RevMetrix Milestone 3

Andrew Watkins, Chris Robinson, David Kyeremeh, Patrick Devine, Ryan Curry, Robert Fields, Sam Diskin, Josh Byers, Brandon Woodward, Zach Cox, Carson Mack, Kyle Fankhauser, James Devine

Presentation Overview

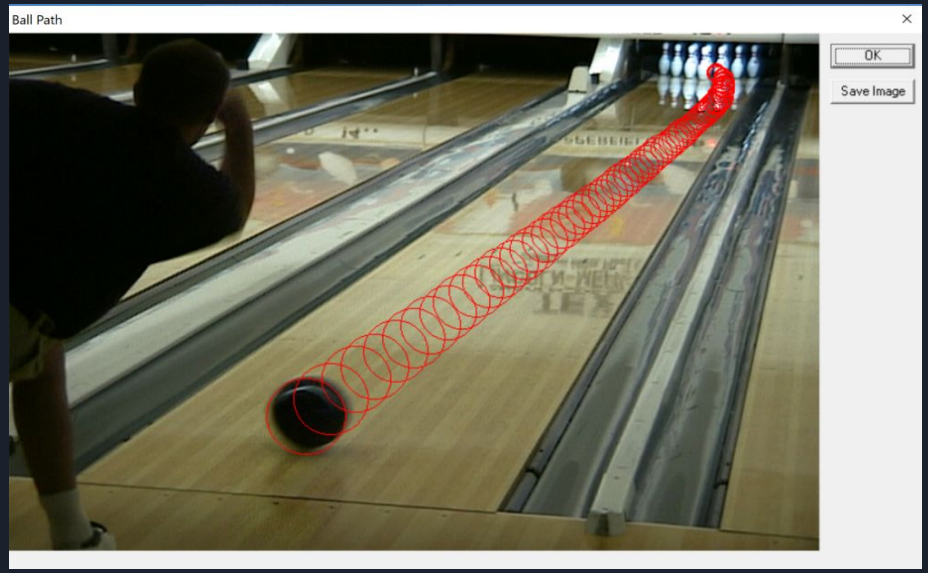
- Background
  - RevMetrix, SmartDot, Ciclopes, Ball Spinner
- Project Overview
- Tools & technologies
- Current Design
  - UMLs and schemas
- Current Implementations
- Demos
- Future implementations
- Conclusion

RevMetrix Background

- Brainchild of Professor Hake and Dr. Babcock
- Combination of Ciclopes and SmartDot Module
- Both compliment each other with supplemental data
- Application:
  - Bowlers
  - Researchers

What is Ciclopes

- Dr. Babcock's contribution
- External bowling analysis



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			

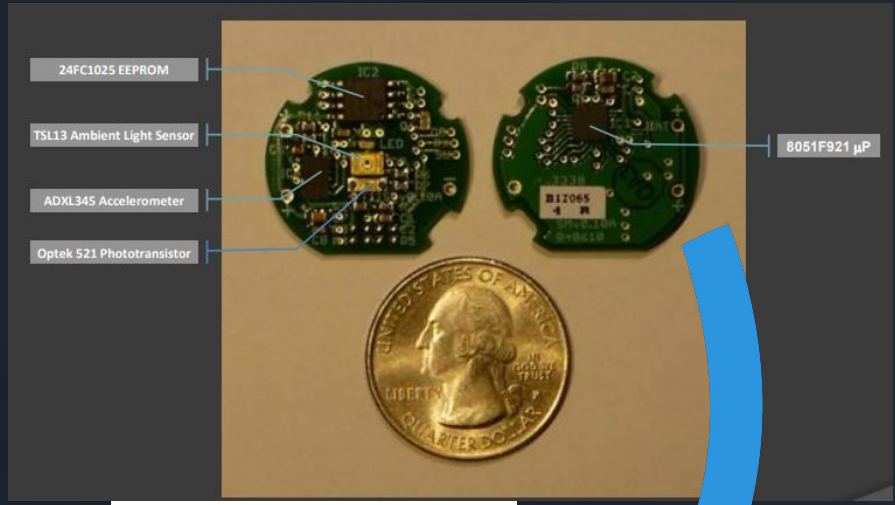
Shot Information

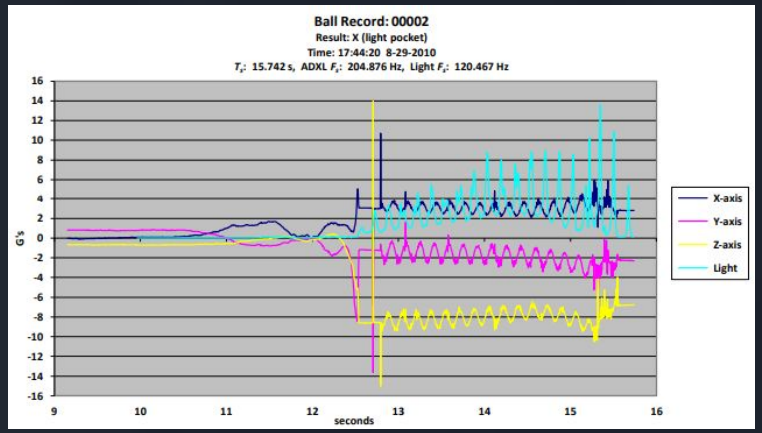
	Shot 1	Shot 2	Shot 3
Initial Speed (mph)	17.03	16.62	16.68
Average Speed (mph)	15.90	15.71	15.37
Launch Angle (deg)	-2.03	-2.09	-2.33
Entry Angle (deg)	6.66	6.83	7.89
Loft Distance (ft)	1.85	4.43	1.03
Laydown Board	21.79	20.52	22.18
Arrow Board	16.19	15.87	15.05
Pindeck Board	16.88	17.34	17.56
Breakpoint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

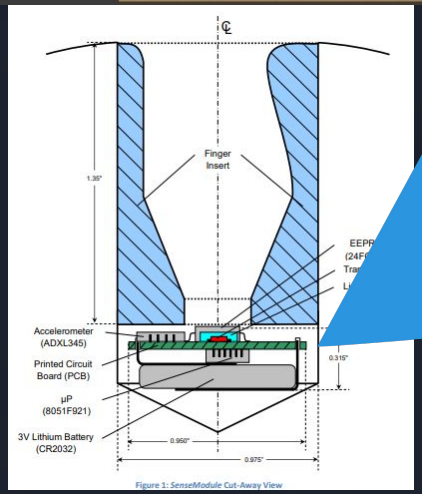
Ready

What is SmartDot

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

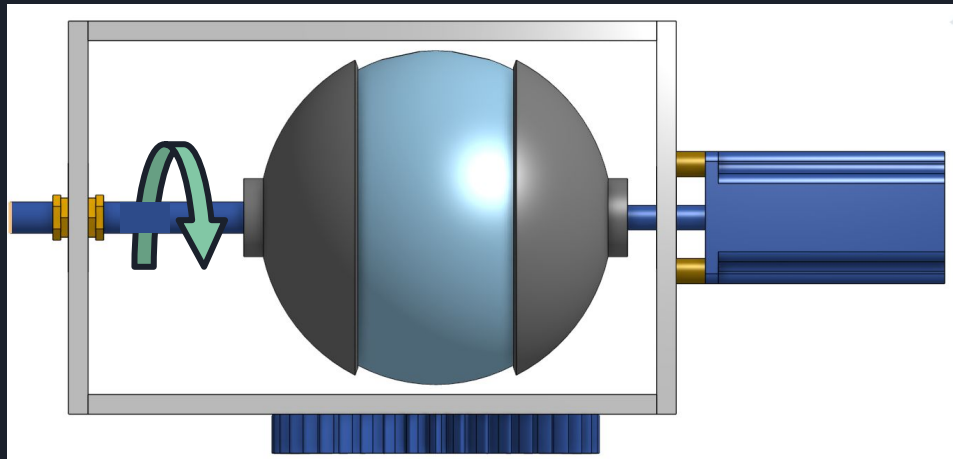






What is the Ball Spinner?

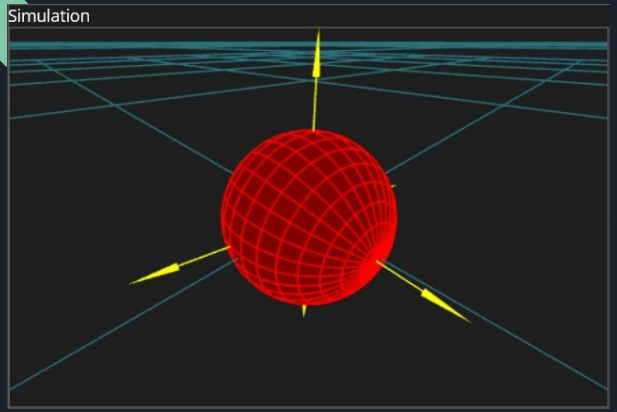
- Device that can emulate a throw down a lane
- Controls movement across 3 orthogonal axes of spin
- External sensors generate data

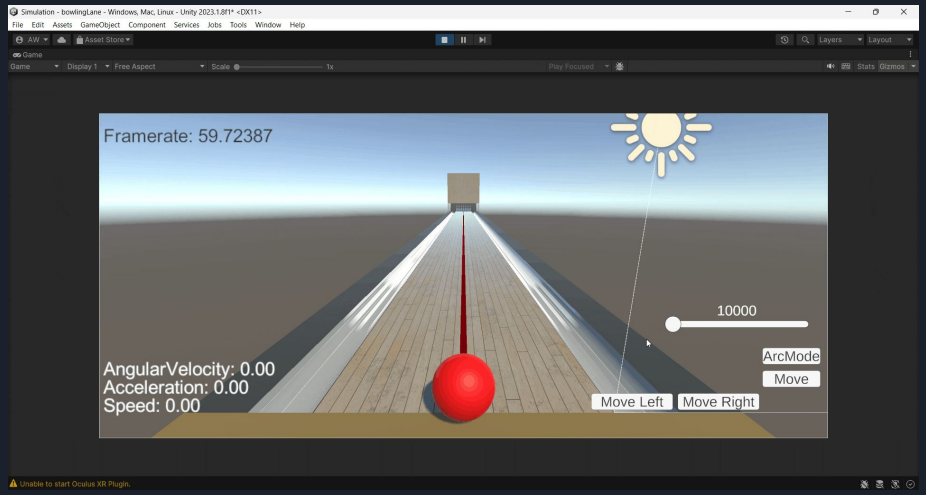


Questions?

Project Overview

Simulation

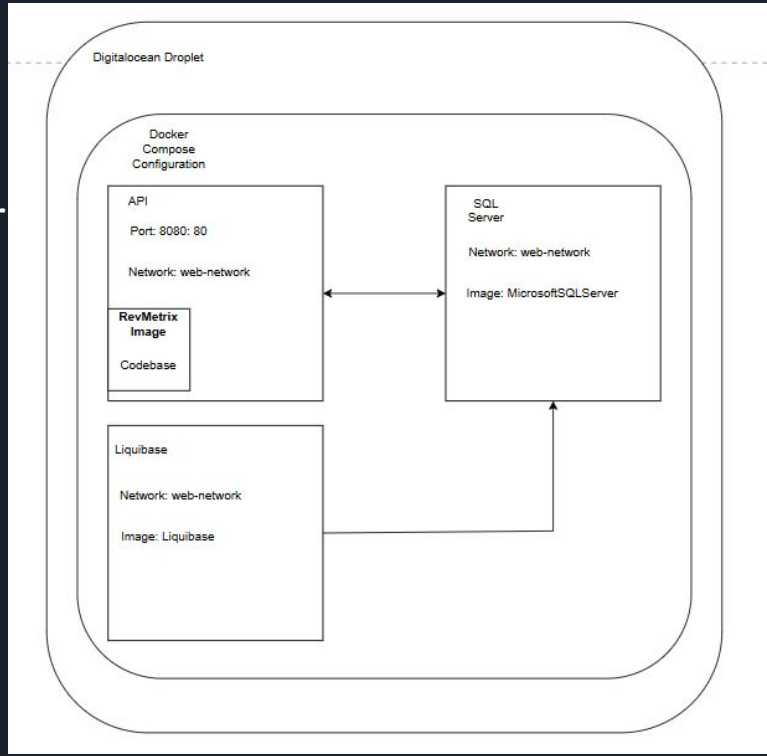




- Shows ball rotation
- Visualizes data through graphs
- Choice between real data from smartdot, and a simulated version of a possible shot
- Unity game with movable shot origin and a possible curve

BSA Backend/Cloud

- BSA:
  - Client-side API methods to interact with API server
  - Processing of incoming/outgoing data
- Cloud:
  - Server-side API methods
  - Administrative tasks for production server
    - CI/CD
    - Server infrastructure



BSA Frontend

- Interface between the User and further system
  - Cloud - save and load users, shots and arsenal
  - Controller - communicate data regarding the motors and handle SmartDot control
- Output of SmartDot sensor data

Ball Spinner Controller

- Connects to the BSA Frontend
- Receives commands from the BSA to control the motors at variable speed
- Connects to Metamotion module over Bluetooth Low Energy (BLE) and forwards data to the BSA
- Has a Human Machine Interface (HMI)
- Implements a TCP messaging protocol with the BSA

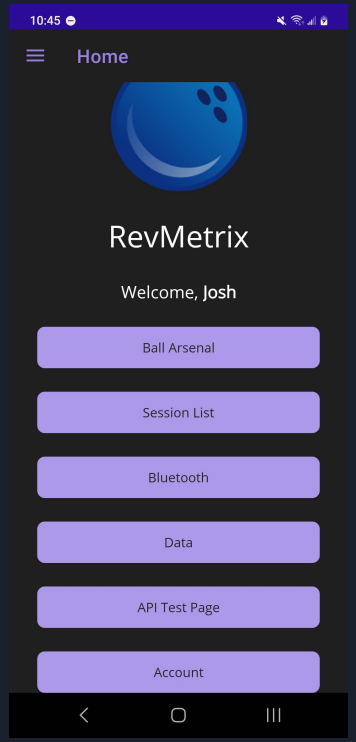
Include Raspberry Pi and python based device. Has GPIO to connect to multiple hardware devices.

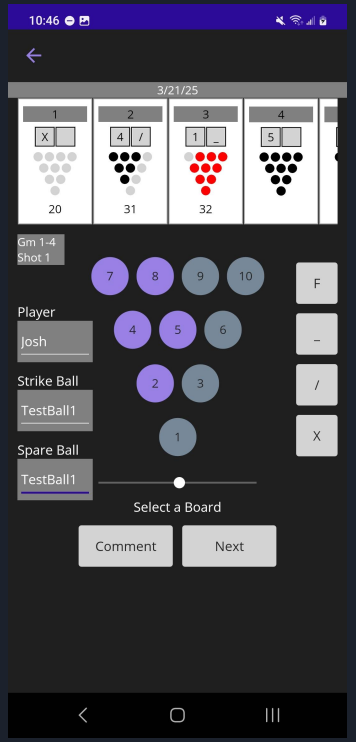
Ball Spinner Team

- Recreates the motion of a bowling ball rolling down a lane
- Motion will be split between 3 DoFs (Degrees of Freedom)
    - Each will be built on the system of the last.
- Will be used to test the accuracy of the SmartDot

Mobile Application Team

- Mobile app for the RevMetrix project
- Allows users to create an account
- Access personal ball arsenal
- Register establishments/ events
- Start sessions and games
- Save shot data





Questions?

