of the localization technology and the ability to manipulate technology over BLE through multiple consoles/centers of code… an experience unbeknown to me in the world of FIRST Robotics

Worked with Nicholas Harty

Software & Product Development

Medibound: An IoMT Management App

Current Branding Standard:

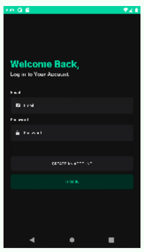
SVG, I made in Illustrator | Gradient Version | App Icon

medibound
Roboto Medium 12px
Lowercase
Font/Text Styling Standard

Why?

Following the Pathways project, the mobile application we developed in the process of that research had broader implications beyond its use of navigation accessibility. After renaming the application component of the project to Medibound, my partner and I redefined our mission: to connect the average user to the larger science of predictive medical health (often found in IoMT NN devices).

Version 1: Continuing to work on the application, we initially focused on increasing the security of medical users by creating a two-way gate between health providers and patients[1] within the app (Using Firebase Auth for login).

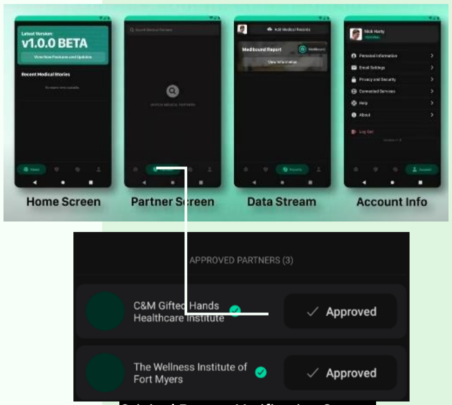


Version 2: However, we eventually dropped the security component, opting to connect with existing security-based medical data-sharing platforms such as Apple/Google Health, and increased our focus on creating a mobile dashboard for managing IoT devices and introducing simplicity to data presentation. Throughout the entire process of designing this application, a Photoshop canvas became a dear friend in solidifying design ideas and fulfilling an internal desire for affordance. We saw this idea as an opportunity to learn more about constructing mobile utilities with React Native and connecting such big data to Firebase, and later the broader IoT Google Cloud platform. (Login/Account page roughly stays the same)

How? & Goals

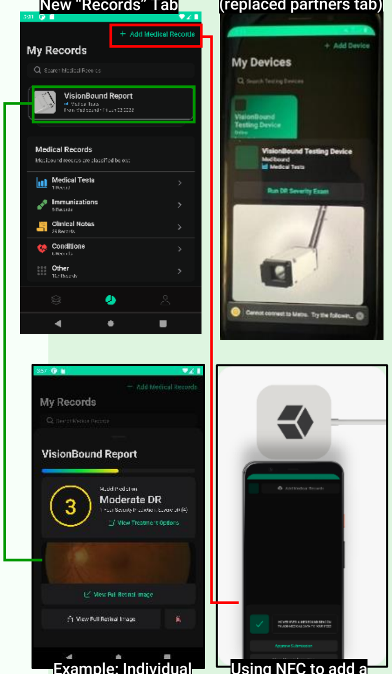
Medibound was developed in React Native and served as the primary introduction, within our education, to NodeJS. I saw NodeJS as a marvel when compared to traditional OOP programming languages. Many of the edits we made to the program would update in real-time without recompiling. The major components in the development of Medibound v1 came with learning Firebase as a data management platform as well as the standards for medical data communication across the web. However, in Medibound v2, feedback received from our local Congressman and the Congressional App Challenge changed our focus to IoMT management. We connected devices over a standard of BLE (using a Raspberry Pi test kit) and incorporated a variety of different methods for extracting data from these devices. These methods included BLE data transfer (for connected devices), Internet/Firebase (dubbed DeviceAPI) transfer, and contactless, anonymous NFC transfer[2]. There was something visionary about connecting these devices to one central system and watching a project that started as a Photoshop sketch transform into something greater.

