# RevMetrix Capstone II

Matt Brown, Josh Byers, Charles Carroll, Zach Cox, Joseph Downey, Gabe Manero, Jakeb Nielsen, Andrew Olvera, Gavin Wentz, and Hunter Wolfe

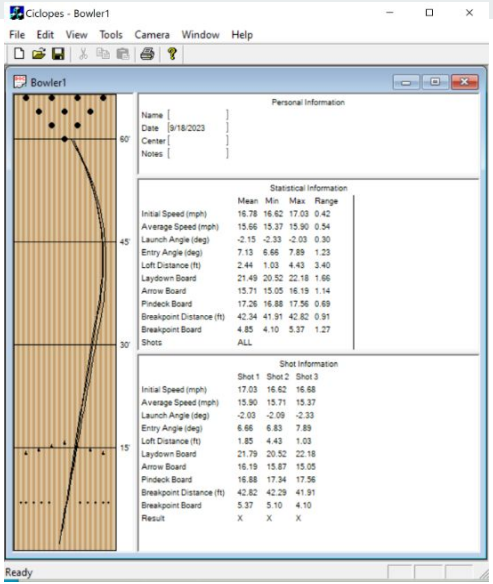
# Background

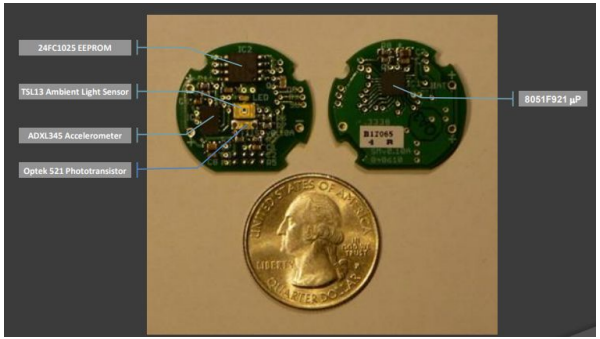
# What problem are we trying to solve?

Absence of a unified, user friendly system for collecting, organizing, and analyzing bowling performance data.

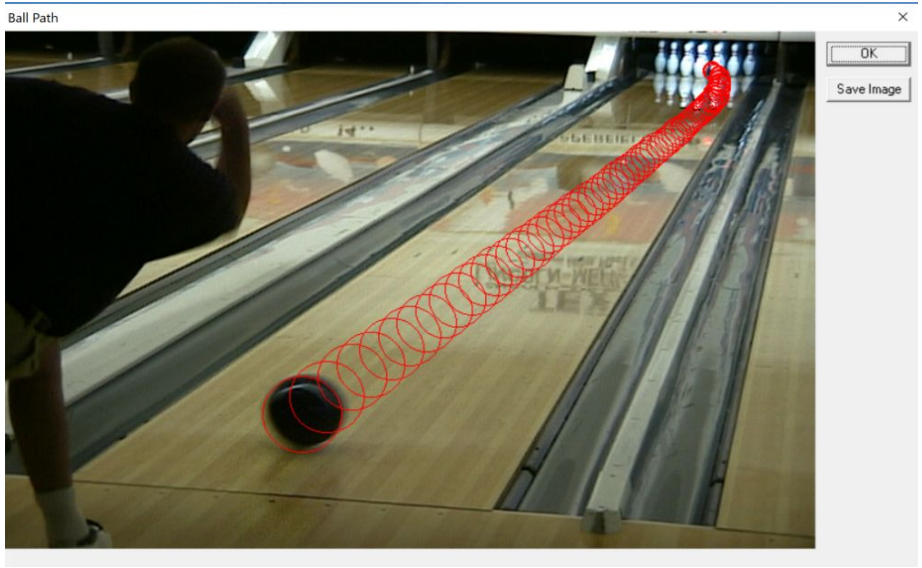
# Founding Technologies

- Creation of Prof. Hake and Dr. Babcock
- Key Components
  - SmartDot module
  - Ciclopes software



# What is Ciclopes?




Ciclopes - Bowler1

File Edit View Tools Camera Window Help

Bowler1

60'
45'
30'
15'

Personal Information

Name [ ]
Date [9/18/2023 ]
Center [ ]
Notes [ ]

Statistical Information

| | Mean | Min | Max | Range |
|---|---|---|---|---|
| Initial Speed (mph) | 16.78 | 16.62 | 17.03 | 0.42 |
| Average Speed (mph) | 15.66 | 15.37 | 15.90 | 0.54 |
| Launch Angle (deg) | -2.15 | -2.33 | -2.03 | 0.30 |
| Entry Angle (deg) | 7.13 | 6.66 | 7.89 | 1.23 |
| Loft Distance (ft) | 2.44 | 1.03 | 4.43 | 3.40 |
| Laydown Board | 21.49 | 20.52 | 22.18 | 1.66 |
| Arrow Board | 15.71 | 15.05 | 16.19 | 1.14 |
| Pindeck Board | 17.26 | 16.88 | 17.56 | 0.69 |
| Breakpoint Distance (ft) | 42.34 | 41.91 | 42.82 | 0.91 |
| Breakpoint Board | 4.85 | 4.10 | 5.37 | 1.27 |
| Shots | ALL | | | |

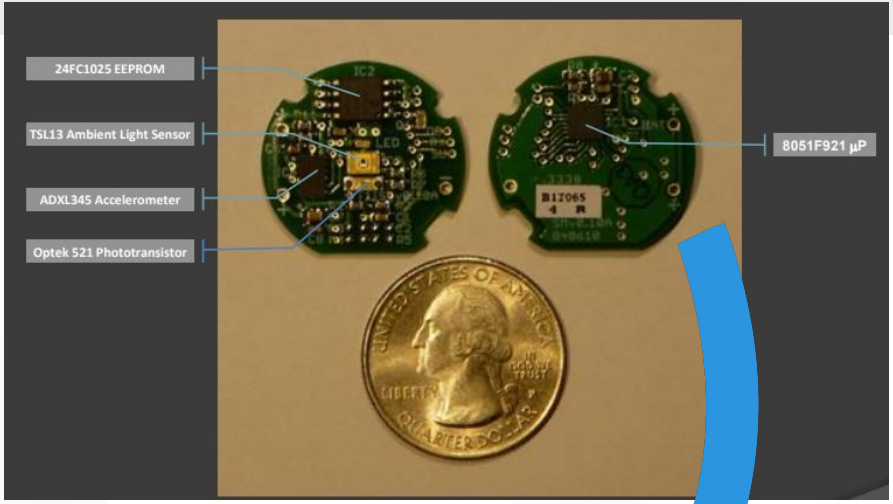
Shot Information

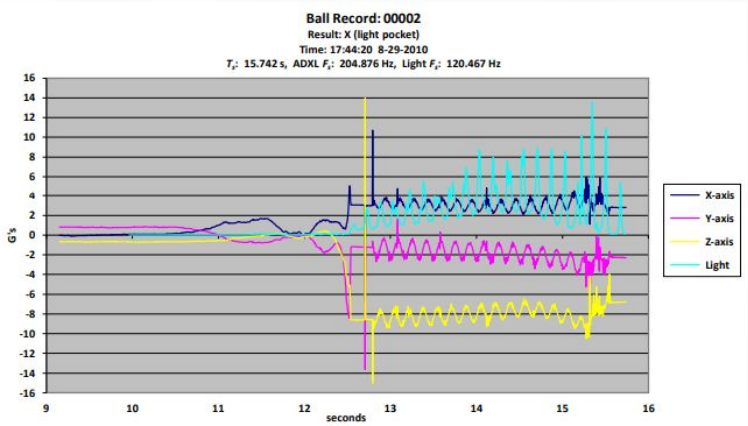
| | Shot 1 | Shot 2 | Shot 3 |
|---|---|---|---|
| Initial Speed (mph) | 17.03 | 16.62 | 16.68 |
| Average Speed (mph) | 15.90 | 15.71 | 15.37 |
| Launch Angle (deg) | -2.03 | -2.09 | -2.33 |
| Entry Angle (deg) | 6.66 | 6.83 | 7.89 |
| Loft Distance (ft) | 1.85 | 4.43 | 1.03 |
| Laydown Board | 21.79 | 20.52 | 22.18 |
| Arrow Board | 16.19 | 15.87 | 15.05 |
| Pindeck Board | 16.88 | 17.34 | 17.56 |
| Breakpoint Distance (ft) | 42.82 | 42.29 | 41.91 |
| Breakpoint Board | 5.37 | 5.10 | 4.10 |
| Result | X | X | X |

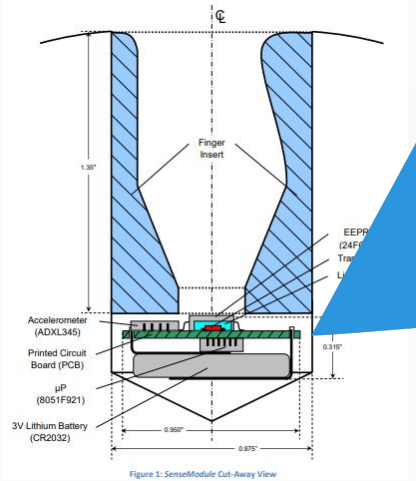
Ready

# What is SmartDot?

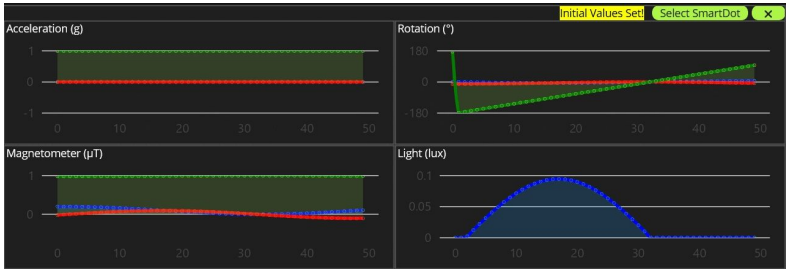
- Professor Hake's Brain-Child
- Light Sensor + Accelerometer

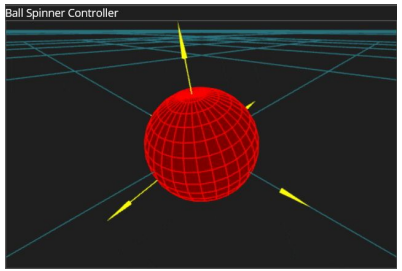


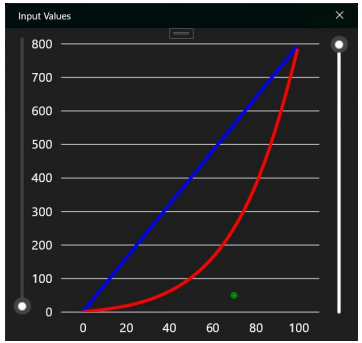



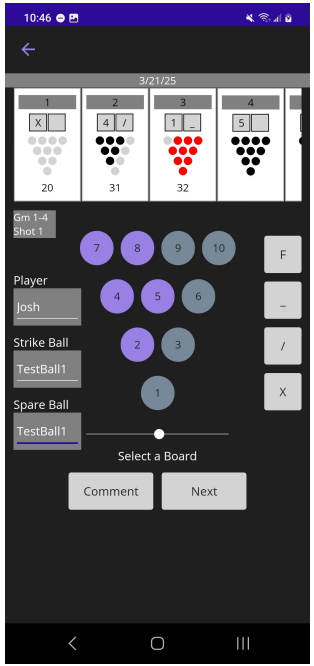



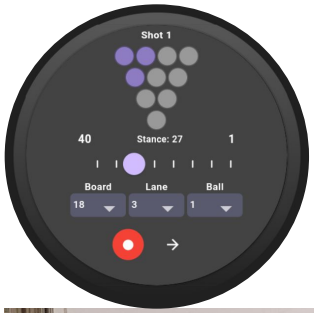
# Where did last year leave us?

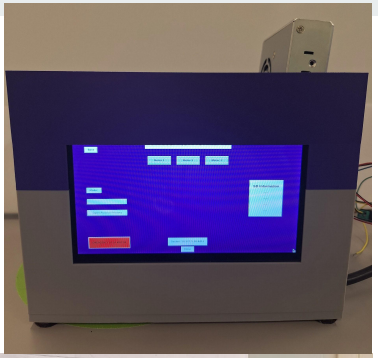


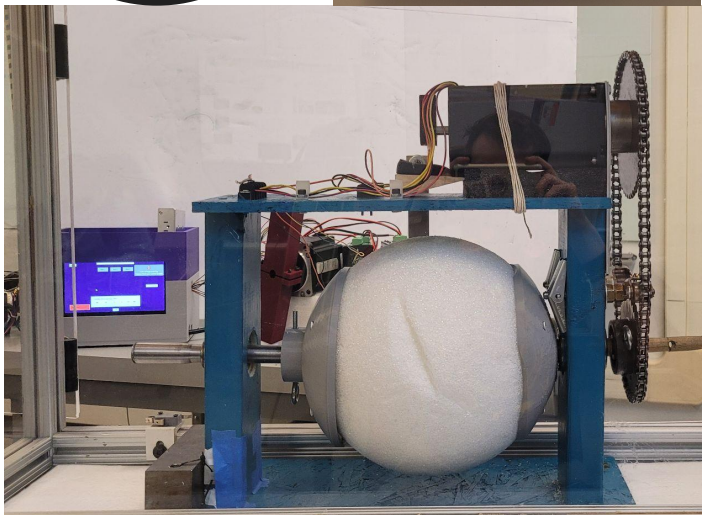



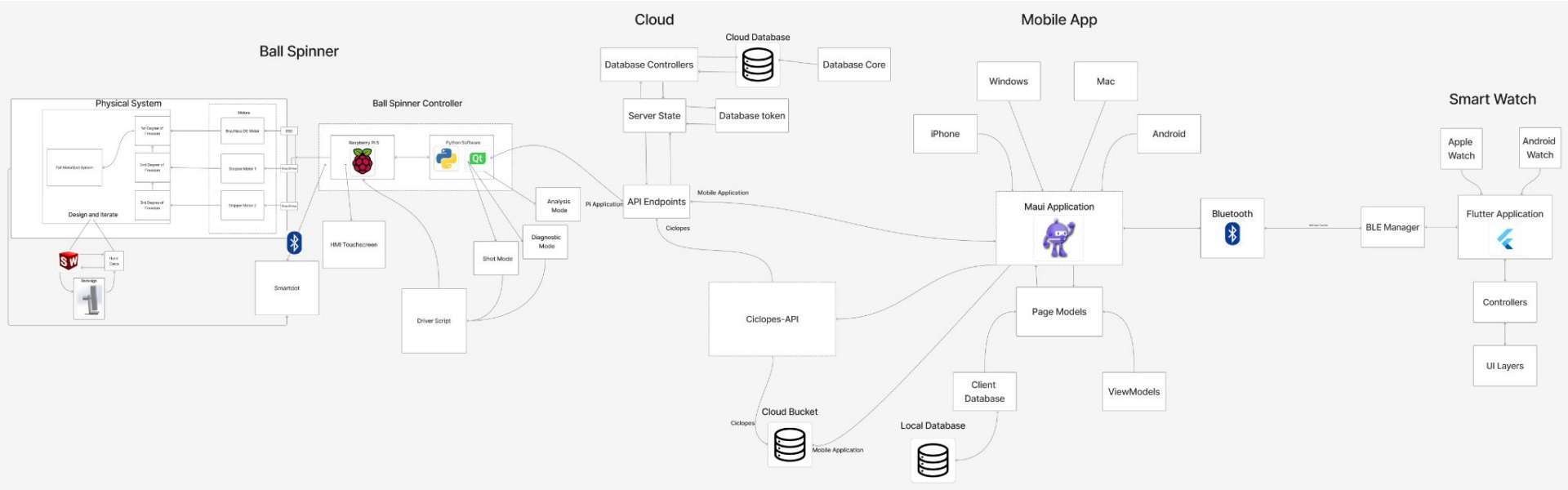





# Diagram of System Overview



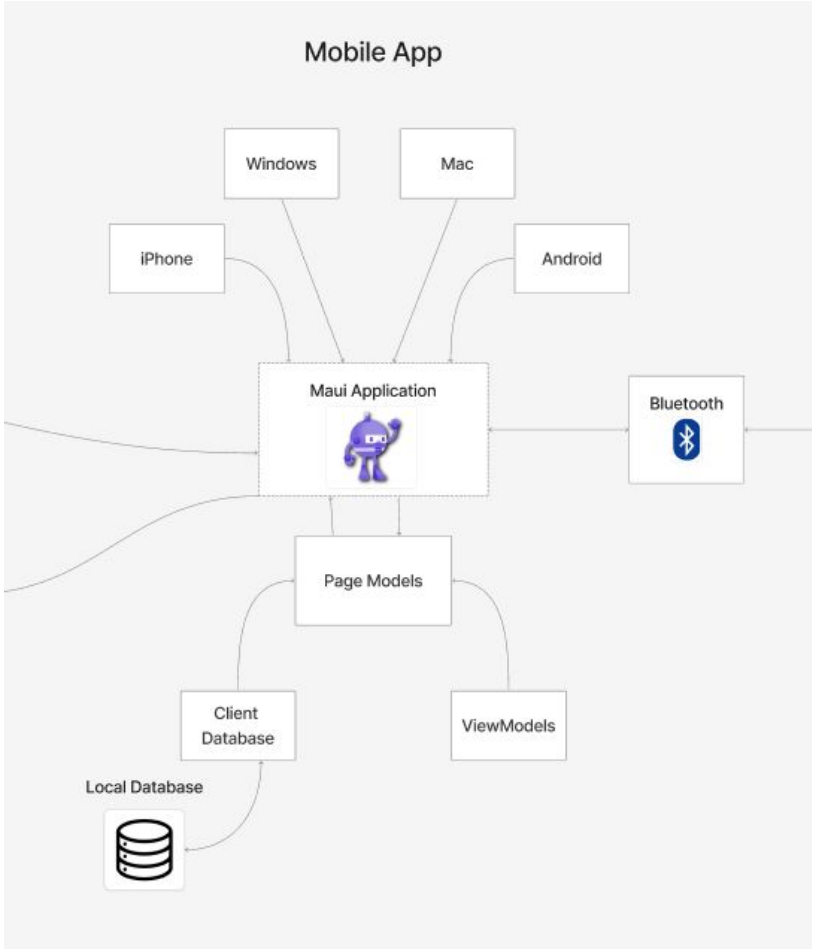
# Unified Demo Watch -> Mobile -> Ciclopes

# Brief Ball Spinner Overview Demo

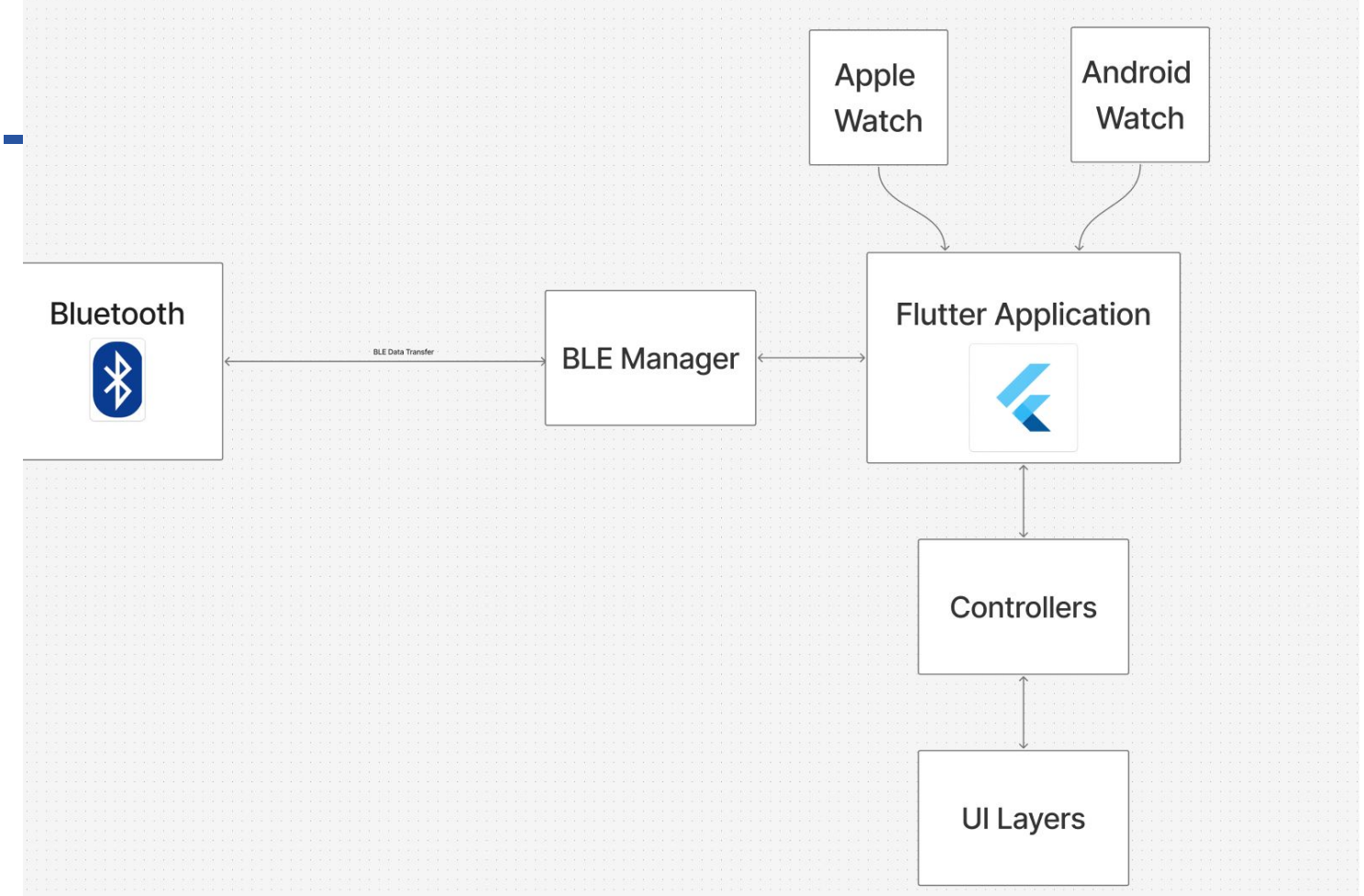
# Questions?