Technologies Used

Application

- .NET, .NET MAUI
- OpenTK(Three.JS)
- Livecharts2(Chart.JS)
- xUnit

Cloud

- .NET
- xUnit
- Docker
- Digital Ocean
- Github Actions

Ball Spinner Controller

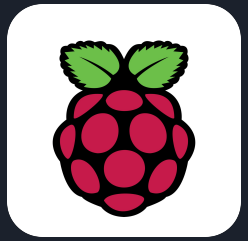
- Raspberry Pi
- KiCad
- Tkinter

Ball Spinner

- SolidWorks
- Onshape

Cellular

- .NET MAUI
- SQLite



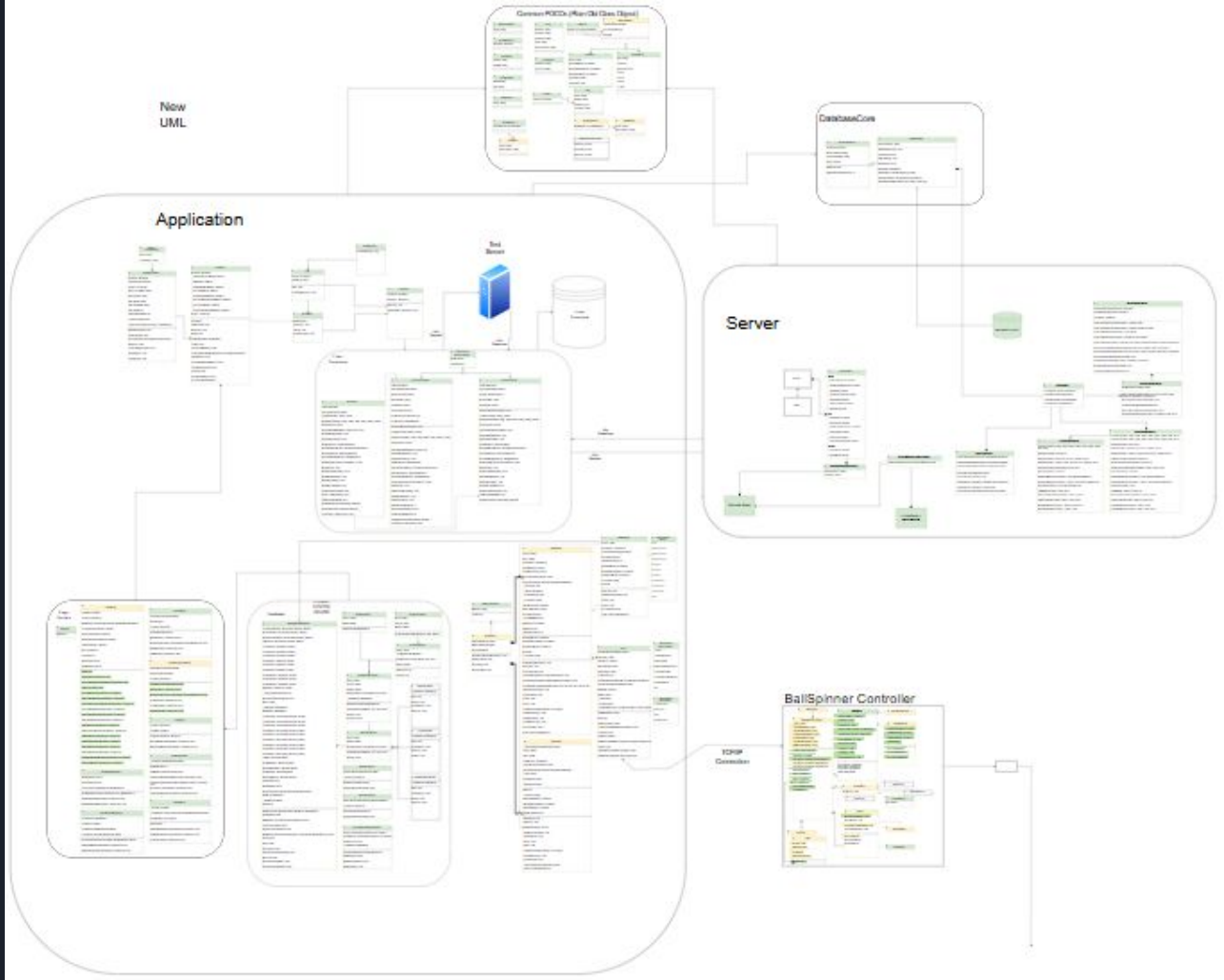
Current Design

Ball Spinner UML

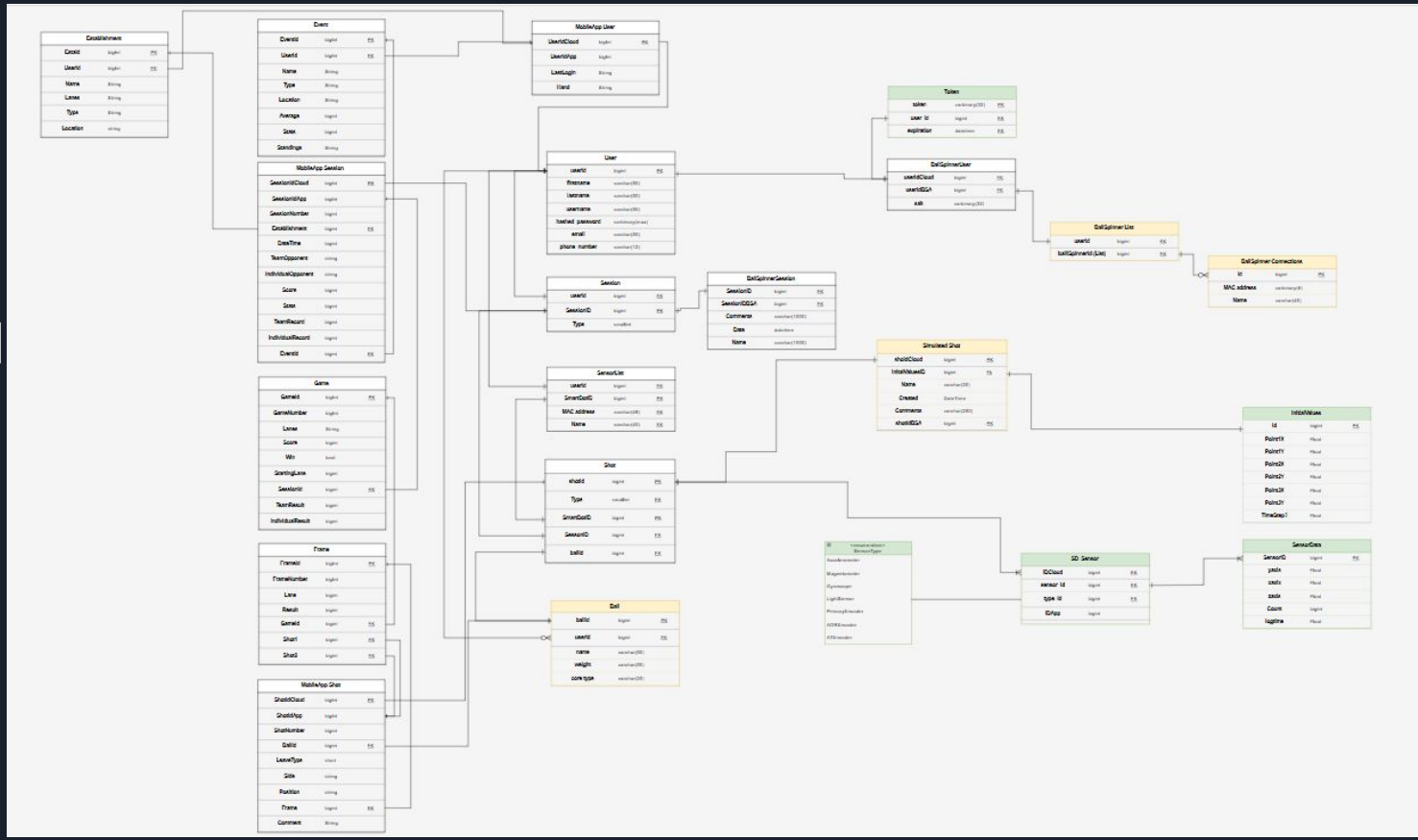
Implemented

Partially Implemented

Not Yet Implemented



Current Database Schema Cloud

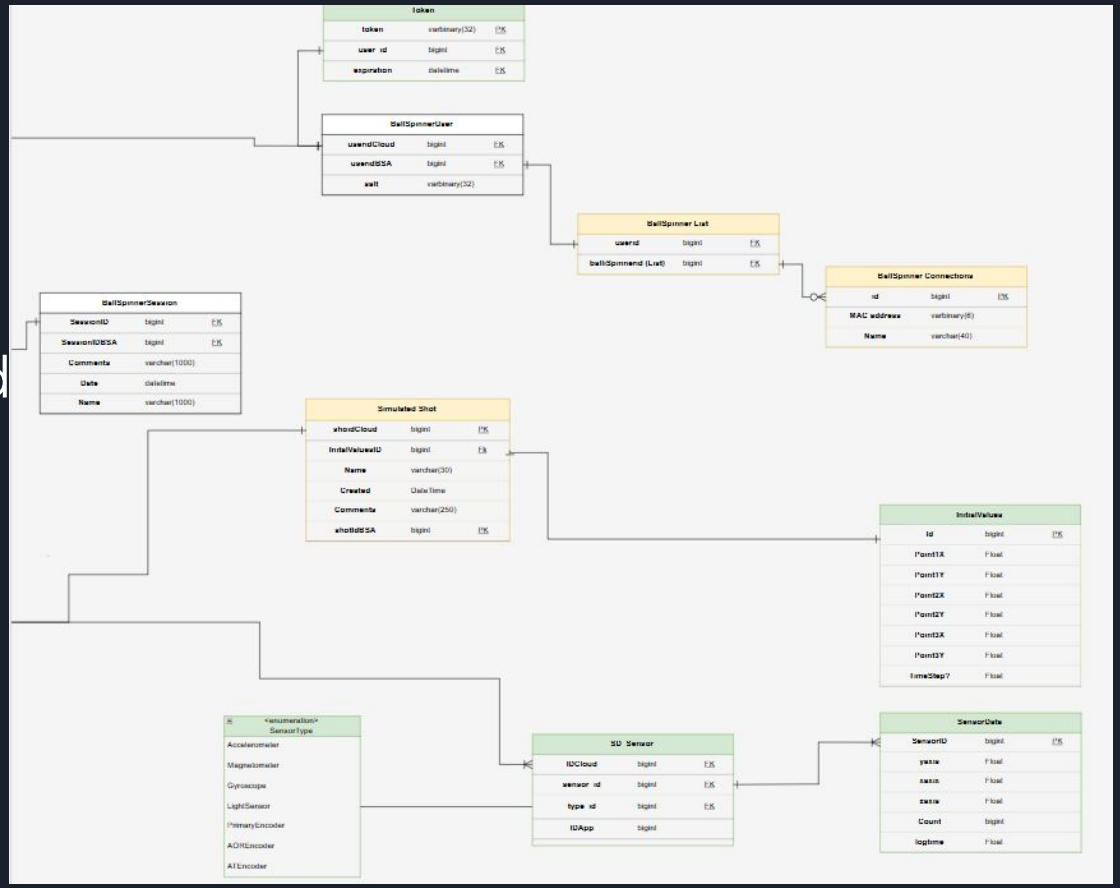


Implemented

Partially Implemented

Not Yet Implemented

Current Database Schema Cloud



Implemented

Partially Implemented

Not Yet Implemented

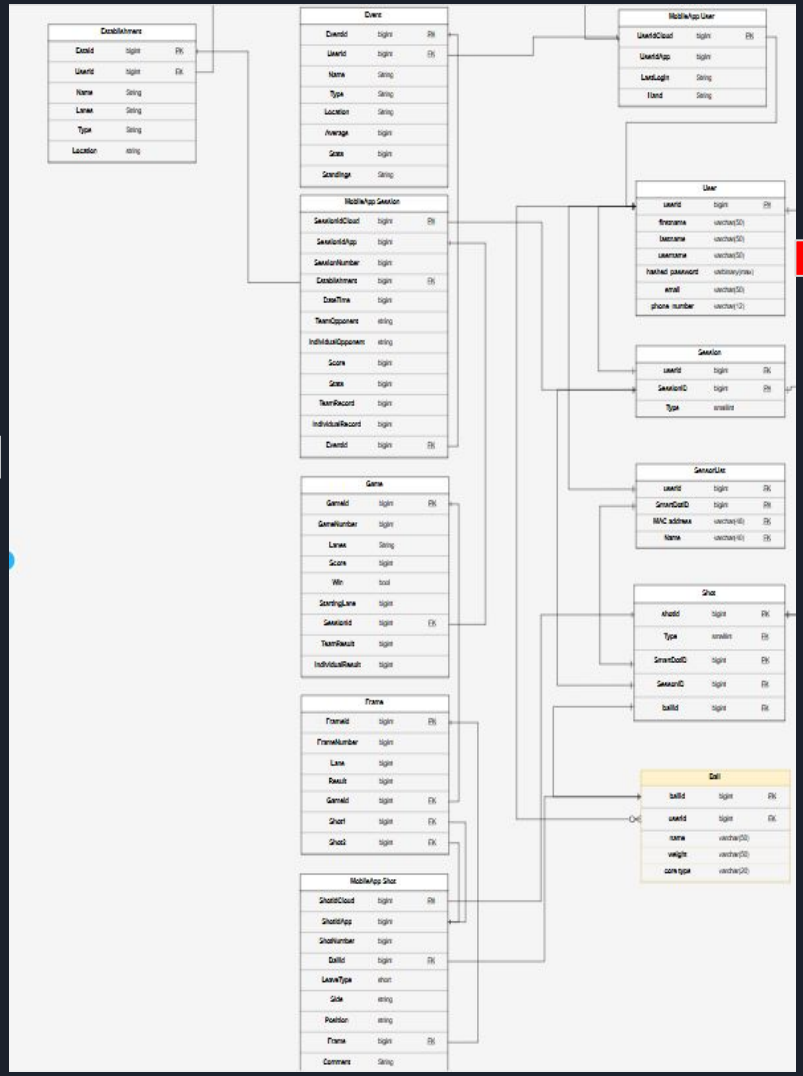
Current Design

Current Database Schema Cloud

Implemented

Partially Implemented

Not Yet Implemented



User

userid	bigint	PK
firstname	varchar(50)	
lastname	varchar(50)	
username	varchar(50)	
hashed_password	varbinary(max)	
email	varchar(50)	
phone_number	varchar(12)	

Session

userid	bigint	FK
SessionID	bigint	PK
Type	smallint	

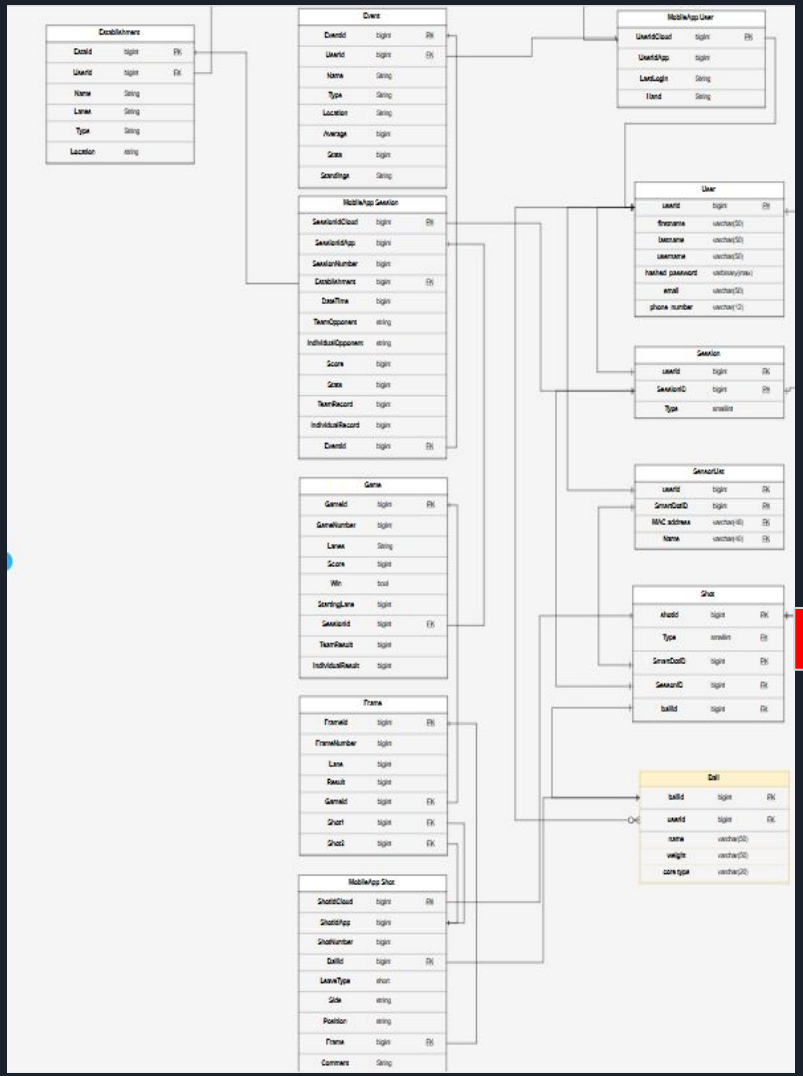
Current Design

Current Database Schema Cloud

Implemented

Partially Implemented

Not Yet Implemented



Shot		
shotid	bigint	PK
Type	smallint	FK
SmartDotID	bigint	PK
SessonID	bigint	FK
ballid	bigint	FK

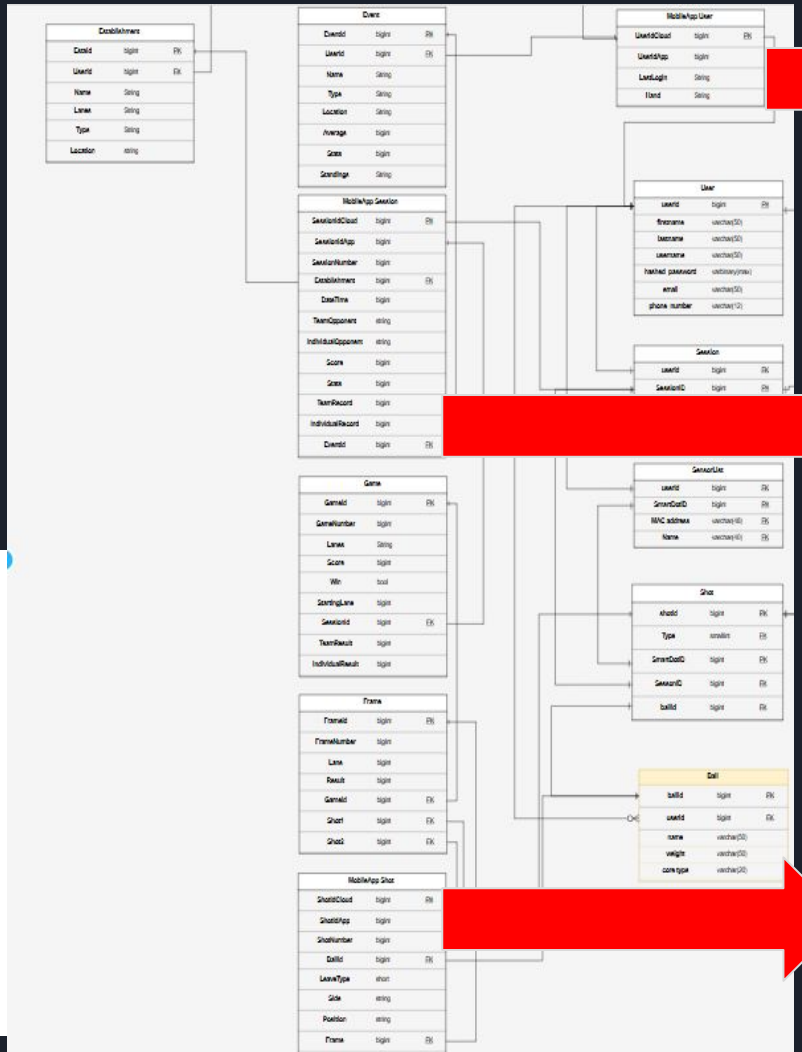
Ball		
ballid	bigint	PK
userid	bigint	FK
name	varchar(50)	
weight	varchar(50)	
core type	varchar(20)	

Current Design

Implemented

Partially Implemented

Not Yet Implemented



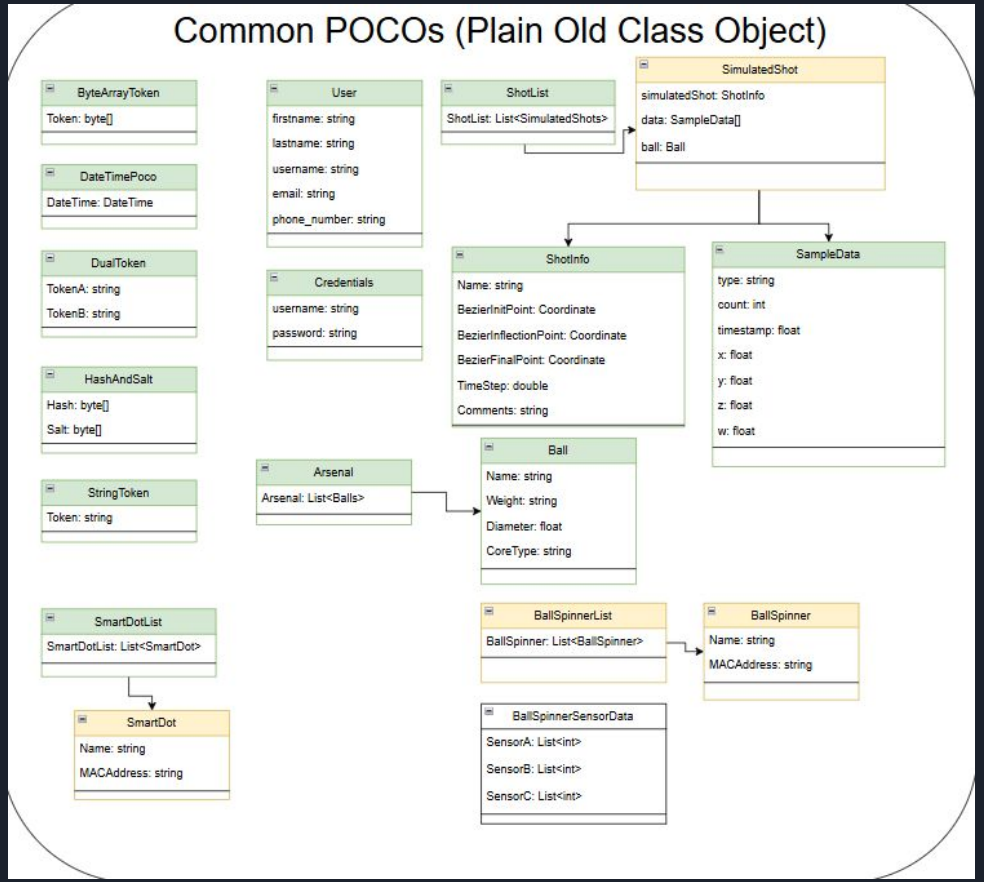
MobileApp User		
UserIdCloud	bigInt	PK
UserIdApp	bigInt	
LastLogin	String	
Hand	String	

MobileApp Session		
SessionIdCloud	bigInt	PK
SessionIdApp	bigInt	
SessionNumber	bigint	
Establishment	bigint	FK
DateTime	bigint	
TeamOpponent	string	
IndividualOpponent	string	
Score	bigint	
Stats	bigint	
TeamRecord	bigint	
IndividualRecord	bigint	
EventId	bigint	FK

MobileApp Shot		
ShotIdCloud	bigInt	PK
ShotIdApp	bigInt	
ShotNumber	bigint	
BallId	bigint	FK
LeaveType	short	
Side	string	
Position	string	
Frame	bigint	FK
Comment	String	