v1



Original Partner Verification System for Sharing Data[1] (currently in the process of being replaced by FHIR)

v2



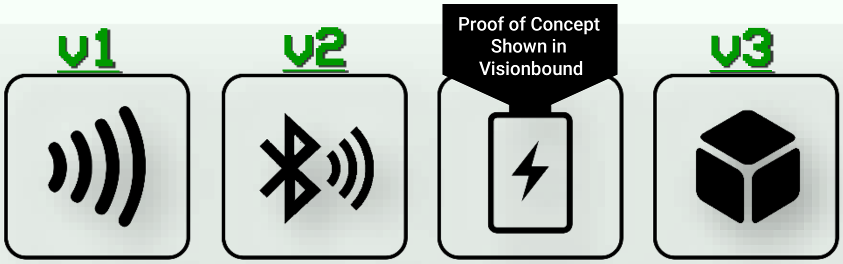
New "Records" Tab

New "Devices" BLE Tab (replaced partners tab)

Example: Individual Record

Using NFC to add a Record[2] (more on process in Visionbound)

Characteristics of a "Medibound" Device (As it stands now):



NFC Contactless / Anonymous Data Transmition (Still working out legality)	BLE Functionality to Connect Device To Medibound Mobile App for Management	Mobility: Battery Powered, Energy Efficient, & Uses Interpretive Neural Networks	Part of the Medibound Artitecture (Device API / Encorporate FHIR)

Challenges	Solutions
The learning curve of moving from a robotics Java environment (or RaspPi) to a NodeJS environment	When first brainstorming our ideas for the application, our team discovered multiple errors within our learning that would need to be overcome if a final product were ever to flourish. Some of these initial obstacles included workflow management and mobile development. Neither of us had experience in proper career programming nor formal cross-platform mobile development; we had come from a background in web-based development. However, we were able to overcome these challenges through the use of project planning/communication software, such as Slack™, and lengthy programming sessions dedicated to learning React Native (our mobile interface framework). We kept a tight schedule as we entered the school year to make sure we didn't fall behind on this learning process. This initial planning certainly laid the groundwork for much of our future collaborations.
Learning Firebase and Server Communication	v1 Further challenges came in the form of server communication and privatizing our users' medical information. After significant research and comparing similar server solutions, we decided to use Google's Firebase server network to hold all of our user information and real-time communication requirements. We ended up resolving upon Firebase because of its expansive API documentation as well as various security utilities that were available across all of our necessary platforms. On top of this, Firebase's ease of use within our NodeJS/Javascript environment allowed for a lesser learning curve and an easier translation of data in the local-server system. v2 However, the advantages of Firebase would also, in some instances, serve as hindrances as we would find bottlenecks within such communication that would, in turn, slow the loading of local elements. To resolve such issues, our team resorted to dividing our IoMT data between Firestore (Firebase's long-form data storage) and Firebase Real-Time as well as transferring less/minimized data between our application and Google's back-end. All in all, these challenges aided our overall learning and helped us become better problem-solvers.

View of React Native Mobile Backend

v1


Explanation of 1st Version of Application:
https://www.youtube.com/watch?v=8DehAuyERTQ

v2


Explanation of 2nd Version of Application w/ Visionbound:
https://youtu.be/QfT6ONe1PDY

Takeaways & What's Next

v3 Going forward, we want to develop a proper API platform (as opposed to our makeshift NextJS one) to allow developers creative freedom in curating different layouts of medical data and ultimately adopt the proper security features to commercially comply with government regulation (this product, although complete in its basic functionality, is far from commercial launch). Learning and incorporating FHIR through their set of standards will surely help with this as we continue to shift into a more industrial-ready product.

π I learned a lot more about the limitations and audience of mobile health solutions as well as the current standards and trends within the emerging mobile IoMT device industry (the subject of a later study we would do).

π I grew ever ambitious in changing the standard of IoMT development to incorporate the data transfer methods we had found to be great benefits to the consumer. In this sense, we grew eager to learn more about predictive health algorithms and developed our own "Medibound" IoMT device.

Worked with Nicholas Harty

Software & Product Development

Visionbound: A Medibound Device for Classifying DR

Why?

