

A circular graphic on the left side of the image. It features two crossed orange softball bats with white outlines, positioned behind a maroon diamond shape. Below the diamond is a white softball with red stitching. The entire graphic is set against a light orange circular background.

**SLOW-PITCH SOFTBALL CAMERA:**

**WEEKLY UPDATES**

SDMAY25-11



# CONSIDERING OUR USERS

To maximize our projects user-compatibility, what features **must** we include, **should** include, and **not** include based on user needs and preferences?



# WHO ARE OUR USERS?



Players practicing or playing recreational slow-pitch softball.

Umpires officiating recreational slow-pitch softball.



# OUR USERS' NEEDS

## PLAYERS

- Fast Response Time
- 

## UMPIRES

- Accurate Readings
- Audible Signals
- Adaptability for different fields
- Long battery life

# OUR USERS' WANTS

## PLAYERS

- View pitch videos
- View pitch statistics
- More readings
  - Speed
  - Spin
  - Trajectory map

## UMPIRES

- Affordability
- Portability
- Simple setup
- Tested as a reliable officary tool
- Device protection for in-game use

# OUR USERS' NO-NOS

## PLAYERS

- Exclusivity for in-game use

## UMPIRES

- Interrupting "Illegal" calls on hits
- Physical components interfering with gameplay



# CONCLUSIONS

- Both users and referees want to preserve the integrity of the game. An accurate, non-intrusive, responsive model is needed for easy gameplay.
- Referees desire an easy setup and long battery life for officiating multiple games
- Players would like an opportunity to view past pitches and their statistics.
- All users would like a cost effective and portable tool.

# DESIGN CONSIDERATIONS



## YOLOV9

Trained object detection model for accurate results



## MOBILE APP

A mobile app allows for an affordability and portability.



## C++

Fast computing language to analyze camera video



## CALIBRATION

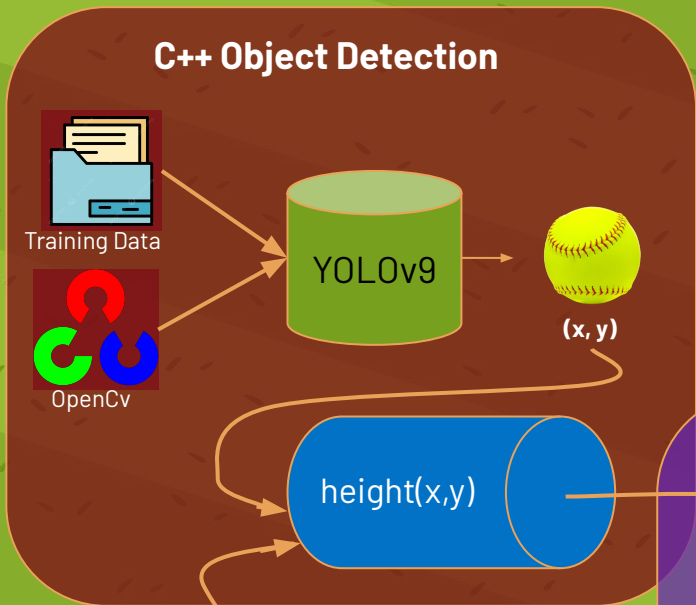
A setup calibration allows for adaptability to each field



## IN-APP STORAGE

Stored videos can be viewed/exported





**OUR CURRENT DESIGN MODEL**



# PROTOTYPE PROGRESS

Here's what we've been up to...



# DATA COLLECTION - YOLOV9

- Taken at different stages of a sunset
- Taken at different viewing angles
- Variety of low/high pitches



# REACT-NATIVE EXPO

- Although React-Native Expo has cross platform compatibility, it cannot run C++ within the application.
- Camera inputs and components are accessed differently with different platforms.