Current Design

POCOs



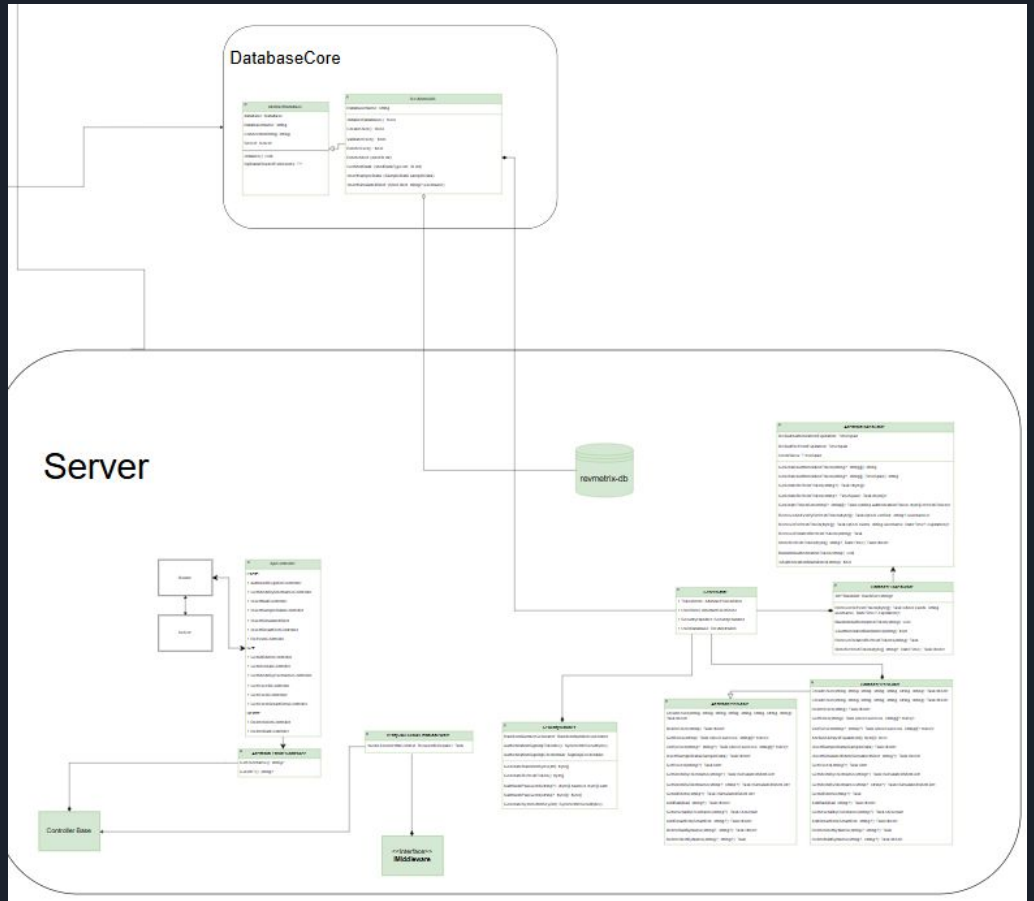
Implemented

Partially Implemented

Not Yet Implemented

Current Design

Cloud



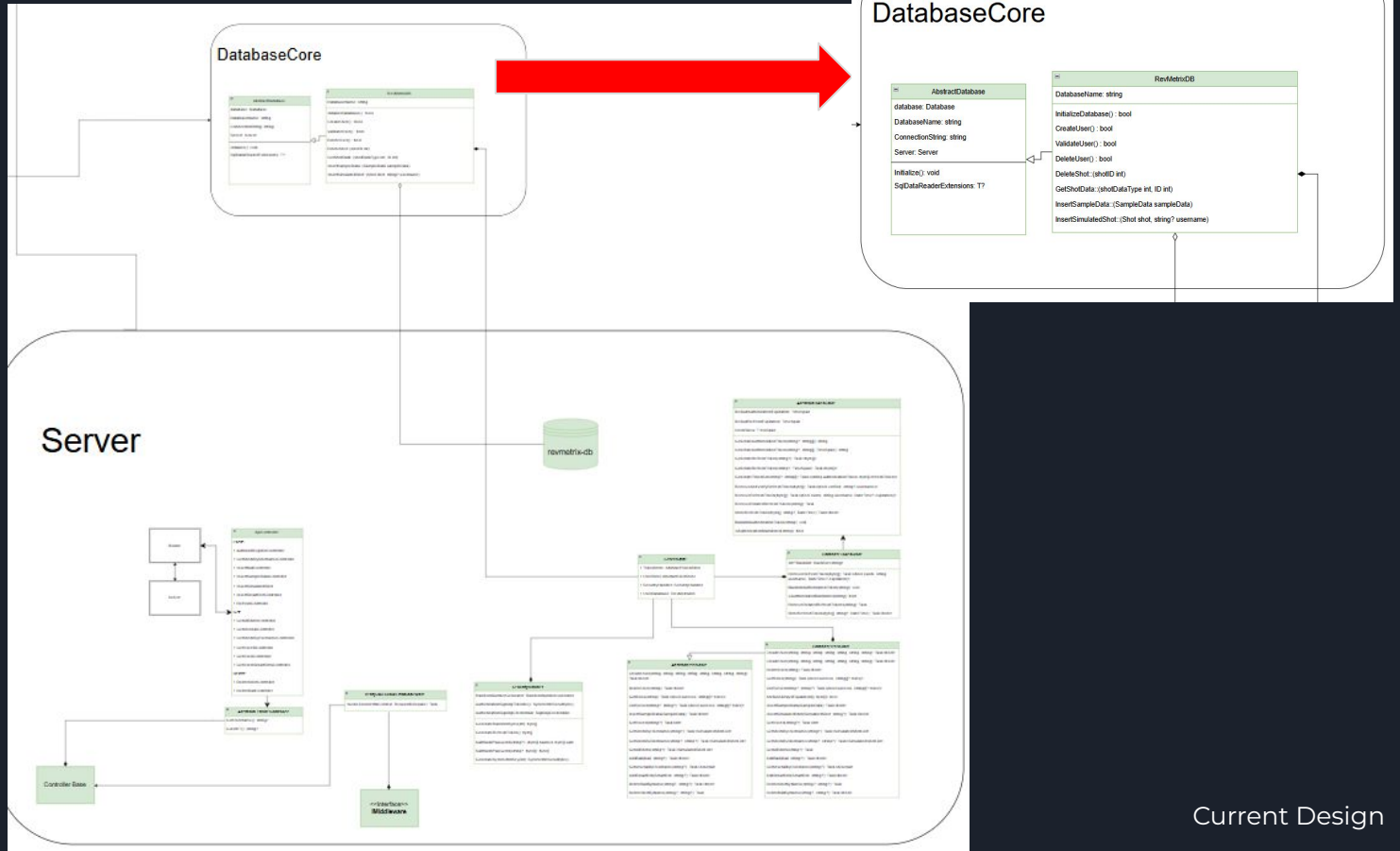
Implemented

Partially Implemented

Not Yet Implemented

Current Design

Cloud



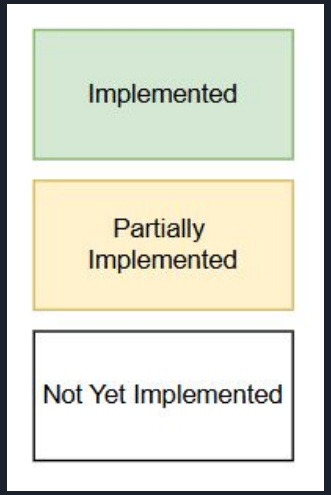
Implemented

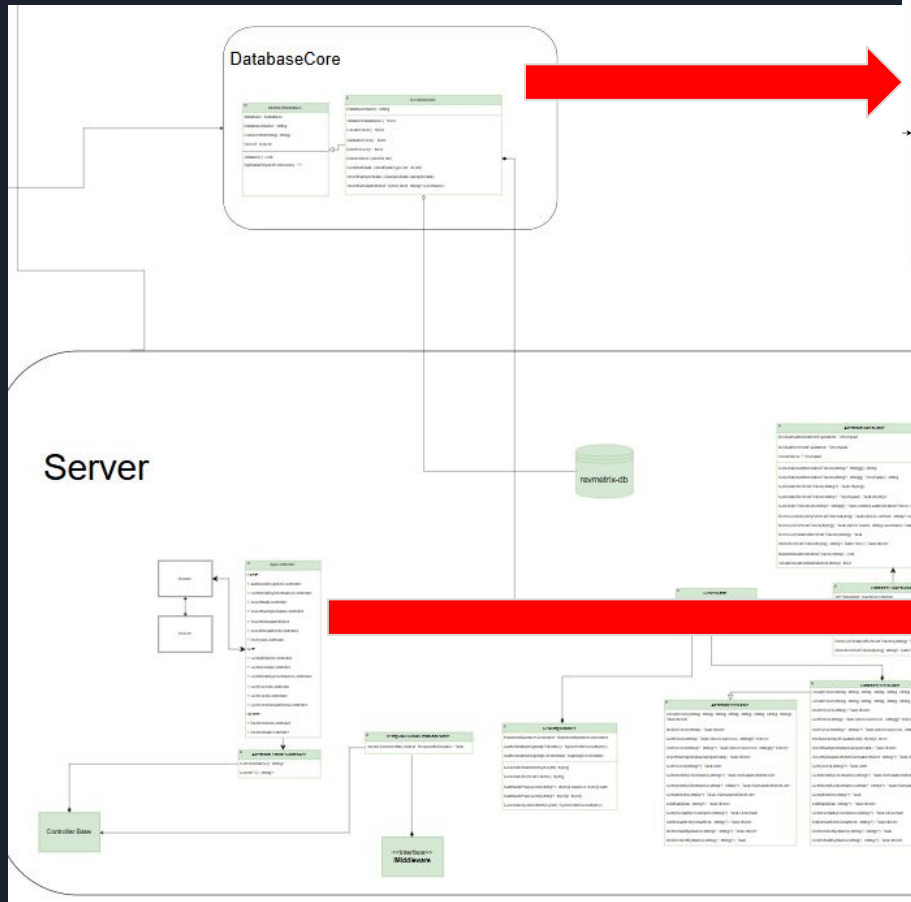
Partially Implemented

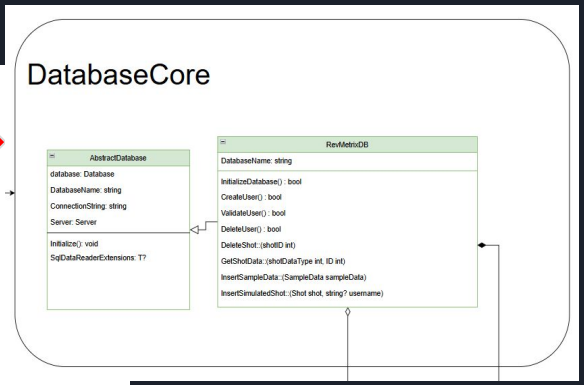
Not Yet Implemented

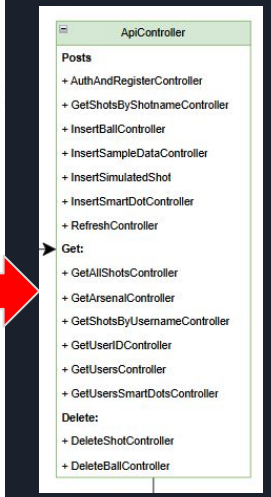
Current Design

Cloud



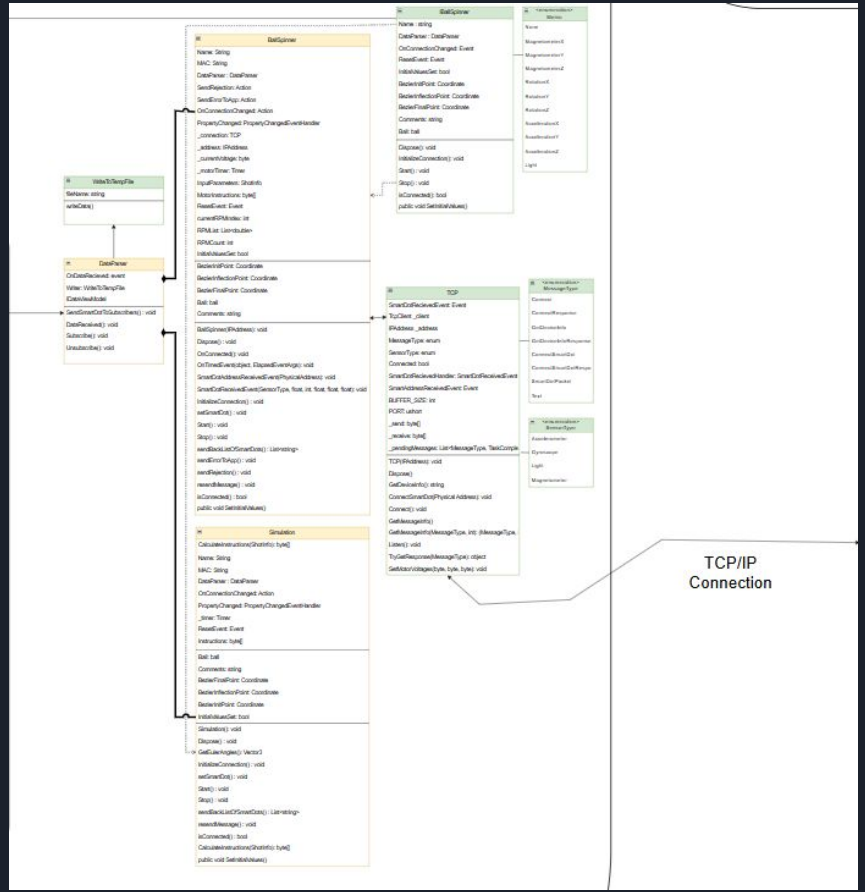






Current Design

BSA Backend

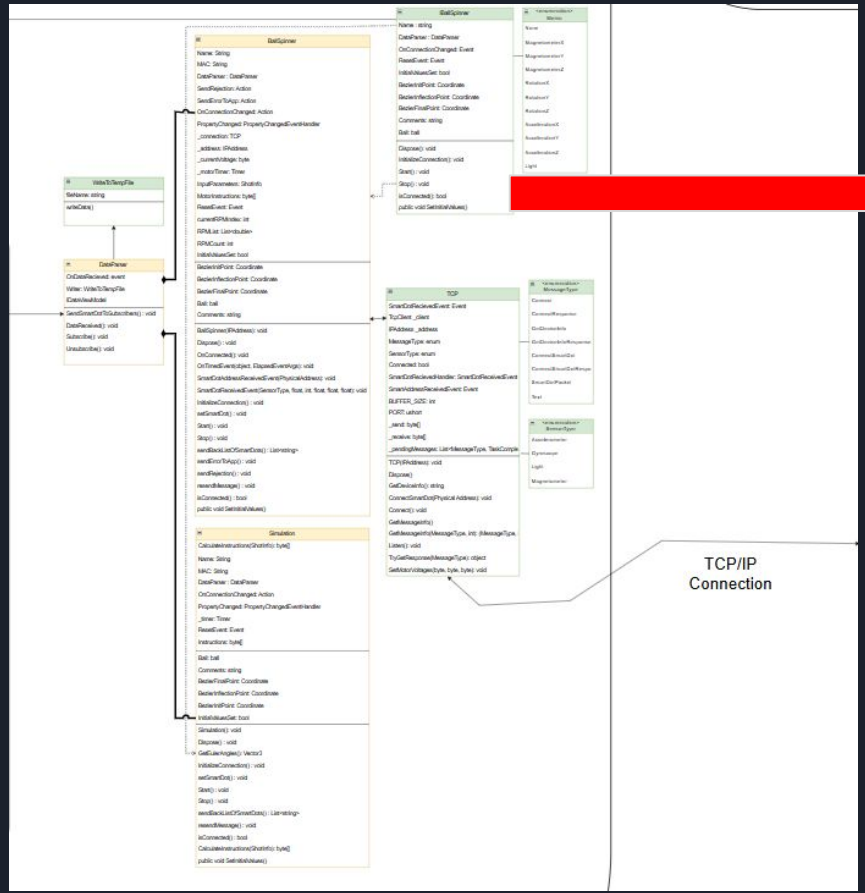


Implemented

Partially Implemented

Not Yet Implemented

BSA Backend



IBallSpinner
Name : string
DataParser : DataParser
OnConnectionChanged: Event
ResetEvent: Event
InitialValuesSet: bool
BezierInitPoint: Coordinate
BezierInflectionPoint: Coordinate
BezierFinalPoint: Coordinate
Comments: string
Ball: ball
Dispose(): void
InitializeConnection(): void
Start() : void
Stop() : void
isConnected(): bool
public void SetInitialValues()

Implemented

Partially Implemented

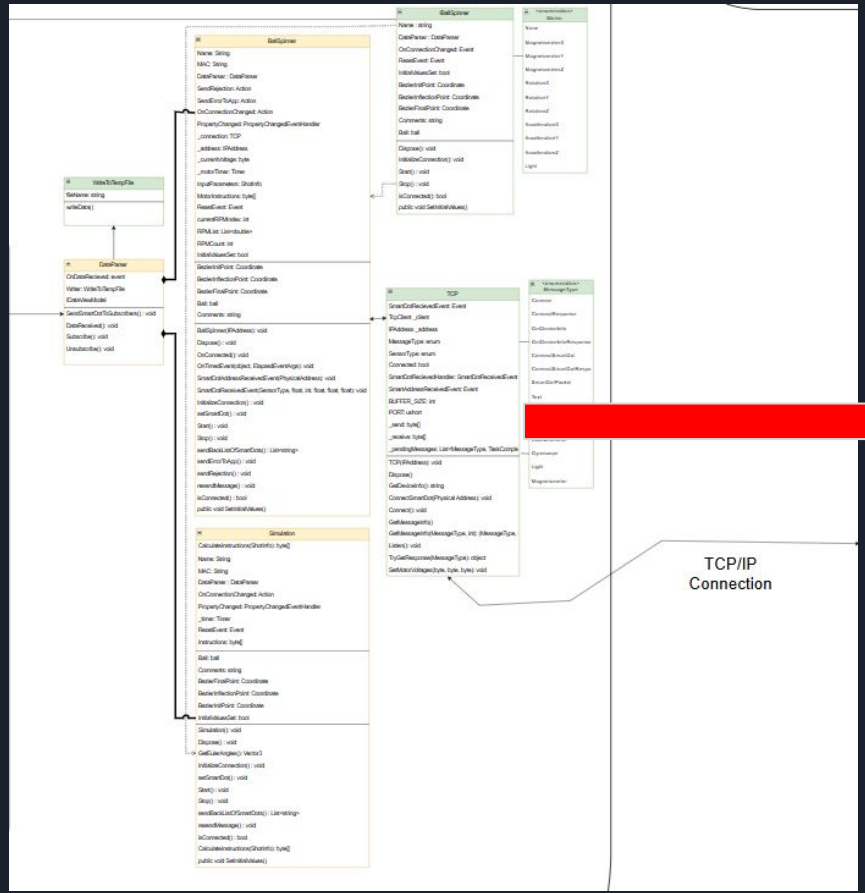
Not Yet Implemented

BSA Backend

Implemented

Partially Implemented

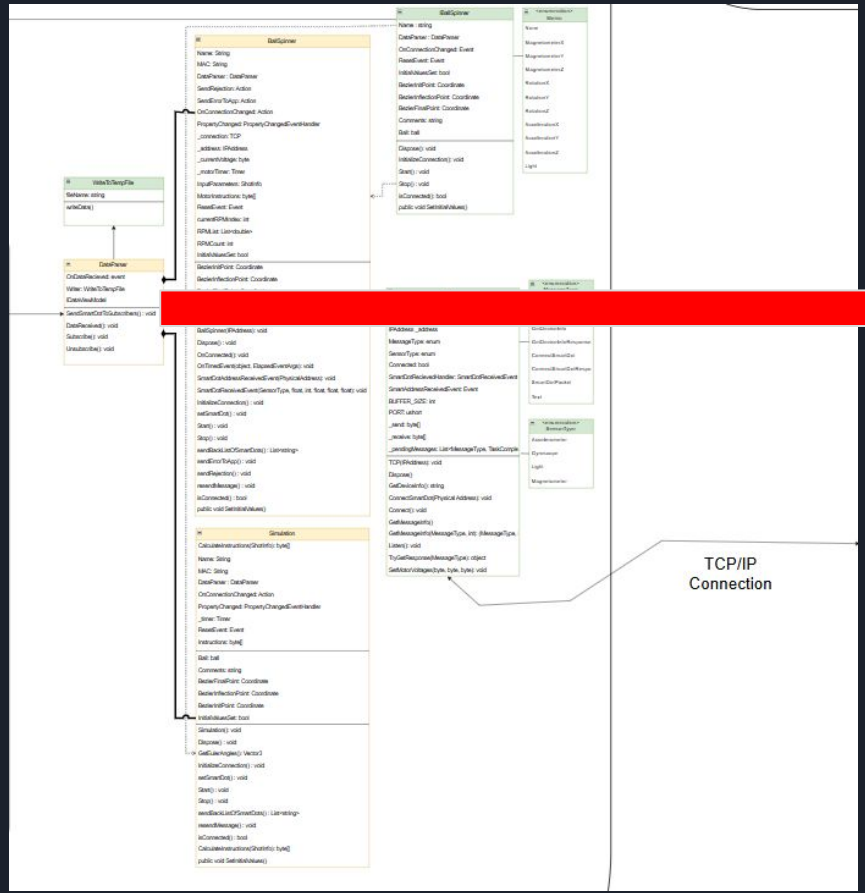
Not Yet Implemented



TCP/IP Connection

TCP
SmartDotRecievedEvent: Event
TcpClient _client
IPAddress _address
MessageType: enum
SensorType: enum
Connected: bool
SmartDotRecievedHandler: SmartDotReceivedEvent
SmartAddressReceivedEvent: Event
BUFFER_SIZE: int
PORT: ushort
_send: byte[]
_receive: byte[]
_pendingMessages: List<MessageType, TaskComple
TCP(IPAddress): void
Dispose()
GetDeviceInfo(): string
ConnectSmartDot(Physical Address): void
Connect(): void
GetMessageInfo()
GetMessageInfo(MessageType, int): (MessageType,
Listen(): void
TryGetResponse(MessageType): object
SetMotorVoltages(byte, byte, byte): void

