liftVectr Design and Development Plan

Praveen Anbu, Micah Bowonthamachakr, Antoine Ferguson, Cole Fisher, Timothy Krause

INTRODUCTION

The liftVectr platform is a wrist device and mobile app pairing that provides measurement, analysis, and tracking tools to weightlifters. It provides consumers of all skill-levels with a low cost alternative to expensive training programs or personal coaches. The liftVectr gives users the guidance to make healthy, continual progress in weightlifting, while maintaining safe and proper techniques to avoid injury.

There are several existing products that aim to track one's bar path as liftVectr does, but they differ in the way that they do so. For example, BarSense [1] is an app that uses the user's camera to record them performing a lift and uses it to track the bar path and extrapolate velocity data. However, this differs from liftVectr in that liftVectr is a device attached to the user's wrist providing more accurate measurements, using the gyroscope in the device, than can be obtained with the camera method.

STATEMENT OF WORK

Performance Expectations

Software:

The mobile application will be created for the Android platform, primarily written in Java. The Android Studio IDE will be utilized for development and testing. Common Android APIs will be utilized throughout the app for common app functionalities such as bluetooth. The Chaquopy [2] python SDK for Android will be utilized to access third-party packages such as numpy and matplotlib for data analysis and visualization. The application will use the Gradle build environment.

Hardware:

The wearable wrist device will consist of a microcontroller and battery, with an outer casing and wrist strap. The microcontroller must possess a 6 axis IMU (3D digital accelerometer, 3D digital gyroscope) and Bluetooth 5.0 module, have an input voltage of 5V or lower, and should ideally be smaller than a square inch in size. The battery should be rechargeable, and meet the microcontroller's input voltage requirement.

Dependencies:

The chosen development board will be the Seeed XIAO BLE nRF52840 Sense [3], which has onboard bluetooth and IMU, reducing inter-hardware dependencies. The board will require a power supply for prototyping, and basic electrical tools for testing/wiring. Software dependencies include integrating Android Studio with Python via Chaquopy, an important task to accomplish as the primary functionality of the mobile app depends on communicating and parsing data from the IMU hardware. Rather than waiting for hardware, the mobile app team will first simulate incoming data and first work on calculations, UI and plotting features.

Team members have been designated sub-teams in order to complete these tasks with minimal dependency, and are not constrained to working only with the sub-team. The Hardware Team includes Praveen, Tim, and Micah, and the Mobile App Development Team includes Cole and Antoine. The initial tasks for the Hardware Team involve setting up the breadboard, programming the microcontroller, establishing bluetooth for the mobile app, and making/printing a 3D-CAD model for the watch chassis. The Mobile App Development Team will first develop a barebones application to communicate with the hardware device over bluetooth, and then work on basic UI and plotting/data-storing procedures for tracking lift sessions and implementing other functions involving mathematical operations with the data.

