Design Draft - SwoleGator

Blackmon, James
Carmona, Maria
Restuccia, Joshua
Sheldon, Jenna
Williams, Mary

Introduction

1. Purpose/Need

Incorporating velocity-based training (VBT) into a strength athlete's training program can lead to an increase in performance. VBT offers many benefits such as precision training, real-time feedback to athletes, objective progress tracking, athlete individualization, injury prevention, and many more. SwoleGator provides an affordable way to track barbell velocity speed for powerlifters and other strength sports athletes by recording accelerometer and gyroscope data from the barbell's path.

2. Domain & Prior Art

SwoleGator is a tracker that falls under the fields of embedded and wearable devices, serving as a way to help powerlifters and athletes incorporate VBT in their training exercises and programs. SwoleGator's objective is to create an affordable and feature-rich device as compared to competitors such as the VELOS-ID [4], Vitruve Fit [1], FLEX [3], and RepOne [2] velocity trackers. These devices possess several disadvantages as compared to SwoleGator including attached cables that interfere with certain exercises, expensive cost, and lack of aesthetic elements.

3. Impact & Risk Assessment

In terms of societal impact, this app has the potential to influence community development within the workout community positively. A social aspect of the app could be an increase in social engagement and motivation for people to increase their training and exercise. It would also take some of the mystery away from an exercise program, velocity training, making strength training more accessible to beginners. The features that allow people to track their progress and outside lifestyle factors could alert them to unhealthy patterns in their lives.

However, the device could bring potential risks. Ethically, tracking apps may promote an unhealthy relationship with exercise. In addition, errors in the app or oversights could lead to injuries from lifters who misuse it. Finally, there is an ethical consideration in that the app may be exclusive to people to whom exercise is not accessible or to whom powerlifting is not possible, whether due to physical or other restraints.

Statement of Work

On the hardware side of the device, we will be utilizing a microcontroller such as an ESP32 interfaced with an IMU peripheral such as the MPU6050, which is popularly used with ESP32s, for detecting user movement while completing an exercise. Bluetooth will be utilized to send the data from the IMU to the app, which will then process it to determine the user's physical performance. For the software side of the design, we are currently considering using React.js to design a web and mobile-compatible application. We are also considering using Node.js to handle the backend and the Chart.js library to implement the velocity tracking element within the application.

1. Performance Expectations/Testing

Among those listed in the table below, the ESP32-WROOM-32 is widely used in wearable device applications because of its compatibility and low power consumption in active mode (ranging from 40 mA to 170 mA). The device will be powered with a single-cell LiPo battery (3.7V), which is lightweight and rechargeable. The MPU6050 has a consumption of 3.9 mA in active mode and a data rate of 1 kHz. To test battery life, we will be measuring the duration for which the tracker can operate on a single battery charge by fully charging the battery, running the tracker continuously until the battery is depleted, and then measuring the average current consumption to estimate the expected battery life.

The maximum data rate of BLE devices is 1Mbps. To test latency, we will record timestamps from when the movement starts and when the tracker detects and responds to the movement. Then, the latency will be calculated by finding the time difference. This will be repeated for the latency of sending data to the mobile app. To test data accuracy, we will perform up, down, and resting movements to determine if the IMU data consistently makes sense (e.g. resting would output 9.8 m/s^2). Additionally, we will graph this data over time for a visual representation of the changes.

ESP32 Microcontrollers			
	ESP32-WROOM-32	**ESP32-WROVER**	**ESP32-PICO-D4**
Processor	Dual-core Tensilica LX6 microprocessor	Dual-core Tensilica LX6 microprocessor	Dual-core Tensilica LX6 microprocessor
Clk Frequency	160 MHz	240 MHz	240 MHz
RAM	520 KB	520 KB	520 KB
Connectivity	Wi-Fi 802.11 b/g/n, Bluetooth v4.2	Wi-Fi 802.11 b/g/n, Bluetooth v4.2	Wi-Fi 802.11 b/g/n, Bluetooth v4.2
GPIO Pins	36	48	18

