progear

Supervisor : T. Shatha Alkhaldi

Group: 17

Names:

Sara Alrahma

Haya Bin Samih

Hessa Almaarik

Table Of Contents

Introduction

Powerlifting depends on essential gear like belts and wraps for safety and performance. Many athletes struggle to manage their equipment, often forgetting items during training or competition.

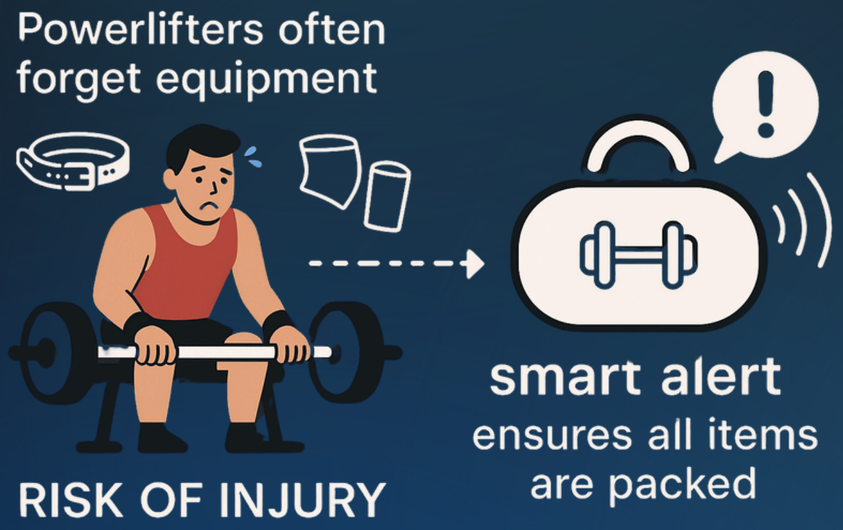
The ProGear Smart Bag offers a smart solution using IoT weight sensors and Bluetooth to track gear and notify users via a mobile app.

This project aims to reduce human error and help powerlifters stay prepared at all times.

Problem Definition

Powerlifters often forget essential gear like belts or wraps due to stress or busy schedules. Relying on memory or checklists isn’t reliable, and forgetting items can lead to injuries or poor performance.

Traditional bags don’t help verify packed items. The ProGear Smart Bag solves this with real-time alerts and weight tracking to ensure all critical gear is packed before training.



Aims & Objectives

The project aims to develop a smart gym bag for powerlifters that uses IoT to detect missing gear in real time and alert users instantly.

Objectives

- Track gear using weight sensors + microcontroller
- Create a mobile app to register gear and receive alerts
- Notify users when any item is missing

Background

Powerlifters rely on essential gear for safety and performance, but traditional packing methods are unreliable—especially under pressure.

The ProGear Smart Bag solves this by using IoT technology to track missing items through weight sensors.

- Load cells detect missing gear.
- ESP32 processes data.
- HX711 converts signals.
- Bluetooth connects to a Flutter app for real-time alerts.

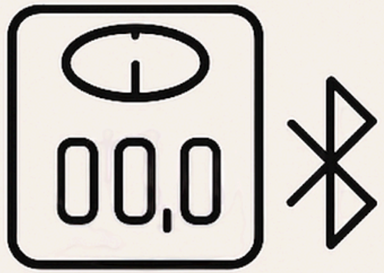
This system ensures athletes never forget key equipment again.

Related Work

SIMILAR SOLUTIONS,
DIFFERENT GOALS

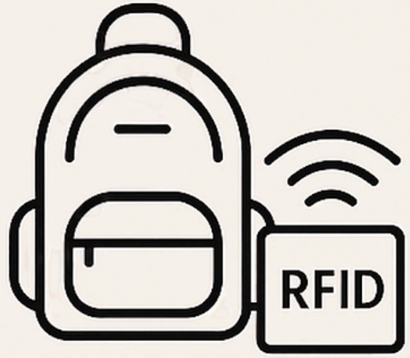
PERFORMANCE
TRACKING

SAME HARDWARE,
DIFFERENT APPLICATIONS



WEIGHT
TRACKING

DIFFERENT
SOLUTIONS, SAMI GOAL



ITEM
DETECTION

Related Work

Many studies explored sensor and Bluetooth technologies for sports and health, but none directly addressed detecting missing gear inside gym bags.

Similar solutions, different goals:

- Bousselmi et al. used load cells for performance tracking, not gear detection.
- Hettiarachchi built an IoT gym tracker for body weight, not equipment management.

Same hardware, different applications:

- Robotique.tech used HX711 + ESP32 for health-focused weight tracking.
- Nordic’s Gymplanner used BLE to monitor equipment usage, not what's inside bags.

Different solutions, same goal:

- Singh & Nigam used RFID to detect missing items in school bags.
- Lenovo’s smart backpack included sensors but lacked weight-based tracking.
- Smart lockers check gear presence but are not portable.

Requirement Elicitation Techniques

Technique we used :

We used **questionnaires** to collect requirements from potential users, specifically powerlifters.

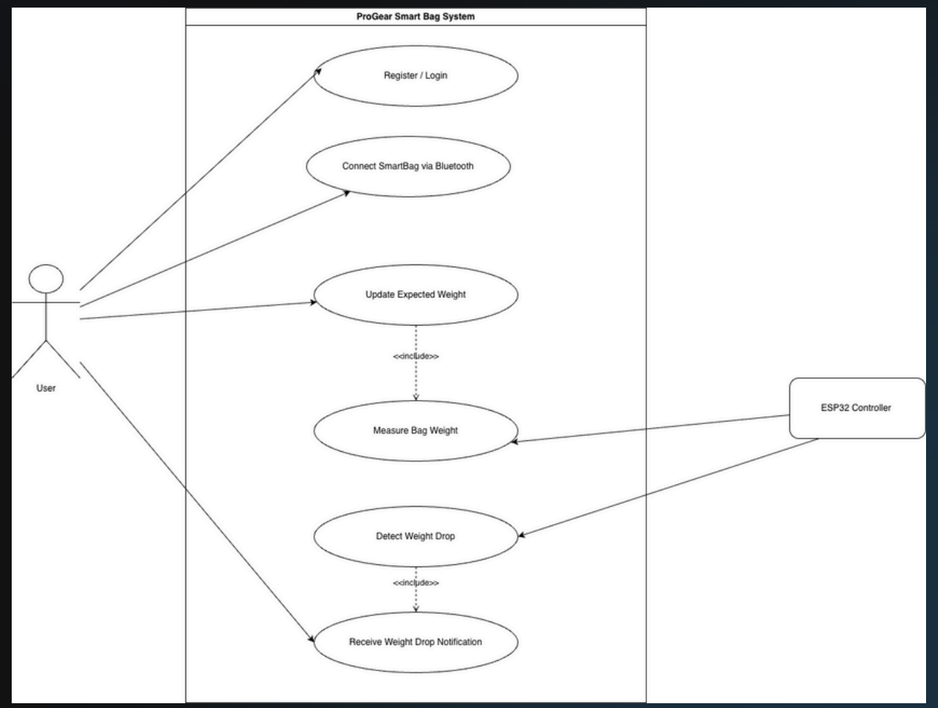
Why this method?