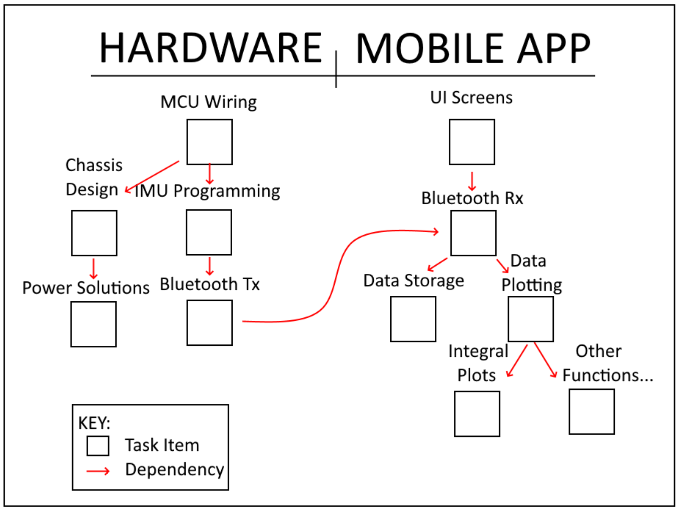
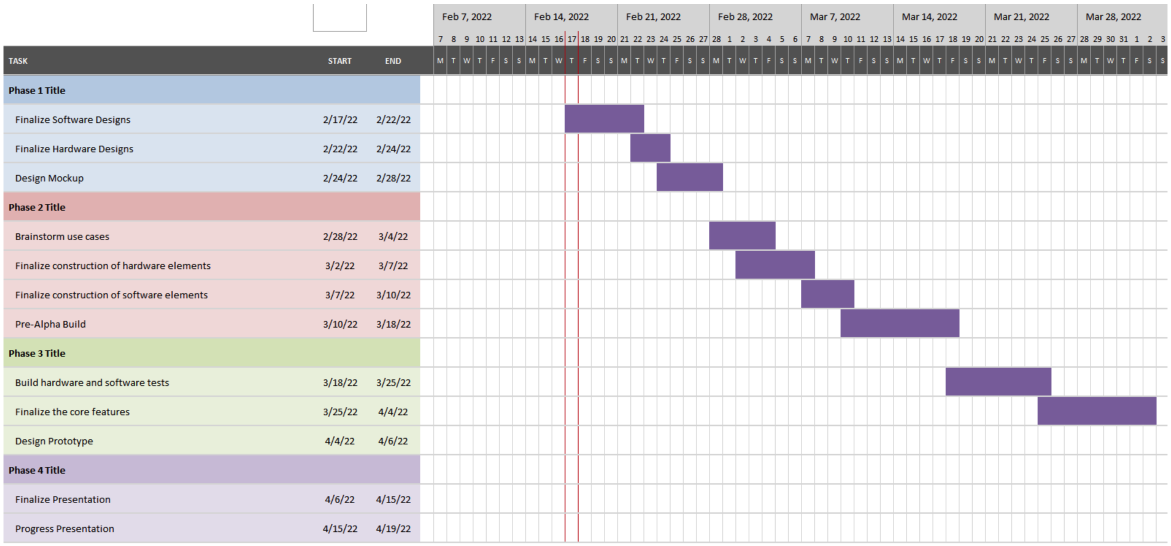


Figure 1: Sub-team tasks and dependency tree. Arrow points from locally independent TO dependent tasks

Deadlines:



TASK	START	END
Phase 1 Title		
Finalize Software Designs	2/17/22	2/22/22
Finalize Hardware Designs	2/22/22	2/24/22
Design Mockup	2/24/22	2/28/22
Phase 2 Title		
Brainstorm use cases	2/28/22	3/4/22
Finalize construction of hardware elements	3/2/22	3/7/22
Finalize construction of software elements	3/7/22	3/10/22
Pre-Alpha Build	3/10/22	3/18/22
Phase 3 Title		
Build hardware and software tests	3/18/22	3/25/22
Finalize the core features	3/25/22	4/4/22
Design Prototype	4/4/22	4/6/22
Phase 4 Title		
Finalize Presentation	4/6/22	4/15/22
Progress Presentation	4/15/22	4/19/22

Figure 2: Gantt Chart representing major project milestone/task deadlines.

DELIVERABLE ARTIFACTS

The team strives to have a working demonstrable product (having a wrist device interfacing with a mobile application) by the progress presentation. At such time, we should be able to deliver the wearable wrist device with all of its intended features fully enabled (rechargeability, bluetooth connectivity, IMU data export) and a mobile app with at minimum the core features implemented (data reception, storage, analysis, and visualization). Afterwards, it should be expected that further progress will be centered around the production of the wrist device and the cleaning and publication of the mobile app.

With our app being developed through the Android platform, we will be able to access and test the app through a mobile phone at any stage during our development phase. This accessibility can carry over to when the app is in production through the Android Play store, through which we can continuously send updated versions to the users. This product is intended to have seamless usability as the bluetooth connection of the device and mobile app should be reliable for data transfer. For maintaining this application, we will need to make decisions on optimizing user data storage with cost of storage resources in future iterations of the mobile application.

References

[1] “Analyze and improve your lifts,” *BarSense*. [Online]. [Accessed: 18-Feb-2022].

[2] “The easiest way to use Python in your Android app,” *Chaquopy*. https://chaquo.com/chaquopy/.

[3] “Seeed XIAO BLE nRF52840 Sense - TinyML/TensorFlow Lite- IMU / Microphone - Bluetooth5,” *www.seeedstudio.com*. https://www.seeedstudio.com/Seeed-XIAO-BLE-Sense-nRF52840-p-5253.html (accessed Feb. 18, 2022).

[4] “Simple Gantt Chart.” *Create More with Microsoft Templates*, 18 Aug. 2021, https://templates.office.com/en-au/Simple-Gantt-Chart-TM16400962.