BSA Backend



Implemented

Partially Implemented

Not Yet Implemented

TCP/IP Connection

DataParser

OnDataRecieved: event

Writer: WriteToTempFile

IDataViewModel

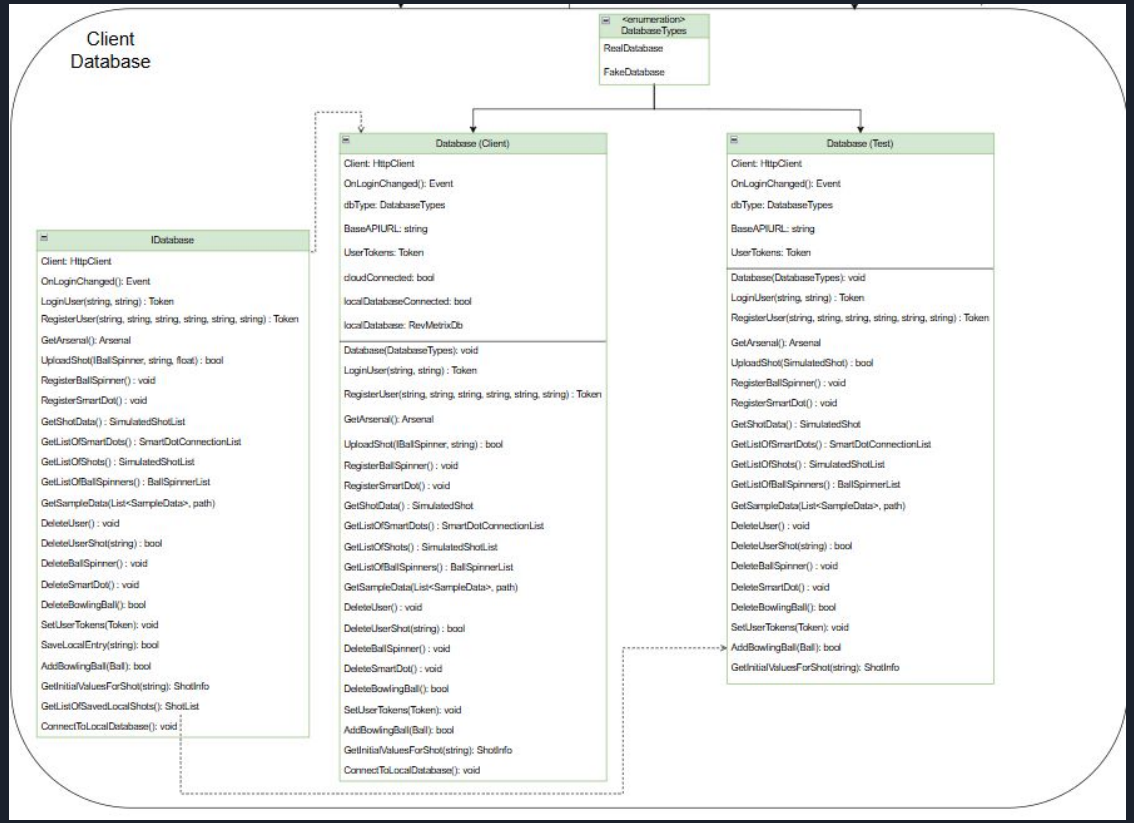
SendSmartDotToSubscribers() : void

DataReceived(): void

Subscribe(): void

Unsubscribe(): void

BSA Backend

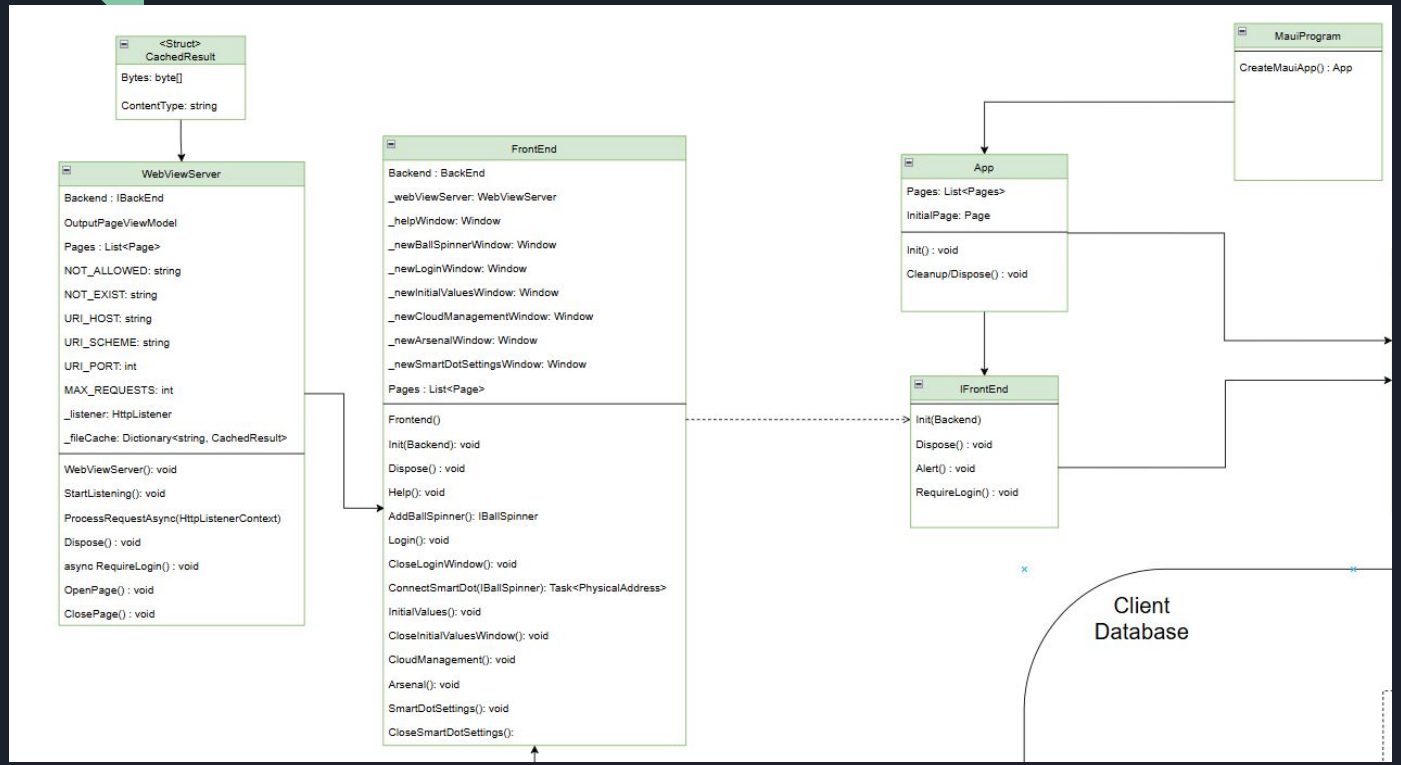


Implemented

Partially Implemented

Not Yet Implemented

BSA Frontend

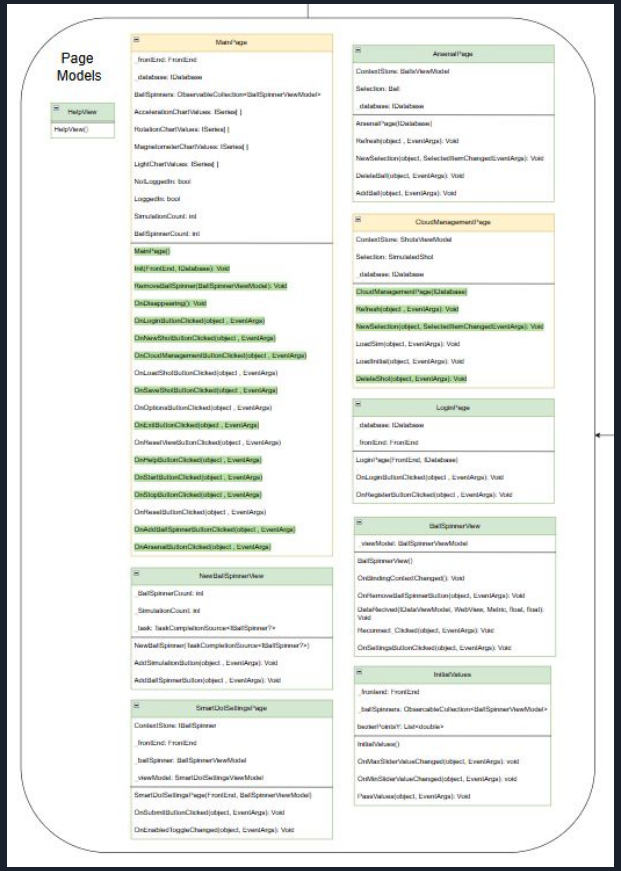


Implemented

Partially Implemented

Not Yet Implemented

BSA Frontend



Implemented

Partially Implemented

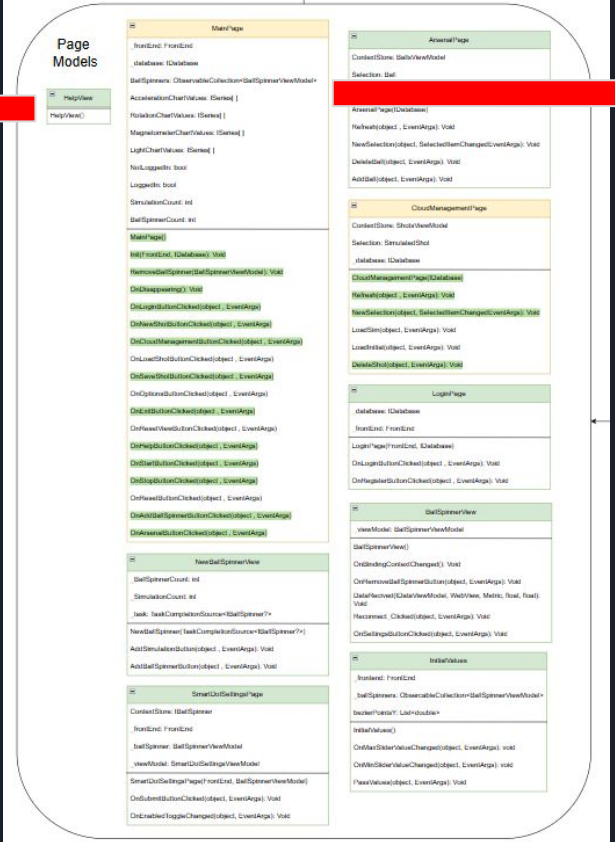
Not Yet Implemented

BSA Frontend

HelpView

HelpView()

Page Models

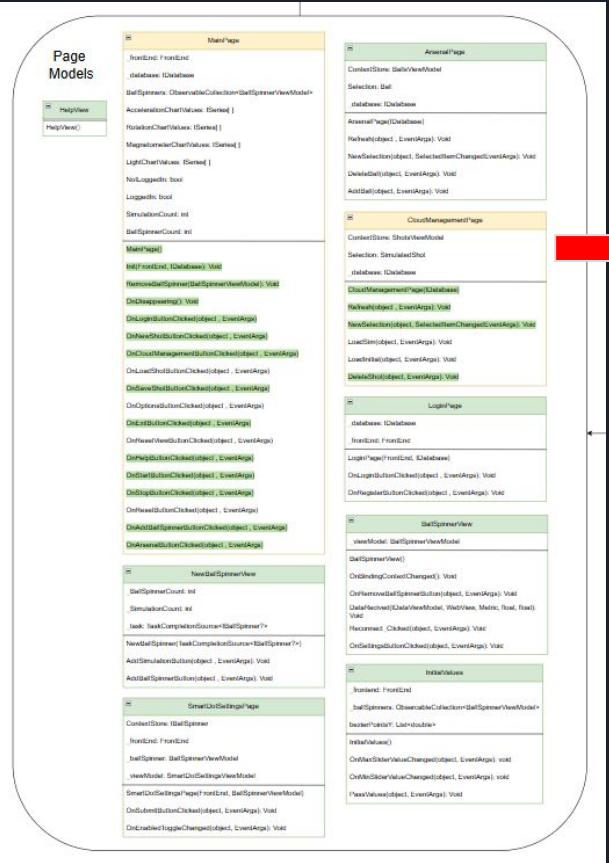


Implemented

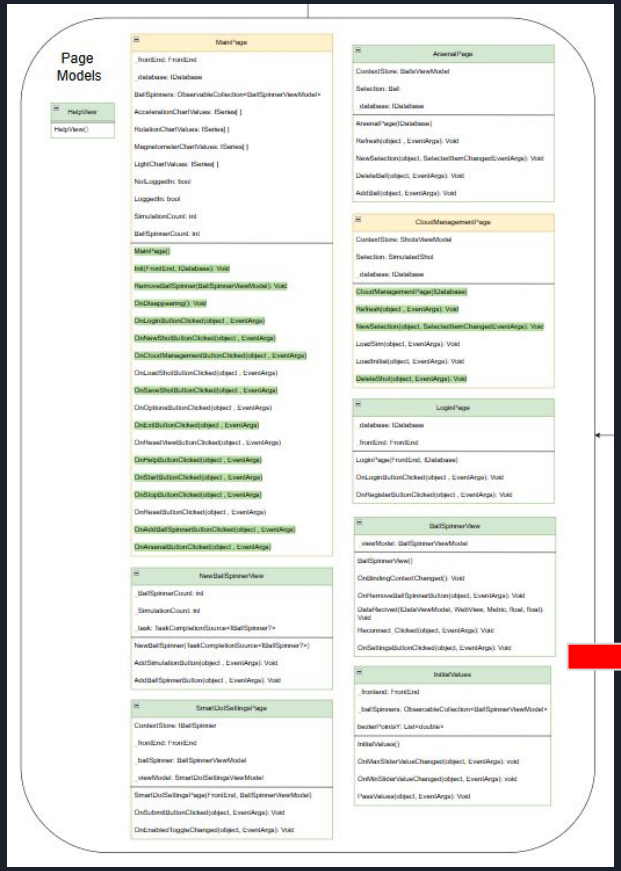
Partially Implemented

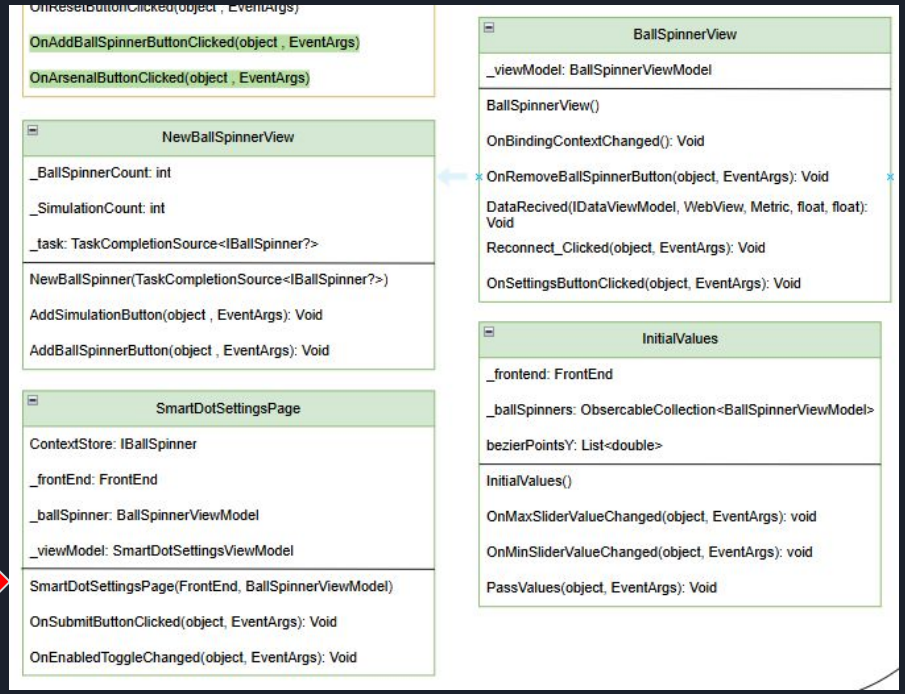
Not Yet Implemented

BSA Frontend

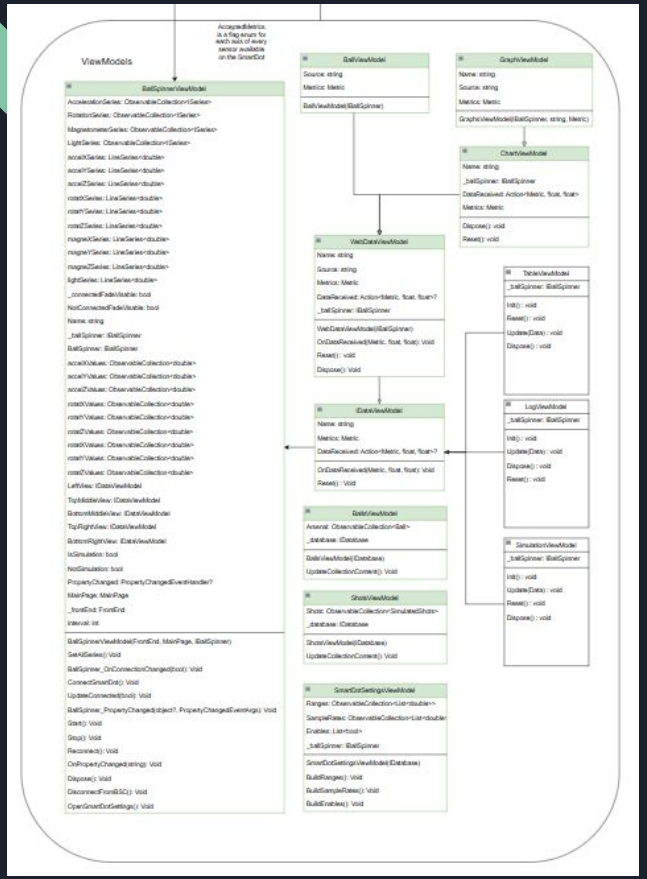


BSA Frontend





BSA Frontend

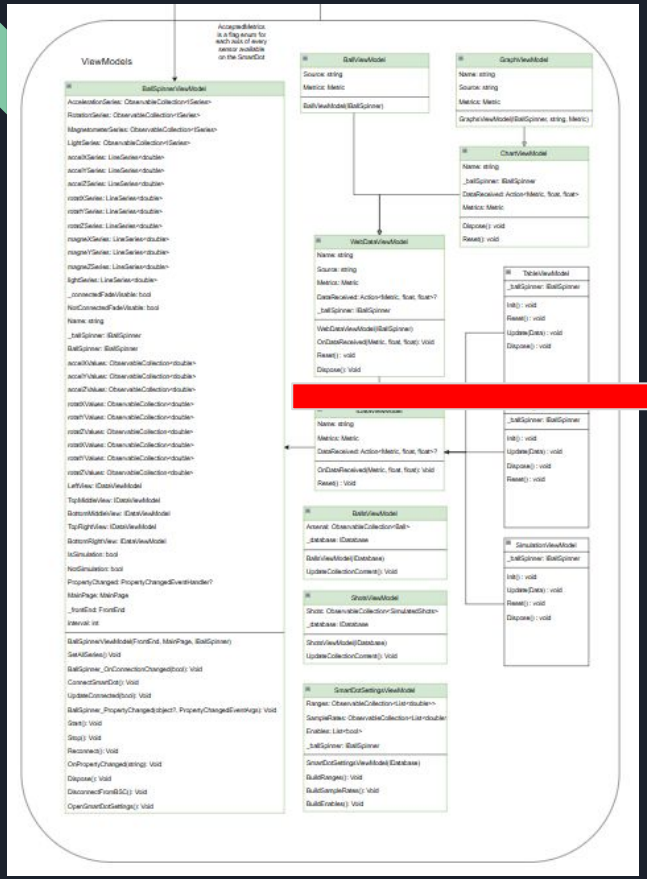


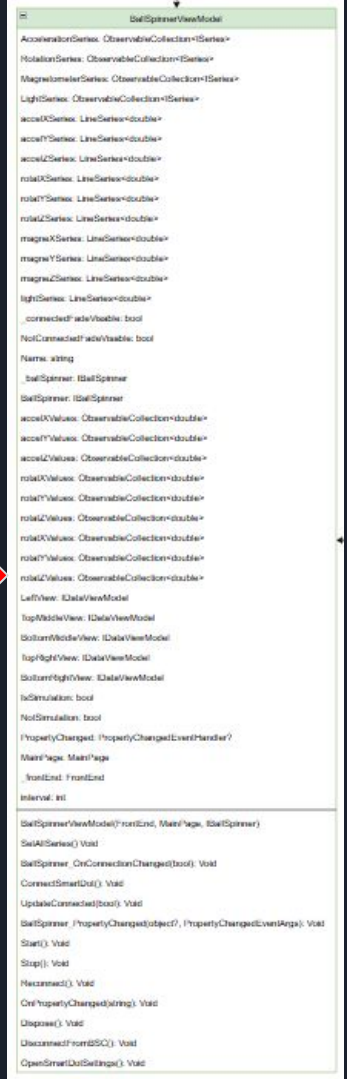
Implemented

Partially Implemented

Not Yet Implemented

BSA Frontend



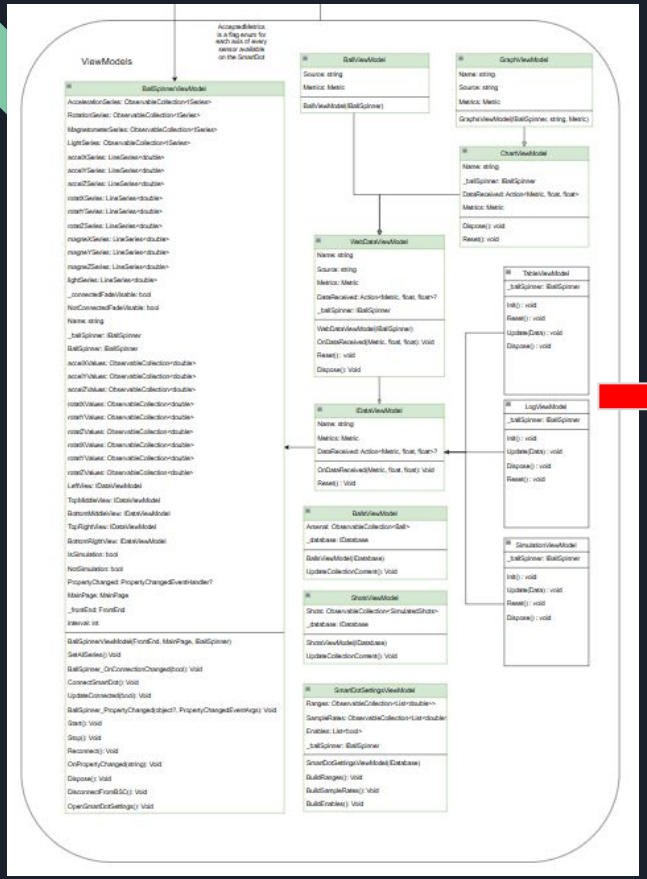


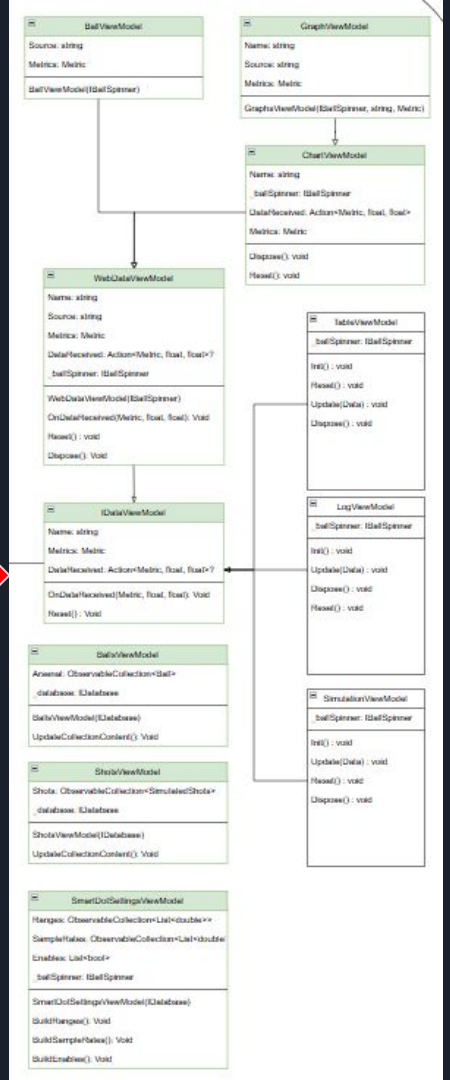
Implemented

Partially Implemented

Not Yet Implemented

BSA Frontend





Implemented

Partially Implemented

Not Yet Implemented

Protocol

1. Setup Connection Messages

a) A_B_INIT_HANDSHAKE: previously (APP_INIT)

