

3. Project Plan

3.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

We are adopting both the waterfall and agile methodology. We are continuously developing the hardware alongside the software and testing each. The software relies on the data from the hardware, so it requires some waterfall thinking.

We are using GitLab to store all code related to the project. We plan on using the *Issues* feature in GitLab to break down things we want to do in small tasks. These issues can have branches created from them, keeping our code separate and helping to avoid conflicts. These issues are contained inside of Issue Boards, which keep track of the status. We currently use Discord for communication and providing updates.

3.2 TASK DECOMPOSITION

Hardware Integration	Connect and test pressure sensors. Ensure that the sensors produce readable and logical data.
Mobile App Development	Design, implement, and test the mobile app (Android).
Wireless Connectivity	Integrate Wifi integration via Raspberry Pico to ensure data transfer between patch and app.
Data Processing	Develop algorithms to process pressure data and trigger alerts.
Testing & Validation	Simulate real-world conditions and conduct user testing with adaptive athletes.

3.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

- Establish wifi communication between phone app and sensor device
 - Can the sensor send data over a wifi signal?
 - Is the app able to receive data from the sensor over an internet connection?
- Establish bluetooth connection between phone app and sensor device
 - Is the app able to receive data from the sensor over a bluetooth connection?
 - Can the sensor send data over a bluetooth connection?
- Get usable data from pressure sensors
 - Is the data received in the form the app expects?
- Develop algorithms for interpreting raw data
 - Can the sensor device turn raw sensor data into usable information?
- Build an Android app that can read and display data from sensor device

3.4 PROJECT TIMELINE/SCHEDULE

[illegible]

Design Document - Project Plan

Conduct extensive testing to ensure the system is robust and ready for deployment. Provide initial support to users and gather feedback for potential future updates or improvements.										
Task	August	September	October	November	December	January	February	March	April	May
Establish wifi communication between phone app and sensor device										
Establish bluetooth connection between phone app and sensor device										
Get usable data from pressure sensors										
Build an Android app that can read and display data from sensor device										
Integrate data from the selected pressure sensors to software via bluetooth										
Develop algorithms for interpreting raw data										

3.5 RISKS AND RISK MANAGEMENT/MITIGATION

1. Sensor Performance and Compatibility

- **Risk:** Sensor accuracy may fall below the required sensitivity, or selected sensors may not be compatible with the microcontroller.
- **Probability:** 0.4
- **Severity:** Moderate
- **Mitigation Plan:**
 - **Alternative Task:** Source higher sensitivity sensors or develop signal-processing code to filter noise.
 - **Off-the-Shelf Solution:** Purchase a more robust, commercially available sensor with known compatibility.
 - **Alternative Technology:** Test multiple microcontrollers to assess compatibility.

Design Document - Project Plan

2. App Functionality and User Experience

- **Risk:** App may not maintain alerts when closed, impacting user responsiveness.
- **Probability:** 0.5
- **Severity:** High
- **Mitigation Plan:**
 - **Alternative Task:** Investigate background service management for persistent notifications on Android.
 - **Off-the-Shelf Solution:** Leverage frameworks or libraries specifically designed for background task persistence.
 - **Alternative Technology:** Explore Power BI's real-time alert capabilities to assist in app functions.

3. Scheduling and Agile Sprints

- **Risk:** Delays in sensor testing may impact agile sprint progress and affect overall schedule.
- **Probability:** 0.4
- **Severity:** Moderate
- **Mitigation Plan:**
 - **Alternative Tasks:** Extend sprints by a week to accommodate testing and identify alternatives that might bypass delay factors if necessary.

3.6 PERSONNEL EFFORT REQUIREMENTS

Hardware Integration	120 Hours	We need to find the right hardware. We need to find how to work with this hardware.
Mobile App Development	160 Hours	We need to develop a UI. We need to connect wirelessly. We need to develop notification logic.
Wireless Connectivity	40 Hours	We will need to figure out how to use Bluetooth.
Data Processing	130 Hours	We need to figure out when to send a notification and what data will need to be sent.
Testing & Validation	80 Hours	We will need to be constantly testing the implementation of hardware and software.

3.7 OTHER RESOURCE REQUIREMENTS

1. Skill Acquisition

- **Kotlin Programming:** Team members will need to learn Kotlin to develop and maintain the Android app efficiently. Resources may include online courses, documentation, and tutorials.

2. Development Tools and Software

- **Android Studio:** Primary IDE for developing and testing the Android app.
- **Database Systems:** Any databases or cloud storage services needed for storing and retrieving user data and sensor readings.

3. Testing Equipment

- **Debugging and Testing Devices:** Access to Android devices (phones or tablets) for testing the app's functionality and real-time sensor data alerts.
- **Simulators:** Android emulators or simulators for testing different device environments and Android versions.