# Analysis and Design

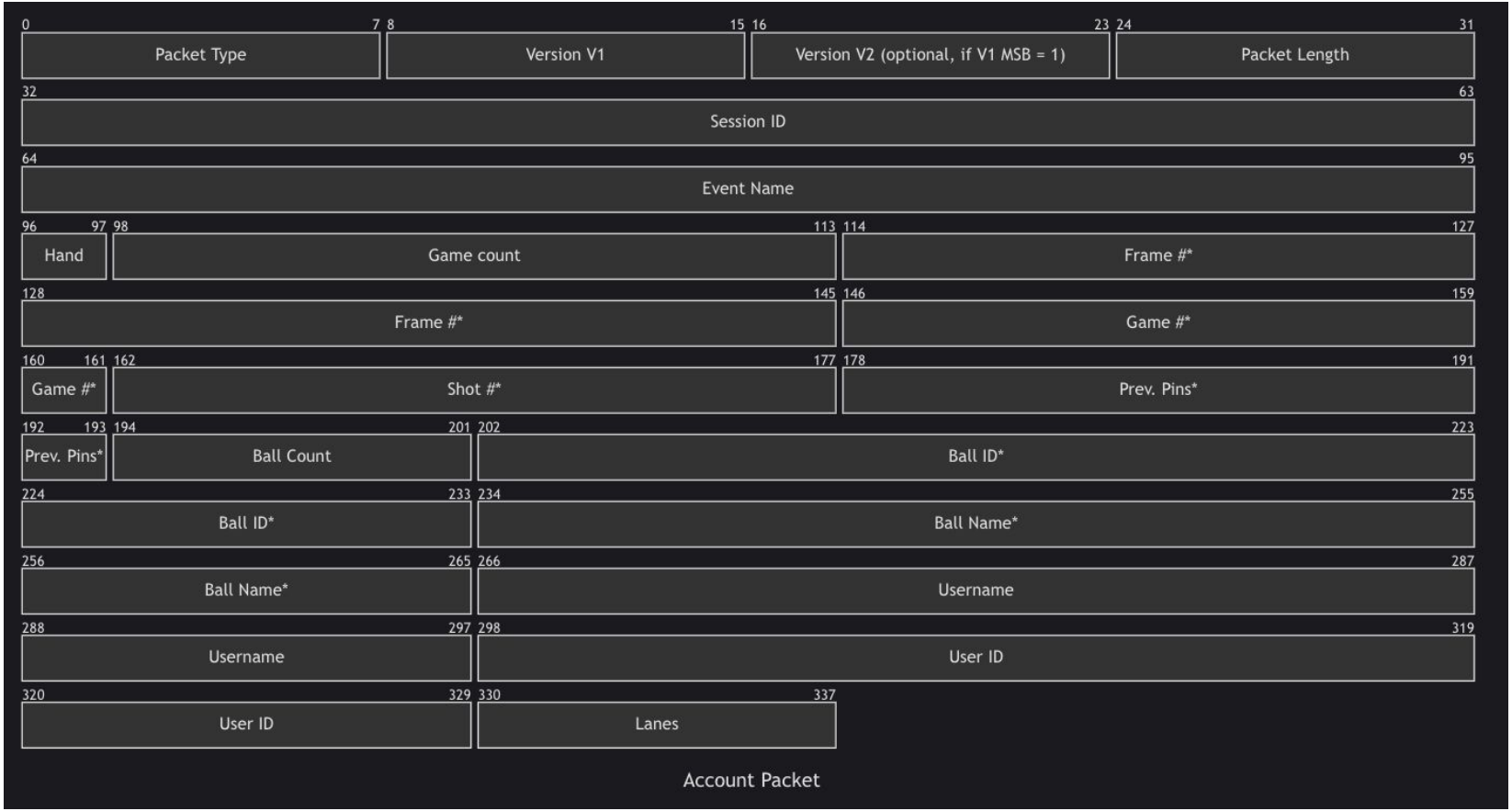
# Mobile: High-Level Overview



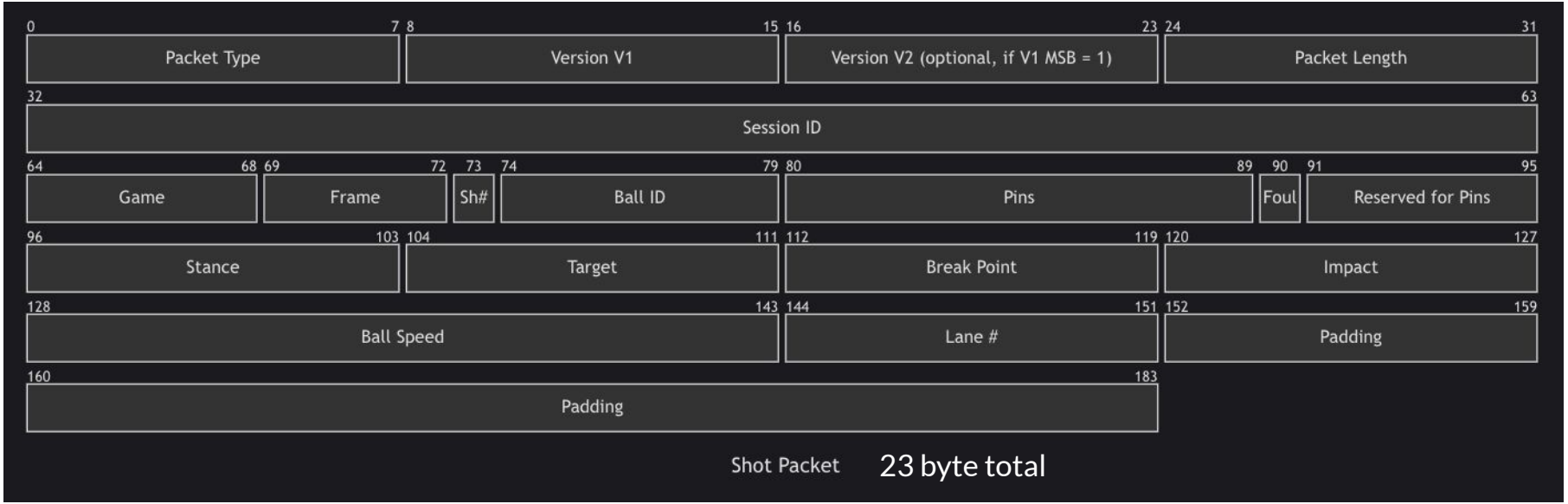
# Watch High-Level Overview



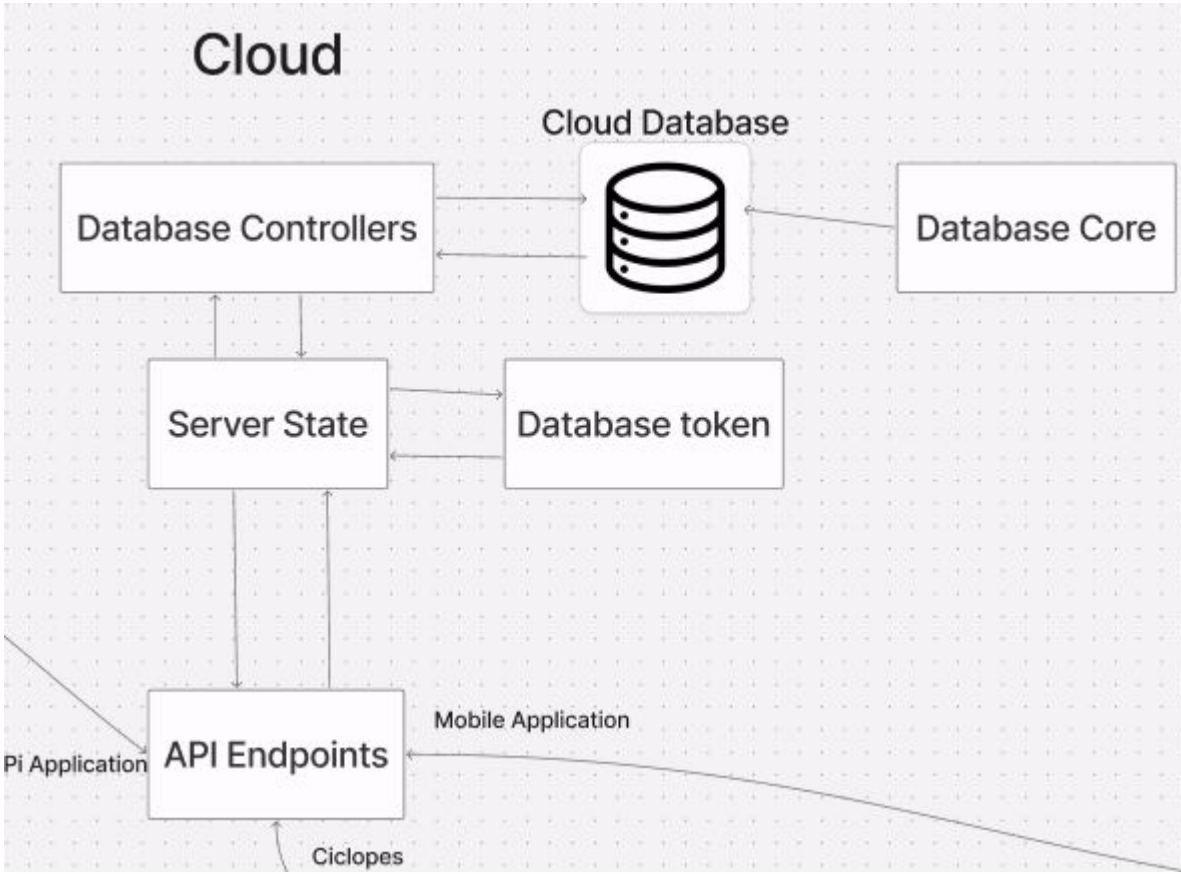
# Account Packet Diagram



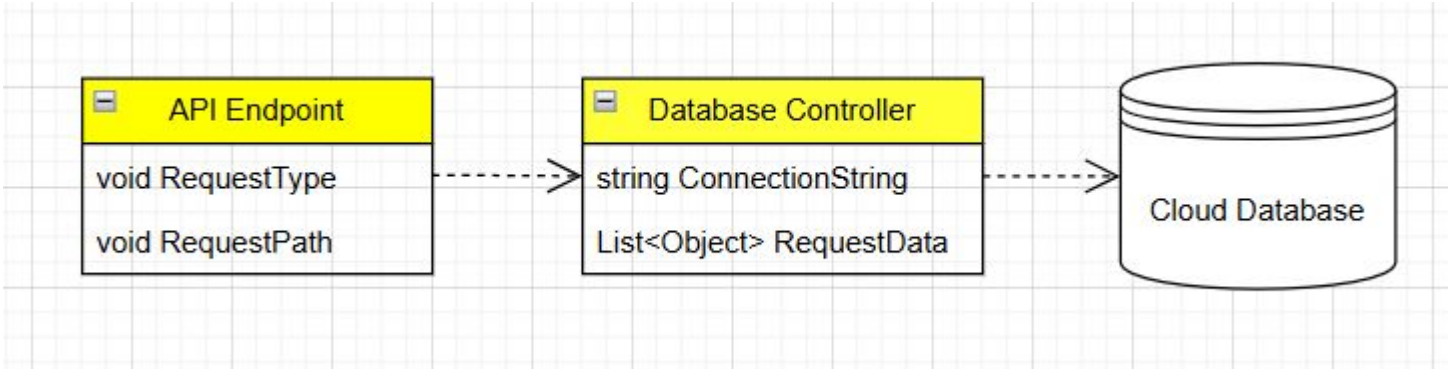
# Shot Packet Diagram



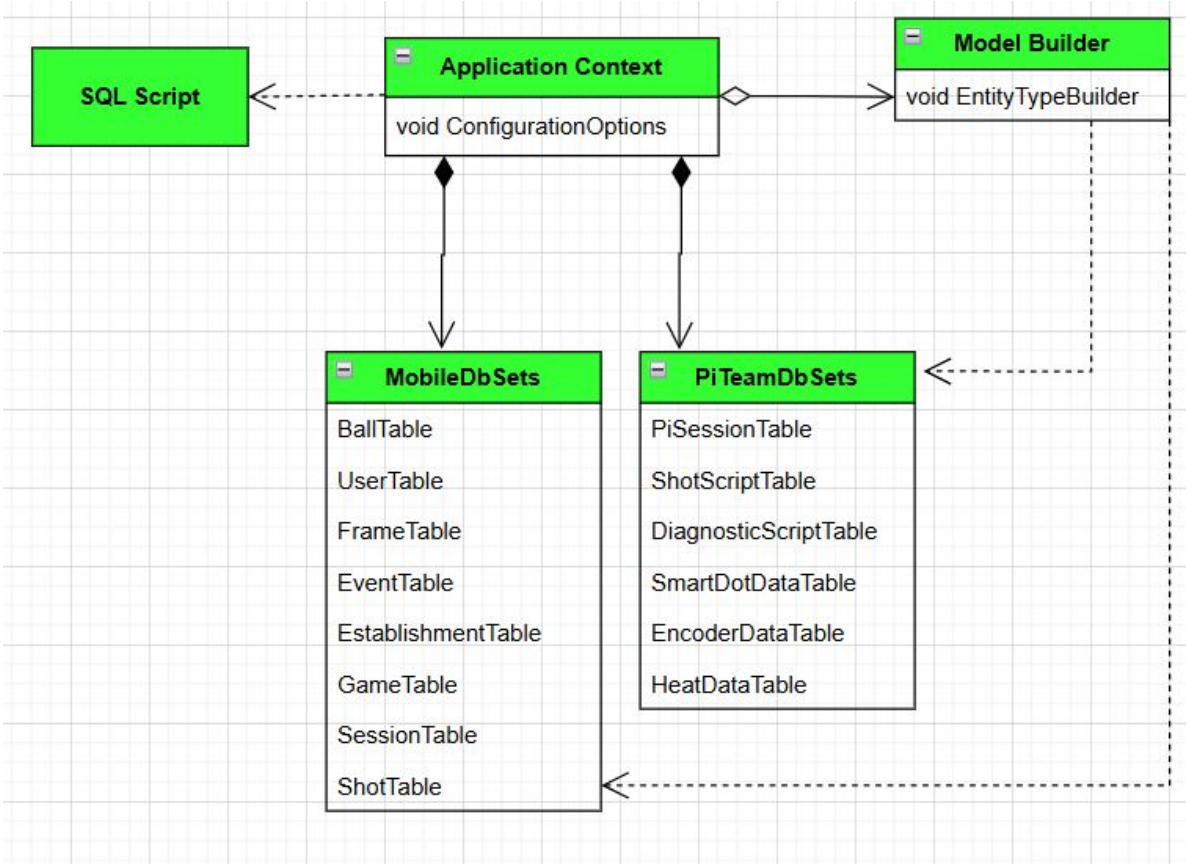
# Cloud
# High-Level Overview



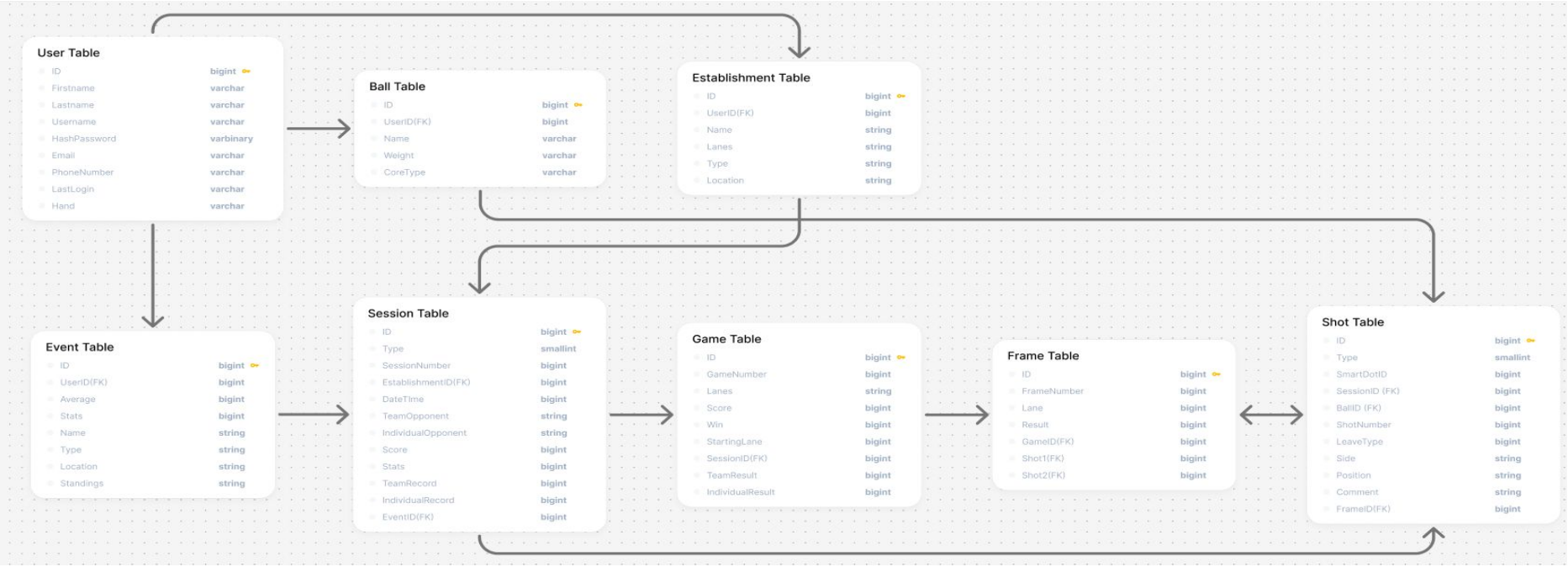
# Cloud - Cloud Application UML Diagram



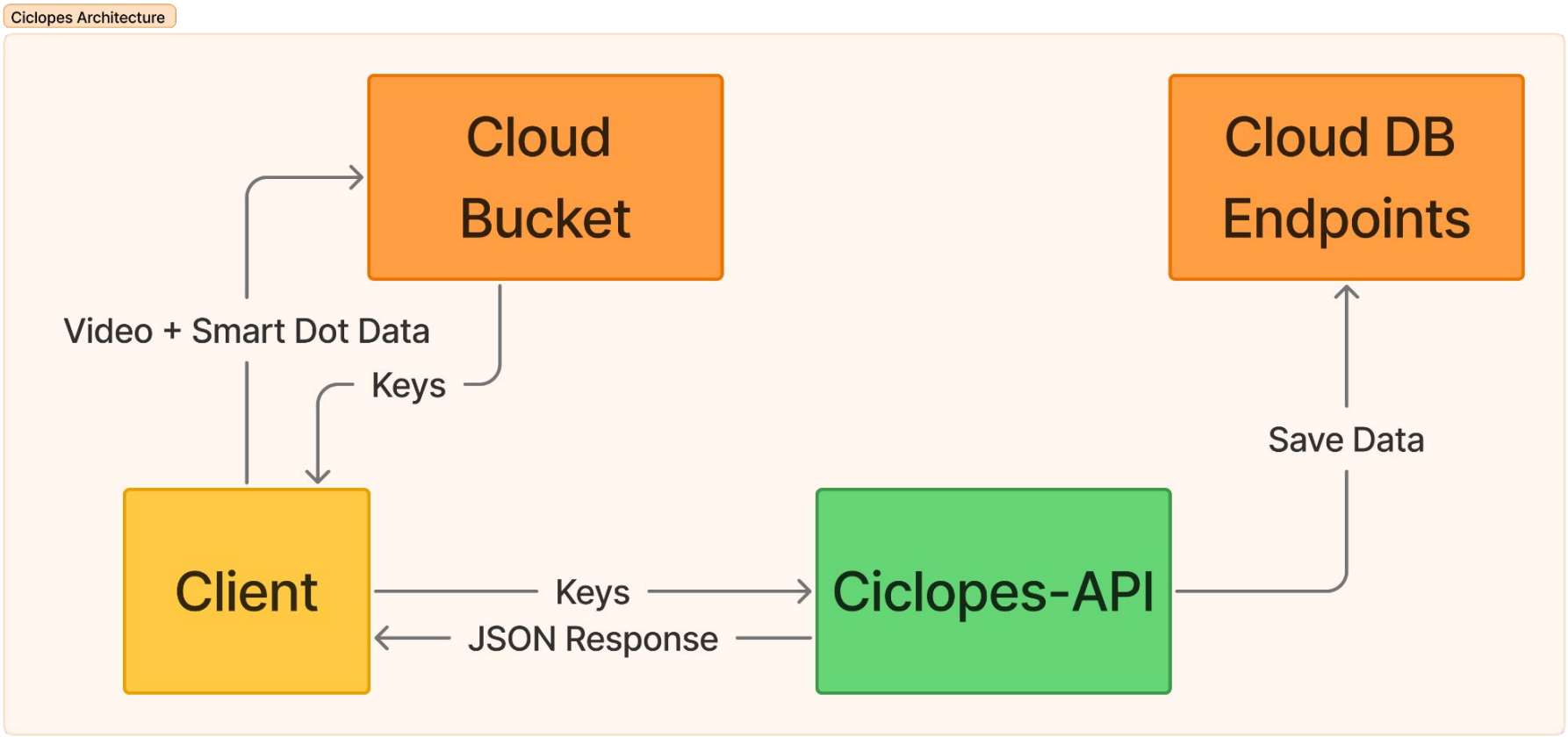
# Cloud - DBCore UML Diagram



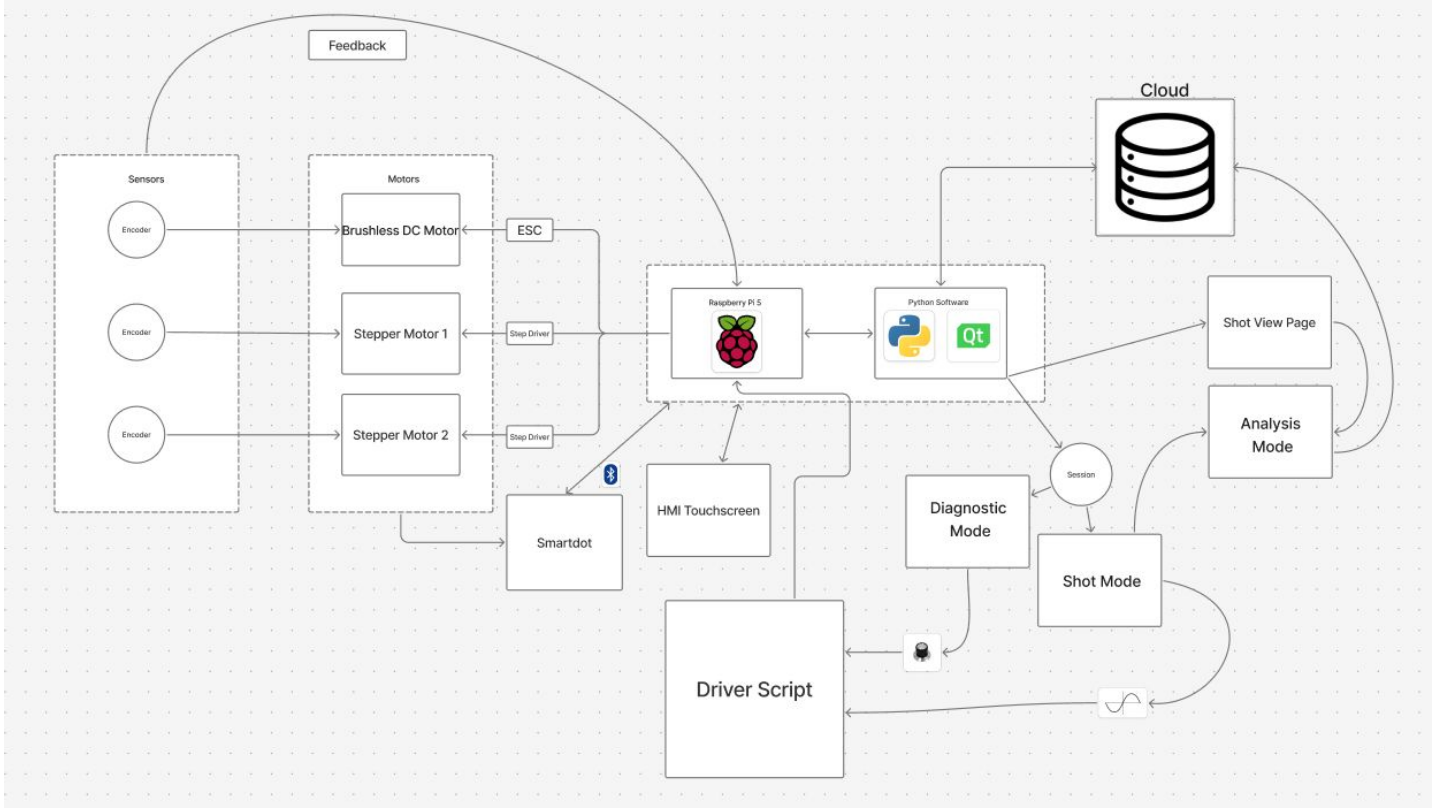
# Cloud - Mobile Database Schema



# Ciclopes - Architecture

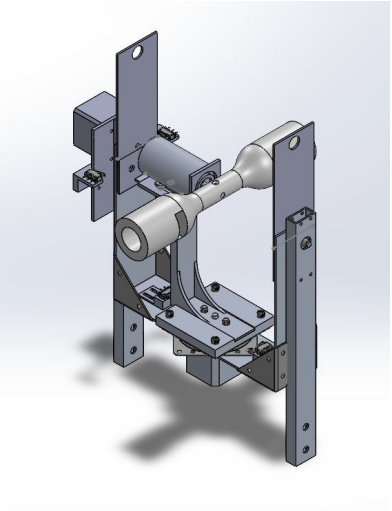


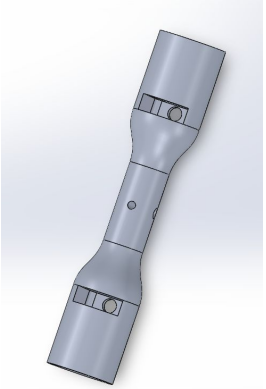
# Ball Spinner Mechanical System High Level Overview