- Fast and easy to distribute.
- Reached multiple users efficiently.
- Gave us direct insight into common gear-related issues
- Helped identify user needs and priorities

The feedback shaped key features like real-time alerts, weight tracking, and mobile app simplicity.

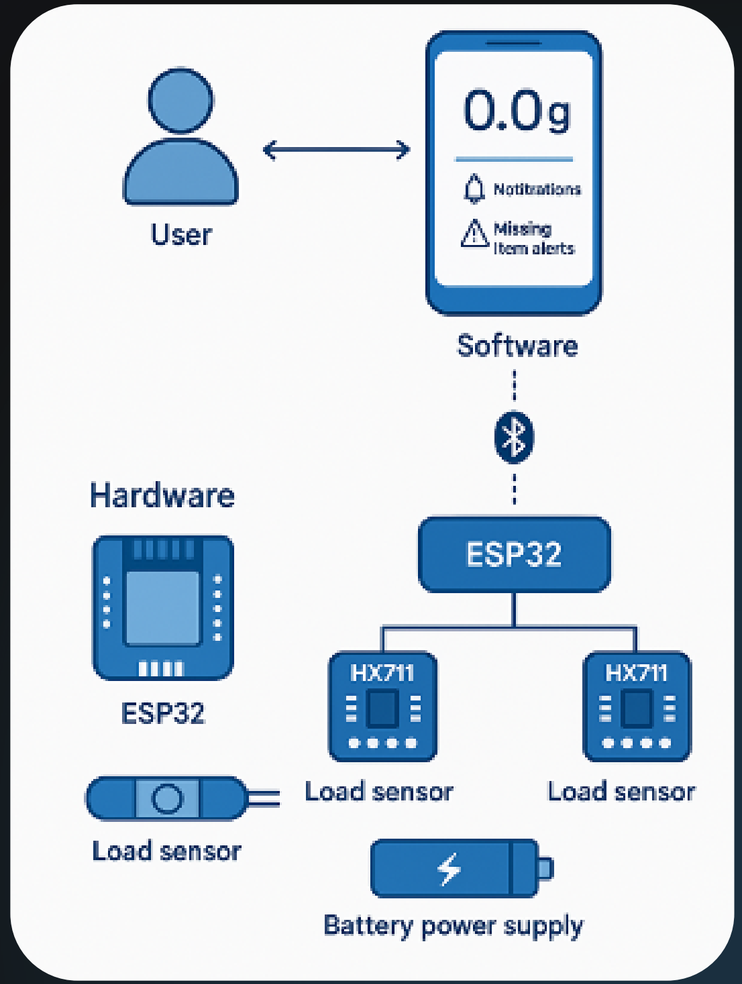
System Analysis & Design

This diagram shows how the user interacts with the ProGear Smart Bag system. It starts with registration and Bluetooth connection, then measures the bag's weight using the ESP32 controller. If a weight drop is detected, the user receives a real-time notification.



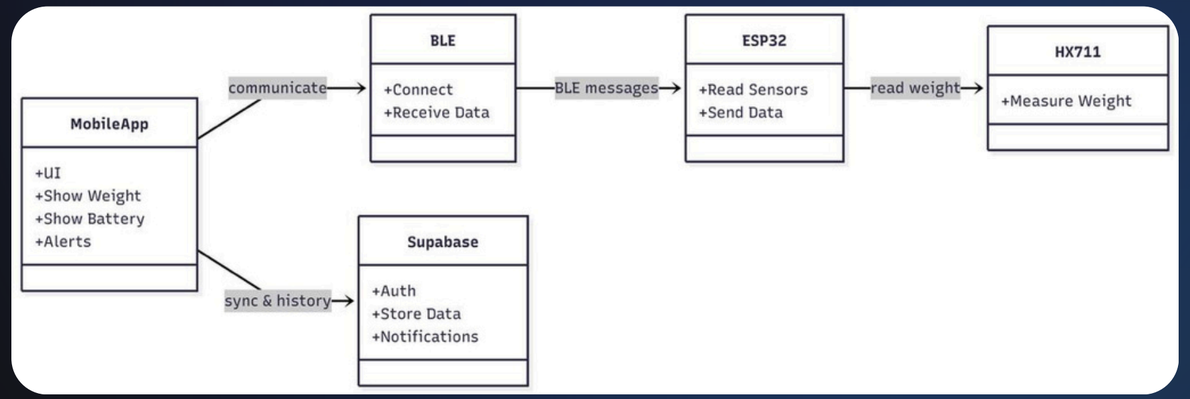
Architectural Design

The ProGear Smart Bag system consists of three main components. The user interacts with a mobile app that displays the gear status and sends alerts when an item is missing. The app communicates with the ESP32 controller via Bluetooth. The ESP32 receives weight data from load sensors through HX711 modules. Based on this data, it detects if any item is missing and sends real-time updates to the app.