Feature/ Milestone	Task	Date	Assigned to Complete
Design Draft Done	Initial Group Meeting & Outline	10/10	All
	Draft Ready for Stakeholder Meeting	10/12	All
	Complete	10/15	All
General Requirements	Create GitHub Repository	10/15	Joshua
	Create Issue Tracking System	10/15	Jenna*
	Create Time Tracking System on Excel	10/15	Maria
Pre Alpha Build Preparation (Hardware)	Order ESP32 and IMU peripheral	10/26	Maria
	Test ESP32 and IMU peripheral individually with simple program on Arduino IDE	10/26	Maria, Joshua, Jenna*
	Test data transmission between ESP32 and IMU	10/26	Maria, Joshua, Jenna*
	Sample program for Bluetooth communication	10/26	Maria, Joshua, Jenna*
	Establish and demonstrate Bluetooth communication between all systems	10/26	Maria, Joshua, Jenna*
Pre Alpha Build Preparation (Software)	Design app interface on Figma	10/19	Mary, James*
Pre Alpha Build	External Interface: mobile app (react)	11/02	Mary & James*
	Persistent State: user profiles/accounts	11/02	Mary & James*
	Internal Systems: ESP32 and IMU interface	11/02	Jenna*
	Communication: Bluetooth for hw/sw connection, app to internet	11/02	Joshua*
	Integrity & Resilience	11/02	Maria*
	Complete!	11/02	All
Design Prototype	Evidence of Soundness	11/09	All
	External Interfaces	11/16	Mary & James*
	Internal Systems	11/16	Jenna & Maria & Joshua*
	Documentation	11/23	All
	Props/ other delivery elements	11/30	All*

Feature/ Milestone	Task	Date	Assigned to Complete
Prototype Presentation	Project State	11/30	All*
	Demonstration	11/30	All*
	Deliverable Plan	11/30	All*
	Present!	12/05	All
Design 2 Begins			
Alpha Build & Test Plan	App works for single user, can connect to hardware and track data	Week 2	Mary & James*
	Hardware works for accelerometer and path tracking	Week 2	Jenna, Maria, Joshua*
	Complete & Tesh Plan Made	Week 3	All
Beta Build & Test Plan, Alpha Results	App has profile & login settings, can store data for each user & posts can be shared (complete app)	Week 5	Mary & James*
	Hardware has passed initial tests and results being communicated to software are accurate and meaningful	Week 5	Jenna, Maria, Joshua*
	Performance Requirements:		
	Test Plan & Alpha Results	Week 6	All*
Progress Presentation	Presentation Complete (tasks TBD)	Week 7	All
Preliminary Report	Have initial report complete (tasks TBD)	Week 10	All
Release Candidate	Tasks TBD	Week 10	All
Final Testing	Complete testing and finishing touches	Week 12	All
Final Report	Revise initial report & complete final (tasks tbd)	Week 13	All
Production Release	Tasks TBD	Week 13	All
Final Presentation	Tasks TBD	Week 14-15	All

* subject to change

Deliverable Artifacts

1. Tracker:

   1.1 Description:

   The tracker will utilize an IMU to detect the user's linear and rotational movement as they complete a barbell exercise. The tracker will be attached around the barbell's shaft and will have an LED to visually alert the user of a completed calibration and a button to turn it on. The tracker will have a built-in battery that will be charged using a USB-C cable. Additionally, the tracker will transmit data to the user's phone app using a wireless connection through Bluetooth.

   1.2 Format/Distribution:

   The tracker will come secured with a velcro strap. Included with the device, is a short manual on getting started with the tracker, calibration, and app connection.

   1.3 Accessibility/Usability/Maintenance Plan:

   The tracker can be calibrated multiple times whenever the user wants to switch barbell exercises or to fix user errors during previously attempted calibrations.

2. Application:

   2.1 Description

   The application interfaces with the ESP32 to receive, process, and display the recorded data from the IMU. This data will be stored in a graph, allowing the users to visualize progress in their lifts over time. There will also be account creation and login functionality, a button to calibrate the device to start the lift, and potentially a social page for the users to share their lifts.