# Ball Spinner Mechanical System

- **3 axes of rotation controlled independently**
  - X-axis: 1st DOF, Smart Dot Holder attached to motor by friction fit and set screw
  - Y-axis: 2nd DOF, rotating plate holding 1st dof
  - Z-axis: 3rd DOF, U-bracket which rotates 1st and 2nd
- **SmartDots mount into SmartDot Holder by a friction fit**
  - Allows for collection of light, acceleration, and positional sensor data
  - Easy removal for charging or replacement
- **Calculated torques based on required speed and angle**



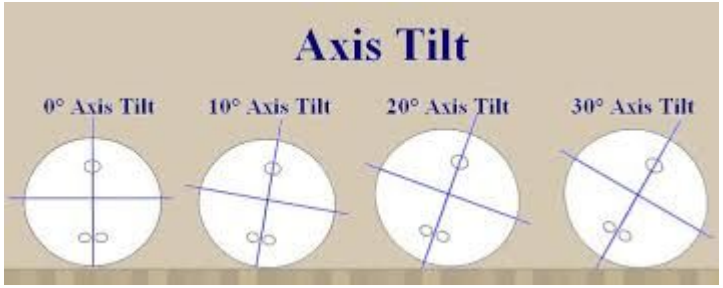


# Ball Spinner Axis Overview

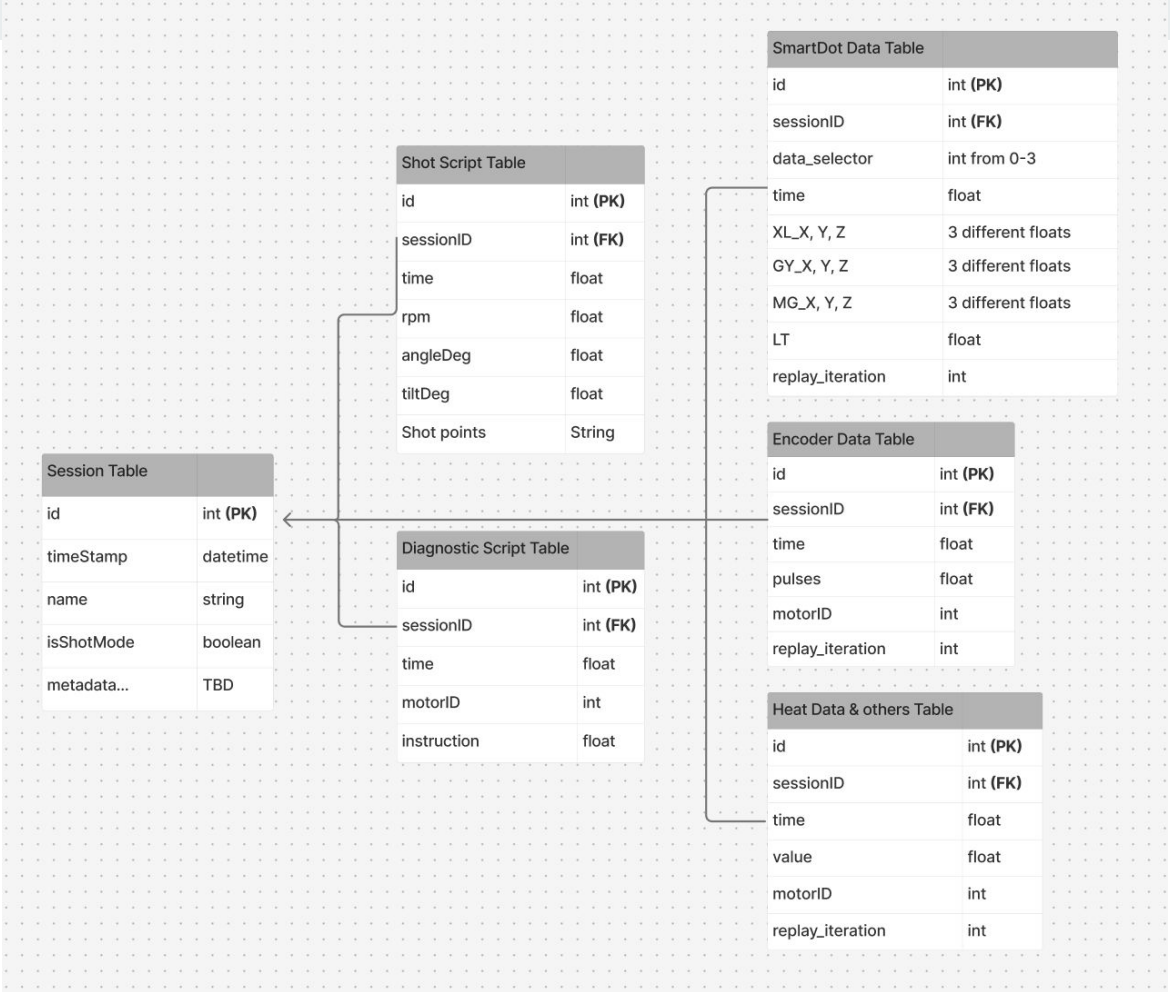
Spin(1st DOF): The Primary axis of rotation

Angle(2nd DOF): The Angle between the direction of rotation and going straight down the lane

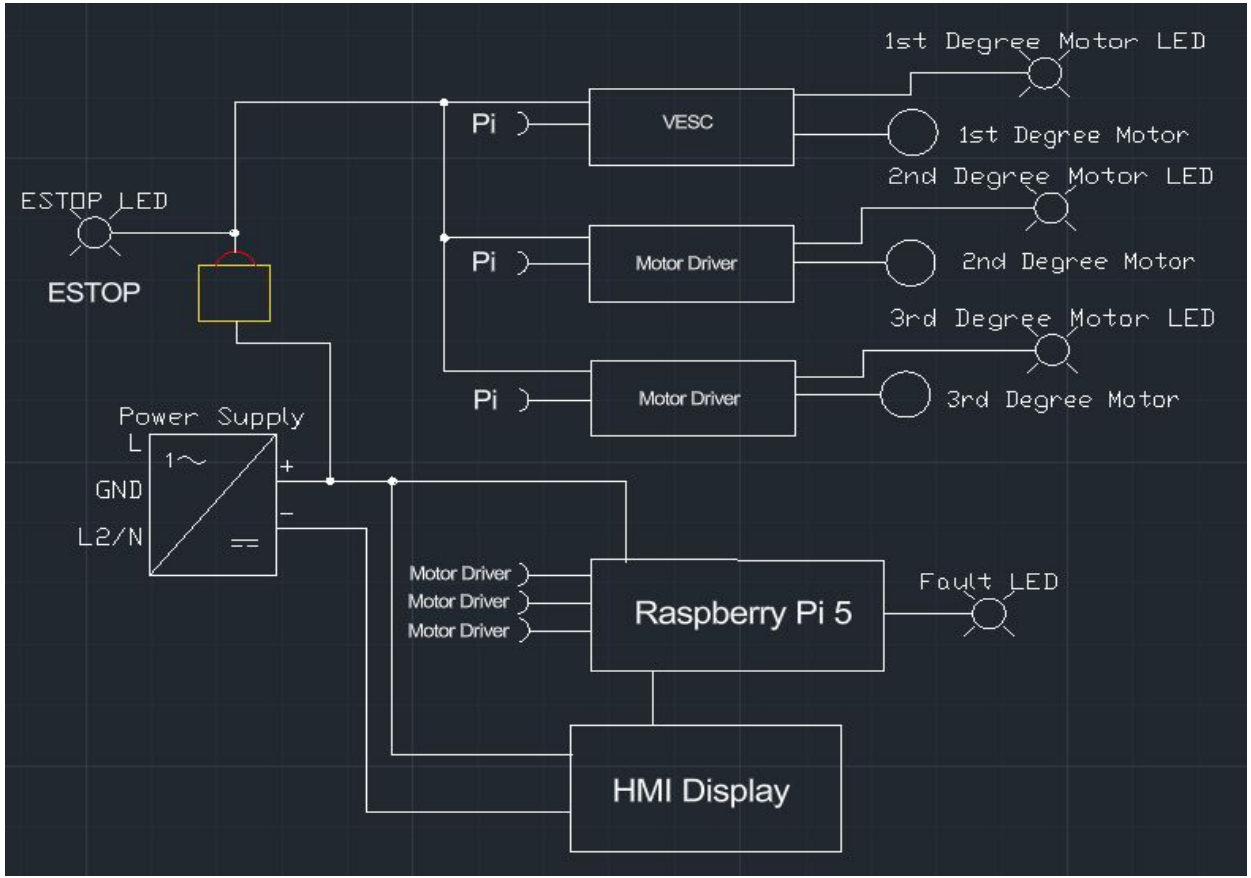
Tilt(3rd DOF): The Angle between the primary axis of rotation and being level with the lane.




# Cloud - Pi Team Schema



# Ball Spinner Controller Block Diagram



# Power and Control System Overview

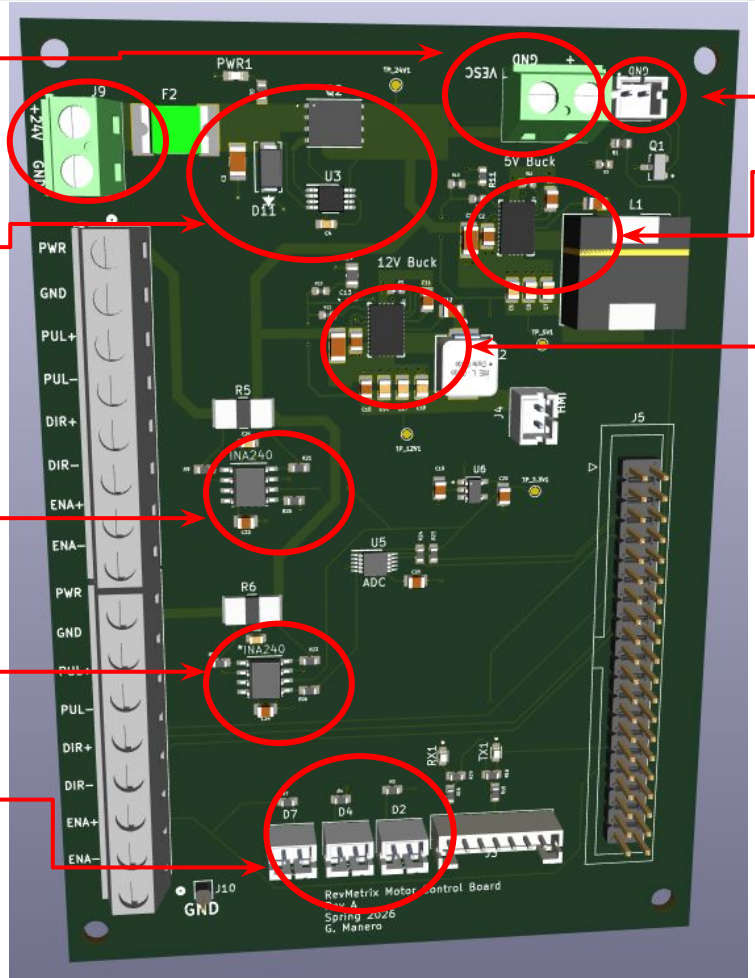
ESC Power

24V INPUT

Reverse Polarity Protection

Current Sensors

Motor Indicator Lights




Fault Indicator Light

5V power

12V power

**Printed Circuit Board (PCB)**

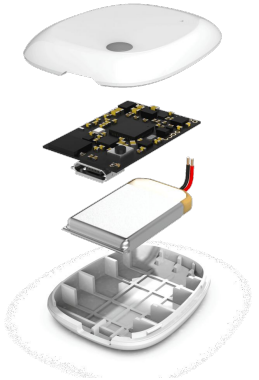
*The Control and Power Hub of The RevMetrix Ball Spinner*

# Questions?

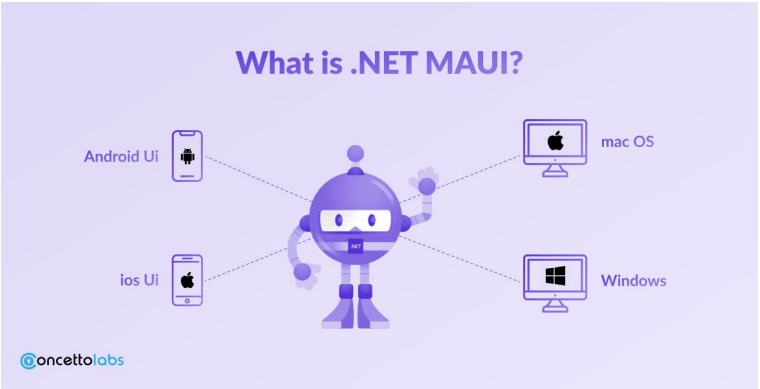
# Tools & Tech

# Mobile App - Development Technologies

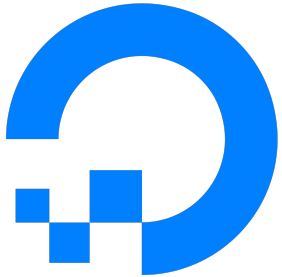
- .NET Maui C#
- MetaMotionS/MetaMotionC
- Bluetooth Low Power (BLE)



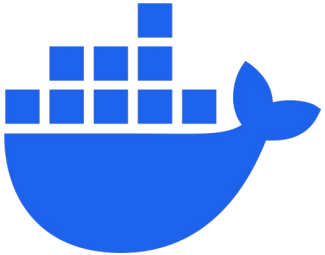
Bluetooth™
Low Energy



# Cloud - Development Technologies



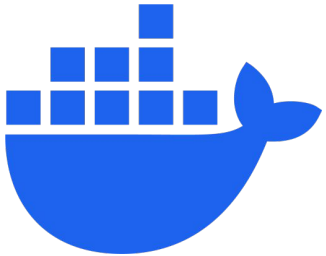
- Postman
- Microsoft EF (Entity Framework) Core
- Liquibase
- Digital Ocean
- Docker
- Swagger Documentation

# Ciclopes - Development Technologies

- Python
- PyTorch
- Docker
- IsaacSim
- NumPy
- SciPy
- OpenCV2
- FastAPI
- NVIDIA GPUs



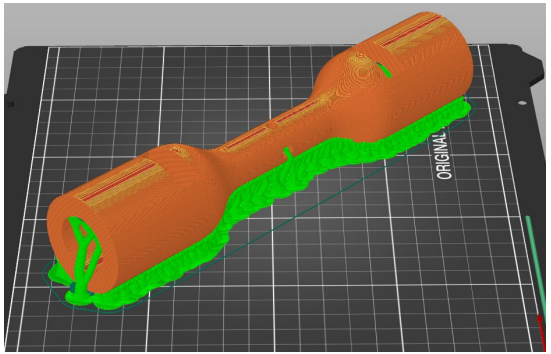



# Watch - Development Technologies

- Flutter
- Dart
- Kotlin
- Visual Studio Code
- Bluetooth Low Energy (BLE)
- Android/IOS Launcher (Emulator)

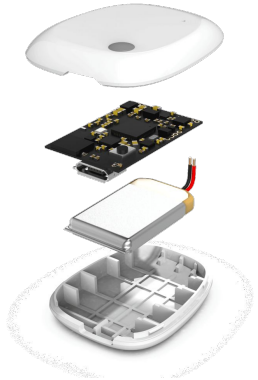

# Ball Spinner Mechanical System

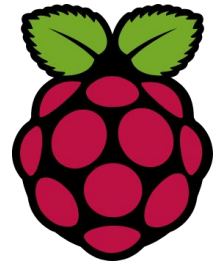
- **SolidWorks**
  - SolidWorks was used throughout the semester to create models of the team's designs as well as to perform FEA testing
- **3D Print Lab**
  - Utilized to rapidly prototype parts for the physical system and produce some final parts
- **Machine Shop**
  - The Machine Shop was used extensively to create physical prototypes for the team to visualize and criticise designs.

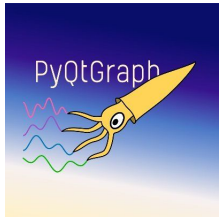



# Ball Spinner Controller Technologies

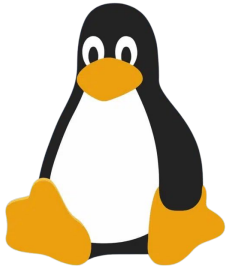
- Linux
- Python
- PyQt6
- PyQtGraph
- PyWavelets
- Raspberry Pi
- MetaMotionS