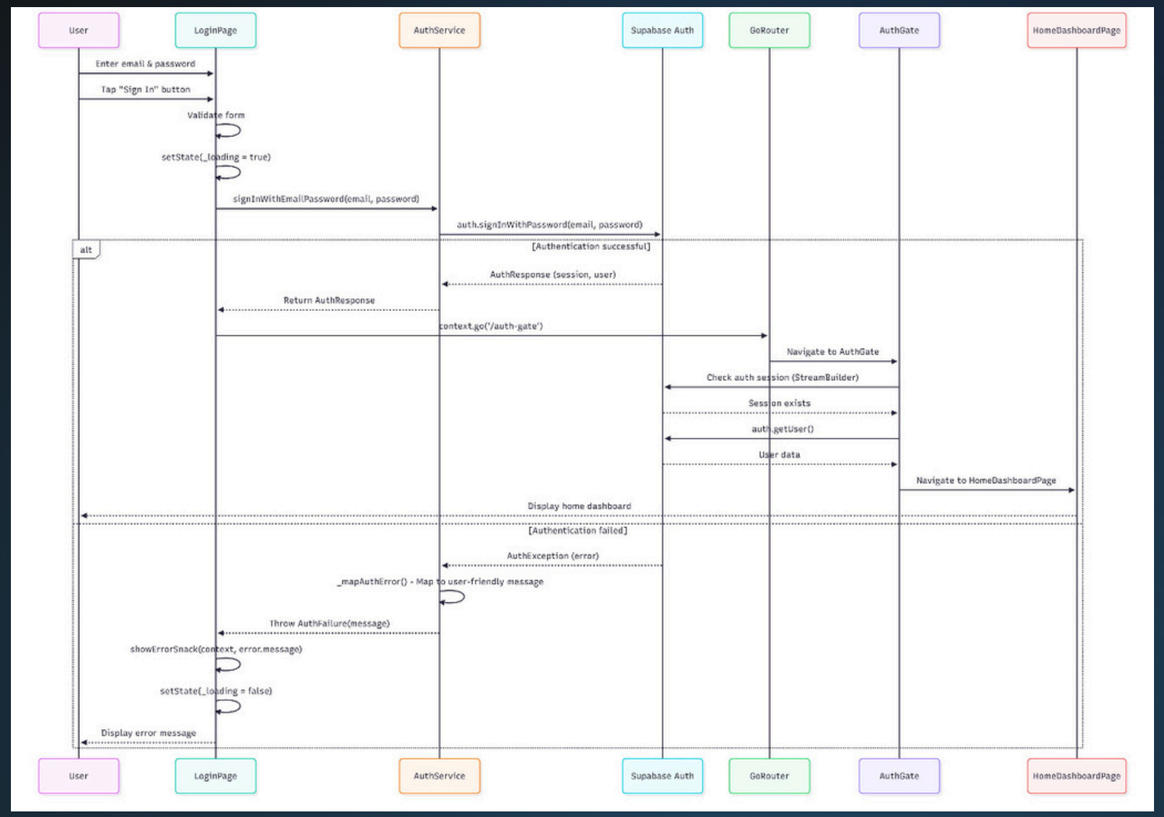


Class Diagram

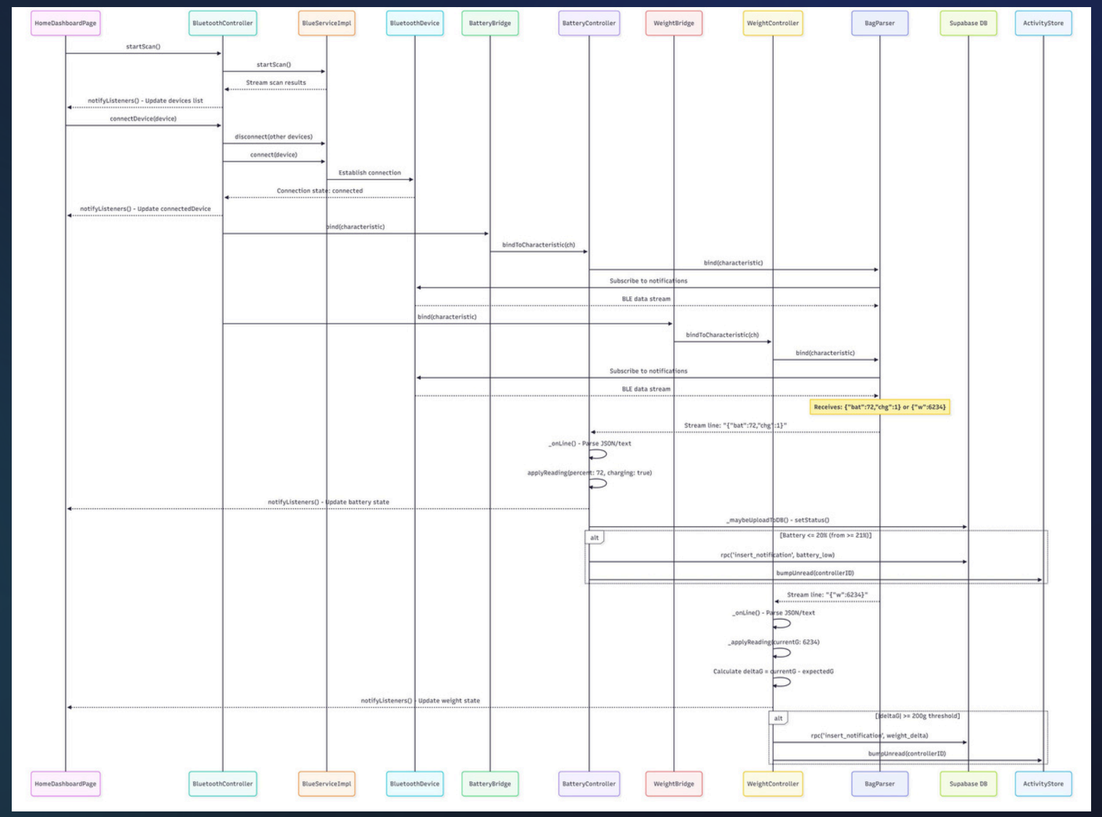
This class diagram shows how the system works. The user logs in through the app. The ESP32 controller reads weights from sensors, compares them to expected values, and sends a notification if something is missing.