   2.2 Format/Distribution

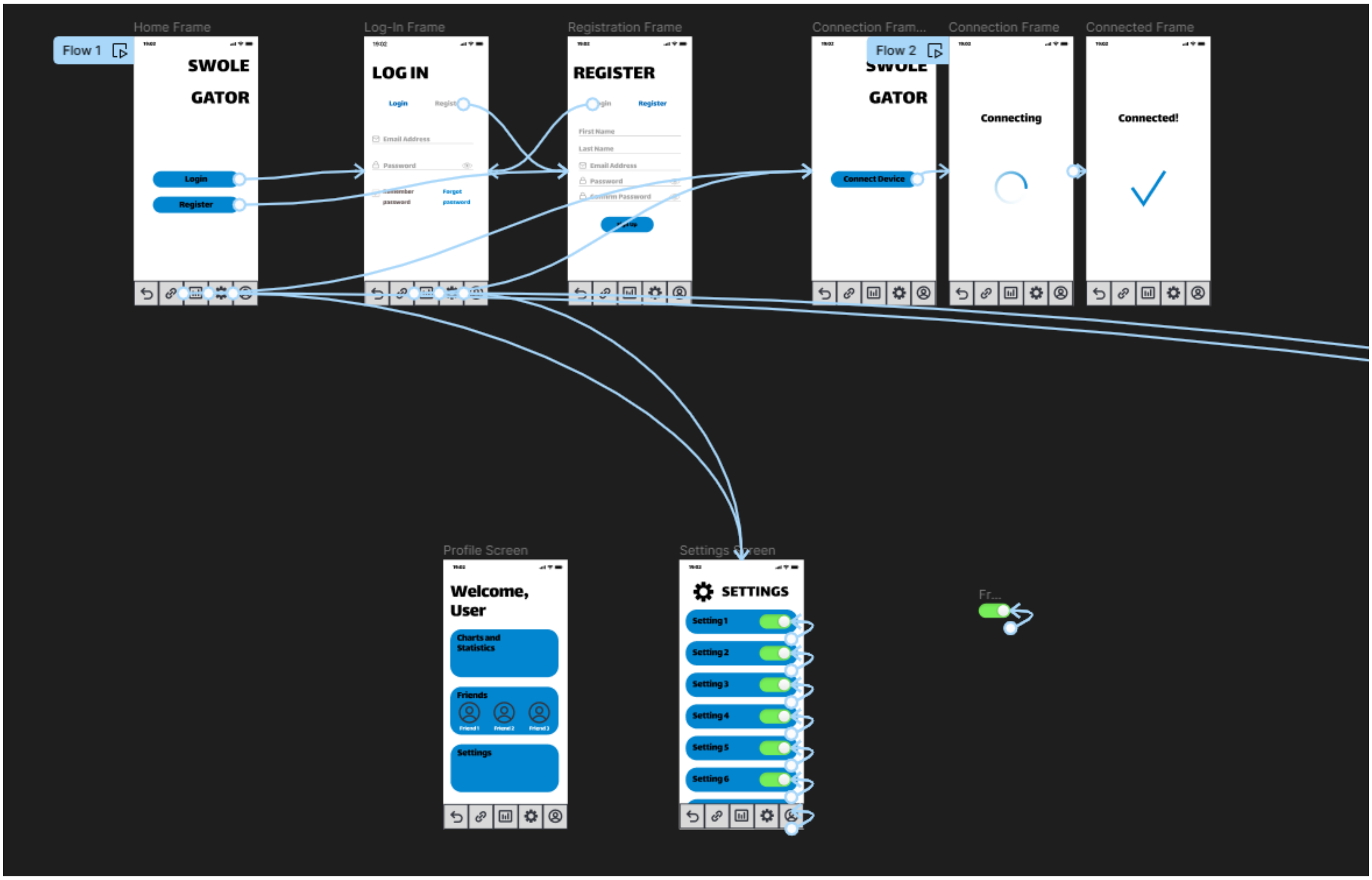
   The application will be created with React.js and distributed by transferring it to a React Native mobile application upon completion. From here, the app will be able to connect and communicate with the tracking device.