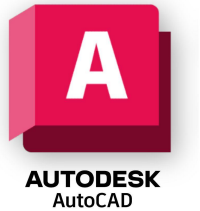




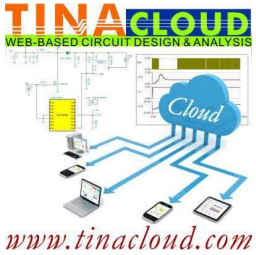


# Ball Spinner Controller Technologies Continued

- LGPIO
- Pyvesc
- AutoCAD
- TINACloud
- OnShape
- KiCad



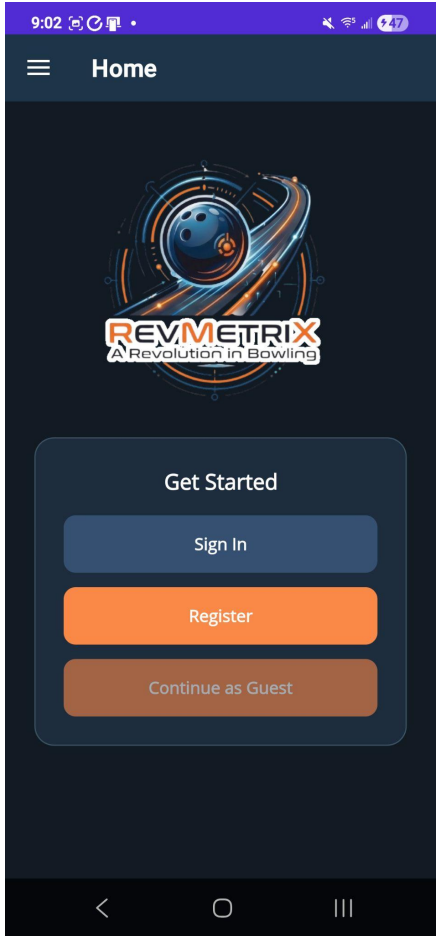


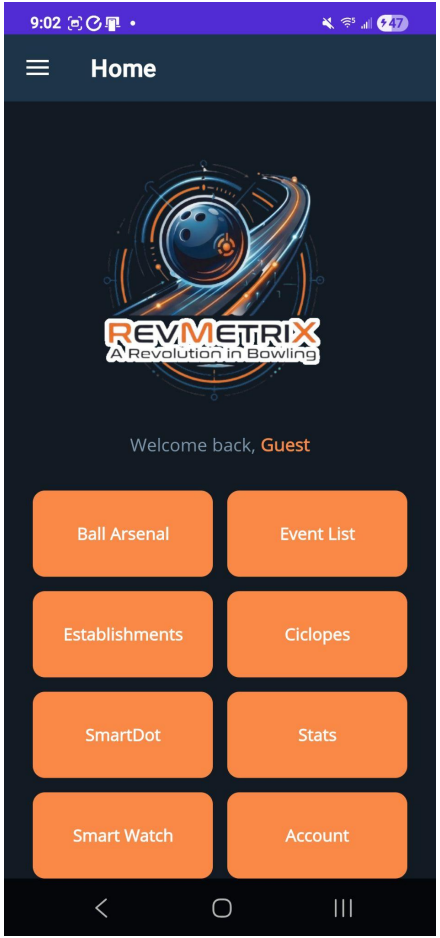



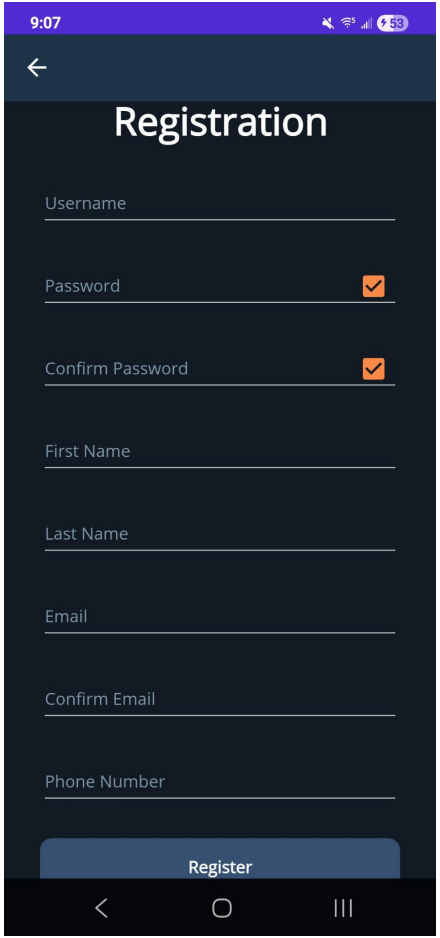
# Implementation

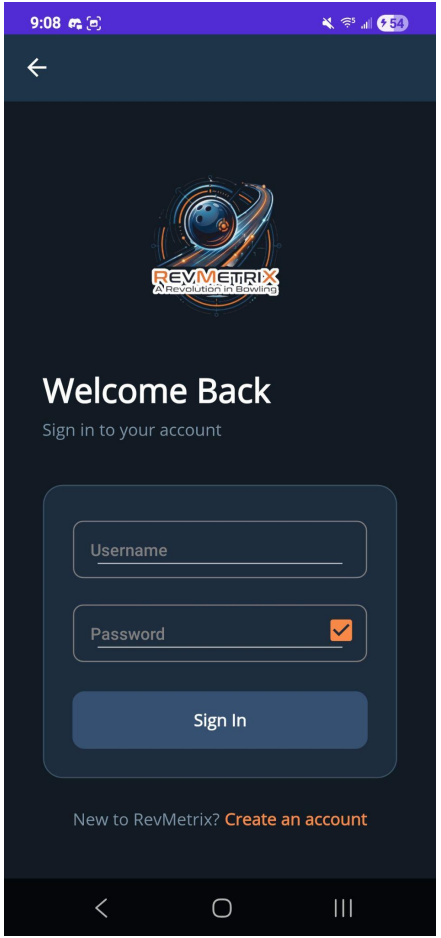
# Mobile Application

# Mobile Pages - Main, Register, Login



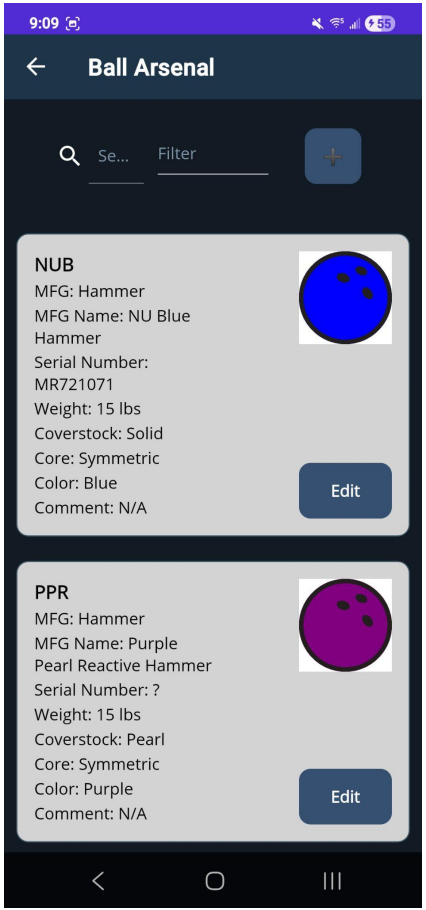



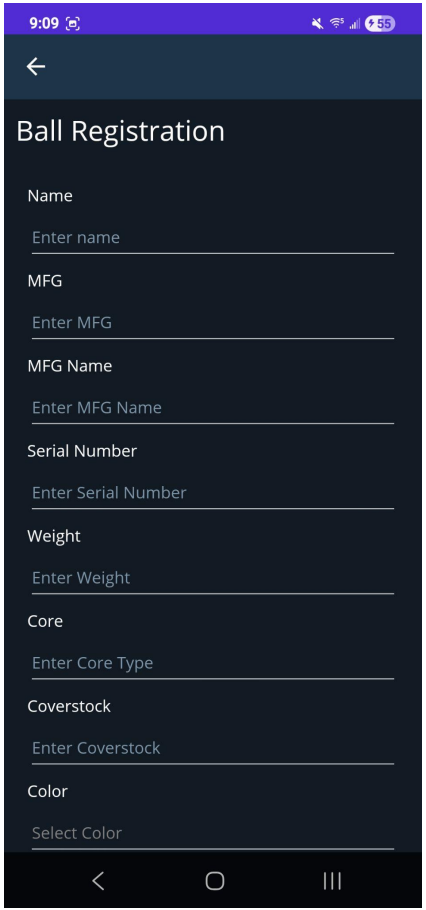



# Mobile Pages - Ball Arsenal

## Page Features

- Input bowling ball information
  - Name
  - Manufacturer
  - Manufacturer Ball Name
  - Serial Number
  - Weight
  - Core
  - Coverstock
  - Color
  - Comment for additional information
- Edit or disable bowling balls in the arsenal

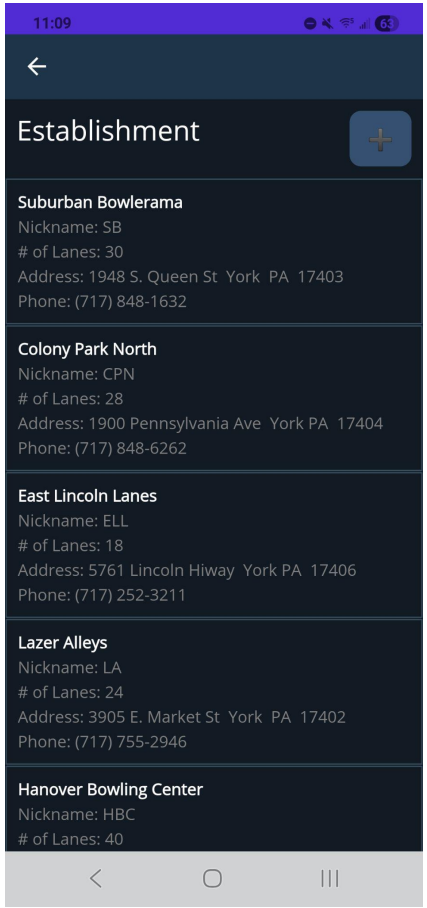


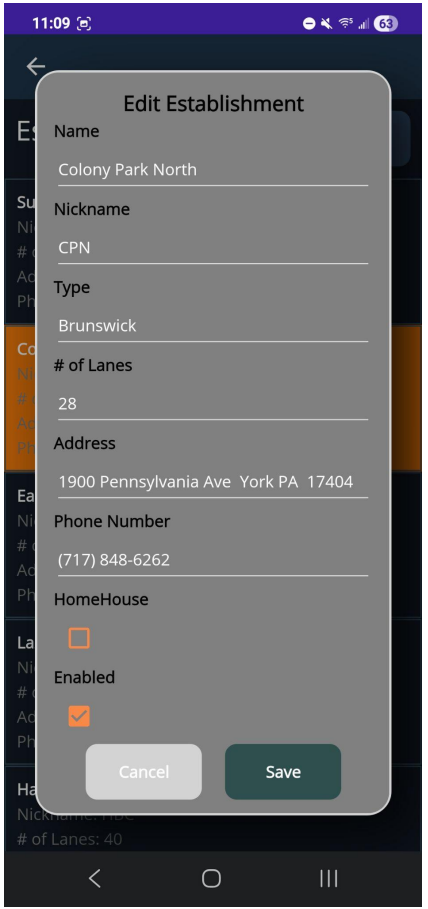



# Mobile Pages - Establishments

## Page Features

- Input establishment information
  - Name/Nickname
  - Type
  - Number of lanes
  - Address
  - Phone Number
  - Home house
- Edit or disable establishments

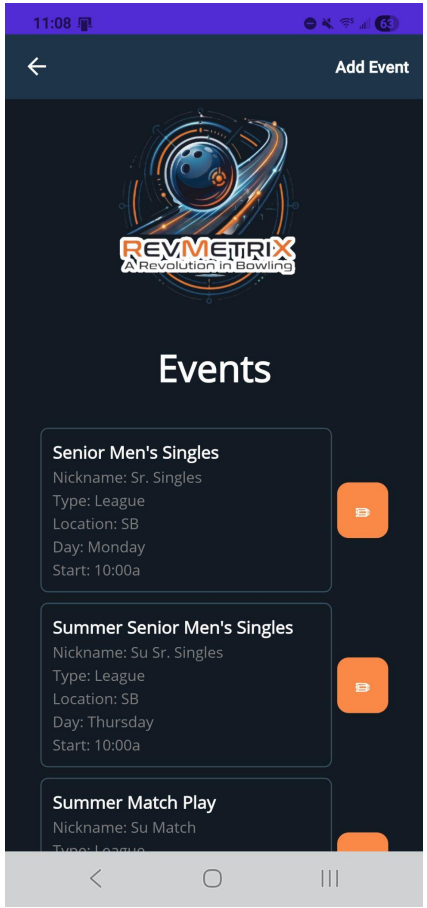


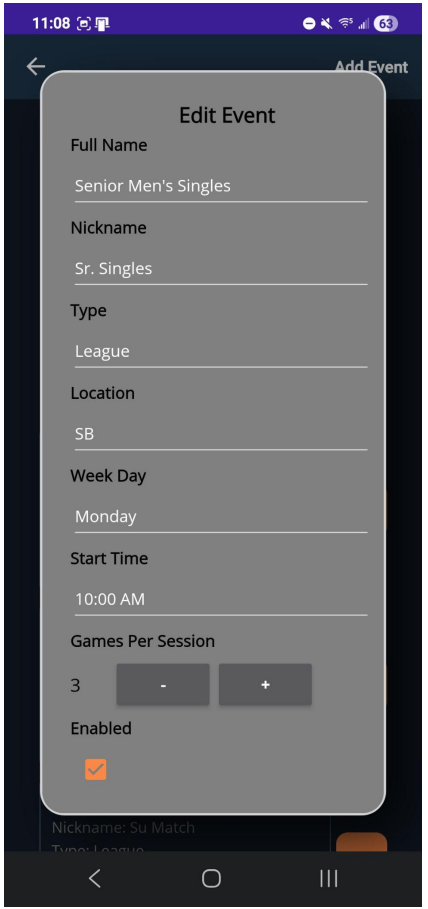



# Mobile Pages - Event List

## Page Features

- Add upcoming events
  - Name/Nickname
  - Type
  - Location
  - Weekday
  - Start time
  - Number of games per session
- Disable events
- Navigation to session lists

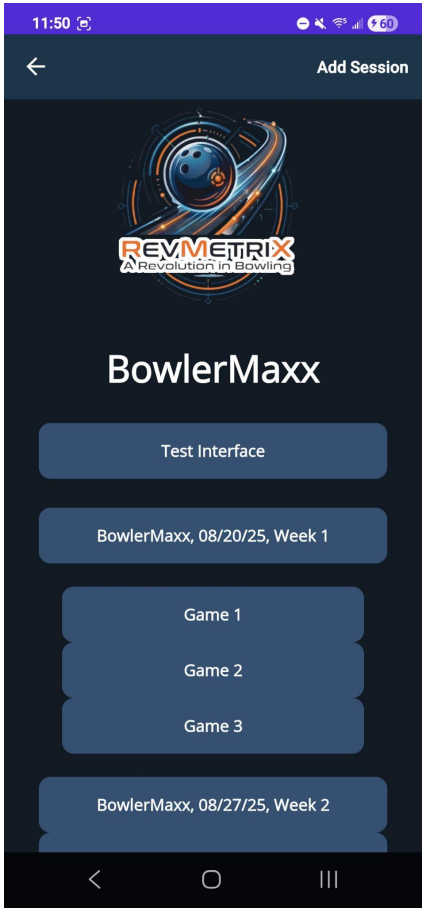


# Mobile Pages - Session List

## Page Features

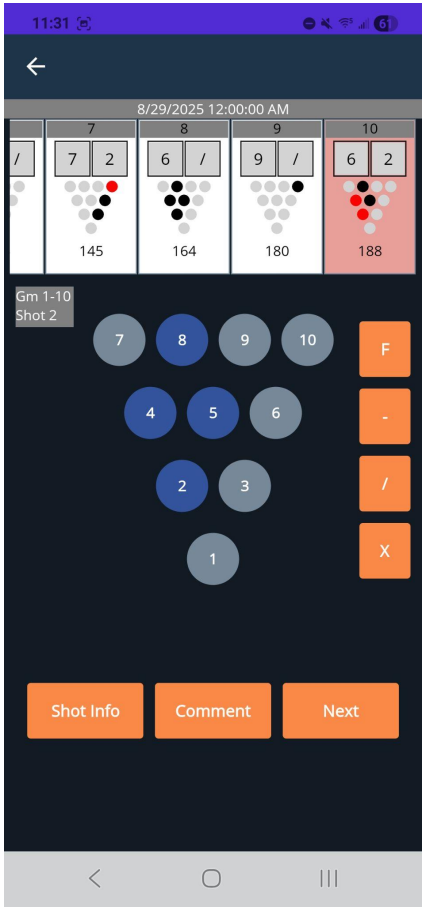
- Create/Add sessions by date and time
- Create/Add games under sessions
- Navigation to shot page for gameplay

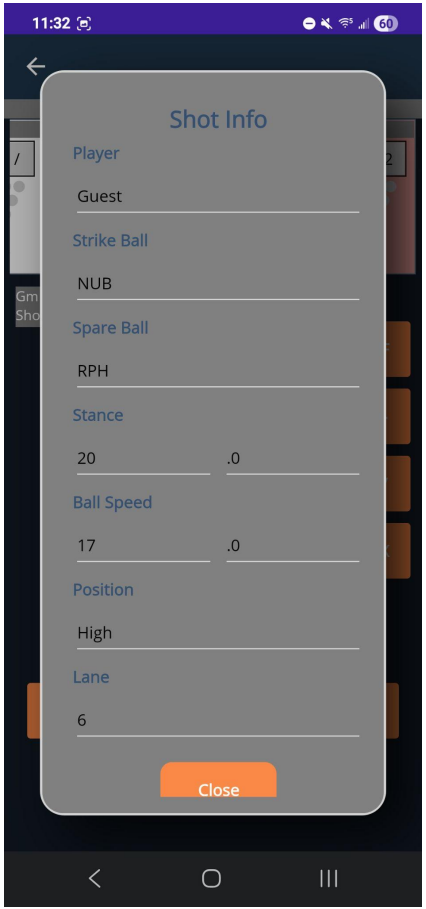


# Mobile Pages - Shot page

## Page Features

- Input shot data
  - Pins left standing per shot
  - Strike/spare balls
  - Stance
  - Ball Speed
  - Position
  - Lane number
  - Comments
- Frame view for previous shot data
- Click on a frame for shot edit mode

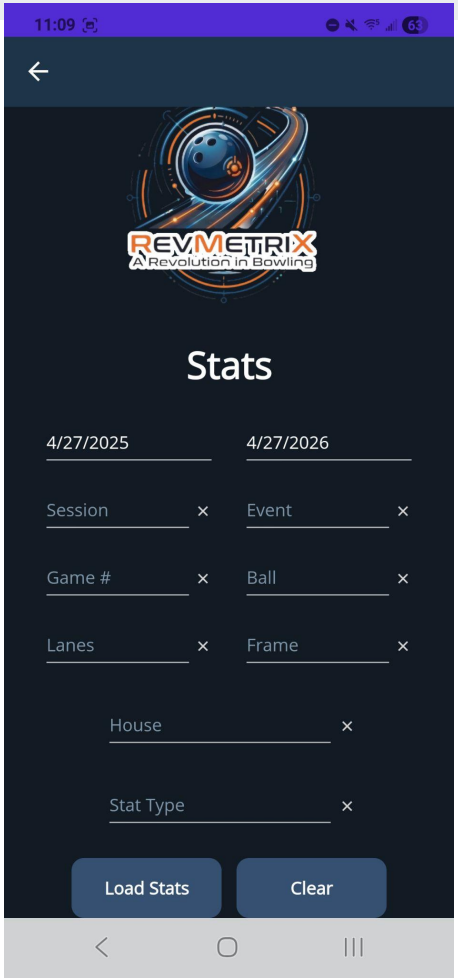


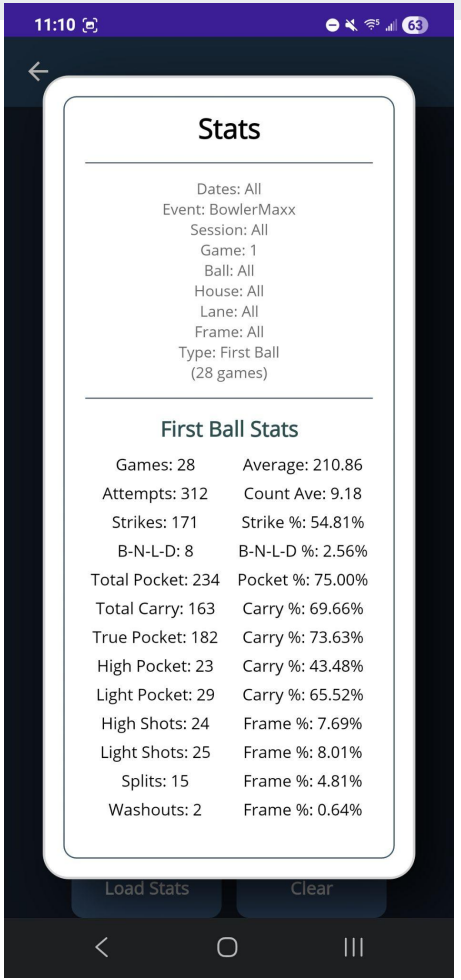



# Mobile - Stats Page

## Page Features

- Custom Query Engine
  - Date Range
  - Session
  - Event
  - Game
  - Ball
  - Lanes
  - Frame
  - House
  - Type
- Stats popup

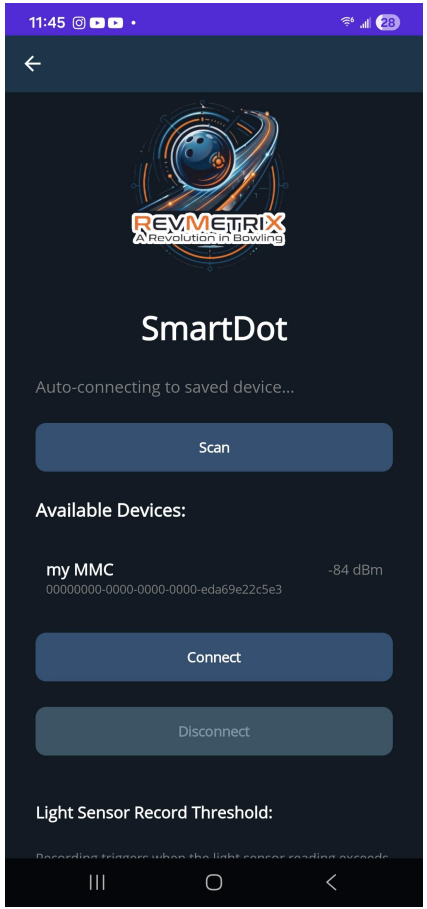


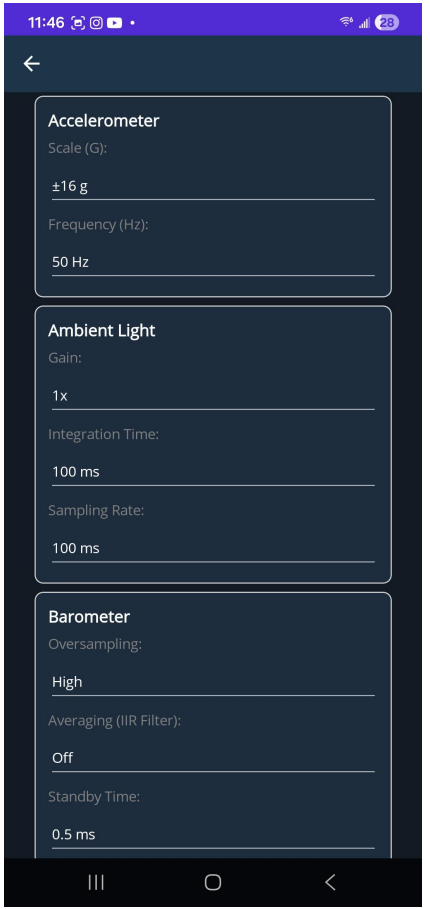



# Mobile Pages -SmartDot Page

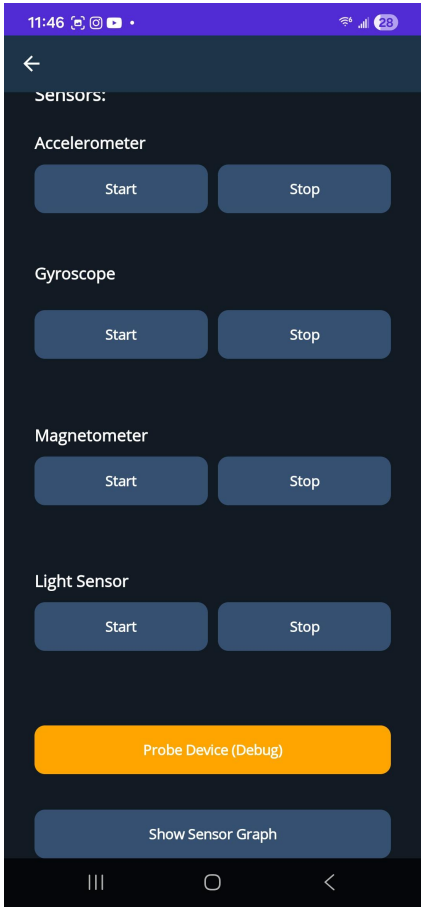
## Page Features

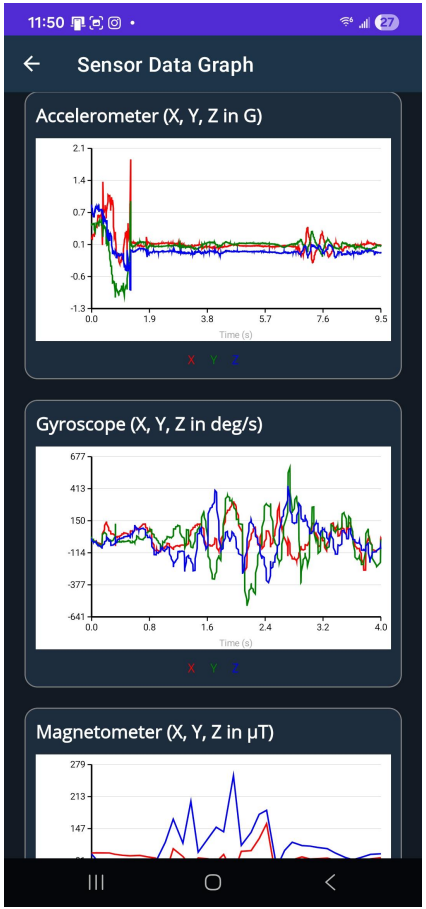
- Scan/Connect to MMC/MMS
- Read Gyro, Mag, Accelerometer, and Light data with live Graph
- Edit MMC/MMS Settings