Sequence Diagram

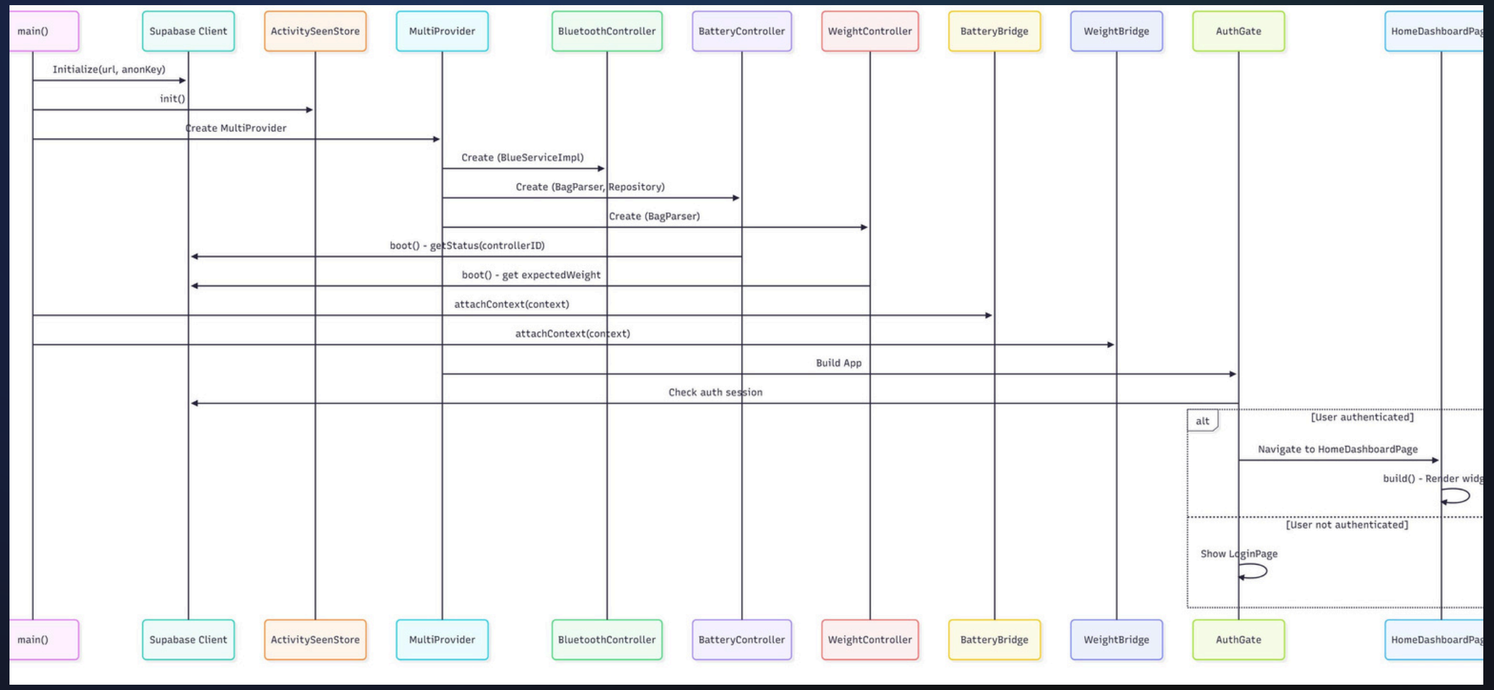


User Login Flow

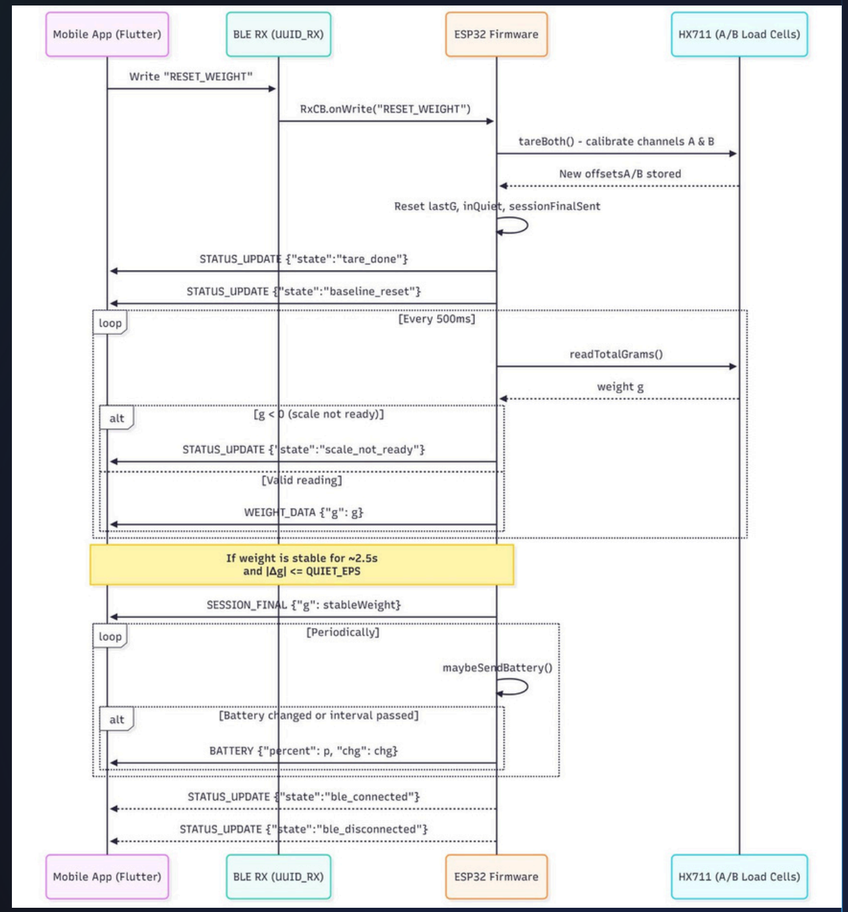


Bluetooth Connection and Data Streaming Flow

Sequence Diagram



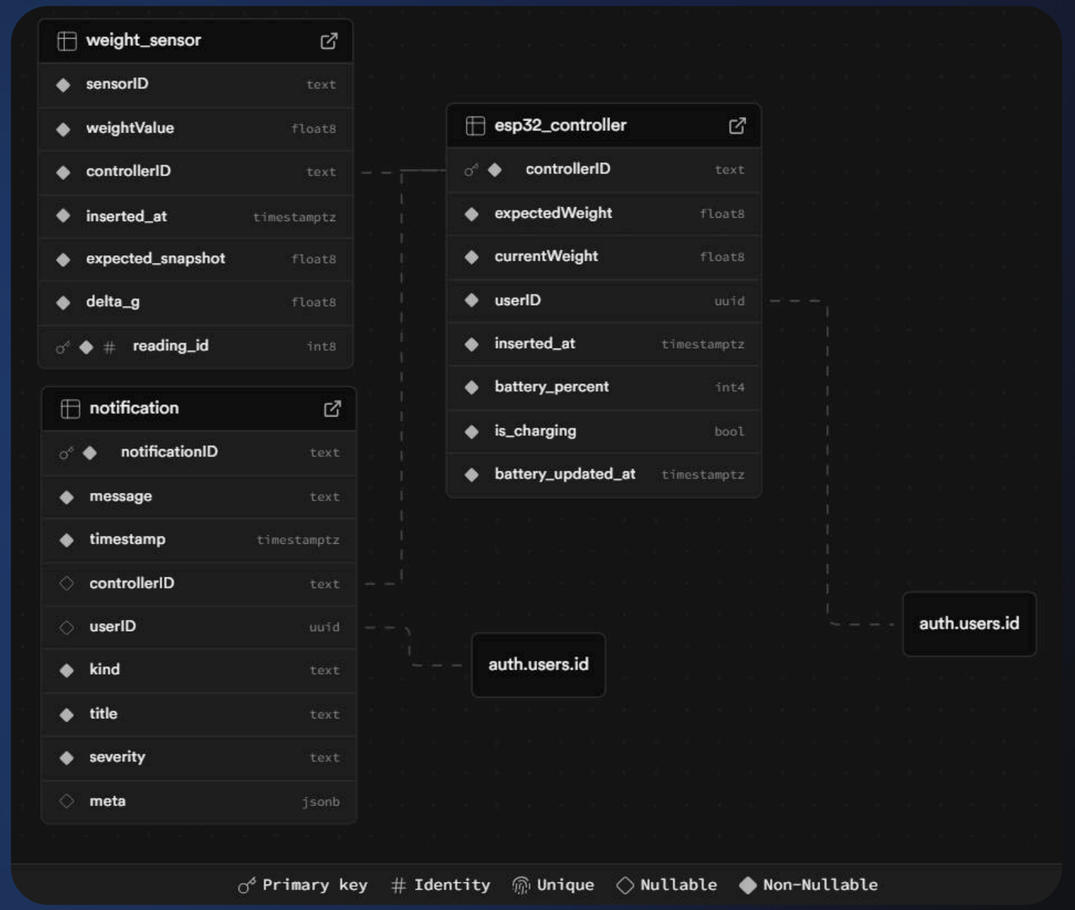
App Initialization and Home Dashboard Flow



Hardware Weight & BLE Streaming Flow

ER Diagram

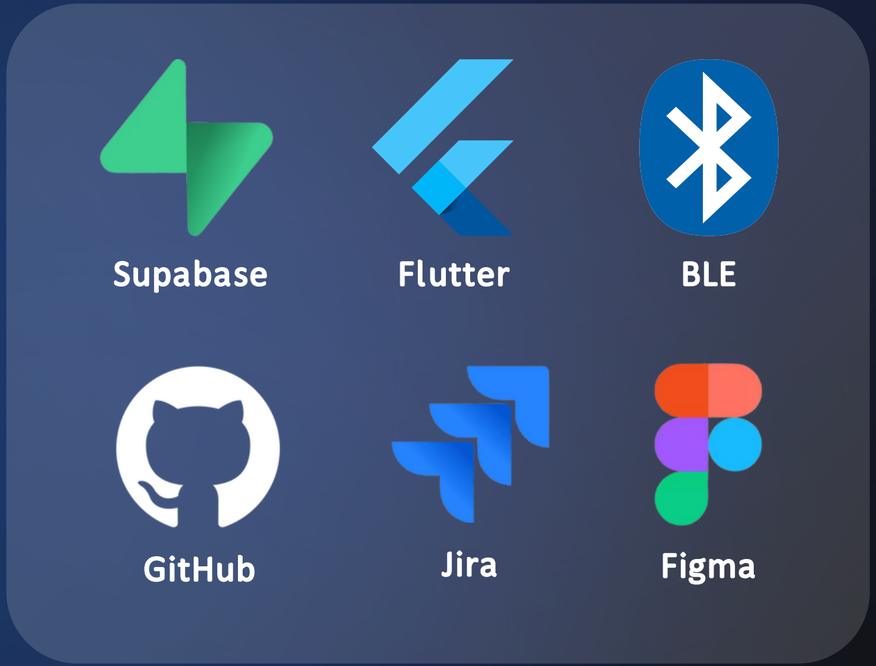
shows how the main database tables in the ProGear system are connected. Each user is linked to one ESP32 controller, which connects to multiple weight sensors. Notifications are linked to both the user and controller, and store messages about missing items along with timestamps. This structure ensures accurate tracking of gear and clear user-alert history.



Implementation

The Implementation phase focused on building the hardware prototype, developing the Flutter mobile application, and integrating both through Bluetooth Low Energy (BLE). This stage transformed our system design into a fully functional prototype.

Software Tools



Implementation

The hardware implementation of the ProGear Smart Bag was built using a set of lightweight and reliable electronic components. The ESP32 served as the main controller, handling sensor readings and BLE communication. Dual load-cell sensors, connected through HX711 amplifiers, were used to accurately measure the weight inside the bag. A rechargeable battery
the entire system, while breadboards, wiring, and connectors supported prototyping and stable assembly of the final circuit.

Unit Testing

Problems

- Delayed initialization of the HX711 sensor.
- Unstable and noisy weight readings.
- Dashboard loaded incorrect data if accessed before Bluetooth connection.

Solutions

- Added a short initialization delay for the HX711 sensor.
- Improved wiring and applied filtering to stabilize weight readings.
- Enforced connection-first workflow so the dashboard displays accurate, real-time data.

Integration Testing

Problems

- Dashboard showed incorrect or zero/default values after Bluetooth pairing.
- Weight and battery data did not update if dashboard opened before BLE connection.
- Rapid item movements caused weight readings to fluctuate.

Solutions

- Restructured workflow: dashboard only accessible after BLE connection.
- System automatically requests weight and battery data once connected.
- Implemented weight-stability check to report final reliable readings.
- Dashboard no longer loads dummy values at any stage.