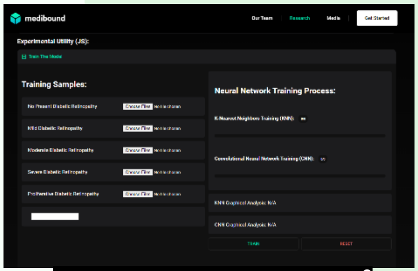
Initially, my grandma's diabetic complications drew me into a Kaggle post on diabetic retinopathy (DR) neural network detection. Although I knew little about neural networks, I was eager to learn and apply such knowledge to my perspective on IoMT devices. Little did I know that this passion project would transform into an entrepreneurial journey that would sweep my Junior and Senior year. After conducting further research on diabetic retinopathy, I learned that 79% of the adults with diabetes reside within low-to-middle income nations and serve a major risk of joining the estimated 93 million people worldwide suffering from diabetic retinopathy. This information changed my perspective on the disease and provided a convergence between my interest in neural networks and the IoMT mobility initiatives of Medibound. Dubbing this project, Visionbound, My partner and I embarked on a journey to compare existing neural network solutions (CNN/KNN) with respect to DR and develop a standard around DR testing for what a "Medibound" device should look like[1].

Final Visionbound Render[1]

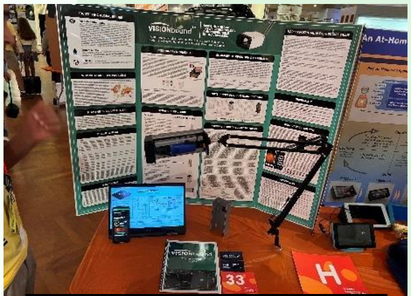
How? & Goals

The Learning Comparison Study (CNN v. KNN):

To study such differences, we developed two closed algorithms through TensorFlow API (ml5.js port to the webJS environment) to serve as an accessible web utility. After preliminary setup, we began training the algorithm with samples at increments and testing their accuracy via their developed online/public platform. This platform was built in HTML/JS that was pasted into an existing WordPress site (https://medibound.com/project)[2]. In essence, its purpose was to take training and testing images and compare the sample size of each training set to its eventual accuracy (properly guessing the severity of a DR sample from 1-5). The experimentation produced accuracies categorized by DR severity and NN algorithm. In both algorithms, there was a general increase in accuracy as the training sample size increased. The CNN algorithm produced greater accuracy across higher training sample sizes ($n \geq 375$) compared to KNN, which produced greater accuracy across lower training sample sizes ($n \leq 250$), supporting our hypothesis. For each training sample size, a two-way ANOVA generated p-values less than the significance level of 0.05. The null hypothesis—no statistical significance between the algorithm type and its average accuracy across training sample sizes—was rejected. We concluded it was a statistically significant relationship, supporting that the algorithm type played a role in its accuracy. These results coincide with the global ophthalmic community that seeks widespread testing on behalf of those in low-income areas who are most at risk for DR[3].



Testing/Training Platform[2]



Prototype Presentation Setup[3]

The Development of a Proof-of-Concept Prototype:

With the information from our previous research, we constructed a prototype that used diabetic retinopathy as a focus for testing and Medibound as a management platform. This product was more of a proof-of-concept as it showed how the rather ambitious features we considered to be essential to "Medibound" devices could be practically applied to an actual medical utility. These features included NFC functionality[4] to serve a larger sample of testing patients and BLE device management for initiating testing. In addition, we also used Fusion360—executing on much of the knowledge we had attained through robotics—to develop an outer encasing for the product. Although this device never performed a live, patient test, its development and design are an accurate representation of how researchers can provide mobility while testing DR in the field[5]. In fact, many products produced around the time of our research indicate the very accuracy of our design as well as the need for a unifying standard of data simplification across IoMT devices.

NFC Incorporation w/ Adafruit ST25DV16K[4]

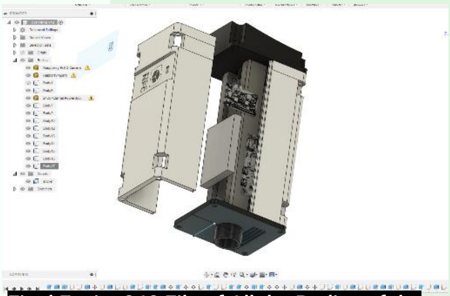
Similar Solution: Working prototype Developed by ITRI (Dec 2021 – 2 months After we finished our nonfunctional prototype)[5]

Challenges	Solutions

Distance prototyping and learning with a friend

There were major challenges associated with this overarching Visionbound project and some of them were attributed with the distance between my research partner and me (He moved to New York, I live in Florida). For example, when using the Kaggle DR library, we often had trouble working together over the computer due to the computational strain of each individual trial. Another instance of this can be found in product development where my partner didn't have access to a 3D printer within his vicinity.