   2.3 Accessibility/Usability/Maintenance Plan

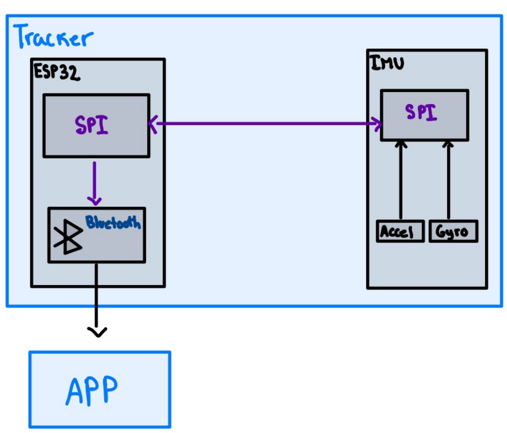
   This application will be especially useful in the long term. The longer a user tracks their lifts, the more tangible progress they will be able to see. Having one place to store this information and reference the data over time is an invaluable tool for the athletes in planning their progressions as well. To maintain the app and keep it growing, it would also be important to update the user interface as client feedback suggests changes or improvements over time.

Mockups

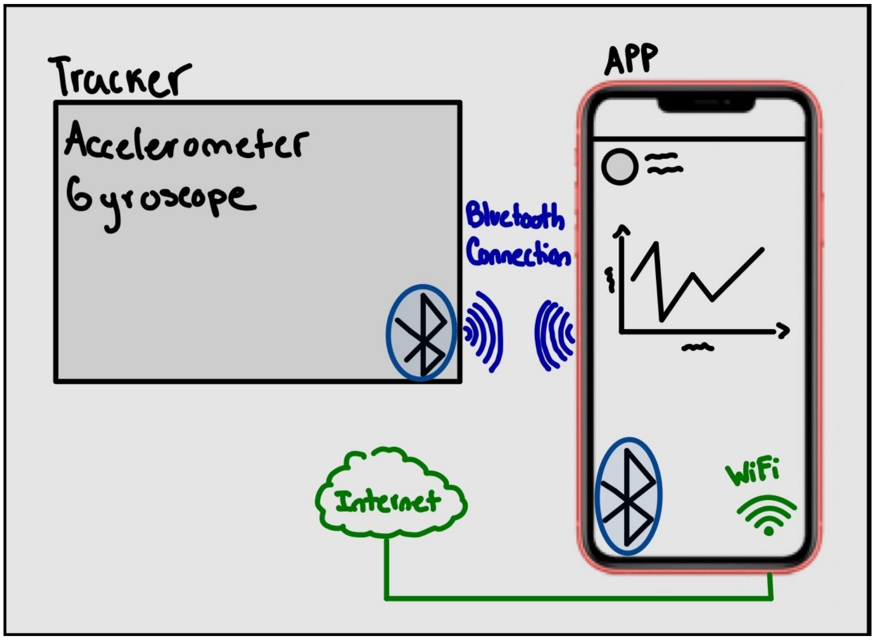
1. Interfaces / Storyboards