User Acceptance Testing

- Tested by three athletes to evaluate accuracy and ease of use.
- Scenarios: powering the bag, Bluetooth connection, live weight & battery monitoring, adding/removing items to check missing-item detection.
- Minor issues (delays, pairing guidance) were resolved with workflow updates and prompts.
- Outcome: Smooth operation, accurate real-time updates, user-friendly interface, fully ready for real-world use and future enhancements.

Tester	**Observation**	**Improvements Applied**
Athlete A	Dashboard showed incorrect values unless Bluetooth was connected first	Added mandatory Bluetooth pop-up and workflow enforcement
Athlete B	Wanted faster updates after connecting	Optimized BLE request timing
Athlete C	The start scan search takes too long	Update the scan process to 10 seconds

Feedback and Observations

Test Cases

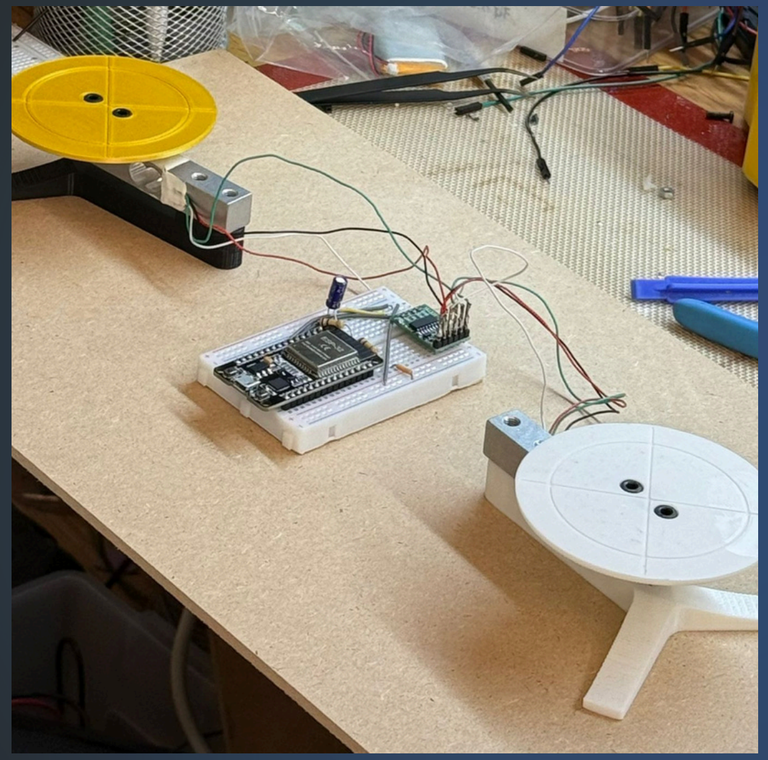
Test Case ID	TC-01
Description	To verify that the user can successfully establish a Bluetooth connection between
Sequence of Events	1. Power the ESP32 by connecting it to the laptop.
Test Data	ESP32 Bluetooth name: 20:E7:C8:68:FE:C6
Testing Environment	Android phone (Android 13), BLE connection.
Expected Result	Connection is successfully established,Dashboard remains
Actual Result	Connection was successful. The system correctly prevented access to the
Status	Passed

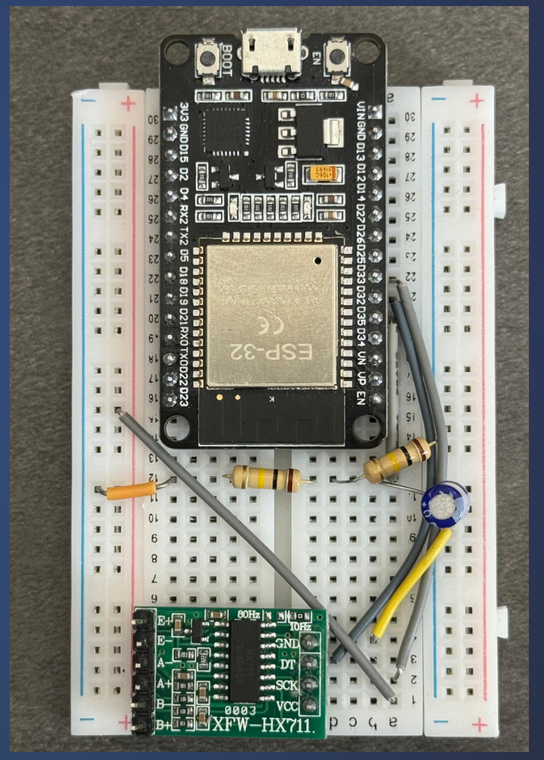
Test Case 1: Bluetooth connection

Test Case ID	TC-02
Description	To verify that the smart bag correctly reads weight values from the HX711 load cell
Sequence of Events	1. Power the ESP32 through the laptop. 2. Ensure Bluetooth connection is
Test Data	• Known weight: 500g sports item • Empty weight: 0
Testing Environment	ESP32 + HX711 load cell, Android phone, BLE connection.
Expected Result	• Weight should update immediately after adding or removing items.
Actual Result	Weight values updated correctly. Readings were stable after calibration,
Status	Passed

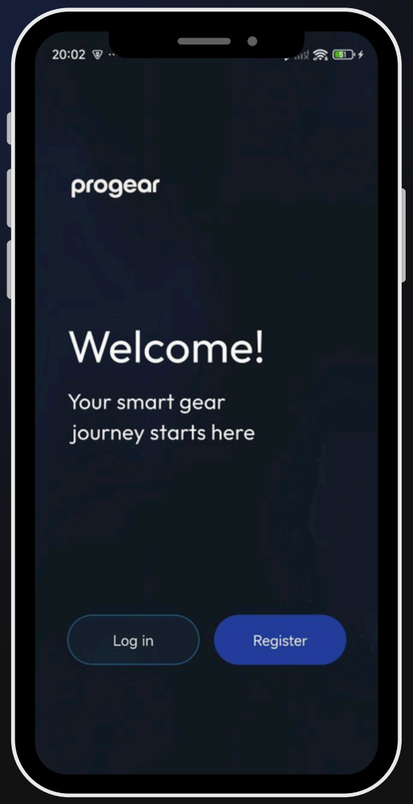
Test Case 2: Weight Measurement Accuracy