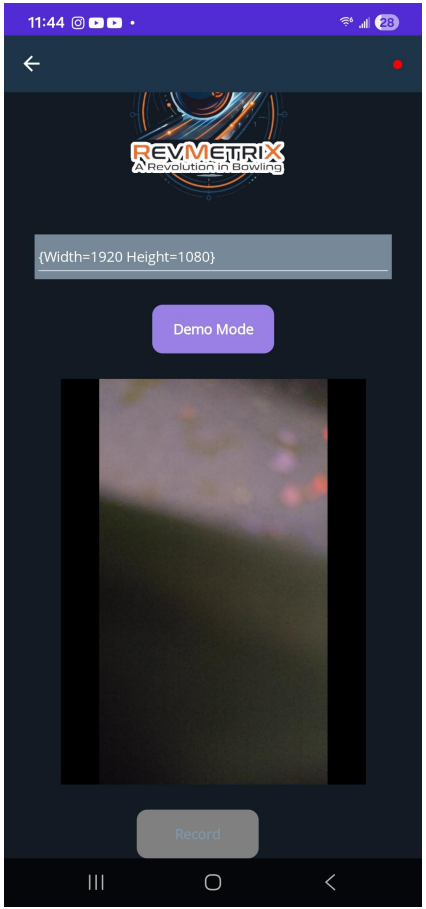



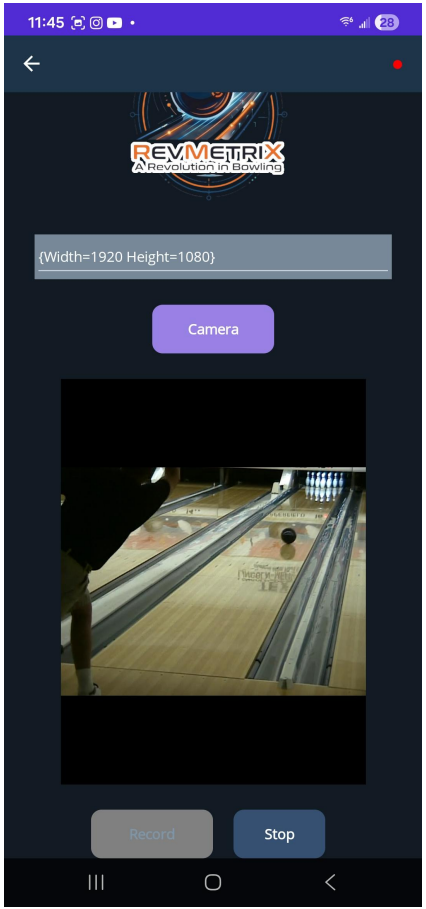
# Mobile Pages -SmartDot Page

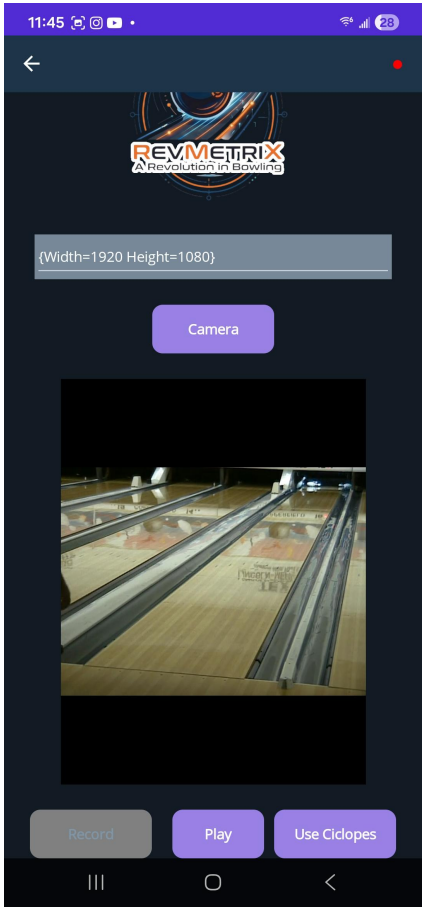


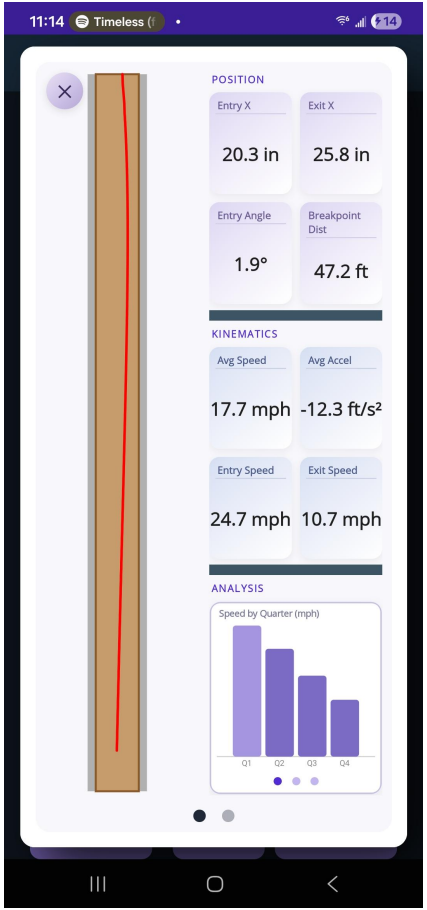



# Mobile Pages - Ciclopes Page



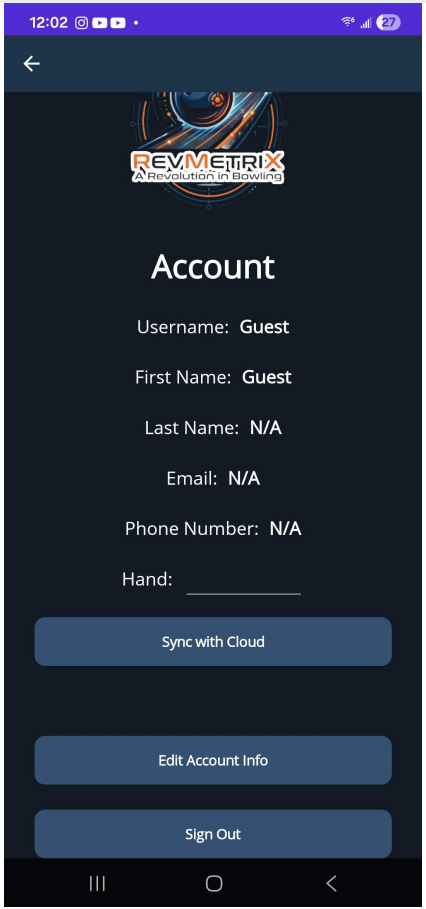



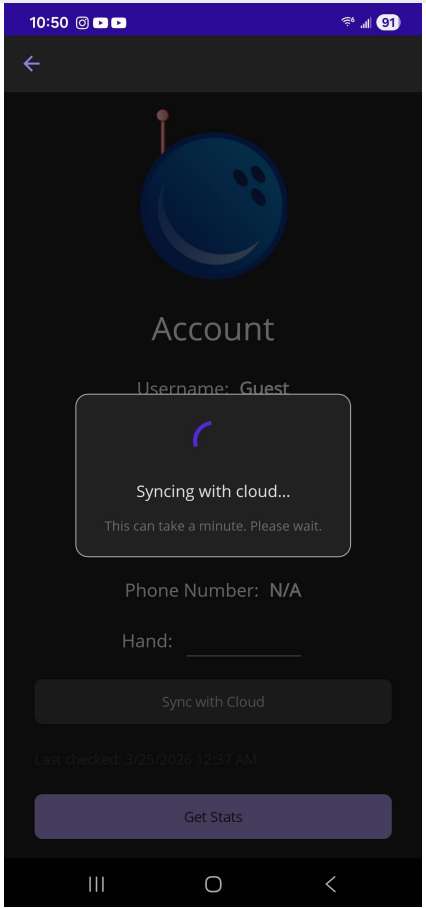



# Mobile Pages - Account

## Page Features

- View and manage user data
- Sync local and cloud data
- Sign out

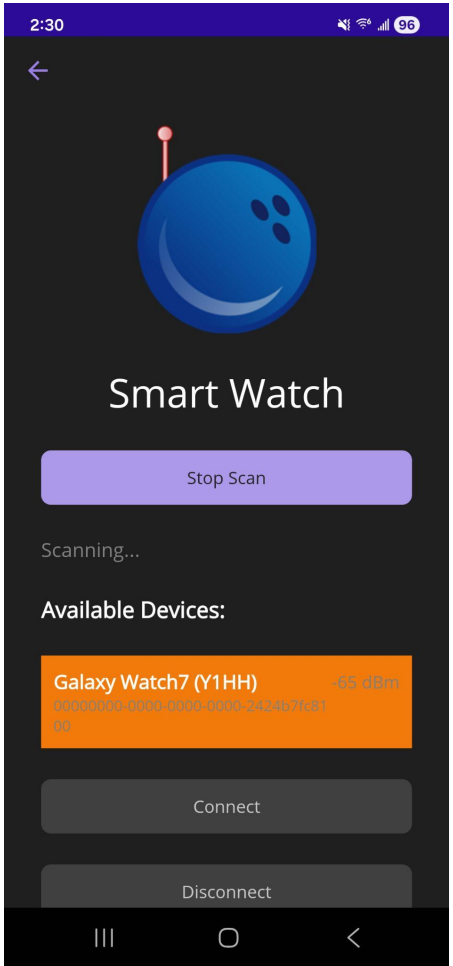


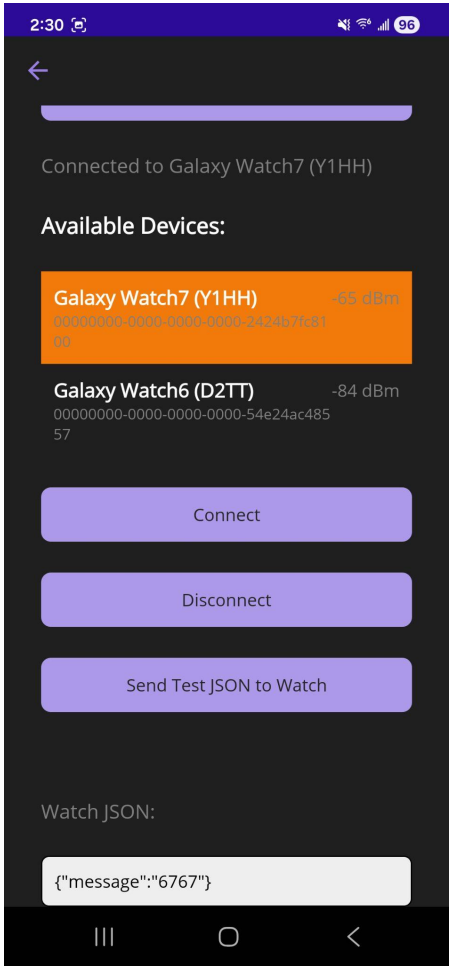



# Mobile Pages - Smartwatch Page

## Page Features

- Scan for available Galaxy Watches
- Connect/Disconnect
- Send/Receive test JSONs

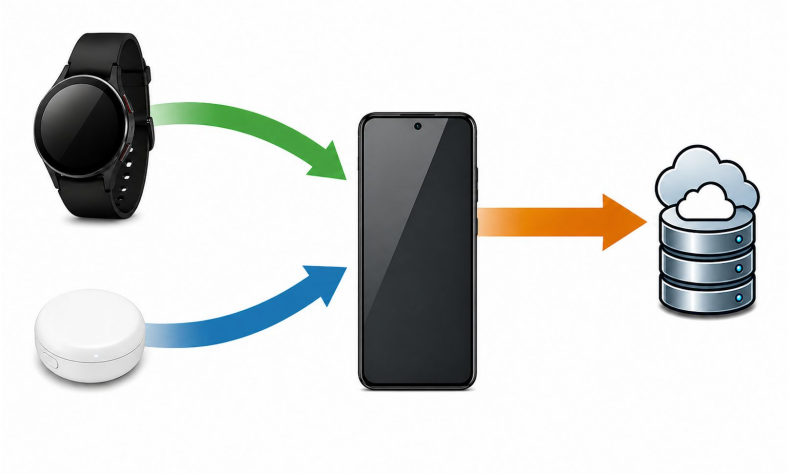


# Watch Application

# Watch - Overview

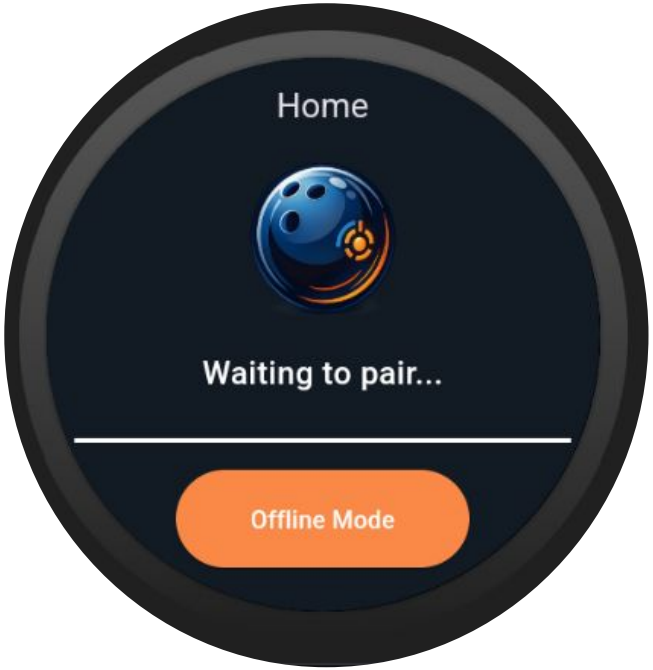
- Acts as a companion to the mobile app
- Collects user input for each frame and shot
- Sends data to the phone using Bluetooth Low Energy (BLE)
- Receives setup data when connected, including:
  - Game details
  - Available bowling balls
- Guides users through the shot workflow for each frame
- Sends shot data to the phone, including:
  - Pins left standing
  - Stance
  - Target
  - Breakpoint
  - Ball speed



# Watch - Home/Bluetooth Page

## Page Features

- First page when loading into application
- Listens to the isConnected reactive variable from BLEManager
- Offline mode
    - Used to play through the application if there is no phone connection
    - Starts an anonymous session
- On connection to the mobile application, auto-navigates to the sessions page



# Watch - Sessions page

## Page Features

- Entry point for starting new sessions from watch
- Displays username of account and 2 session selection options
- Anonymous session button
    - Creates a brand new session with unique session ID
- Event session button
    - Shows event name associated with the session that was sent
    - Closest session to the current time/date

# Watch- Game page

## Page Features

- Displays game count and number (based on session data received from phone)
- Access to move between games through game carousel
- Settings cog for access to system settings
- Add and remove buttons
    - Adds a new game to the active session
    - Removes the current game with confirmation

# Watch - Settings page

## Page Features

- **Sync:** updates watch with latest data from mobile app
- **End:** sends watch back to session page with new session loaded for selection
- **Log Out:** Disconnects phone and watch completely, returning watch to home page



On connection : account packet

Session page populates

Start games

Frame page

Settings page

Re-initializes UI with updated session data

Sync selected

Logout selected

End selected

Update session page with newest session

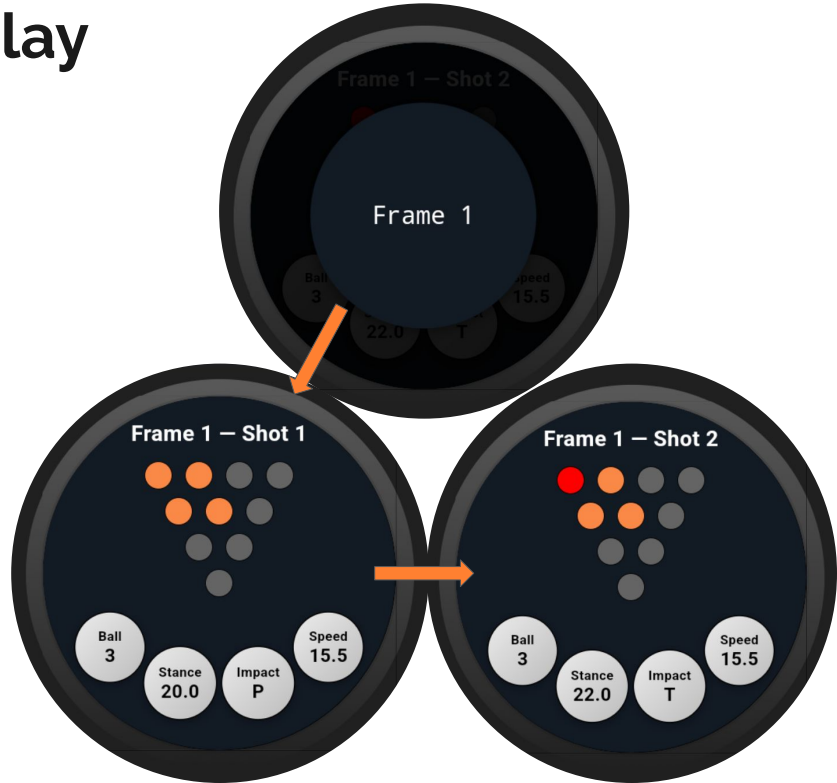
Disconnect app and watch

Return home

# Watch - Frame Selection/Display

## Page Features

- Preview frame and shot number
- Displays pins left standing for current shot
- Info bar at bottom showing 4 shot metrics
    - Ball used
    - Stance for shot (in boards)
    - Impact of shot
    - Ball speed
- Access to previous frames through frame carousel