As a result of this, I printed out 3D components at Florida Gulf Coast University after we finalized the CAD files we had designed[6]. This process was repeated multiple times until the outer encasing perfectly fit together, in which case I assembled and soldered together the necessary electronic components (Raspberry Pi Zero W, 12 MP HQ Camera, a 2465 Adafruit PowerBoost 1000C Rev B, an Adafruit ST25DV16K I2C RFID EEPROM Breakout, a lithium-ion polymer battery (3.7v 2500mAh), a USB-C cable, and an outer shell casing).



Final Fusion360 File of All the Bodies of the Prototype (w/ Scaled Placeholders for Components)[6]
https://medibound.com/VisionboundDevice.f3d

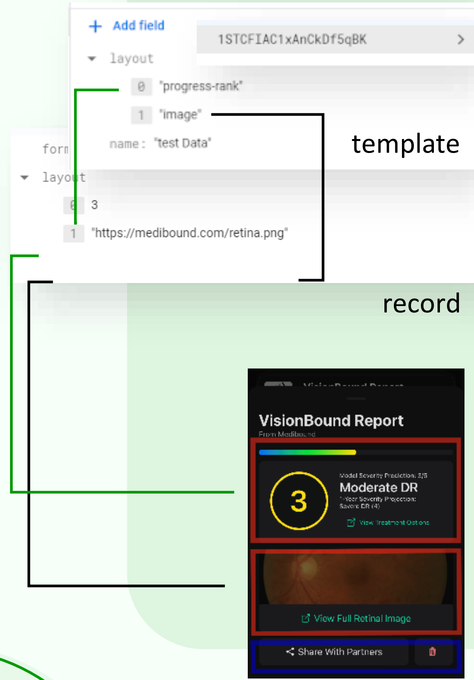
Visionbound – Medibound communication of data

v2

Much of the data transmission between Medibound and Visionbound is split into two forms of commands: Action commands (such as "run a test") and data commands (such as retrieve an array from NFC).These transmissions were crucial to prototype as we originally provided zero data protection. However, one difficult point about this data transfer was figuring out how to display it on a phone. To do this, we created a very rough form template type that would serve as the basis for how to display data on the mobile application.

v3

However, the main problem with this prototype is that there is currently still no way to create these templates, from a provider standpoint, without directly editing Firebase (which is not feasible), and there is no form of standardization/security of which we could communicate this data. After a recent discussion with Eric Just, a health technology advisor, we have begun to incorporate a medical data standard called FHIR into our application and create a service, either through Postman or Retool, to create templates and manage data presentation from a developer/provider standpoint.



Takeaways & What's Next

- π Although we learned quite a bit about the internal structure and adjustable metrics of the K-nearest neighbors' algorithm and convolutional neural networks, I recognize that we only scratched the surface of research dating back to Alan Turing on the intricacies of machine learning systems. I will further explore such wonders in depth as I continue to connect medical utilities to a largescale marketplace.
- π We discovered a larger (emerging out of COVID-19) industry of IoMT devices that has the potential to merge under one universal management. We took notice of the movement to remove the duties of physicians within many low-income testing initiatives through the development of NN confidence. Such stemming interest, referring all the way back to our Pathways project, would inspire an entrepreneurial research project with FGCU to analyze the market trends amongst these emerging technologies and the resulting standards that would best serve low-income patients.
- π Learning doesn't always present itself in similar shades. In robotics, I grew used to evident problems with seemingly definite solutions. Through moving out of that groove, I have learned that end-all-be-all projects, such as Pathways, Medibound, or Visionbound in hindsight, can reveal themselves to be pieces in a longer puzzle of a more niche interest. In some ways, my research into IoMT correlates much with my more abstract film projects. They both reveal a magic that excites people and changes the perspective of many as to what is possible.