Msg Type (0x01)	Msg Size (0x0001)	Random Start Byte (1 Byte)

b) B_A_INIT_HANDSHAKE_ACK: previously (APP_INIT_ACK)

Msg Type (0x02)	Msg Size (0x0001)	Repeated Random Byte (1 Byte)

c) A_B_NAME_REQ: previously (BSC_NAME_REQ)

Msg Type (0x03)

d) B_A_NAME: previously (BSC_NAME)

Msg Type (0x04)	Msg Size (2 Bytes)	ASCII NAME (Up to 255 Bytes)

2. Setup SmartDot Connection

a) A_B_START_SCAN_FOR_SD: previously (SMARTDOT_SCAN)

Msg Type (0x05)

b) B_A_SCANNED_SD: previously (SMARTDOT_SCAN)

Msg Type (0x06)	Msg Size (2 Bytes)	BLE MAC Address (6 Bytes)	Name in ASCII (Up to 255 Bytes)

c) A_B_CHOSEN_SD

Msg Type (0x07)	Msg Size (2 Bytes)	BLE MAC Address (6 Bytes)	Name in ASCII (Up to 255 Bytes)

d) B_A_RECEIVE_CONFIG_INFO

Msg Type (0x08)	Msg Size (2 Bytes) 0x0008	Accel Sample Rates (2 Byte)	Gyro Sample Rates (2 Byte)	Mag Sample Rates (2 Byte)	Light Sample Rates (2 Byte)

Protocol

3. Sending Run Data

a) A_B_RECEIVE_SD_CONFIG

Msg Type (0x09)	Msg Size (0x0004)	Set Config For SD 2 Bytes

b) A_B_SD_TOGGLE_TAKE_DATA

Msg Type (0x0A)	Msg Size (0x0001)	On or off byte (1 byte)

c) B_A_SD_SENSOR_DATA

Msg Type (0x0B)	Msg Size (0x0013)	Sensor Type (1 Byte)	Sample Count (3 Byte)	In-between Time (4 Byte)	"X-axis" data (4 Bytes)	"Y-axis" data (4 Bytes)	"Z-axis" data (4 Bytes)

d) A_B_MOTOR_INSTRUCTIONS

Msg Type (0x0C)	Msg Size (0x0003)	Motor1 Speed (4 Bytes)	Motor2 Angle (4 Bytes)	Motor3 Angle (4 Bytes)

4. Terminate Processes

a) A_B_STOP_MOTOR

Msg Type (0x0D)

b) A_B_DISCONNECT_FROM_BSC

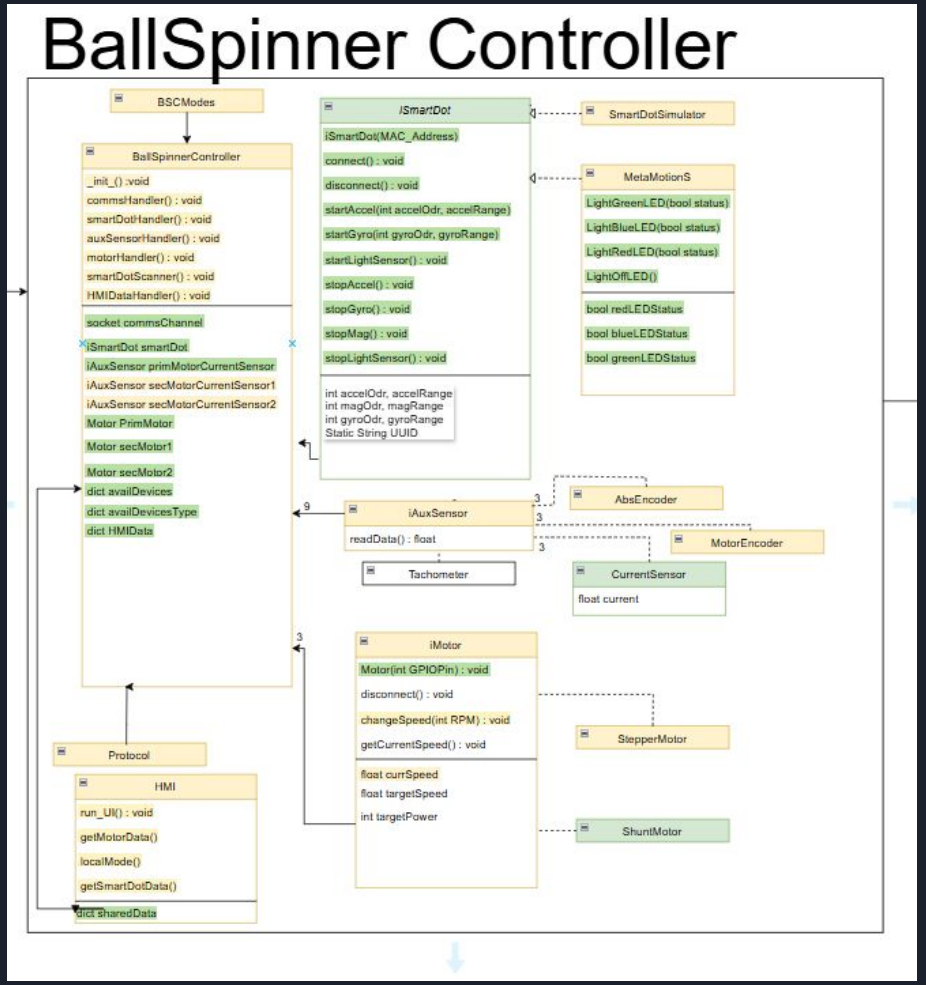
Msg Type (0x0E)

Ball Spinner Controller

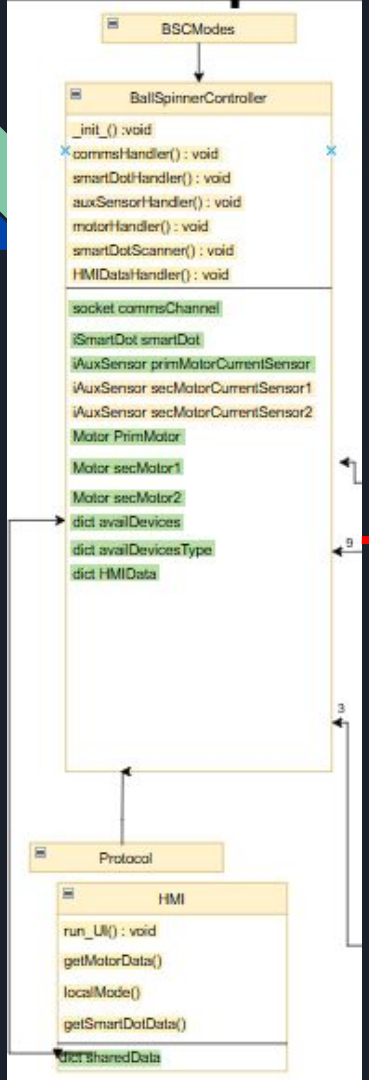
Implemented

Partially Implemented

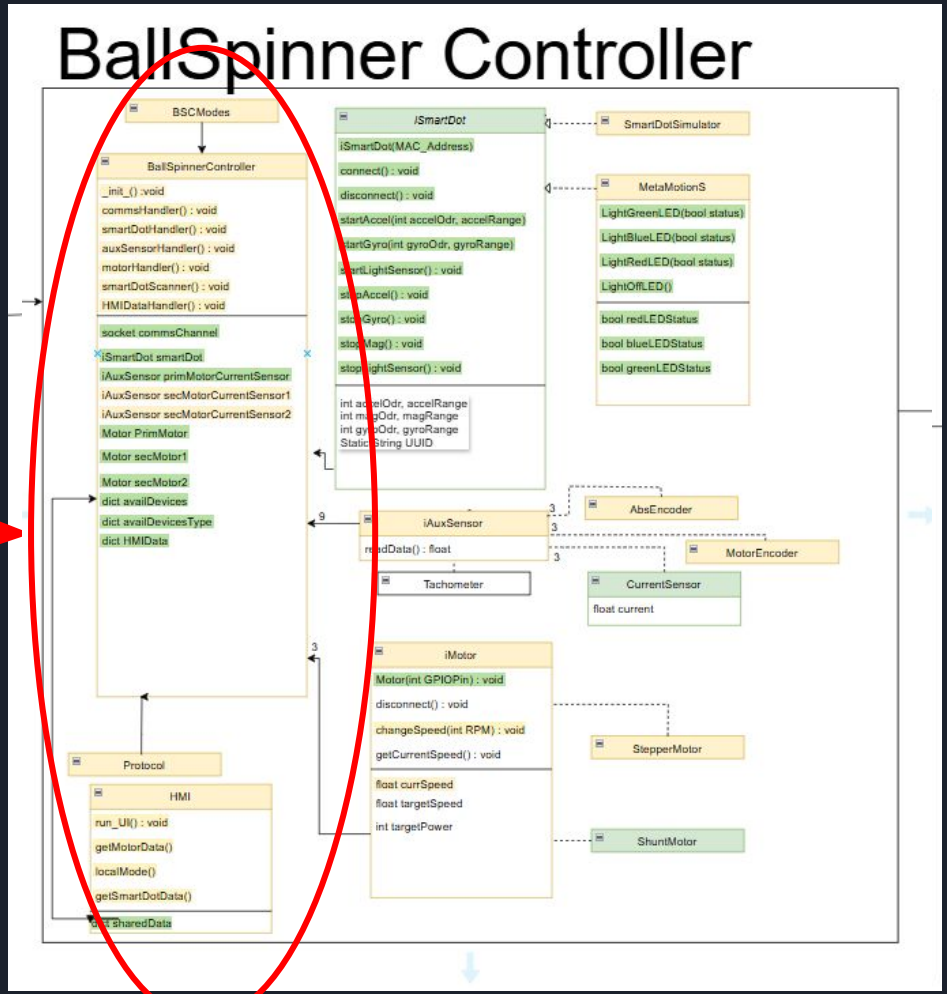
Not Yet Implemented



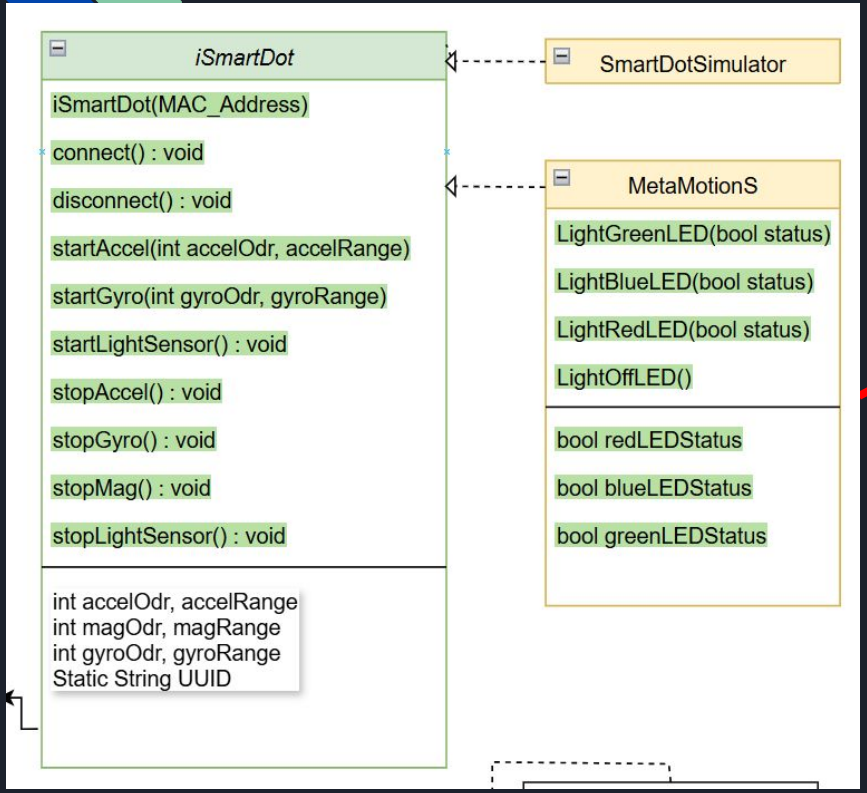
Current Design

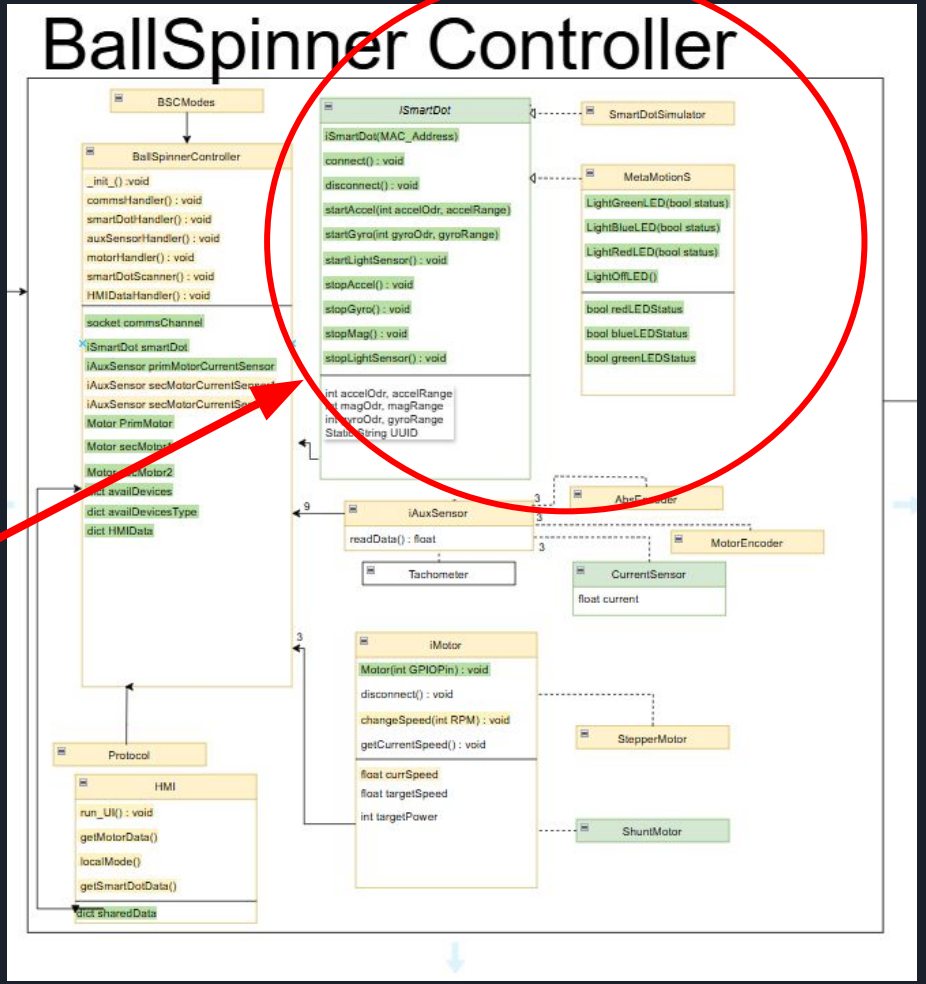


BallSpinner Controller

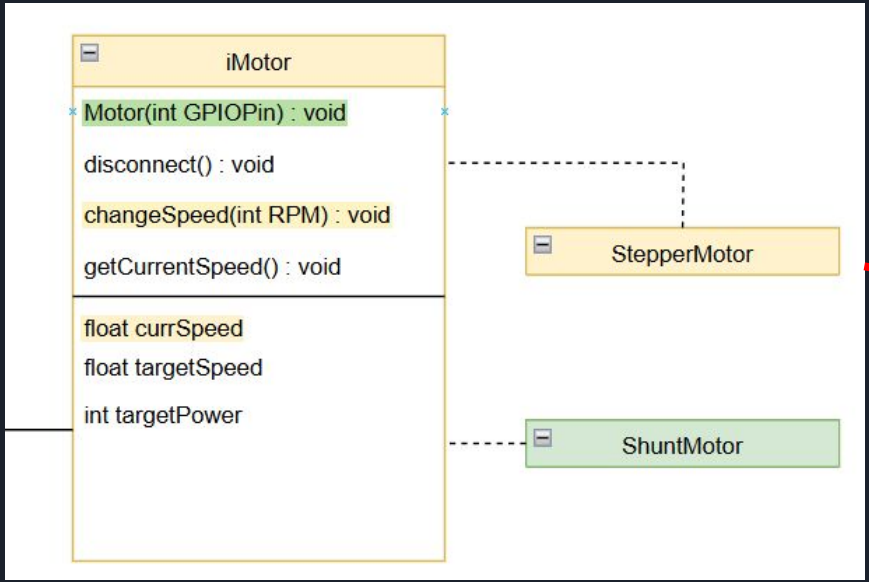


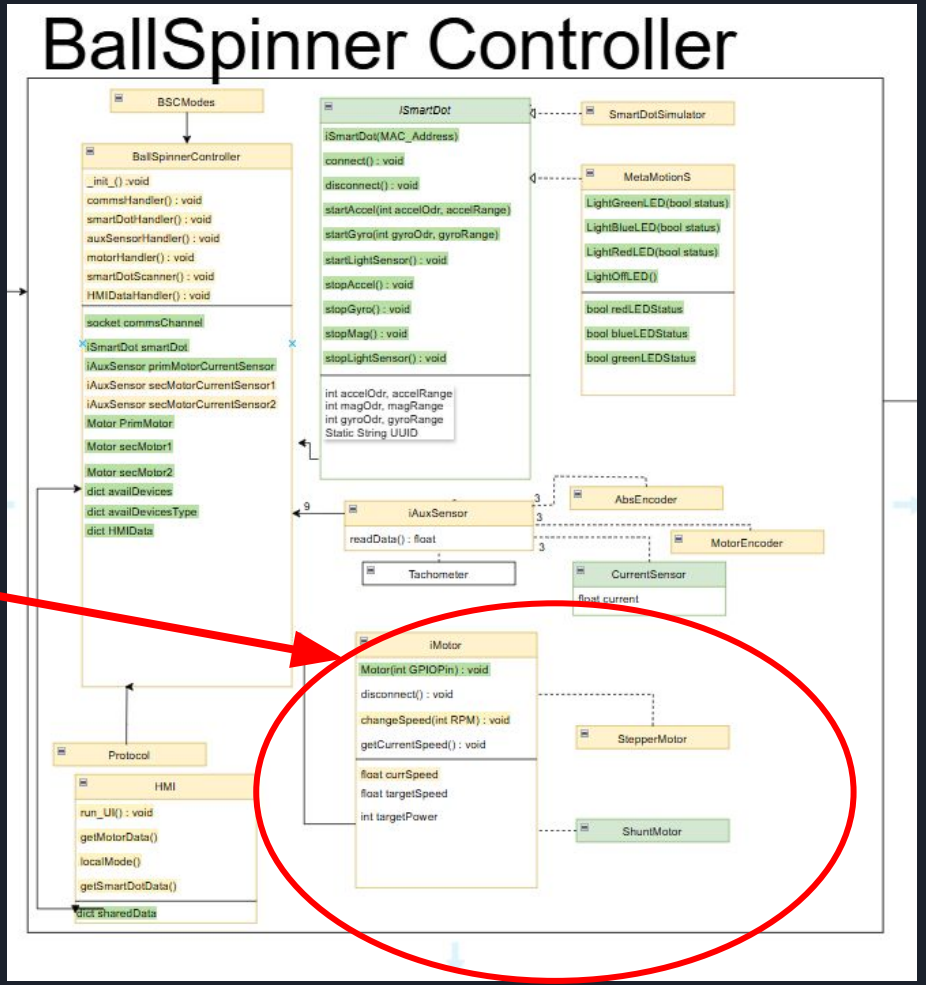
Current Design


iSmartDot
iSmartDot(MAC_Address)
connect() : void
disconnect() : void
startAccel(int accelOdr, accelRange)
startGyro(int gyroOdr, gyroRange)
startLightSensor() : void
stopAccel() : void
stopGyro() : void
stopMag() : void
stopLightSensor() : void
int accelOdr, accelRange
int magOdr, magRange
int gyroOdr, gyroRange
Static String UUID
SmartDotSimulator
MetaMotionS
LightGreenLED(bool status)
LightBlueLED(bool status)
LightRedLED(bool status)
LightOffLED()
bool redLEDStatus
bool blueLEDStatus
bool greenLEDStatus