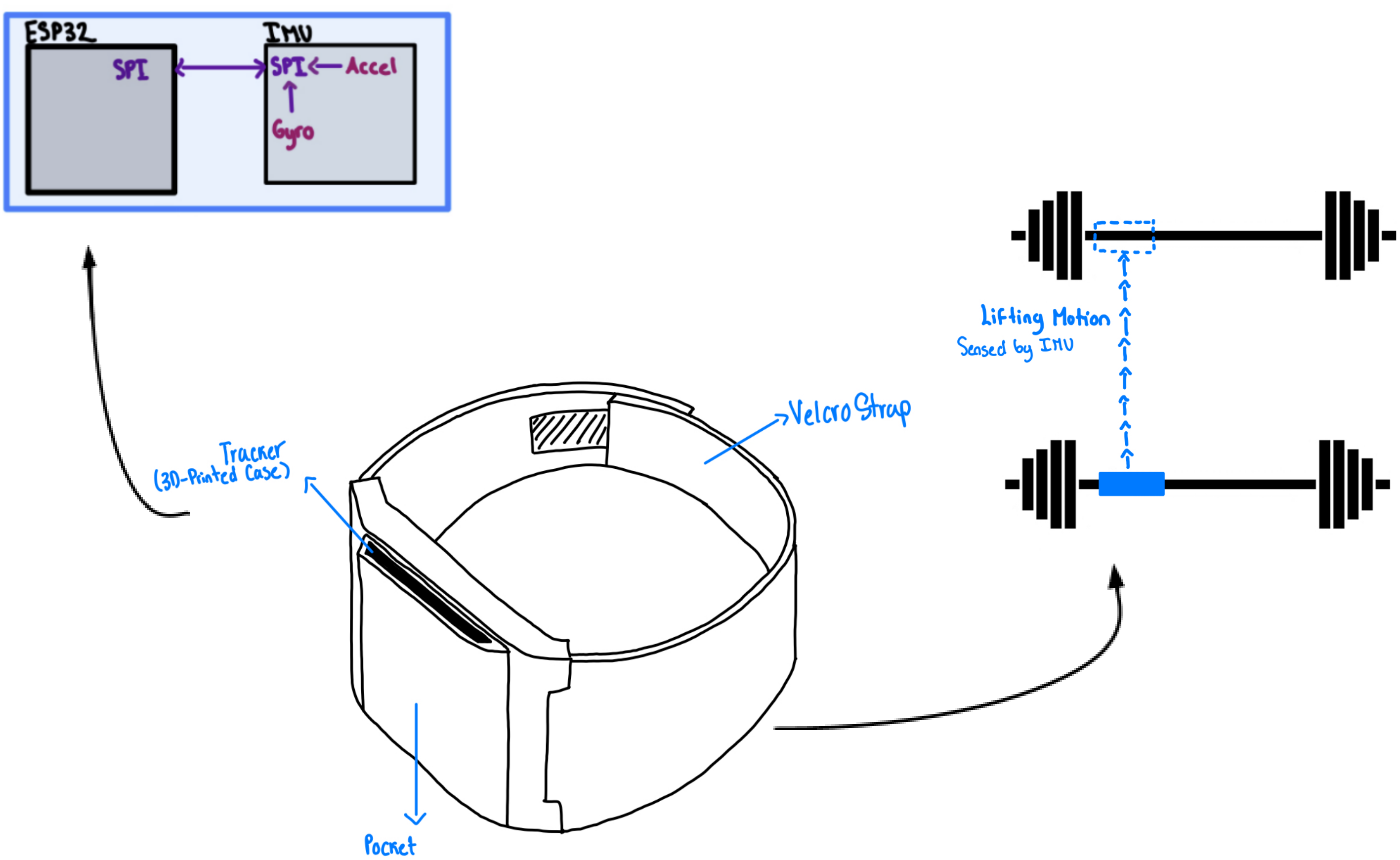
2. Systems



3. Networking



4. Draft Schematics



References:

[1] “Barbell velocity tracker for strength training,” Vitruve, https://vitruve.fit/blog/barbell-velocity-tracker/ (accessed Oct. 15, 2023).

[2] “RepOne strength,” RepOne Strength, https://www.reponestrength.com/ (accessed Oct. 15, 2023).

[3] “Velocity-based training Made simple,” FlexStronger, https://www.flexstronger.com/ (accessed Oct. 15, 2023).

[4] “Velos-ID v3 barbell tracker for velocity based training (VBT),” Sheiko Gold, https://sheikogold.com/product/velos-id-v3-barbell-tracker-for-velocity-based-training-vb/ (accessed Oct. 15, 2023).