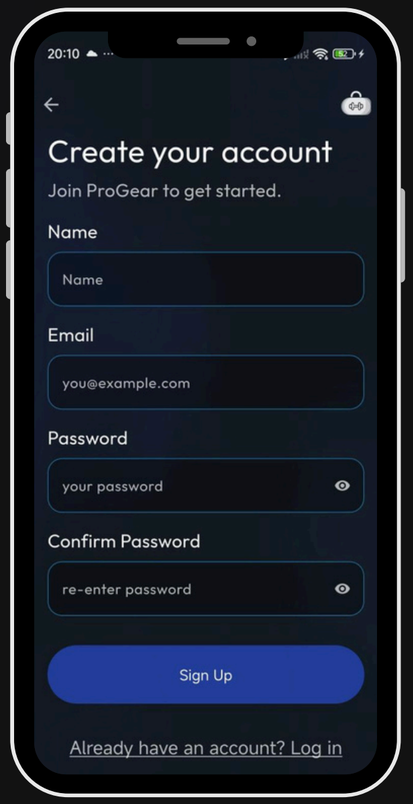
Hardware

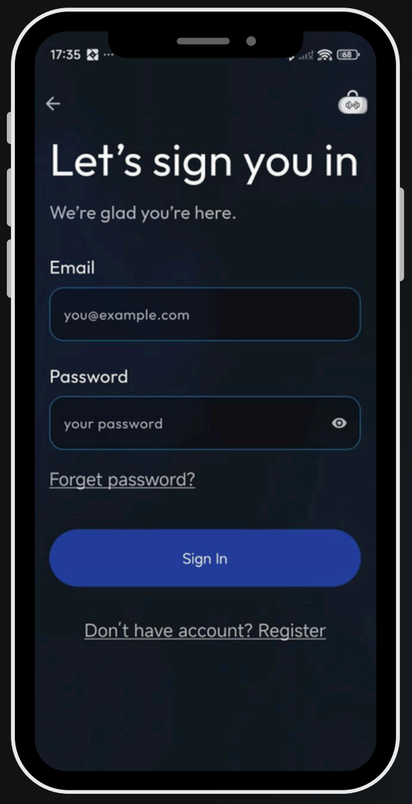


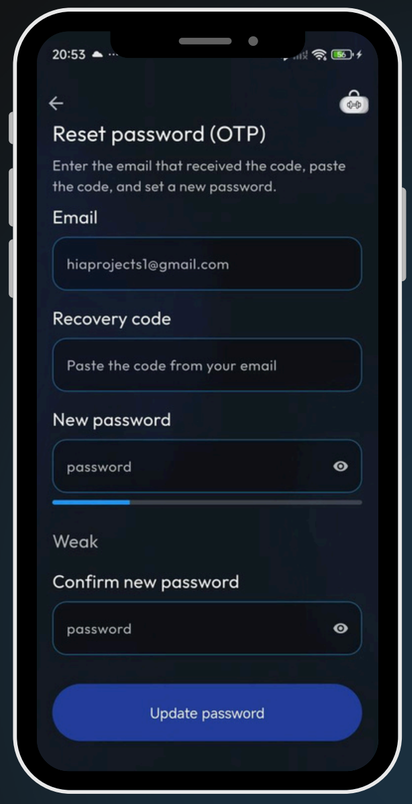


Mobile App Interfaces

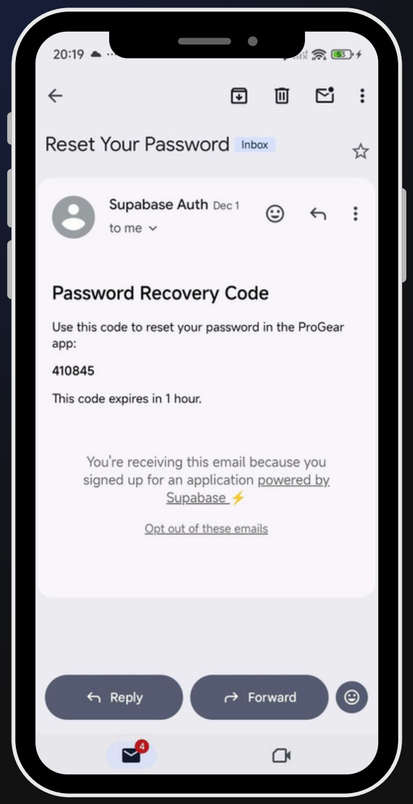


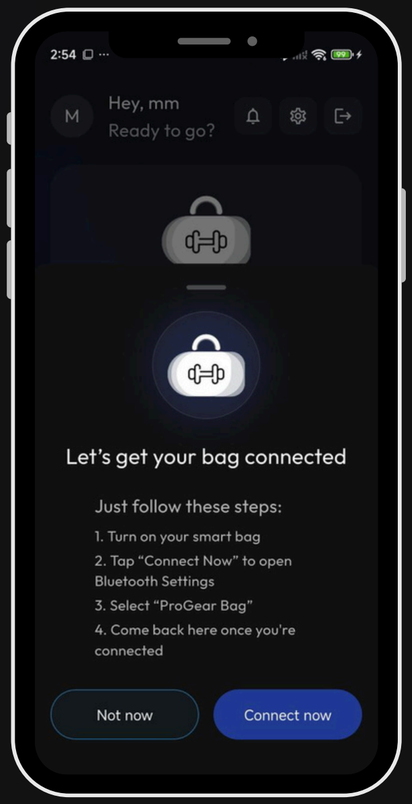


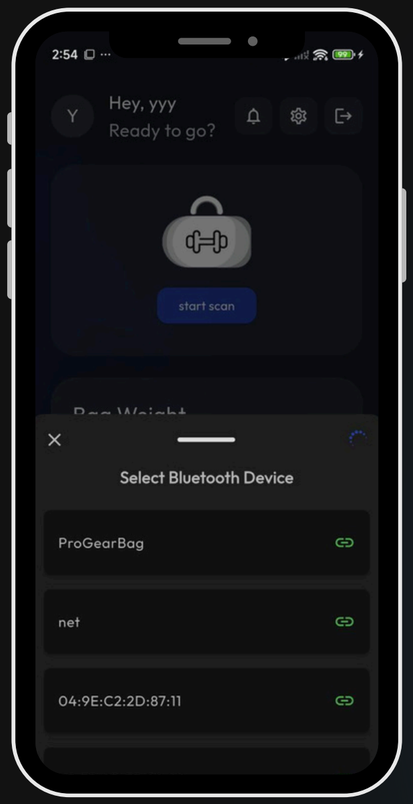


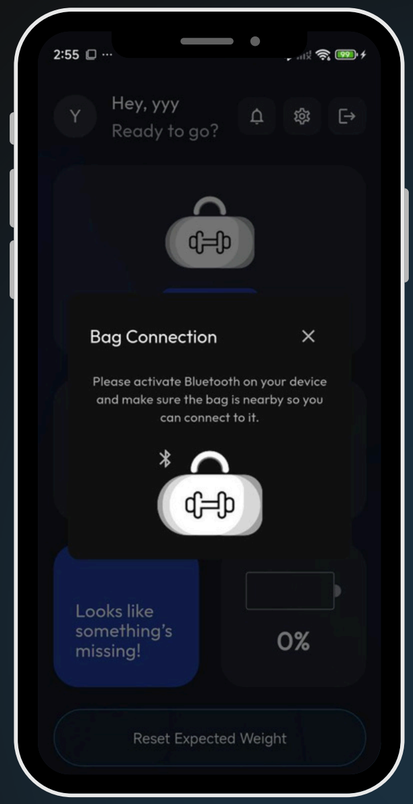


Mobile App Interfaces

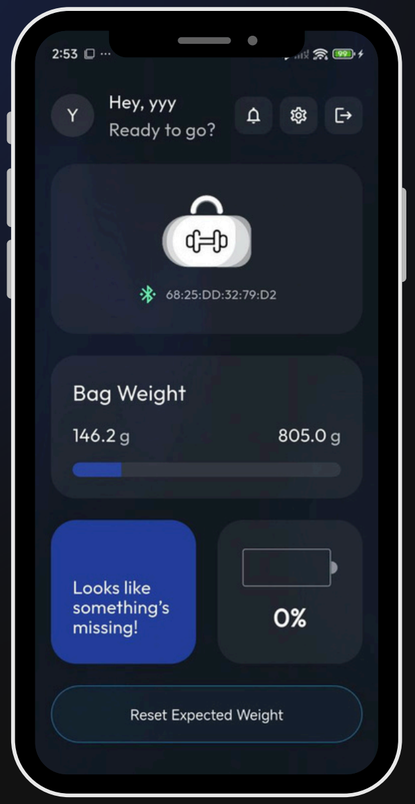


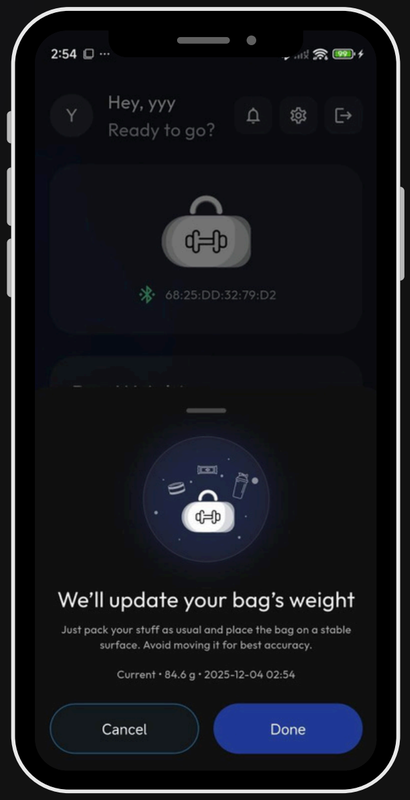


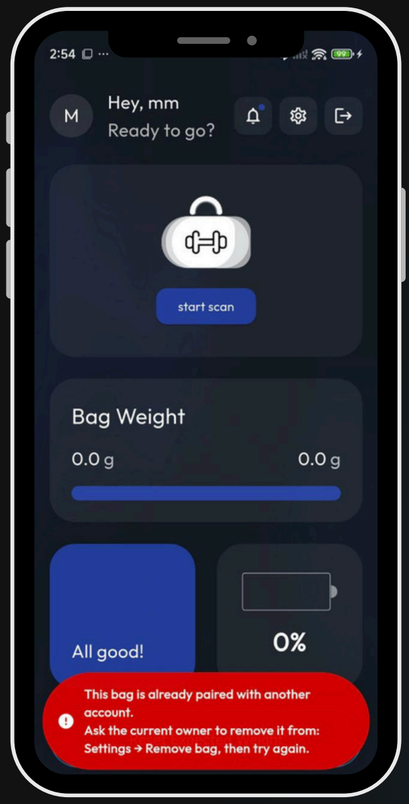


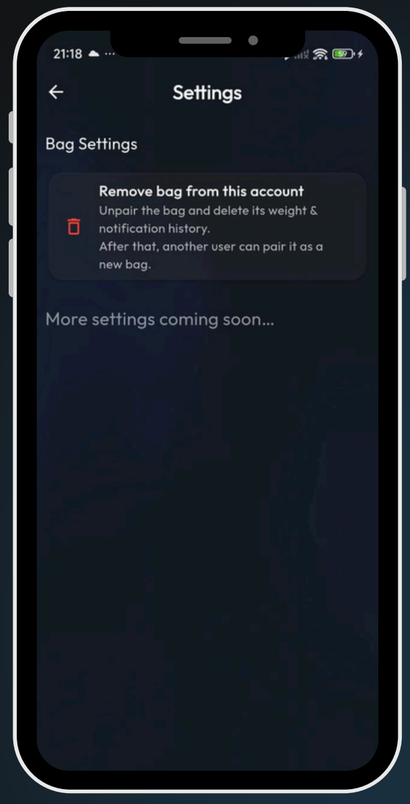


Mobile App Interfaces









Conclusion & Future Work

ProGear it’s more than just a smart bag; it's a step toward smarter, safer, and more focused training. By addressing a real problem with a practical and user-friendly approach, our project demonstrates how simple technology can make a meaningful impact in athletic preparation.

Future enhancements may include multi-sensor support, per-item identification using machine learning, adaptive thresholds, AI-based motion filtering, cloud integration for analytics, smarter notifications, and expanded mobile app features.

References

Thanks for listening !

Any Questions ?