BallSpinner Controller
BSCModes
BallSpinnerController
init() :void
commsHandler() : void
smartDotHandler() : void
auxSensorHandler() : void
motorHandler() : void
smartDotScanner() : void
HMIDataHandler() : void
socket commsChannel
iSmartDot smartDot
Motor PrimMotor
dict availDevices
dict availDevicesType
dict HMIData
iSmartDot
iSmartDot(MAC_Address)
connect() : void
disconnect() : void
startAccel(int accelOdr, accelRange)
startGyro(int gyroOdr, gyroRange)
startLightSensor() : void
stopAccel() : void
stopGyro() : void
stopMag() : void
stopLightSensor() : void
SmartDotSimulator
MetaMotionS
LightGreenLED(bool status)
LightBlueLED(bool status)
LightRedLED(bool status)
LightOffLED()
bool redLEDStatus
bool blueLEDStatus
bool greenLEDStatus
iAuxSensor
readData() : float
Tachometer
MotorEncoder
CurrentSensor
float current
iMotor
Motor(int GPIOPin) : void
disconnect() : void
changeSpeed(int RPM) : void
getCurrentSpeed() : void
float currSpeed
float targetSpeed
int targetPower
StepperMotor
ShuntMotor
Protocol
HMI
run_UI() : void
getMotorData()
localMode()
getSmartDotData()
dict sharedData

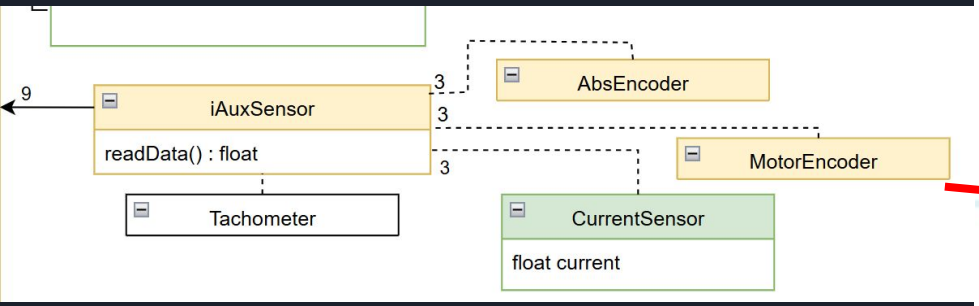
BallSpinner Controller

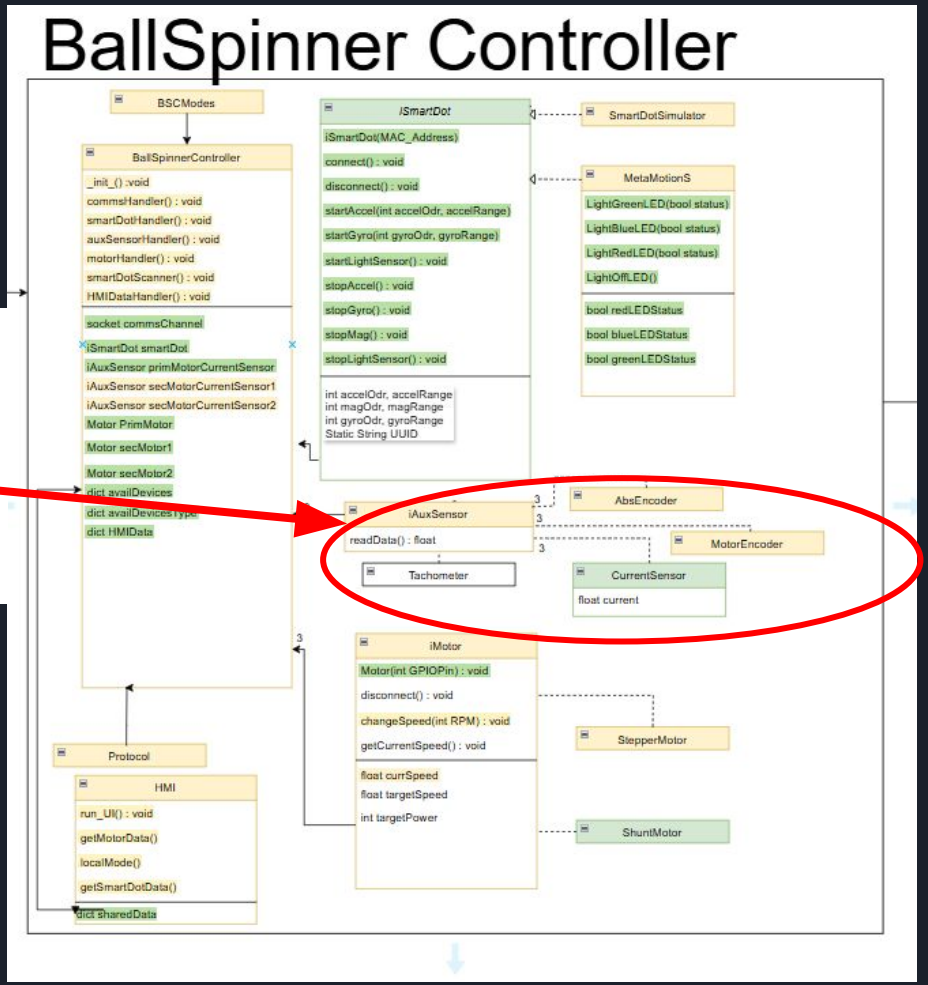




Current Design

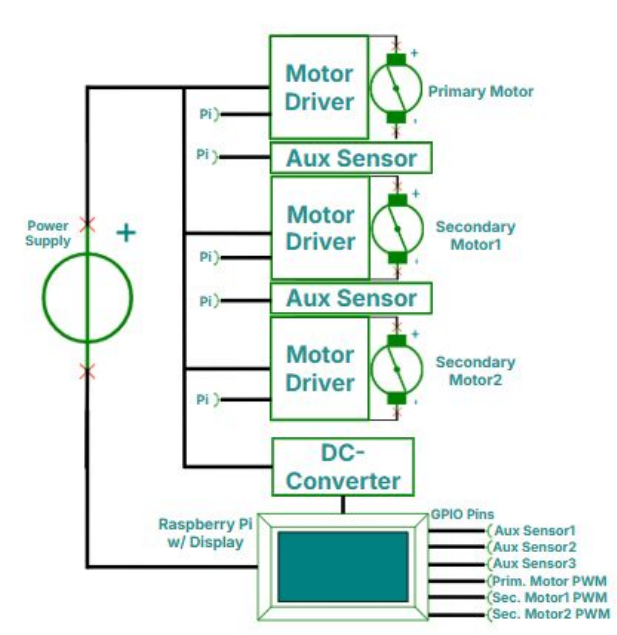
BallSpinner Controller



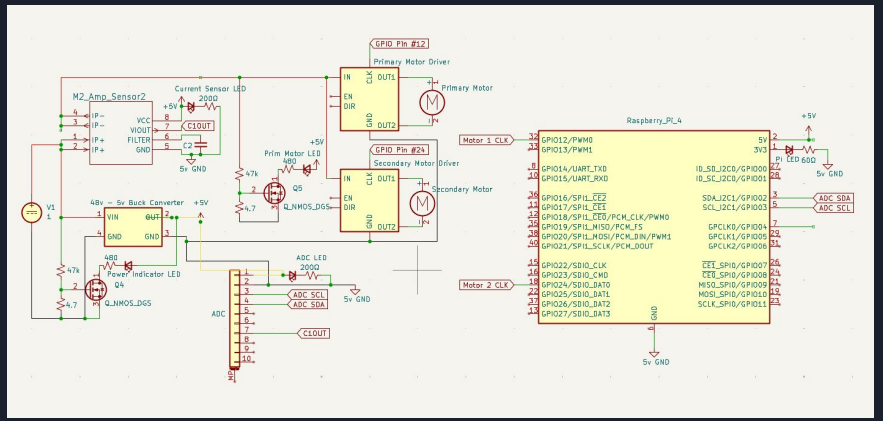


Current Design

Ball Spinner Controller

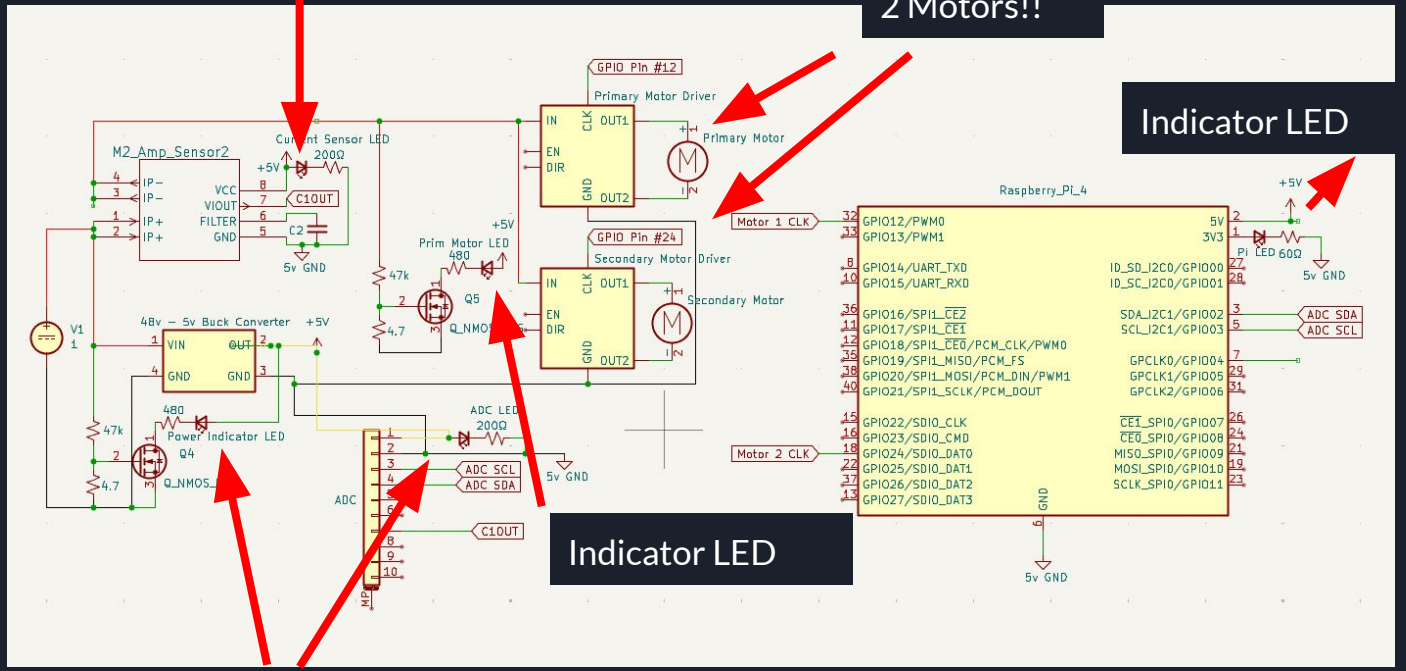


Stepper Motor Electrical Block Diagram



Current Electrical Schematic

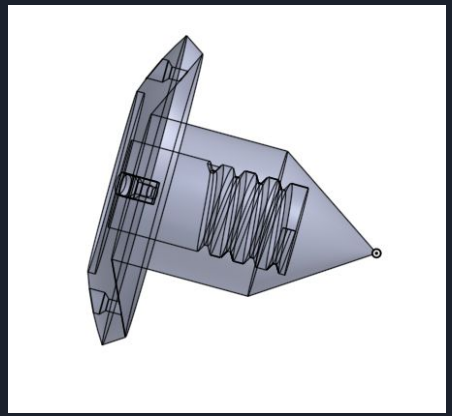
Ball Spinner Controller

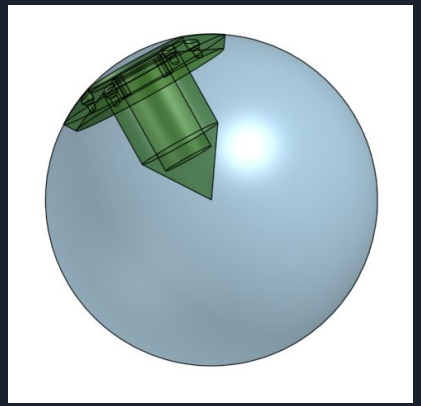


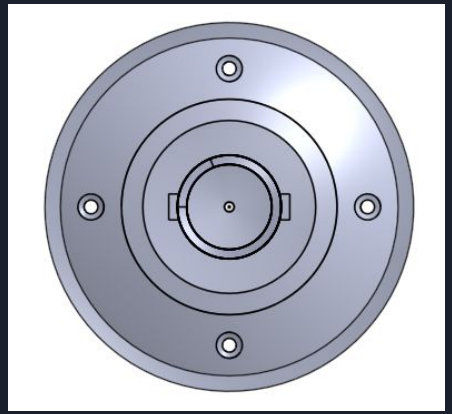
Current Design

Ball Spinner Team

- Print and implement SmartDot module insert
  - Attaches to styrofoam test ball
  - SmartDot container will screw in axin to a lightbulb to resist centripetal
- 2nd and 3rd DoF axis of rotation
  - A miniature showcase of a current manuel design for the 2nd and 3rd DoF