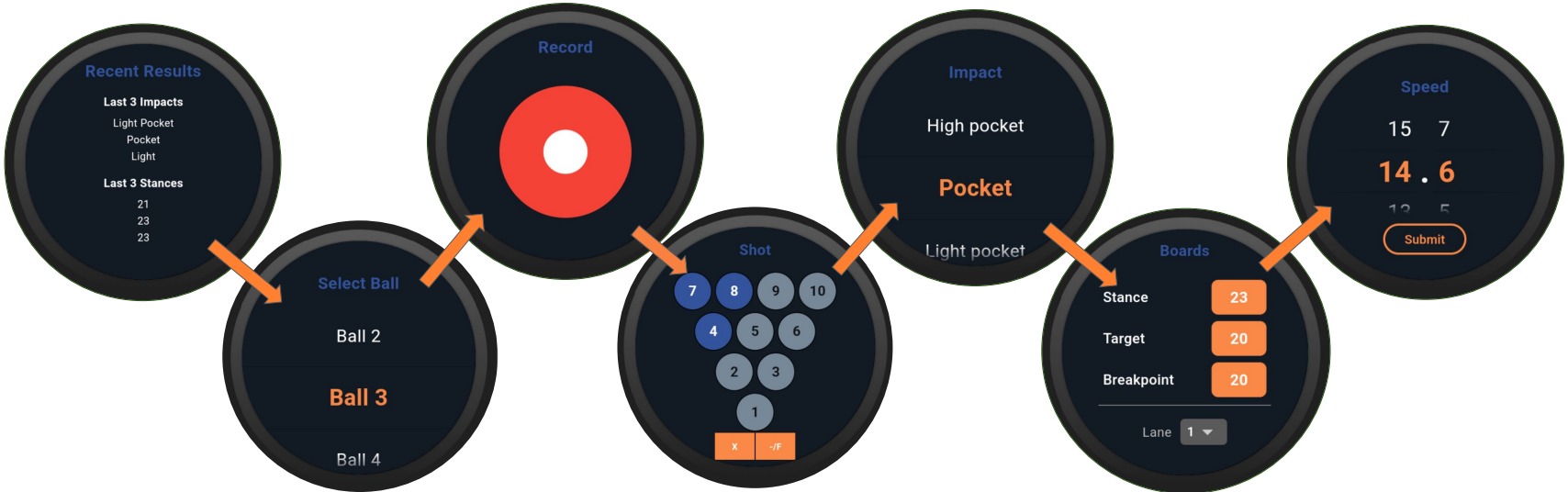


# Watch - Shot page & Workflow

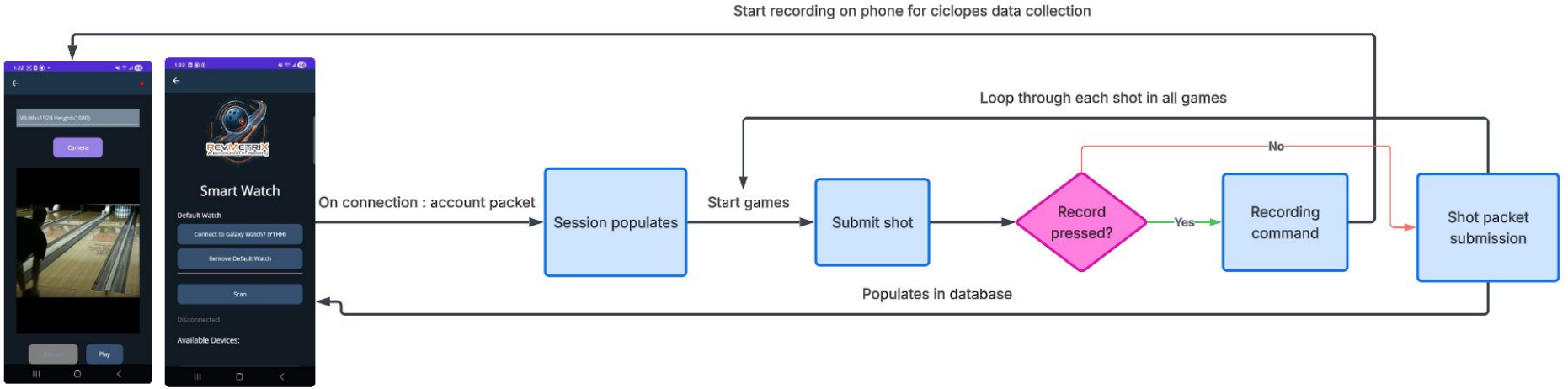
## Page Features

- Scaled down version of phone UI
- Submit sends a shot packet to phone

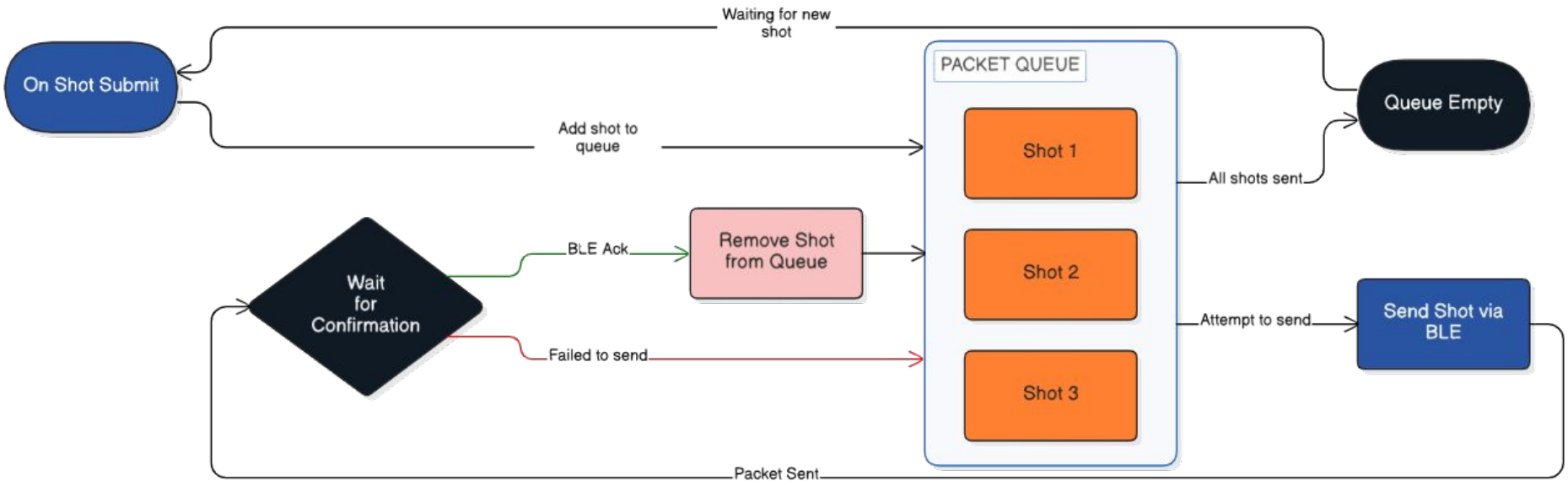
- 7 stage input including:
  - Recent results
  - Ball selection
  - Recording,
  - Pins left standing
  - Impact selection
  - Stance/target/breakpoint selection
  - Speed selection



# Watch - Cellular side connection



# Watch - Queue

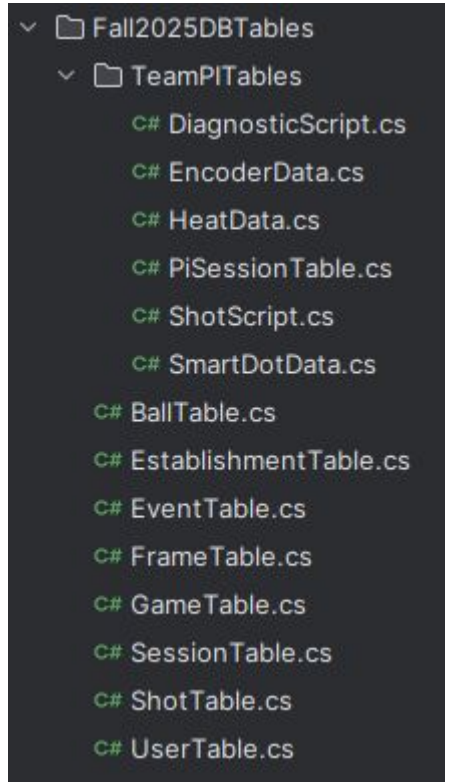

Waiting for new shot
On Shot Submit
Add shot to queue
PACKET QUEUE
Shot 1
Shot 2
Shot 3
Queue Empty
All shots sent
BLE Ack
Remove Shot from Queue
Wait for Confirmation
Failed to send
Attempt to send
Send Shot via BLE
Packet Sent

# Cloud Database

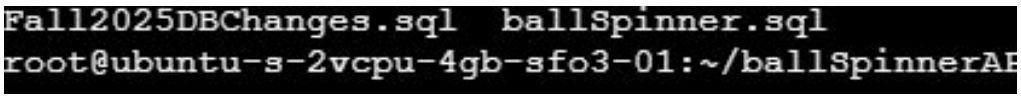
# Cloud - Migration Tools

- Safe SQL code generation
- Tables are organized into C# classes
- Lower likelihood of corrupting DB with SQL changes

C# files containing table information



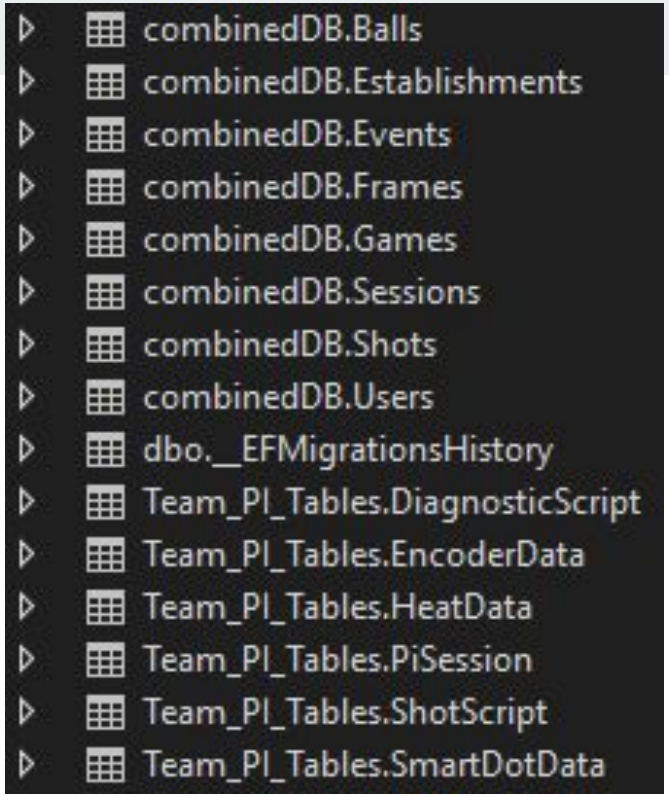

Generated script file in Droplet



# Cloud - Database

- Added 15 new tables under 2 new schemas
- Migration History to track DB updates
-

Screenshot from Visual Studio Server Explorer




| | MigrationId |
|---|---|
| 1 | 20251021160455_AddTables |
| 2 | 20251029015335_UsersEstabGamesSeshShots |
| 3 | 20251104170247_AddPiTables |
| 4 | 20251116200753_AddedPiTables |
| 5 | 20251117231619_AddDataSelectorToSmartDotTable |
| 6 | 20251204020317_AddedReplayIterationColumnToEncoderSmartDotHeat |
| 7 | 20251204023329_DataSelector |
| 8 | 20251204023631_ReplayIteration |
| 9 | 20251204024742_AddedReplayIteration |

Screenshot of migration history from this semester (stored in DB)

# Cloud - API

- Added 30 new API endpoints
- Create/Edit endpoints for both the Cellular and Pi Teams

| | |
|---|---|
| GET | /api/gets/GetAppUsers |
| GET | /api/gets/GetAppShots |
| GET | /api/gets/GetAppSessions |
| GET | /api/gets/GetAppGames |
| GET | /api/gets/GetAppEstablishments |

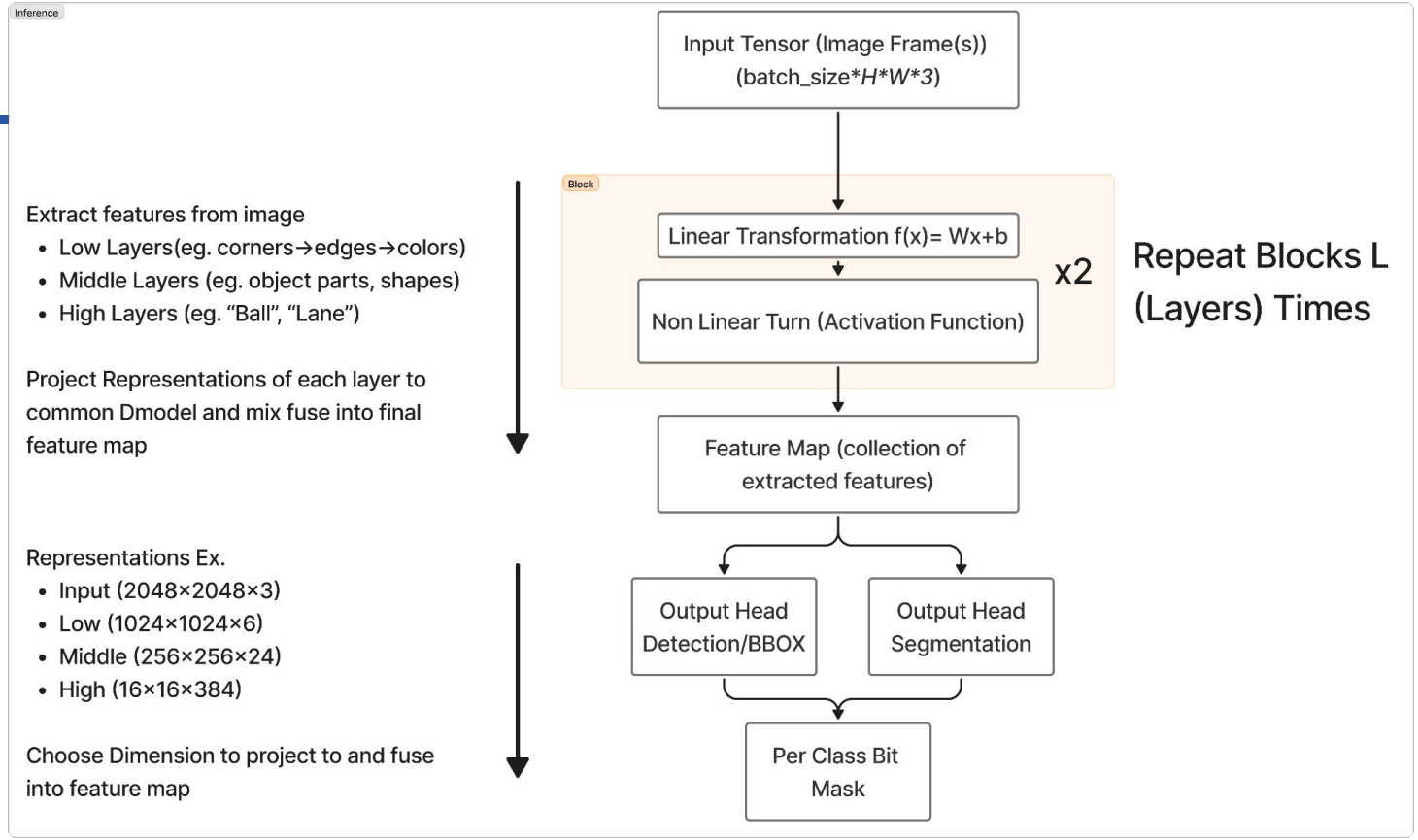
| | |
|---|---|
| GET | /api/gets/GetBallsByUsername |
| GET | /api/gets/GetEventsByUsername |
| GET | /api/gets/GetFramesByGameId |
| GET | /api/gets/GetShotsByUsername |
| POST | /api/posts/PostBalls |
| POST | /api/posts/PostEvent |
| POST | /api/posts/PostFrames |

| | |
|---|---|
| GET | /api/gets/GetAllPiDiagnosticScriptBySession |
| GET | /api/gets/GetAllPiEncoderDataBySession |
| GET | /api/gets/GetAllPiHeatDataBySession |
| POST | /api/gets/GetAllPiSessions |
| GET | /api/gets/GetAllPiShotsBySession |
| GET | /api/gets/GetAllPiSmartDotDataBySession |

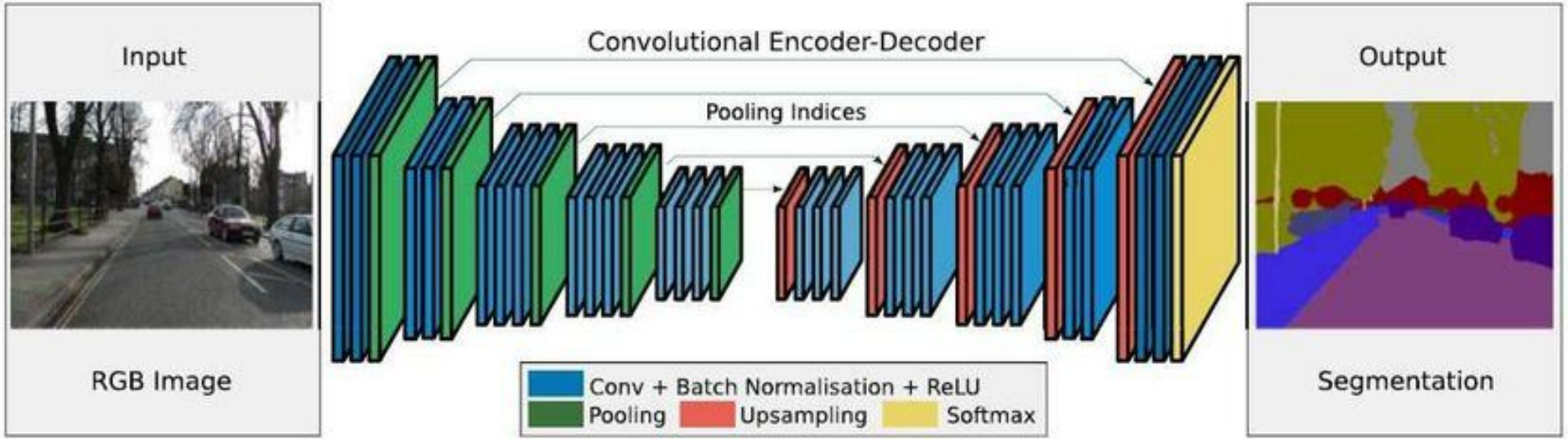
| | |
|---|---|
| POST | /api/posts/PostPiDiagnosticScripts |
| POST | /api/posts/PostPiEncoderData |
| POST | /api/posts/PostPiHeatData |
| POST | /api/posts/PostPiSessions |
| POST | /api/posts/PostPiShot |
| POST | /api/posts/PostPiSmartDotData |

# Ciclopes

# Ciclopes - Deep Learning Overview

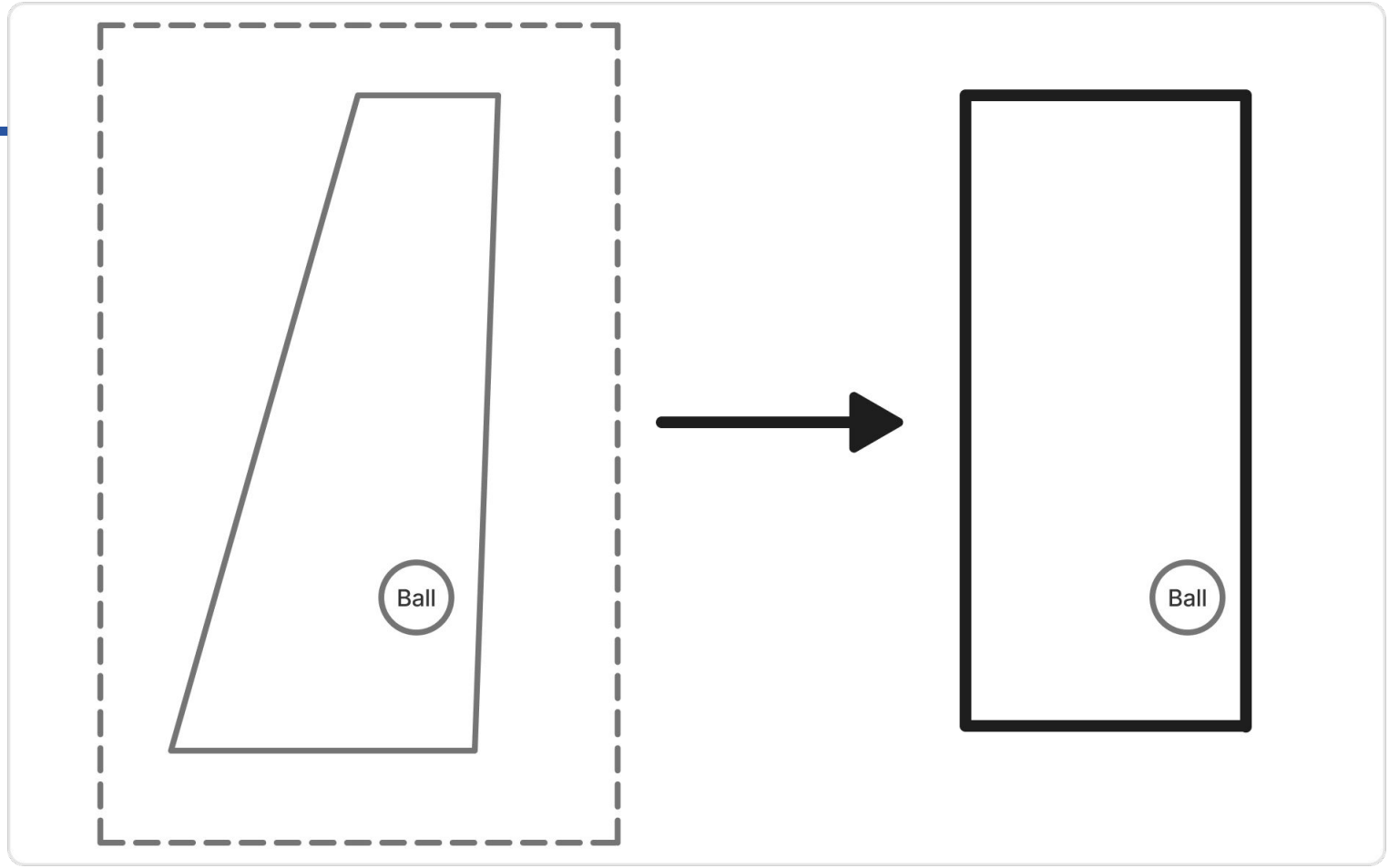


# Ciclopes - Segmentation Architecture Visualization

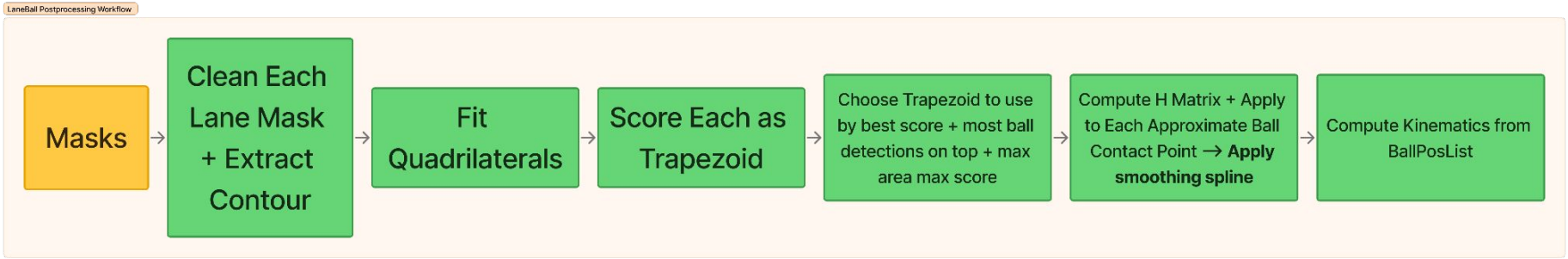


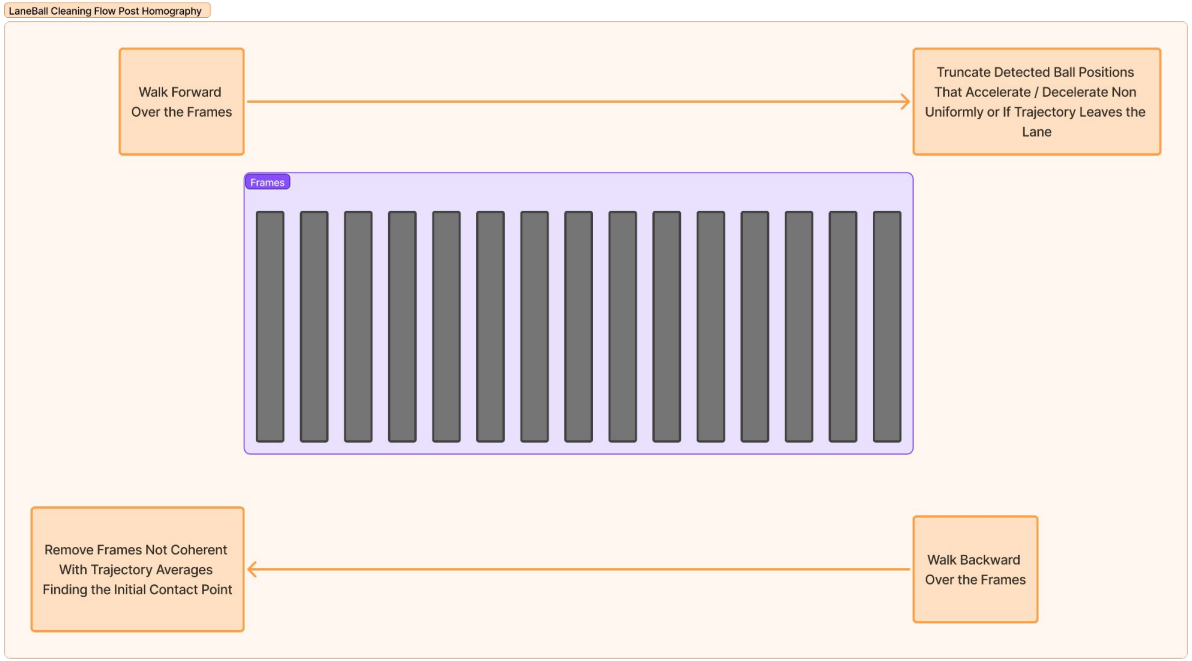
# Ciclopes - Segmentation and Lane Fit Performance