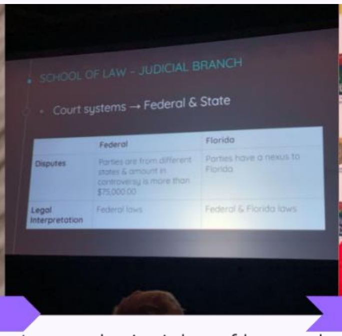
Worked with Fellow Delegates, Dr. Cindy Banyai for Congress FL19, Howard Sapp for Florida House District 78

Community Outreach

Political Involvement

Last summer, I had the honor of being elected as the Florida American Legion Boys State Governor. But I wasn't always this politically driven. In fact, I initially wasn't sure if I would run, but that's what Penn's Political Union and Boys State offers. It offers the chance to put yourself out there and experience the inner workings of government firsthand. I gained a deeper understanding of why politics matter and how they impact our everyday lives, while working with over 400 students—a family I still connect with today.

Selected as Delegate from Fort Myers.

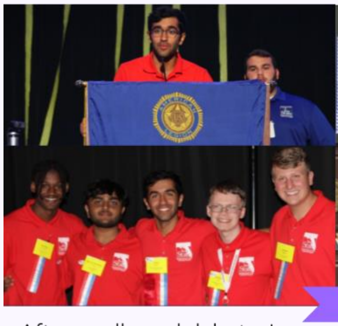


Learned principles of law and civic rights and responsibilities throughout the program.

Elected as Mayor, met with Tallahassee Mayor John Dailey, and led city simulations.

Played taps as a Bugler, was elected County Commissioner Chair, led county simulations, and created a budget.

Won county nomination and party nomination for Governor.



After a rally and debate, I was elected Governor, making it the first time in 11 years that a Federalist won the position. My party also made history by taking every officer role.

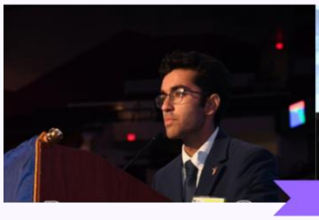
Sworn in by Judge Robert E. Long.

Addressed the Senate.

I was honored to speak at the Florida Vietnam Veterans' Memorial.

Decided on bills with my Cabinet. See interview with the Florida Channel:

https://drive.google.com/file/d/171qnsre7NXdb8vRVR0HF5G59C0ohkOnj/view?usp=sharing



Gave final remarks at the awards ceremony.

Pictured with Director Andrew Satterlee and Florida American Legion Dept. Commander Eunice Butts.

Most recently, I spoke at an American Legion meeting, advocating for Boys State. I am forever thankful to the program, and look forward to returning as Governor and a counselor this summer. I also continue to campaign for a better future.

Worked with Robotics Team

Community Outreach

Raspberry Pi Soccer Robots

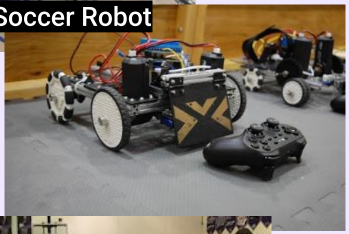
After our first FTC season ended, the team and I decided to set a goal for the summer. The objective was for each of our members to build a set of fully functional soccer robots to present at community events.

From building a test robot, which eventually turned into my personal robot, I was able to determine that each robot needed a solenoid, solenoid hat, two hex motors, and a Raspberry Pi 3B. I then turned to the programming side to figure out how to unify control across all the robots. Each team member needed their robot to move forward and backward, be able to turn, and fire its solenoid.

I ended up programming the Pi's in Python, using a BLE library and an Xbox controller driver to link any available controller immediately upon boot.

Once these customized robots were finalized using plexiglass frames and Rev FTC components, we were excited to showcase them at STEMtastics (a Lee County STEM event), an Imaginarium exhibit, and other mentoring sessions to communicate the principles of engineering to elementary and middle school students.

My Final Soccer Robot



Worked with Fort Myers High School students

Hurricane Ian Relief Efforts

I was lucky to be minimally affected after the hurricane, only losing power and enduring some damage to my home. Now, I devote my efforts to aiding the less fortunate.

CNN Feature

Charlie Crist and I were in Harlem Heights at the same time

Drywall Removal

Cleanup at Canterbury, one of our fellow Southwest Florida schools

I admire this quality about Fort Myers—we come together in times of hardship. We rebuild as a community and work together despite our differences. I am eager to continue that same drive and passion throughout my life, and I hope the attached photos can highlight that.