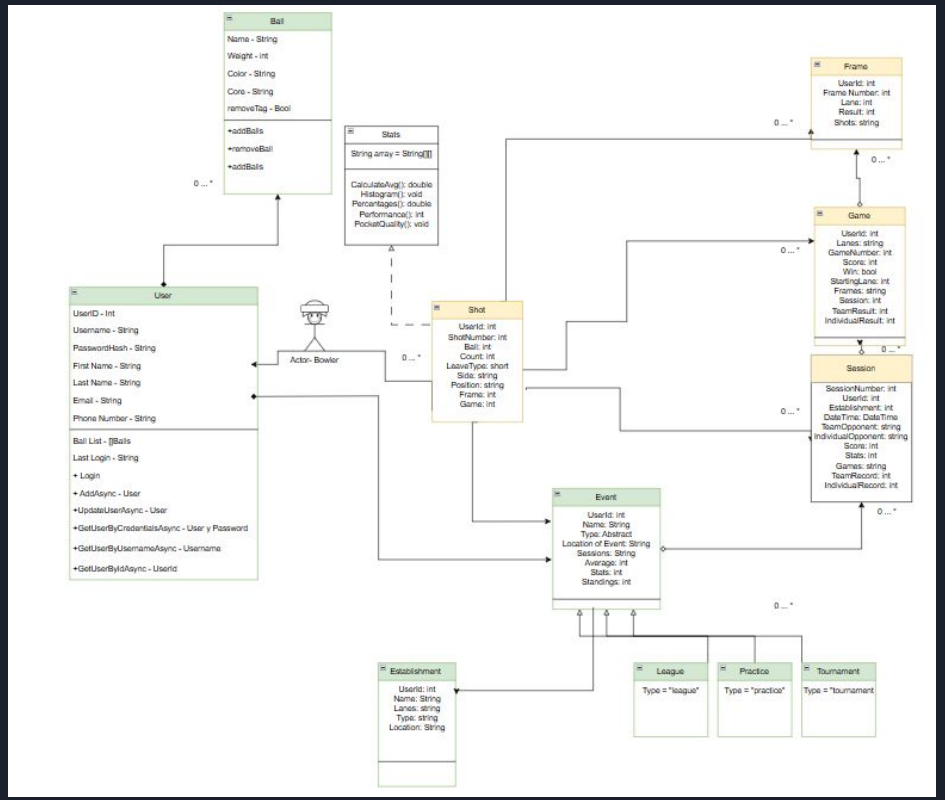
Current Design

Mobile Application Team

Implemented

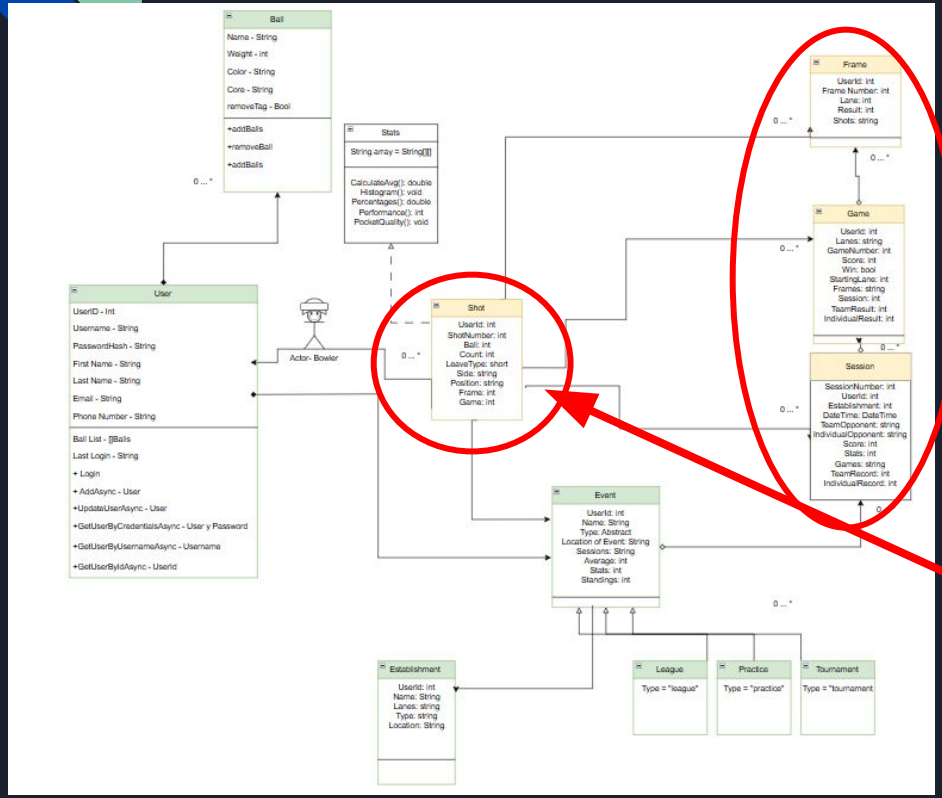
Partially Implemented

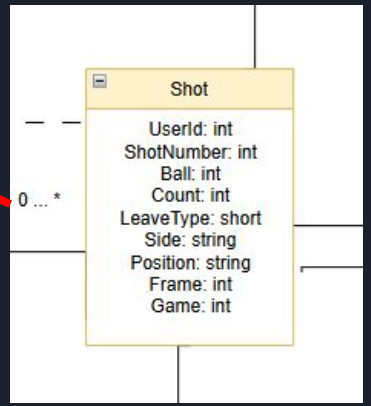
Not Yet Implemented

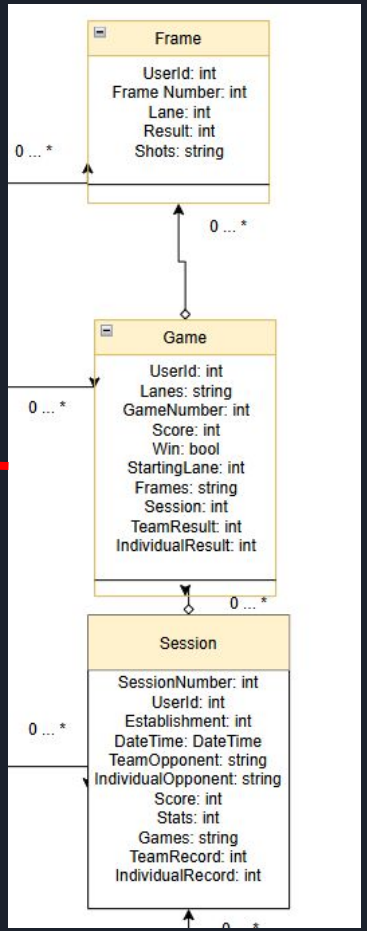


Current Design

Mobile Application Team

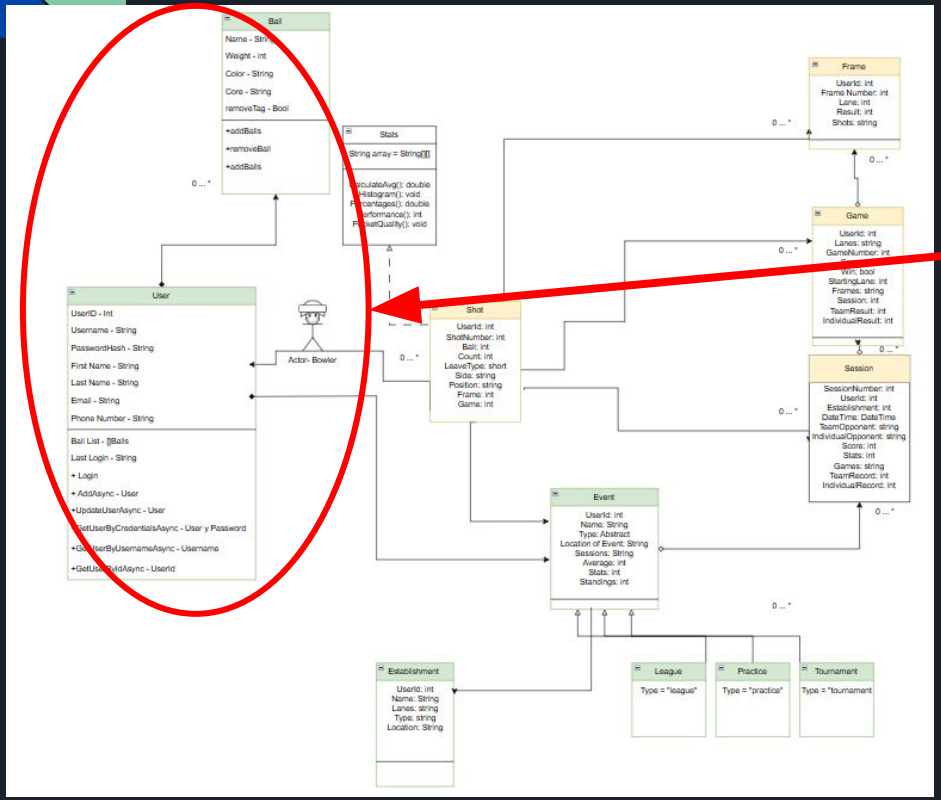


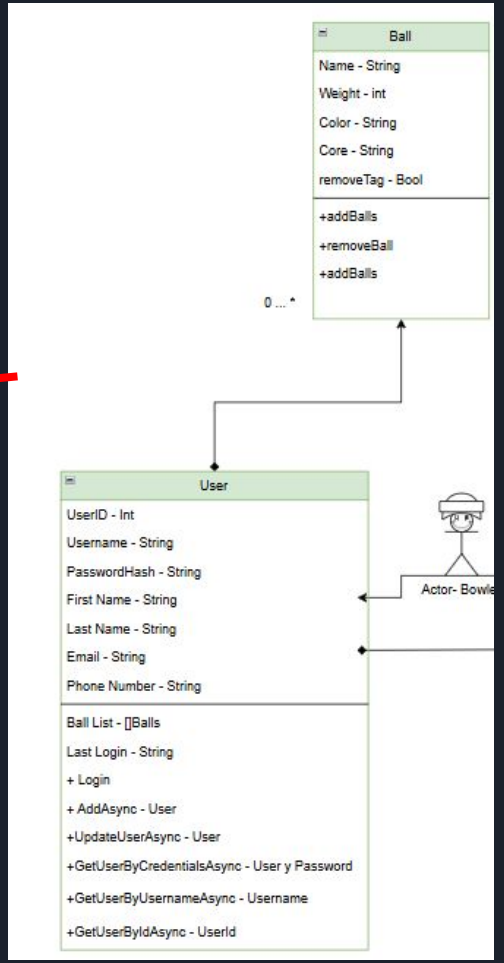




Current Design

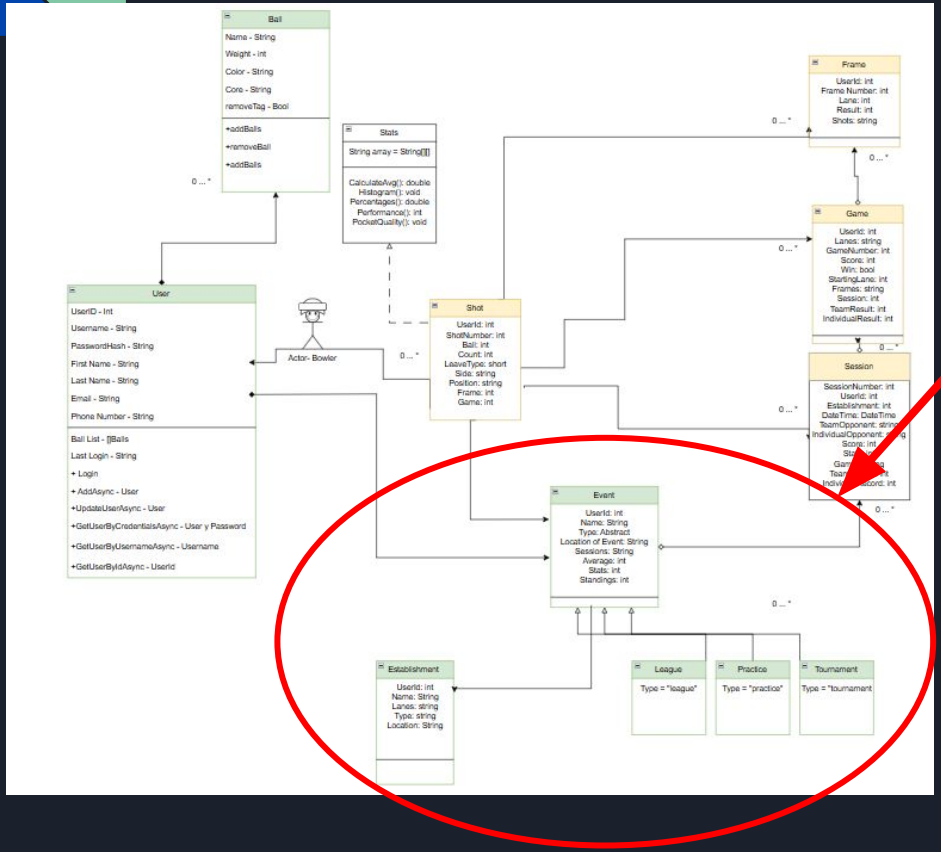
Mobile Application Team

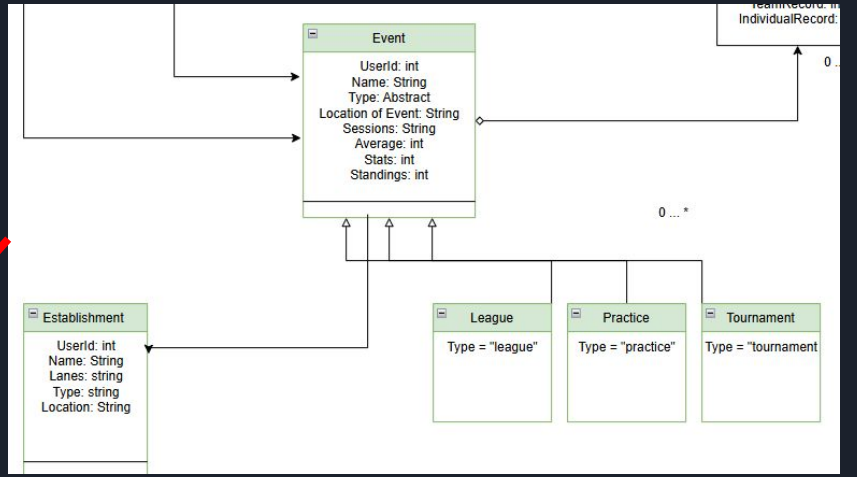




Current Design

Mobile Application Team



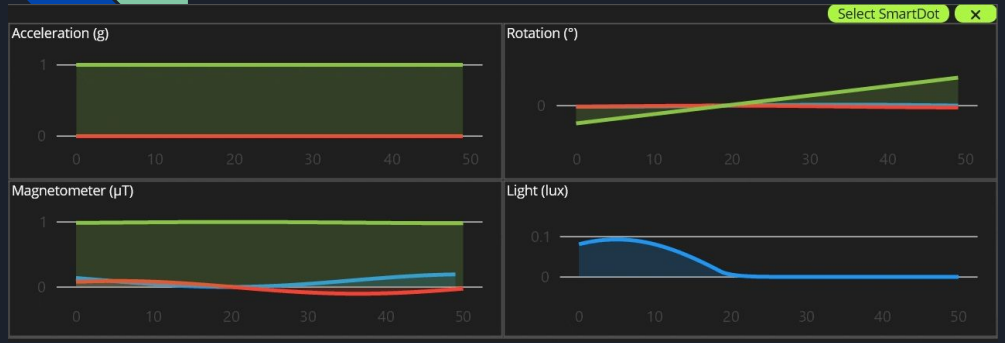


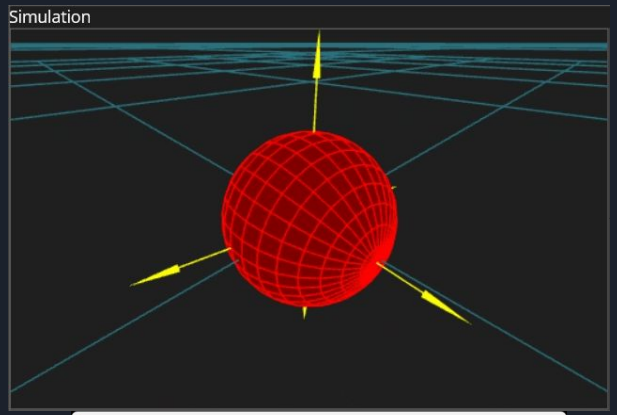
Current Design

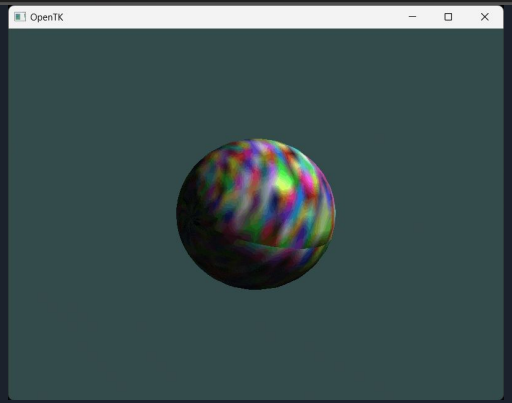
Questions?

Current Implementation

Simulation (MS3 Goals)





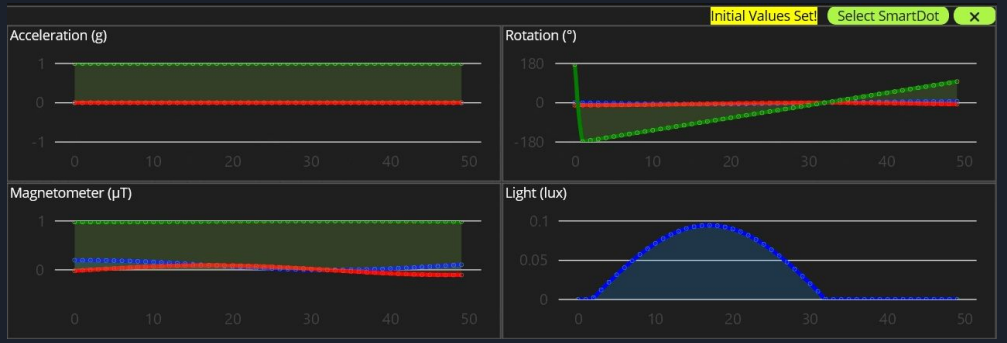


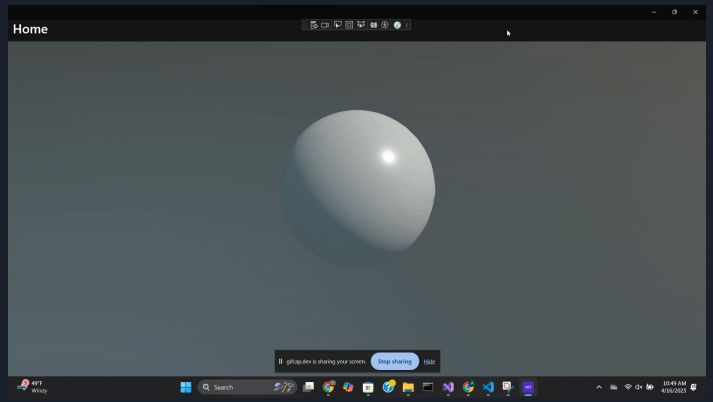
- Port new simulation into BSA
- Axis and units for graphs
- Bezier inflection point automatic movement/controlled movement
- Update the Unity game to work with the current version of MAUI

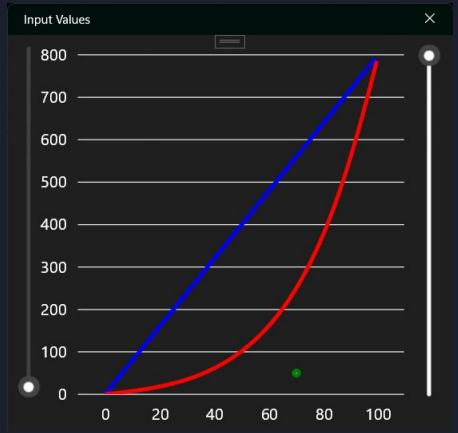
Current Implementation

Simulation (MS3 Achievements)

- Graphs updated (again!)
  - Y Axis forced
  - Timer for X Axis framework
- Inflection point settable
  - Font and color update
- Ball visualization switch from Three.js
  - OpenTk
  - Evergine
- Three.js moving with Motor instructions and encoders



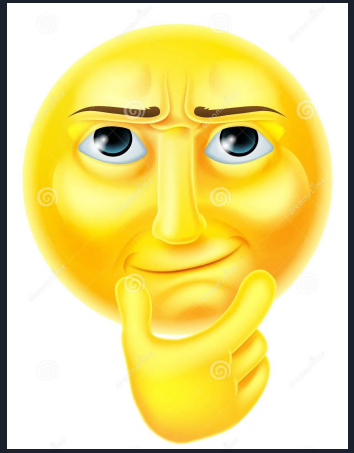




Current Implementation

BSA Backend/Cloud (MS3 Goals)

- Motor instructions
- Replace CSV with memory mapped files
- Save motor encoder data in database
- Replay shot feature with data from database
- Work with mobile team to get mobile schema in cloud



BSA Backend/Cloud (MS3 Achievements)

- Memory mapped files replace CSV for caching SD data
- Encoder data/current sensor data can saved/sent to Simulation
- Simulation spins based on initial RPMs
- Secondary axis motor instructions can be sent to BSC
- Worked with mobile team to create comprehensive cloud database schema

Size	Type	Size	Time	Size	Sample Count	Size	X	Size	Y	Size	Z

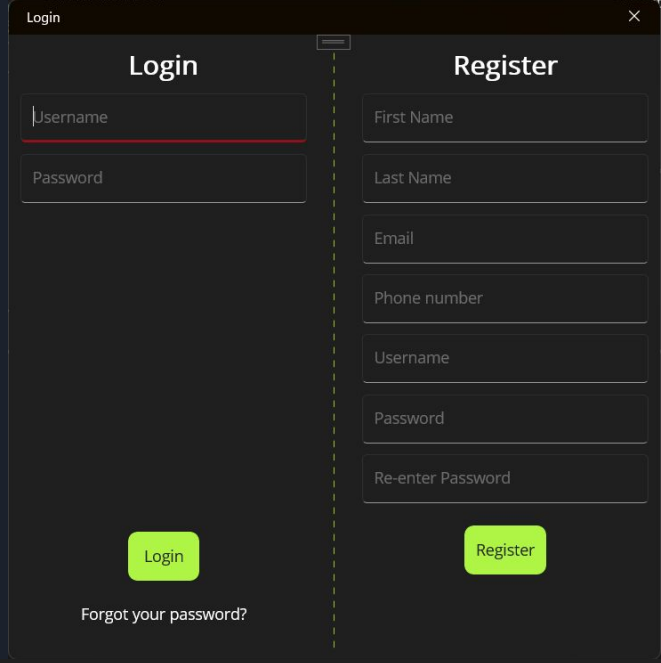
BSA Frontend (MS3 Achievements)

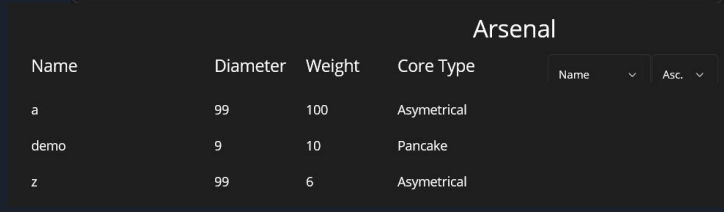
Goals

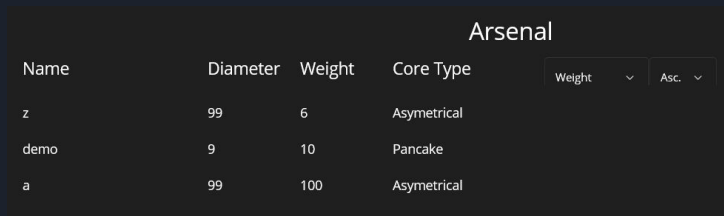
- Update Registration page
- Sort Arsenal page and Cloud Management page
- Prepare the Ball Spinner Application frontend for next semester

Achievements

- (Mostly) Revised Login/Registration page
- Sorting on Arsenal & Cloud Management pages
- Continued work on in-code documentation







Ball Spinner Controller (MS3 Goals)

- Improve encoder accuracy with Motor encoder
- Integrate encoders into system
- Finalize PCB Traces and print for Expo (Find Relays and fuses)
- Restructure Protocol Messages (Msg Types)

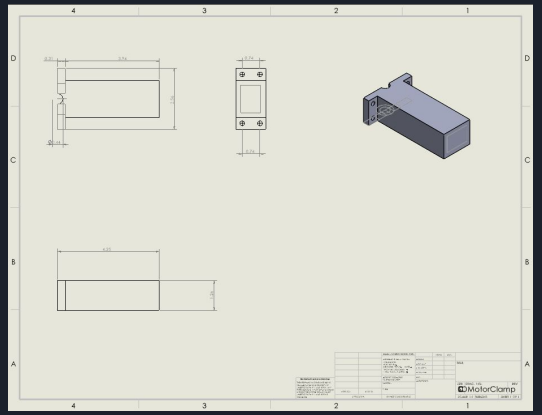
Current Implementation

Ball Spinner Controller (MS3 Achievements)

- Motor Encoders in final stages of integration process

- PCB ordered for Expo:
  - 2 Motor Drivers
  - 2 Motor Encoders

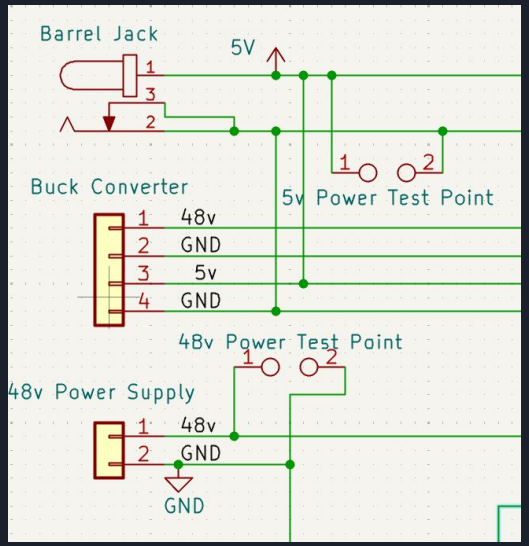
- Design Prototype MMS motor clamp



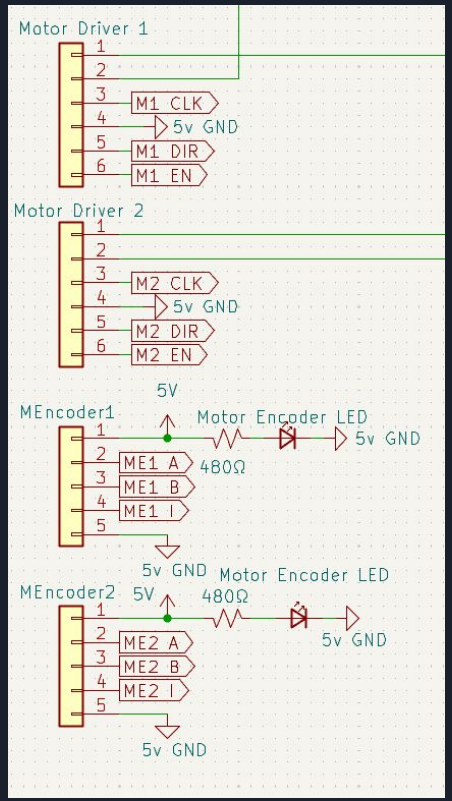
$$Encoder\ RPM = \frac{Number\ of\ A\ Signals}{Time\ Between\ each\ Impulse\ (Seconds) * CPR} * \frac{60\ Seconds}{1\ minute}$$

$$Software\ Indexed\ Encoder\ RPM = \frac{Number\ of\ A\ Signals}{CPR} * \frac{60\ Seconds}{1\ minute}$$

Ball Spinner Controller (PCB Features)



Feeds in both 48v and 5v



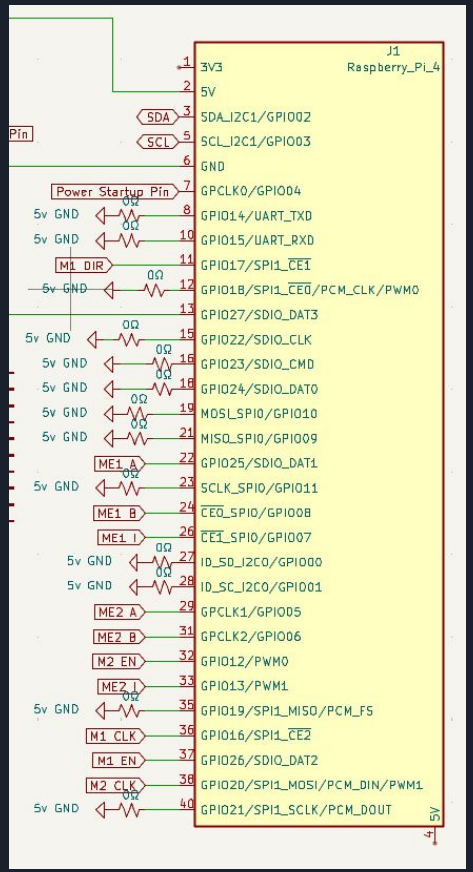
Allows for:

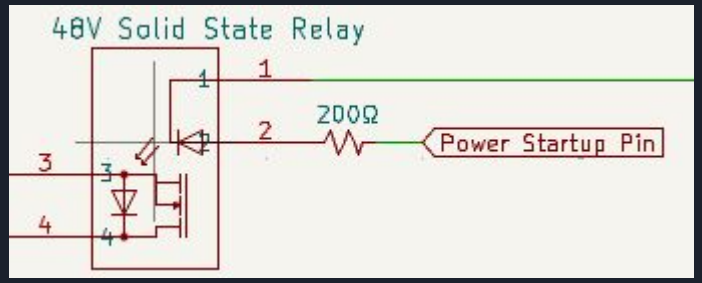
- 2 Motor Control
- 2 Current Sensor Readings
- 2 Motor Encoder Readings