# Ciclopes - Homography Example

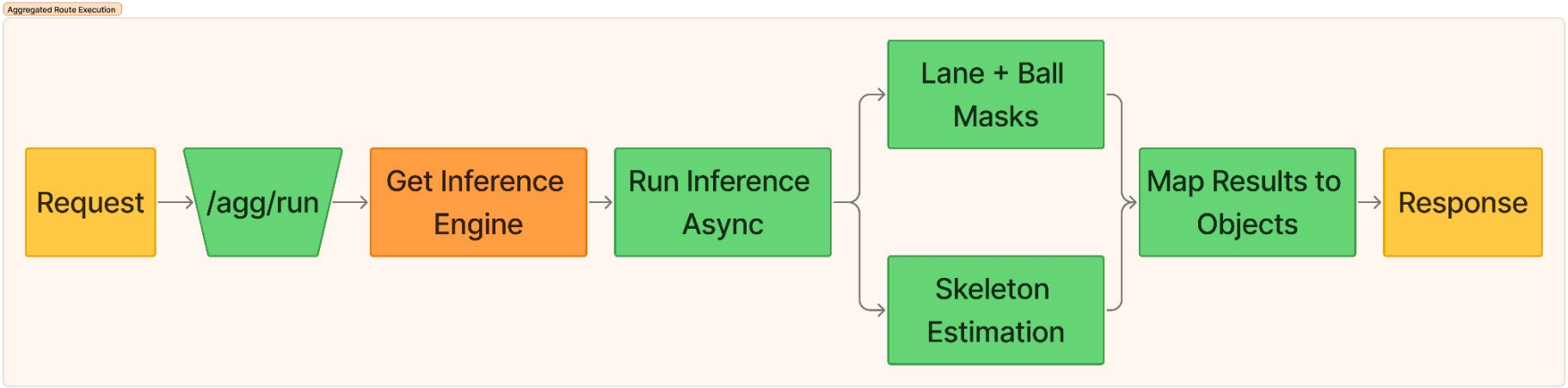


# Ciclopes - Algorithm Control Flow

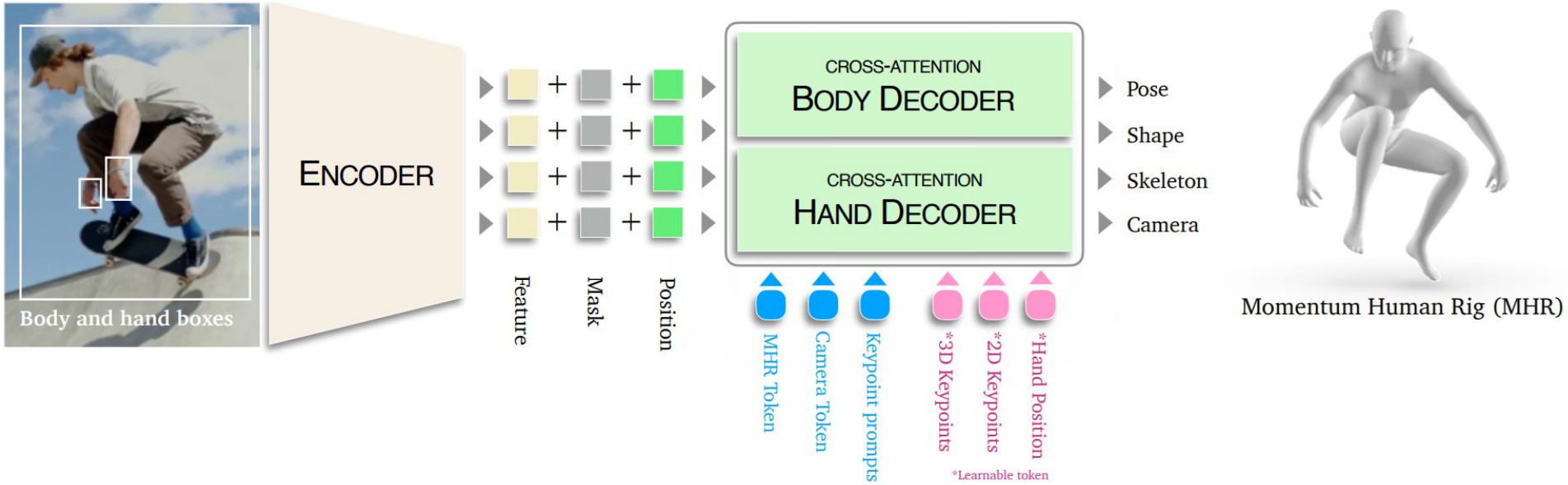


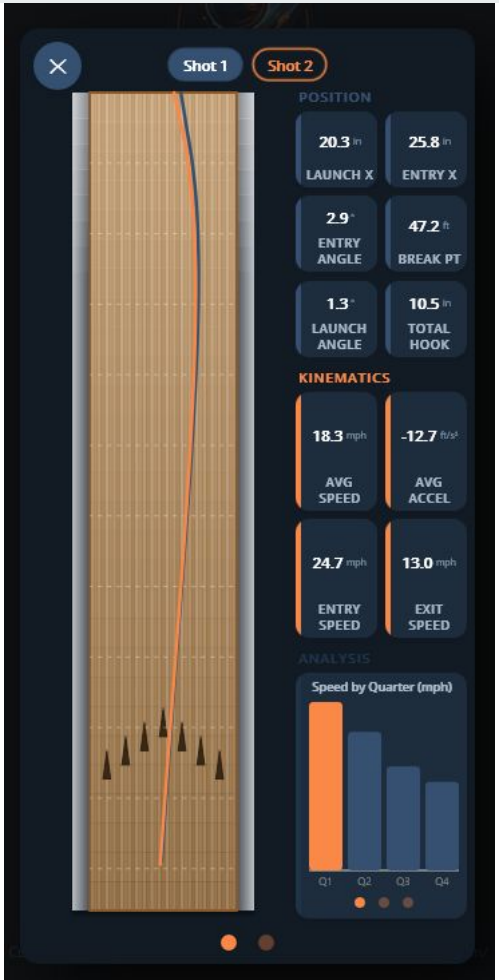
# Ciclopes - Request Process Flow



# Ciclopes - Pose Estimation Architecture Visualization

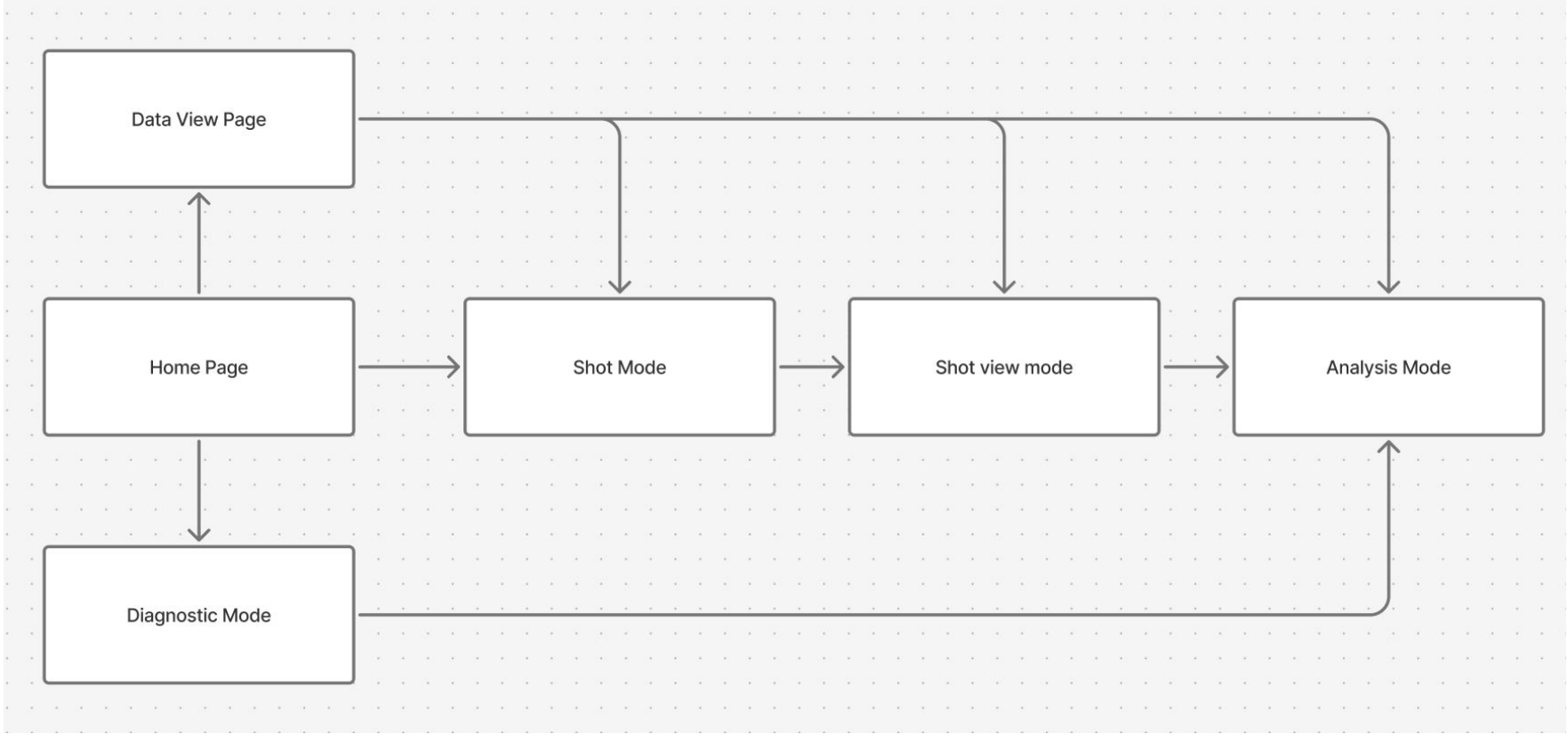


# Ciclopes - Data Visualization




# Ball Spinner Controller

# Ball Spinner Controller Operation flow



# Ball Spinner Controller - BSC Class

- Core singleton class for communicating between pages
- It holds all of the most important objects
  - CloudAPI
  - Session
  - Data Controller
  - Motors

# Ball Spinner Motors and Drives

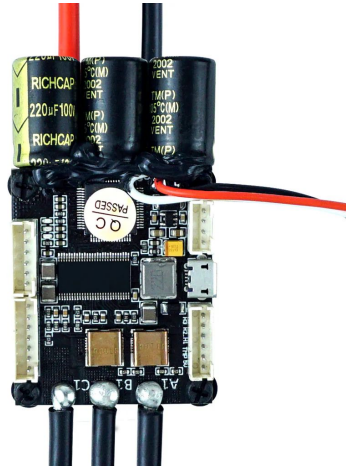
- Primary axis - Sensored BLDC motor +Flipsky VESC controller
  - RPM capabilities
  - Hall sensors provide consistent acceleration
  - ESC provides current limiting, temperature monitoring and fault detection
- Second and third axis - Nema 17 stepper motor + DM542 driver
  - Angle control
  - High holding torque for tilt and rotation
  - Reliable industrial grade step driver


*NEMA 17 stepper motor*

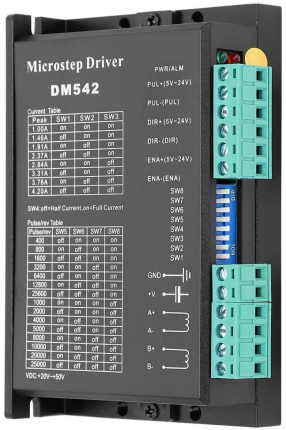
*E3665 2500KV inrunner motor*

# Ball Spinner Controller Continued...

- Primary axis motor configuration
  - Hall sensor detection and motor characterization in VESC tool
  - Pi sends commands over USB to the ESC
  - Controlled via PWM (Pulse Width Modulation)
- Second & third axis stepper motor configuration
  - Microstepping and current dip switches configured to 1.46A an
  - Pi generates controlled STEP/DIR pulses with an adjustable frequency
  - Created angle based control routines




*Flipsky VESC 4.20*




*DM542 Microstep Driver*

# Ball Spinner Power System and Safety

*P1-150-24 Power Supply*

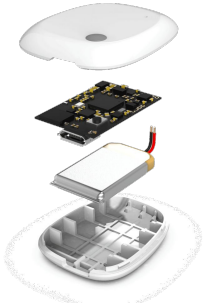
- 12V Power
  - HMI, Enclosure Fan Power
- 5 V  Power
  - Raspberry Pi power
- Estop Button
- Status indicators
- Modularity

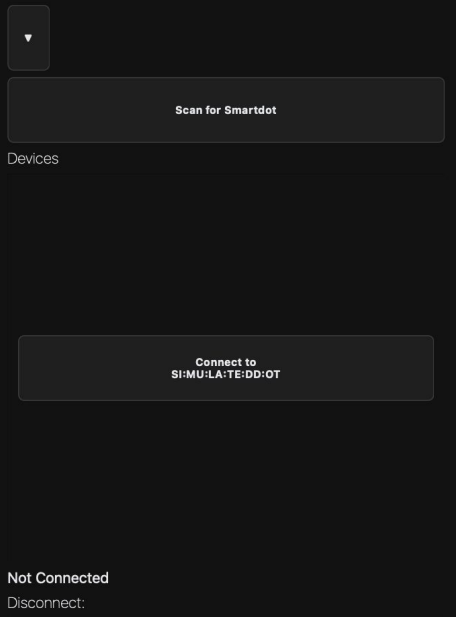
*Limit Switch*

*Estop Button*

# Ball Spinner Controller - SmartDot

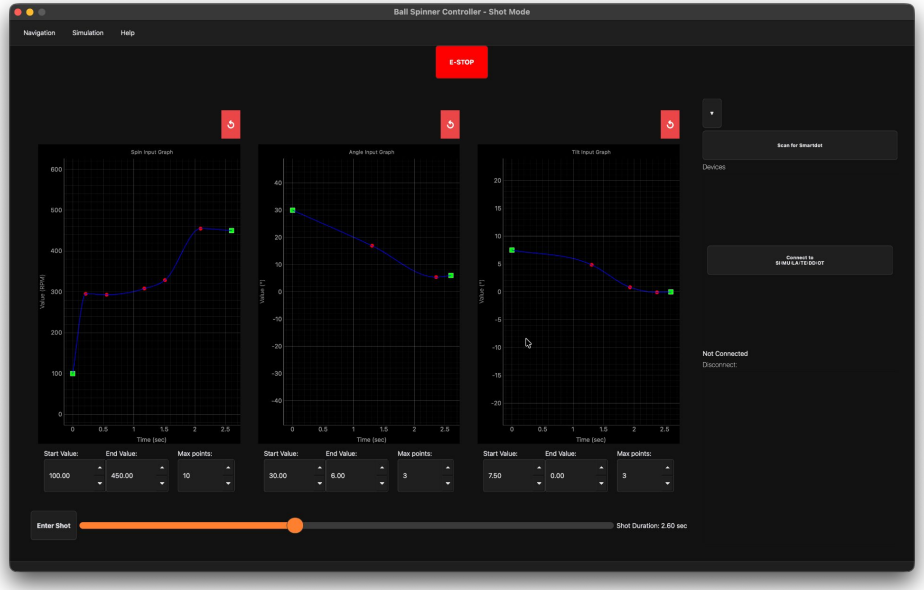
- iSmartDot
  - MetaMotionS
  - Simulated Smart Dot
- MetaWear Implementation
- Scan, Connect, and Disconnect
- Smart Dot Connection Manager





# Ball Spinner Controller - Shot Mode

- Primary method of simulating a shot
- 3 input graphs allow user to make shot instructions
- Slider lets user select length of shot

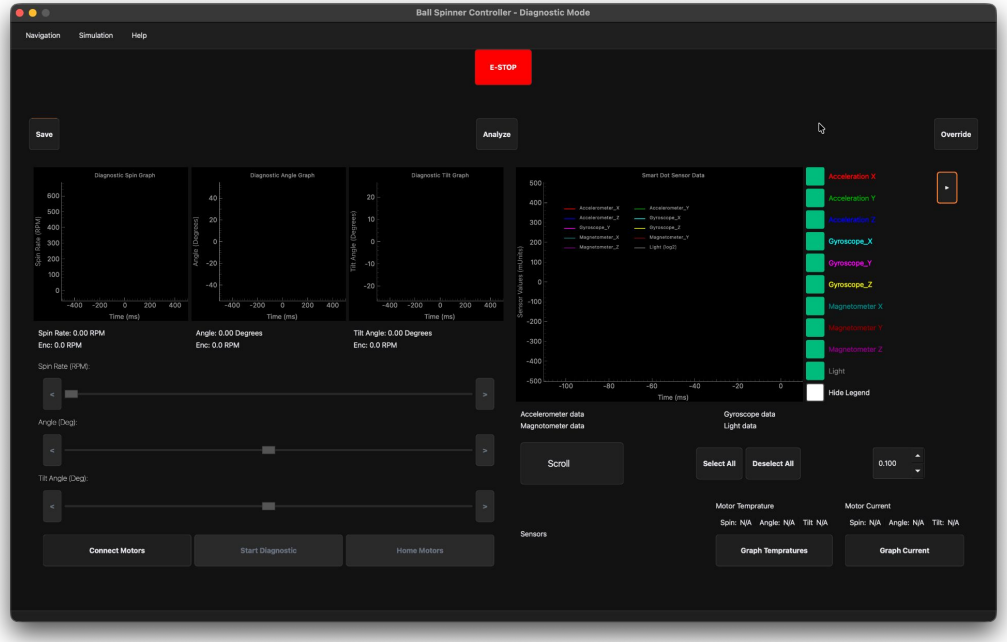


# Ball Spinner Controller - Analysis Mode

- Smart dot and motor data are loaded in from the Data Controller
- Data is passed into the SmartDot graph and Motor Graph
- Can perform various Analysis Operations.

# Ball Spinner Controller - Diagnostic Mode

- Individual motor control
- Real-time graphical feedback
- SmartDot connectivity for quick data collection
- Motor current and temperature data.



# Ball Spinner Controller - Models

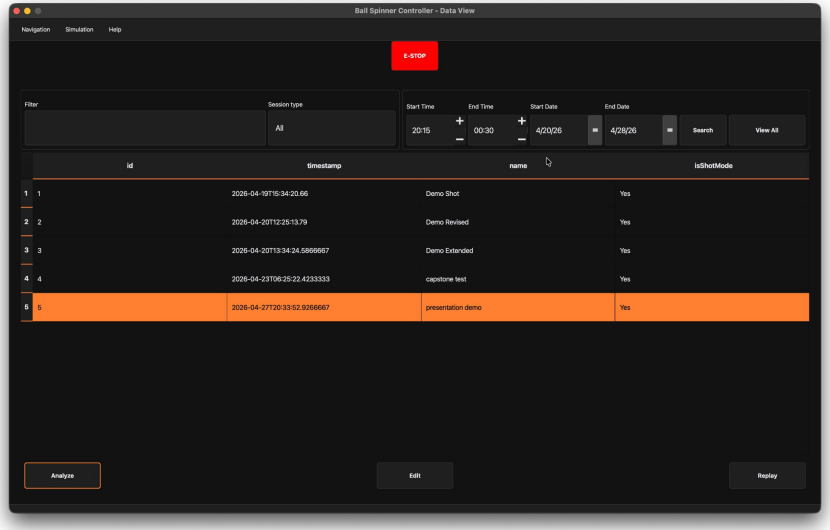
- Data Controller
  - Session
    - Diagnostic Script Data
    - Shot Mode Data
    - Smart Dot Data
    - Encoder Data

# Ball Spinner Controller - CloudAPI

- APIUtils
- HTTPS
- Object Serialization

# Ball Spinner Controller - Data View Page

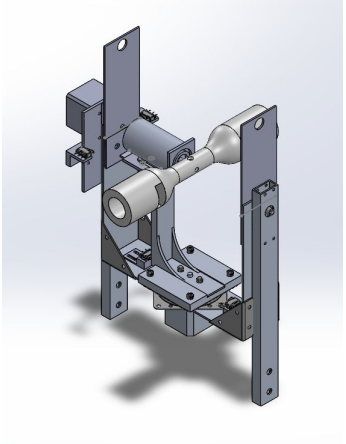
- Data is loaded from cloud into table
- User can filter data by date
  - User can further refine data by searching across fields
  - User can also filter for session type
- User can replay, analyse, or edit a previously created shot

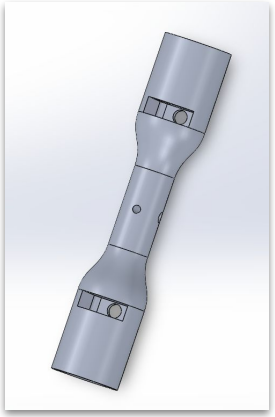


# Ball Spinner Mechanical System

# Ball Spinner Mechanical System

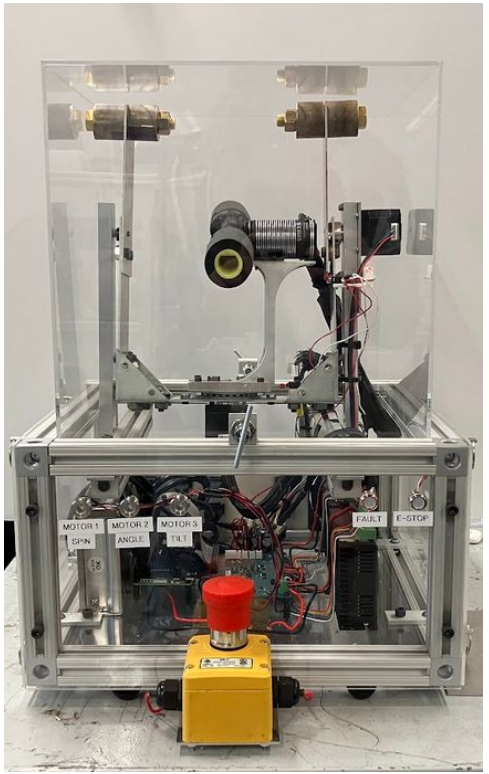
- **Model developed based on feedback**
  - Aluminum selected as material
  - Smaller and lighter to reduce required torque and power consumption
  - Components reinforced with gussets
  - Safety features implemented
- **SmartDot holder**
  - Design changed as the system requirements changed
  - 3D printed in polycarbonate to improve system lifespan