# FLUTTER

- Google's toolkit for cross-platform UI development
- Written in Dart, similar in syntax to Java and C++
- C++ can be directly ran within the application
- Camera plugin allows for integrating the camera feed into the YOLOv9 C++





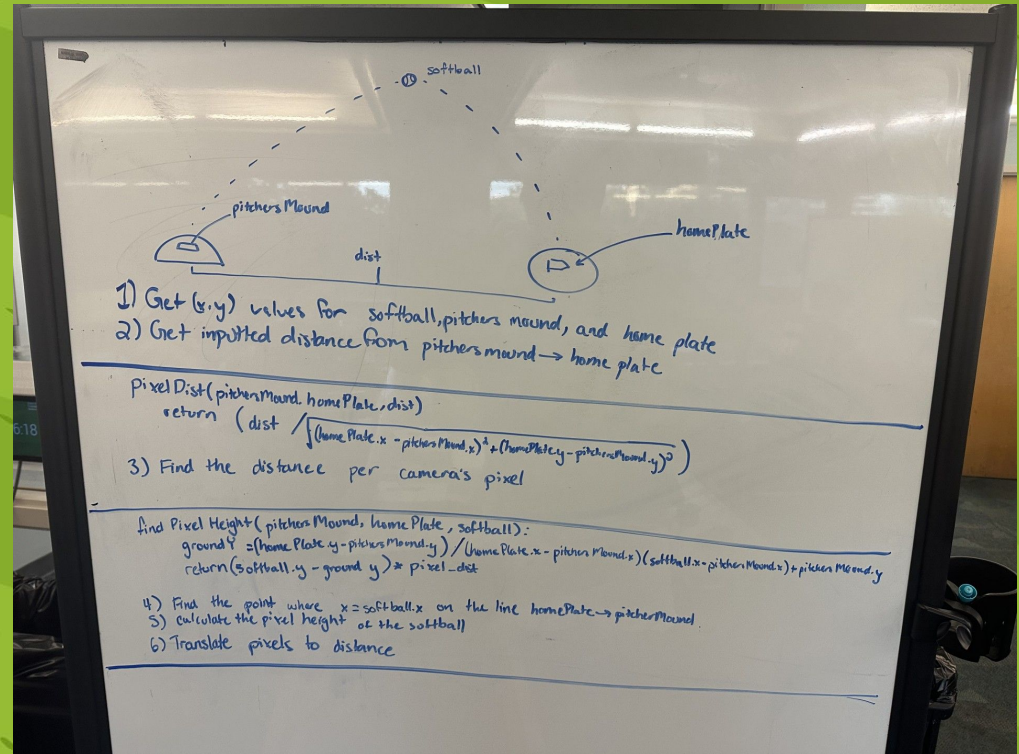
# NEXT WEEK...

Here's what we will work on next week



# C++ OBJECT DETECTION TESTING

- Using the research data, training a YOLOv9 model to detect softballs from a constant camera feed.
- Once the program can successfully and reliably detect a softball, we must find how to record trajectory and calculate height with a constant video stream from the camera



# MAKING A FLUTTER APP

- Need to configure and set up the repository for a Flutter application
- Have testing instructions to run on Android or iOS
- Make screen sketches for all screens and user interactive portions.



# GITHUB ISSUES

## Flutter App

Open Issue created 35 minutes ago by joshhyde

Making a flutter app to be able to code on one app for both IOS and Android. Due Date is a rough estimate

- Create a flutter app
- Edit the README.rd for installation instructions
- Install the camera plugin

0 of 3 checklist items completed · Edited just now by ethgru



## Object Detection

Open Issue created 36 minutes ago by joshhyde

A working object detection model that can accurately detect a softball consistently. The due date is estimated.

- Create a model to consistently and accurately detect a softball on the screen (x,y)
- Begin working on calculating height

0 of 2 checklist items completed · Edited just now by ethgru



## Mobile App User Experience

Edit ⋮

Open Issue created 5 days ago by ethgru

- Decide what features will be included in the app
- Design the screens associated with each feature
- Design the dashboard for navigating through the app
- Assign each page a module in the app

0 of 4 checklist items completed



Create merge request ▾