Current Implementation

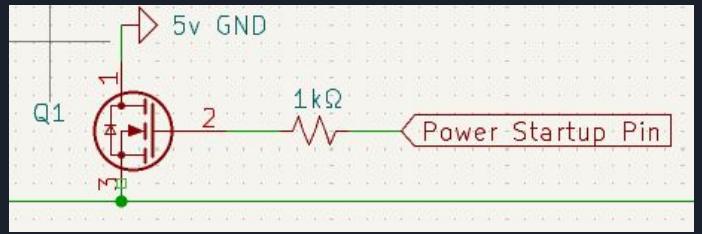
Ball Spinner Controller (PCB Features)

Open GPIO Pins for future implementation



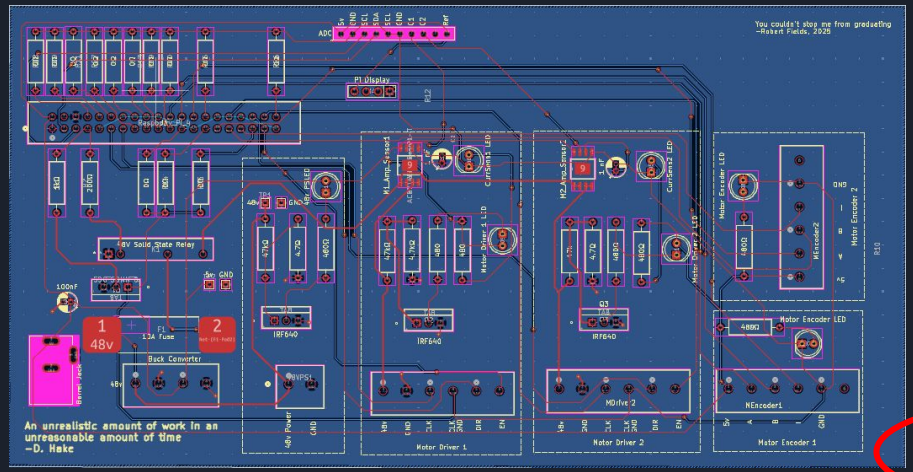


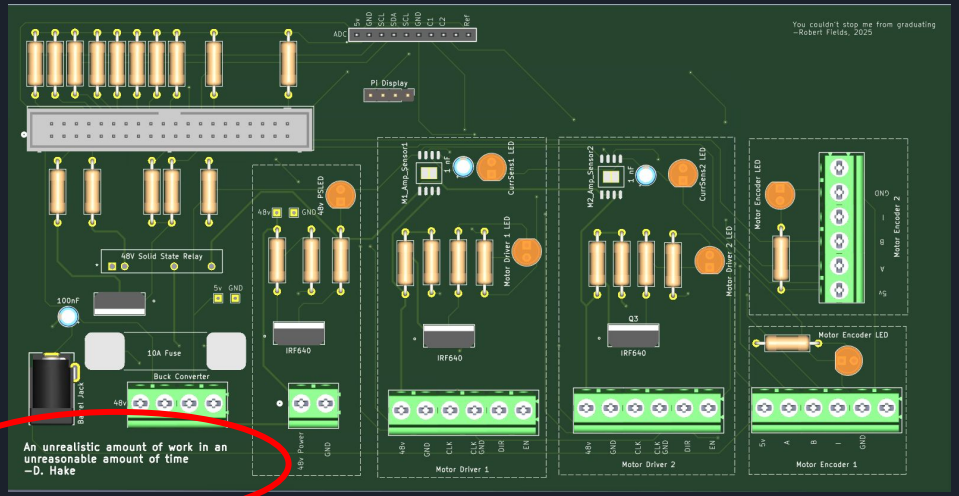
Has Pi control Power Startup



Current Implementation

Ball Spinner Controller (PCB)





An unrealistic amount of work in an unreasonable amount of time
–D. Hake

Current Implementation

Ball Spinner Controller (MS3 Achievements)
HMI

- Motor Encoder Value appears in the HMI (Local + BSC)
- SmartDot Connection in Local Mode
- SmartDot Configuration in Local Mode
- SmartDot Graphing in Local Mode

Ball Spinner Controller HMI

plementation

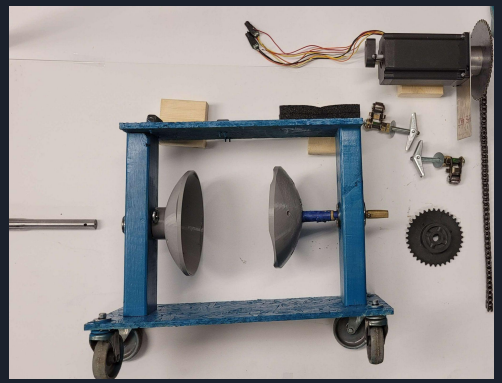
Current Implementation

Current Implementation

Ball Spinner Team

Proposal

- Create miniature 2nd and 3rd DoF prototype
- Print and implement the SmartDot module insert
- Finalize preparations for the Capstone Expo



Reality

- Finished printing of 2nd (axle) cup to stabilize 1st DoF
- Drilled access hole for wiring through the safety box
- Finished miniature 2nd and 3rd DoF prototype
- Created designs for the SmartDot module insert
- Painted parts for the Capstone Expo
- Clamping axle cut, finished, and mounted
- Cut out RevMetrix Emblem (to be mounted)

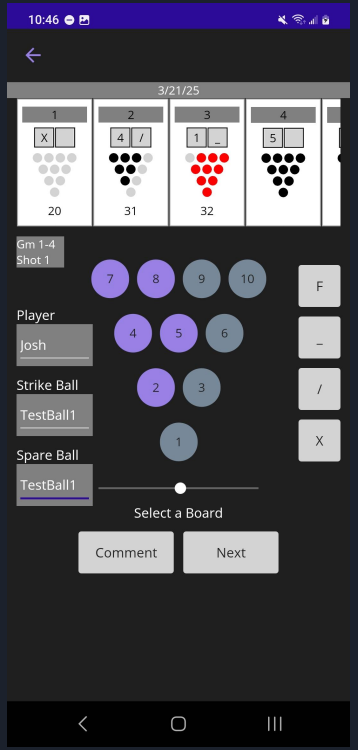
Mobile Application Team

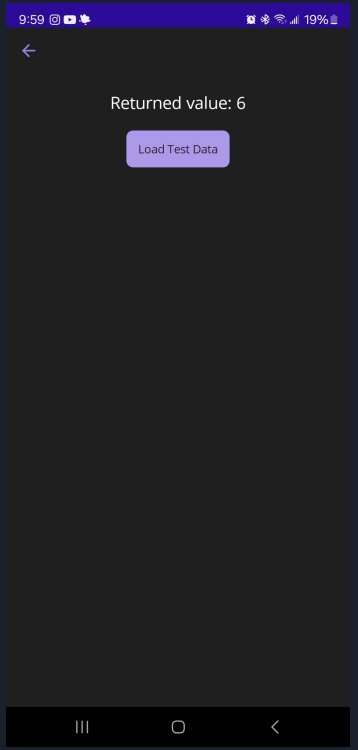
MS3 Goals

- Improved Game functionality
- Connection to Cloud Database
- Bluetooth/MMS (Stretch Goal)

MS3 Achievements

- Improved game functionality
- Connection to Cloud Database

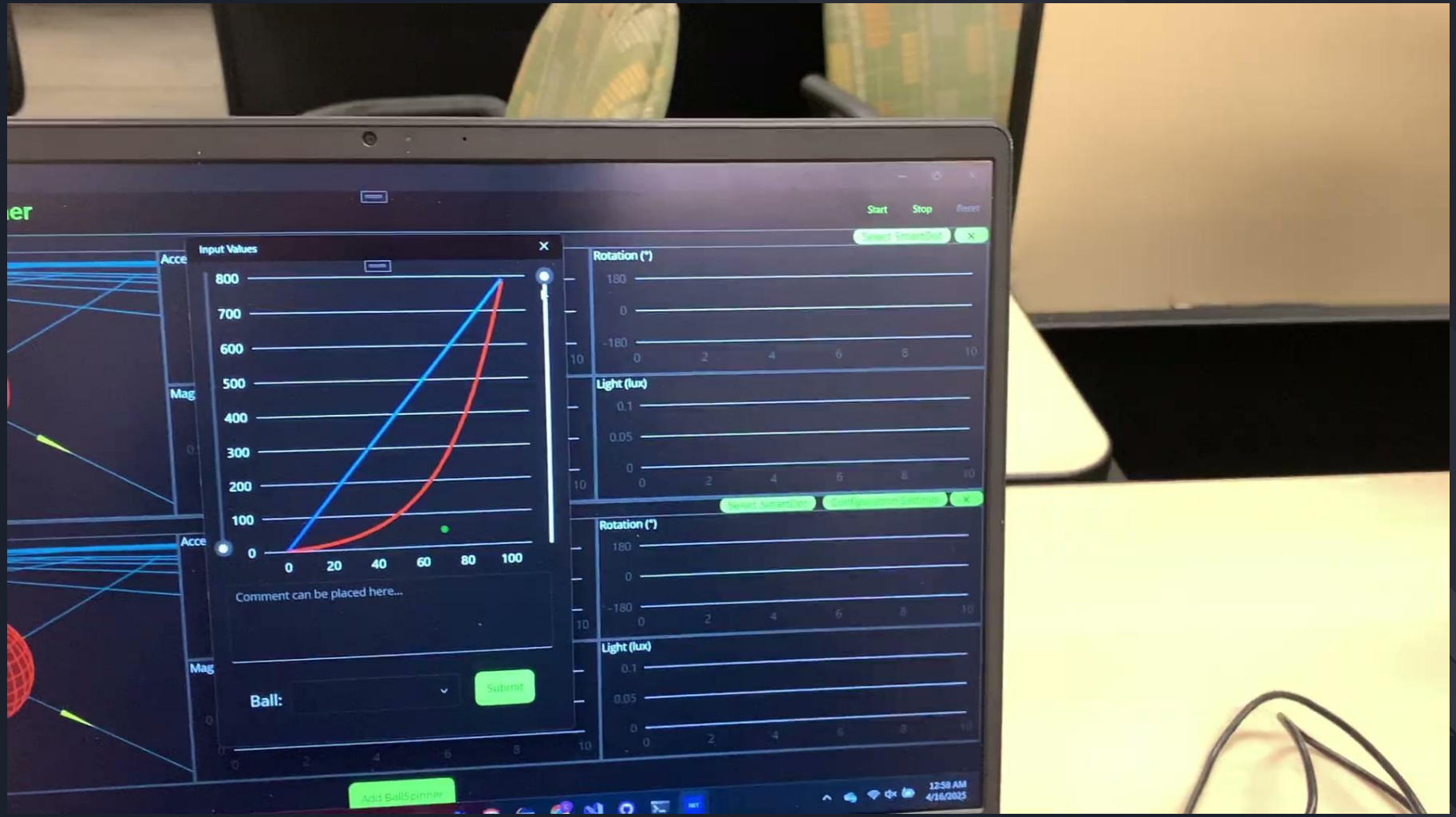


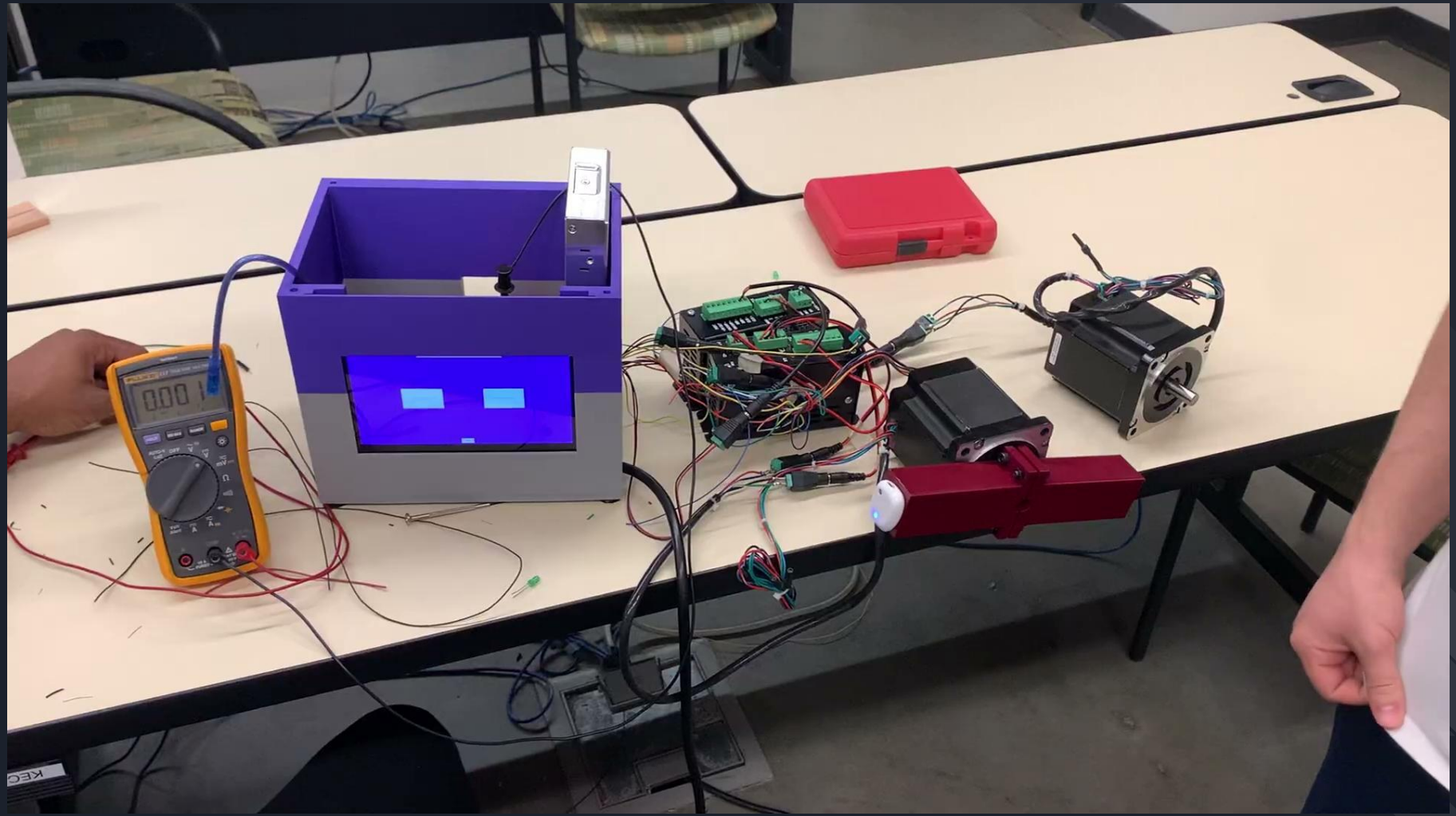


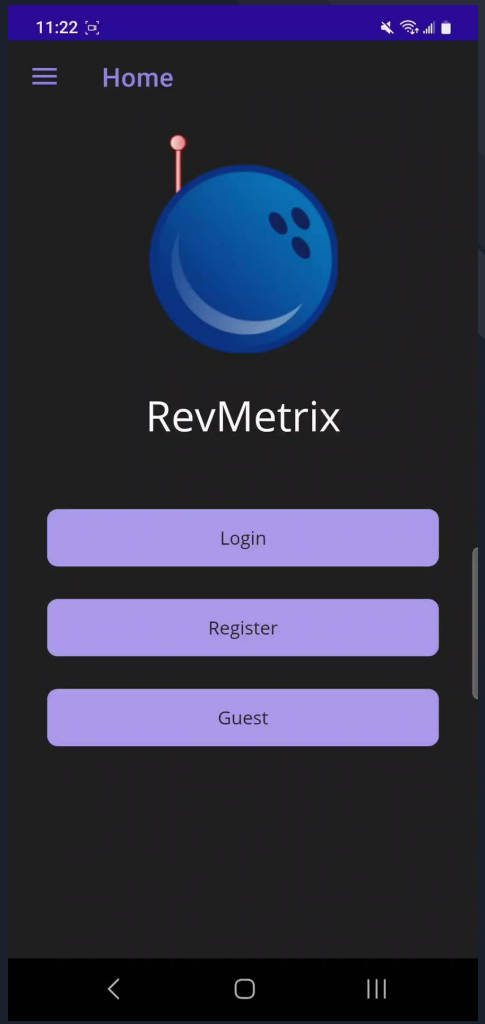
Current Implementation

Questions?

Demos


Start
Stop
Input Values
800
700
600
500
400
300
200
100
0
0 20 40 60 80 100
Comment can be placed here...
Ball:
Submit
Rotation (°)
180
0
-180
Light (lux)
0.1
0.05
0
Add BallSpinner
12:58 AM
4/16/2025




11:22
Home
RevMetrix
Login
Register
Guest

Mechanical Team Demo

Future Implementations

Simulation

- Implement timer
- Slight bezier calculation refactor
- Update the Unity game to work with the current version of MAUI

BSA Backend/Cloud

- Testing
- Bug fixes
- Documentation
- Poster
- Removing diameter from Ball POCO

BSA Frontend

- Prepare for the expo
  - Poster
  - Testing
- Continue Documentation
- Fix Register

Ball Spinner Controller

- Integrate PCB
- Determine Motor Encoder issue
- Add Reverse Rotation of Second Motor

Ball Spinner Controller HMI

- Circular buffer on the protocol history
- Reposition local SmartDot/make local mode layout cleaner.
- Add popup windows to the back button.
- Add optimizations to the graphing + fix GY data.

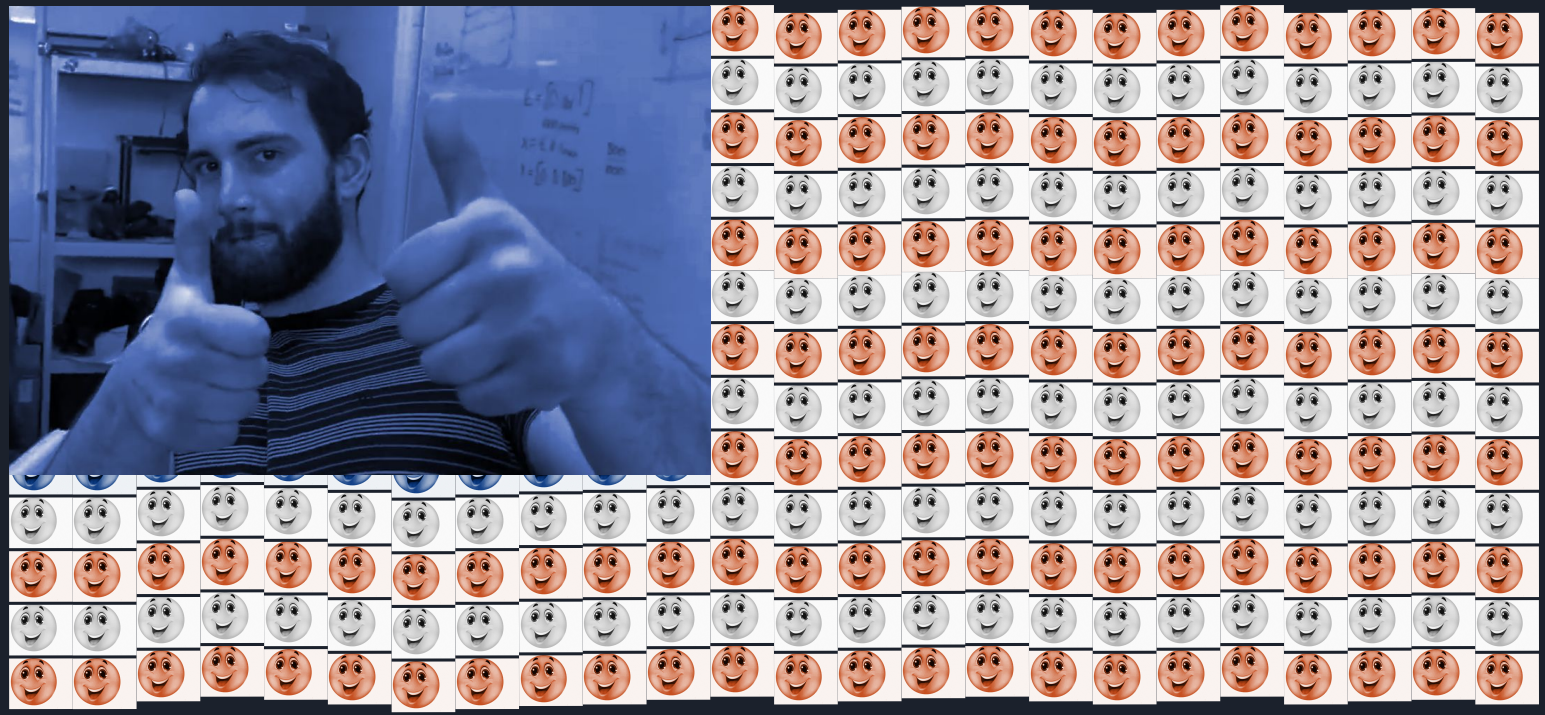
Ball Spinner Team

- Visualize full physical system
- Finalize design for insert for smart module
- Capstone expo
  - Finalize poster

Mobile Application Team

- Testing/ Bug fixes
- Wiki updates
- Capstone expo
  - Finalize poster

Questions?



Please let Robert Fields graduate on May 17, 2025