# Ball Spinner Mechanical System

- **BSC Enclosure and Platform**
  - Houses Motor drivers, PCB, Raspberry Pi, and Power Supply
  - Made of T-Slotted Aluminum Channel
    - Machined for pin and slot mounted acrylic side panels
  - Holds Acrylic safety enclosure fastened by T-handle spring plungers
  - Polycarbonate sheet to support electrical components

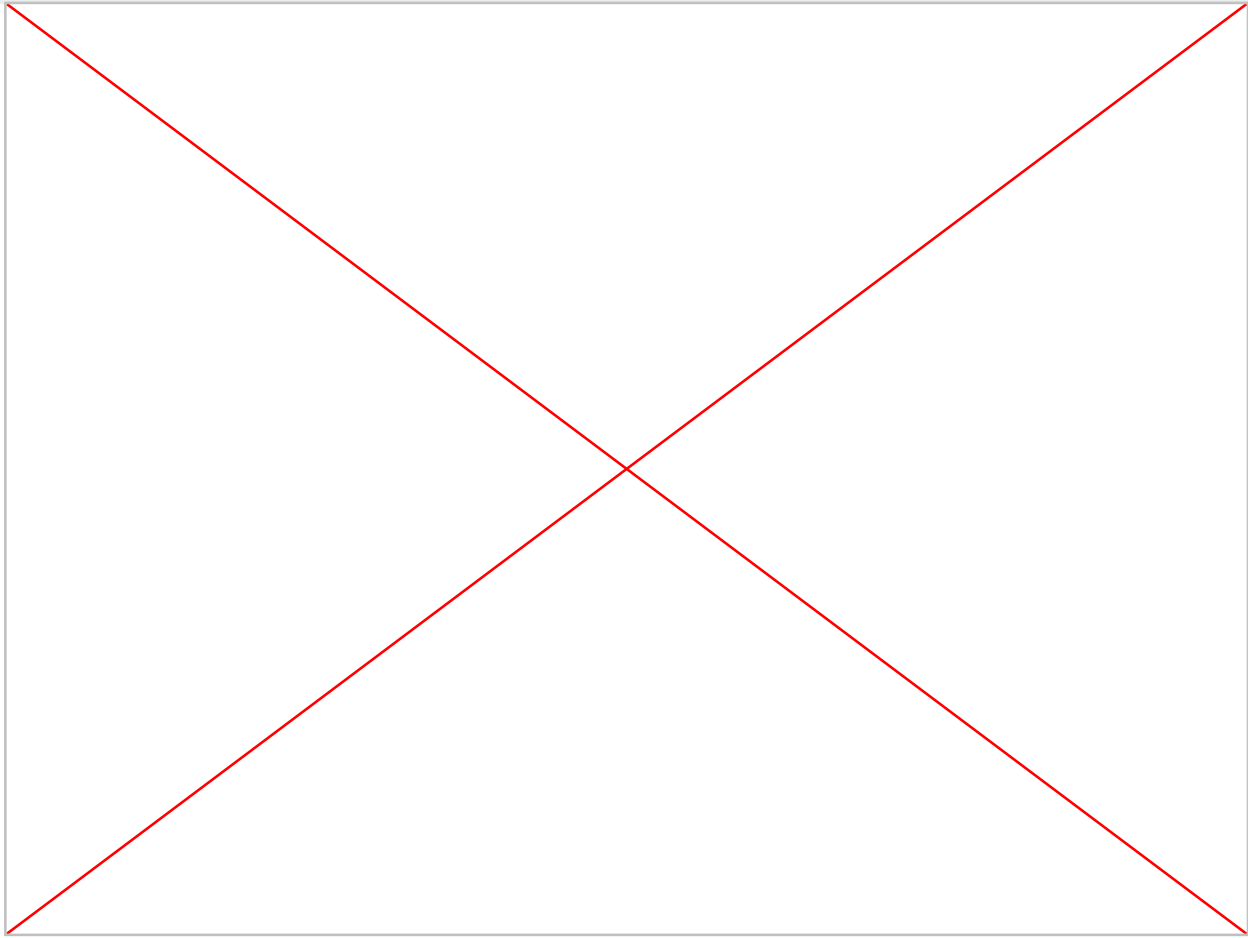


# Project Demonstrations

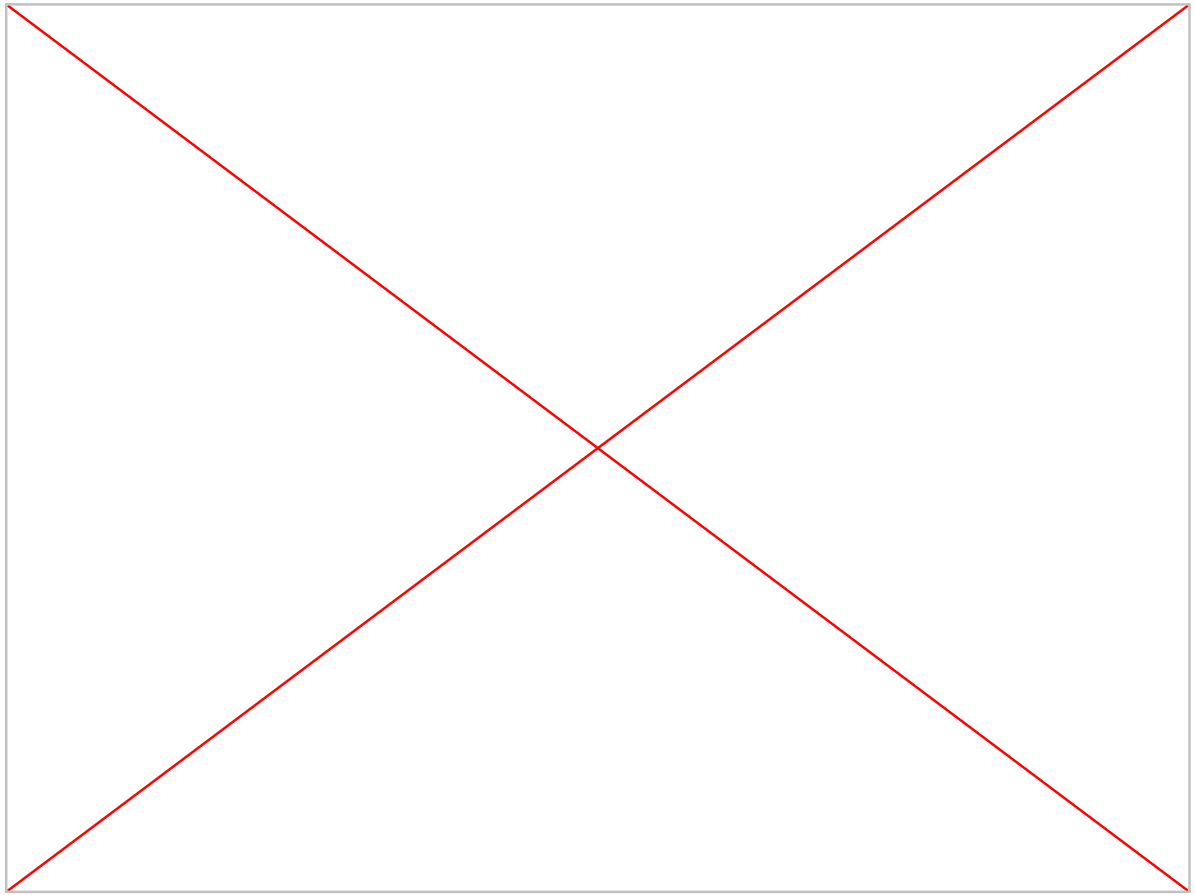
# Mobile App Demo

# Watch

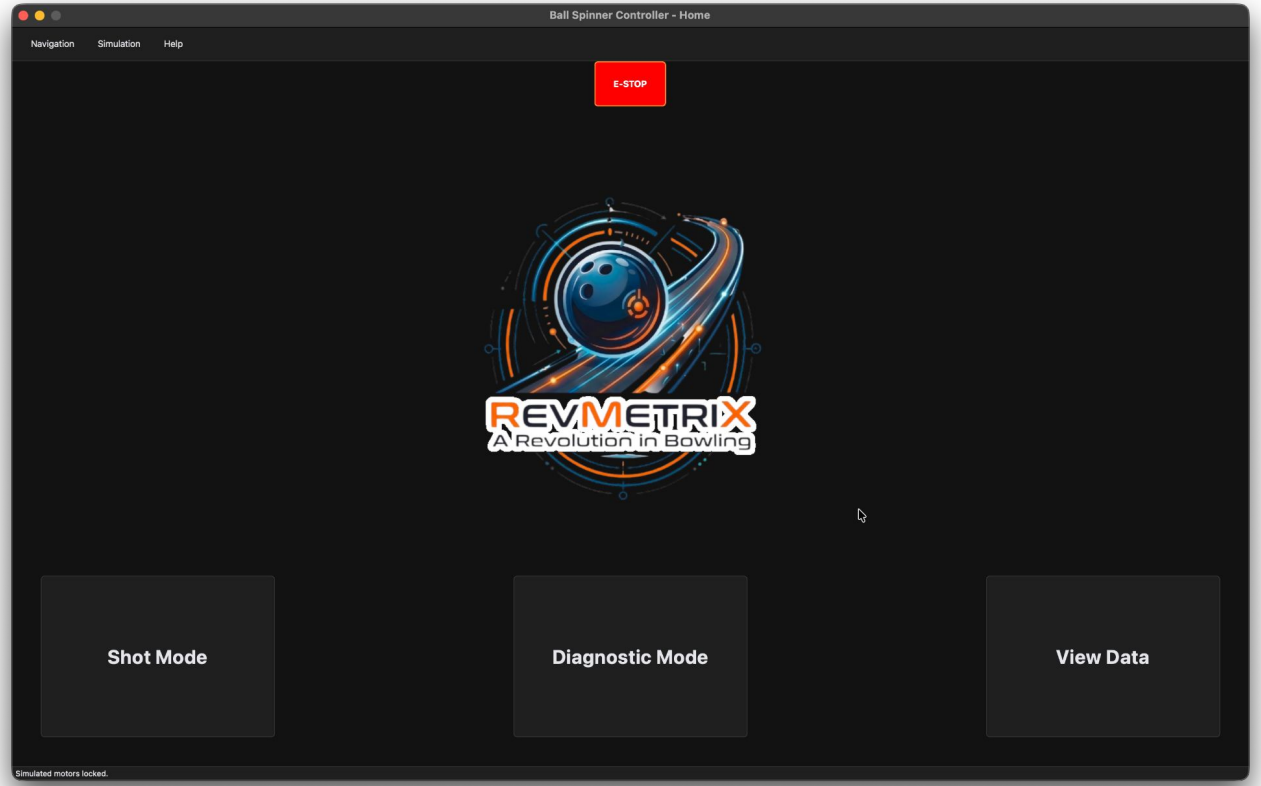
# Cloud - Pi and Mobile API



# Cloud - Mobile App



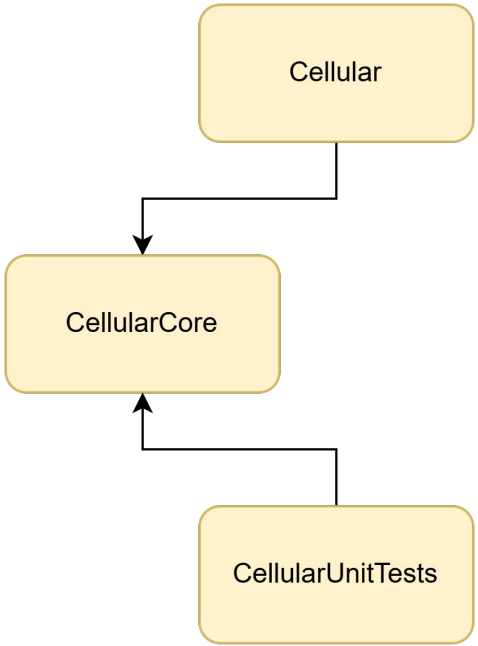
# Ball Spinner Demo

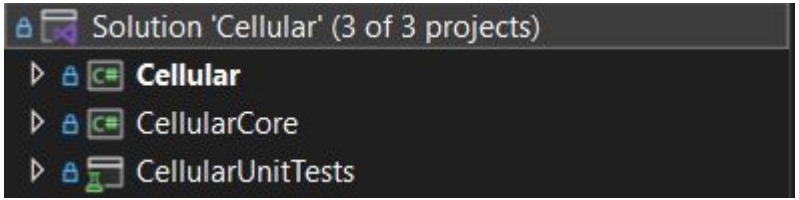


# Automated Testing Framework

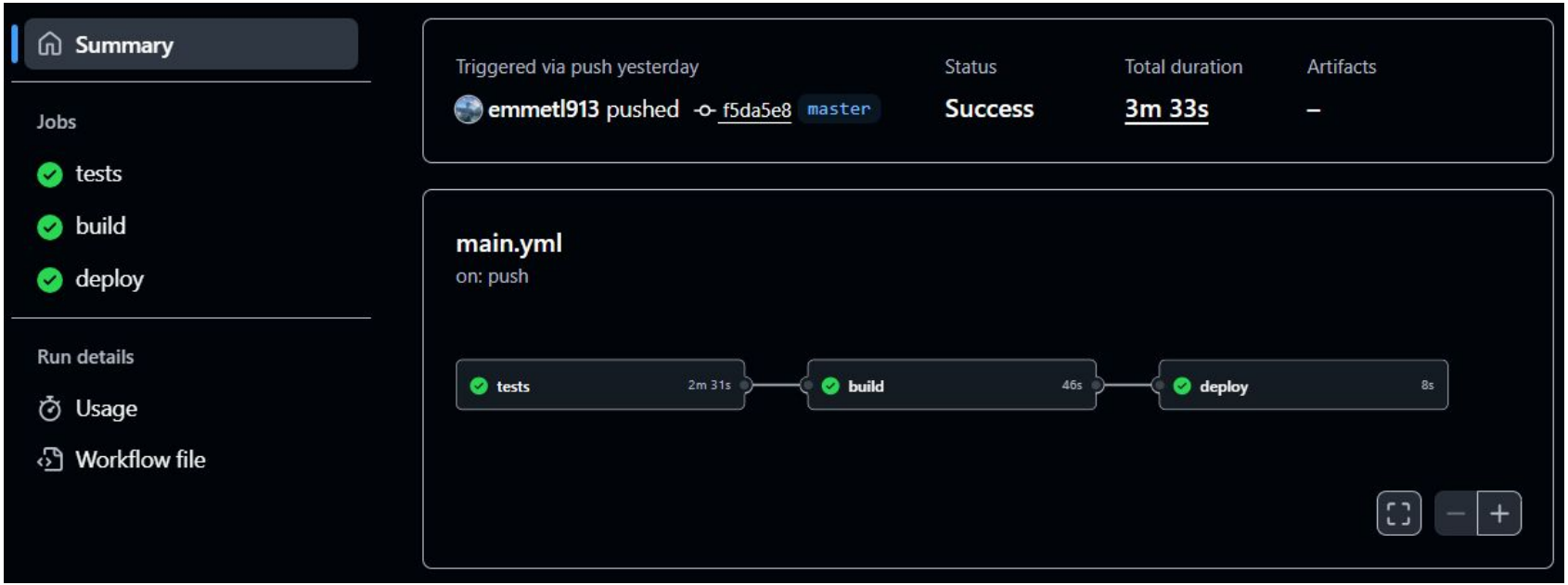
# Mobile App

- Cellular
  - Main App Functionality
- CellularCore
  - Contains core functions for Cellular
- CellularUnitTests
  - Tests CellularCore functions

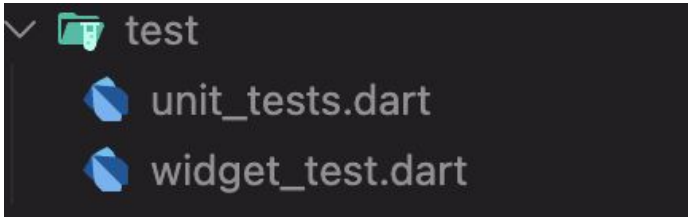


# Cloud - CI/CD Pipeline



# Watch Tests




```
charlescarroll@DESKTOP-C7TGMLH flutter_prototype % flutter test test/unit_tests.dart
00:03 +31: All tests passed!
```

```
charlescarroll@DESKTOP-C7TGMLH flutter_prototype % flutter test ./test/widget_test.dart
00:01 +3: All tests passed!
```

# Ball Spinner Controller Tests

```
ballz@Ballz:~/BallSpinner-Controller-v2 $ source venv/bin/activate
(venv) ballz@Ballz:~/BallSpinner-Controller-v2 $ pytest
============================================= test session starts ==============================================
platform linux -- Python 3.13.5, pytest-9.0.2, pluggy-1.6.0
PyQt6 6.9.1 -- Qt runtime 6.9.2 -- Qt compiled 6.9.0
rootdir: /home/ballz/BallSpinner-Controller-v2
configfile: pytest.ini
testpaths: tests
plugins: qt-4.5.0
collected 217 items

tests/cloud/test_cloud_api.py ....                                                                    [  1%]
tests/integration/test_gui_interactions.py sssssssssssssssssssssssss                                  [ 13%]
tests/integration/test_main.py ...                                                                    [ 15%]
tests/test_smartdot_data.py ....                                                                      [ 17%]
tests/unit/test_ads1115_current_sensor.py ...                                                         [ 18%]
tests/unit/test_analysis_mode_bandpass.py .                                                           [ 18%]
tests/unit/test_apiutils.py ............                                                              [ 24%]
tests/unit/test_bdcmotor.py ...                                                                       [ 26%]
tests/unit/test_bsc.py ......                                                                         [ 29%]
tests/unit/test_cloudapi.py ..............                                                            [ 35%]
tests/unit/test_data_controller.py ....................                                               [ 45%]
tests/unit/test_data_models.py ..........................                                             [ 57%]
tests/unit/test_gui_interactions_unit.py ....                                                         [ 58%]
tests/unit/test_gui_windows.py ..........s.ss........................                                 [ 76%]
tests/unit/test_imotor.py ..                                                                          [ 77%]
tests/unit/test_ismartdot.py ..                                                                       [ 78%]
tests/unit/test_motor_test_analysis.py ..                                                             [ 79%]
tests/unit/test_package_smartdot.py .                                                                 [ 80%]
tests/unit/test_shot_script_data.py ...                                                               [ 81%]
tests/unit/test_simmotor.py .......                                                                   [ 84%]
tests/unit/test_simsmartdot.py ...                                                                    [ 86%]
tests/unit/test_stepmotor.py ss                                                                       [ 87%]
tests/unit/test_utils.py .........                                                                    [ 91%]
tests/unit/test_utils_bandpass.py ..                                                                  [ 92%]
tests/unit/test_utils_functions.py ......                                                             [ 94%]
tests/unit/test_wavelet_dialog.py .....                                                               [ 97%]
tests/unit/test_wavelet_dialog_order.py ..                                                            [ 98%]
tests/unit/test_wavelet_dialog_outliers.py .                                                          [ 98%]
tests/unit/test_wavelet_helper_widget.py ...                                                          [100%]

============================================== warnings summary ==============================================
<frozen importlib._bootstrap>:488
  <frozen importlib._bootstrap>:488: DeprecationWarning: builtin type SwigPyPacked has no __module__ attribute

<frozen importlib._bootstrap>:488
  <frozen importlib._bootstrap>:488: DeprecationWarning: builtin type SwigPyObject has no __module__ attribute

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
========================================= 186 passed, 31 skipped, 2 warnings in 49.49s =========================================
```

# Future Work

# Mobile

- Improve Custom Query Engine/ Stats Page
  - More rigorous testing
  - Add additional statistics
- Shot Page
  - Indication for splits
  - Disable editing for completed games
  - Landscape compatibility
- UI/UX Improvements
- Update Session List Page
- Create Settings page
  - Account, Ciclopes, Ciclopes Test, API test, Smartwatch, Smartdot, SQL Database
- Limit guest account features
- Continue to Test and Improve
  - SmartDot Integration
  - Ciclopes Integration
  - Cloud Integration

# Watch

- **Relative sizing for different watch types**
- Continue wiki page
- **Cross platform integration**
  - **Apple watch deployment**
- **Replay video recorded from watch**
- **Ciclopes integration**
- **Implement local database**
  - **Cloud sync**
- Access to stats from watch
- Live scoring from mobile application
- BLE Acknowledgements
  - For offline data storage/proper dequeue

# Cloud

- API Improvements and Additions
- Automated Testing Improvements
- Seperate database migration application

# Ciclopes

- Tune algorithms for robustness in real world use
- Full cloud integration for persistent user data in an efficient manner
- Implementation of on device inference
    - Real time overlay and calculations
    - Opens up new User Experience(UX) workflows
- Ciclopes-API serving in the cloud for widespread use
- UI/UX improvements based on bowlers' real workflows
- Potential IoT camera array development for use in bowling alleys
    - Specto functionality at a fraction of the price
    - Direct integration into the application for a better bowling experience

# Ball Spinner Controller - Software

- Smart Dot connection robustness & data fidelity
- Implement the current sensors
- Improve Primary Motor Tuning
- Use UART via GPIO for primary motor

# Ball Spinner Controller - Hardware

- Enhanced Safety features
- Communication Expansion UART or CAN BUS
- 2nd and 3rd degree motor encoders for  accurate positioning

# Ball Spinner Mechanical System

## Final Aluminum Prototype

- **Physical stops to prevent interference**
  - 2nd Degree: No further than ± 45 degrees
  - 3rd Degree: No further than ± 22.5 degrees
- **Improve motor mounting**
  - Prevent motor slip
- **Higher Torque 3rd degree motor**
  - Different system orientation for larger motor
- **Improve Polycarbonate 3D printing methods**
  - Prevent warping and improve tolerances

# Ball Spinner Mechanical System

Enclosure Platform

- Improve Ventilation
- Improve back plate for better port connection
- More room for motor encoders
- Wire routing from motors to the Ball Spinner Controller

# Questions?

# Thank you!