

Florida International University

Fall 2023 Capstone 2 Project

SkillCourt

Students: *Johnny Casares*

Mentor: *Gummi Oxndal, Product Owner*

Instructor/Faculty: *Masoud Sadjadi*, Florida International University

PROBLEM

The SkillCourt mobile application faced challenges due to incomplete development and UI implementation errors, characterized by redundant code and unexpected behavior. Notably, the UI element code required abstraction for enhanced codebase usability. Addressing navigation methods was crucial to enhance both user experience and performance. Noteworthy features that needed implementation included Play Set Up, enabling players to customize game settings, and Timer Page, where players could test their skills against time.

CURRENT SYSTEM

The app is made in Dart using Flutter, leveraging its capability to facilitate the development of a unified codebase targeting multiple platforms, ensuring a consistent UI across them all. This approach not only streamlines development but also offers significant time savings for developers. Substantial progress has been made, bringing the app closer to its final product.

VERIFICATION

To verify the process, we made sure the new features closely resembled the vision established in the Figma designed provided. I successfully implemented the UI for the Play Set Up and each of its parameters, including the new and saved game sections; and the Play Timer page. Additionally, I created a provisional database helper that allows the user to save the games that are created.

SUMMARY

In this project, I played a key role in the initial research and decision-making process for the chosen communication protocol, which resulted in the adoption of WebSockets. I contributed to code optimization by reducing boilerplate and implementing design patterns such as singletons and fragments, thereby enhancing code readability and maintainability. A significant focus was placed on navigation improvements, where I utilized my expertise in Java Android development to engineer an efficient bottom navigation bar, minimizing unnecessary UI reloads for smoother transitions between pages. Additionally, I created a route generator class to streamline route management and parameter handling, reducing redundancy in the main function. In the context of Play Set Up, I developed a comprehensive feature allowing users to configure game settings and save setups for future use. Although the Play Timer implementation is in progress, foundational work has been laid out, with basic UI elements in place and plans for further method implementation.

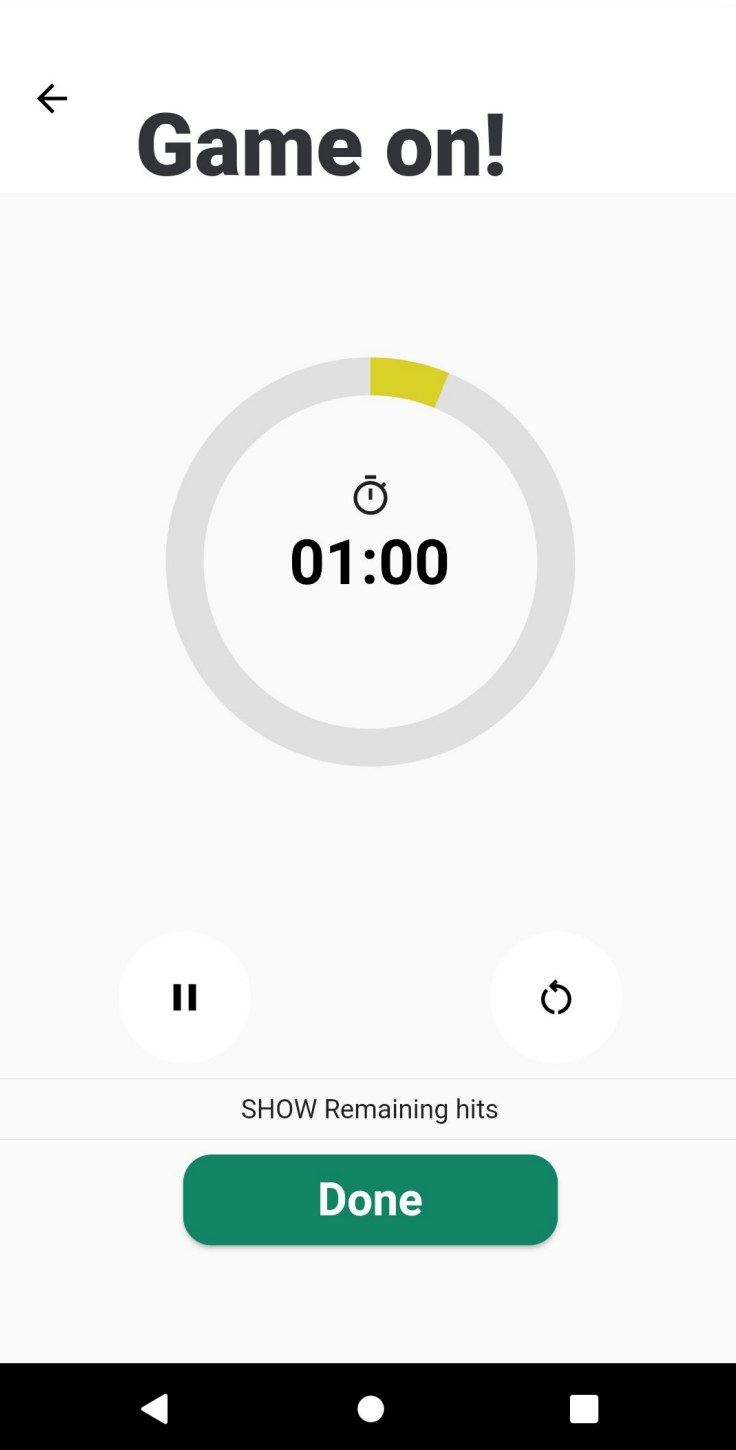
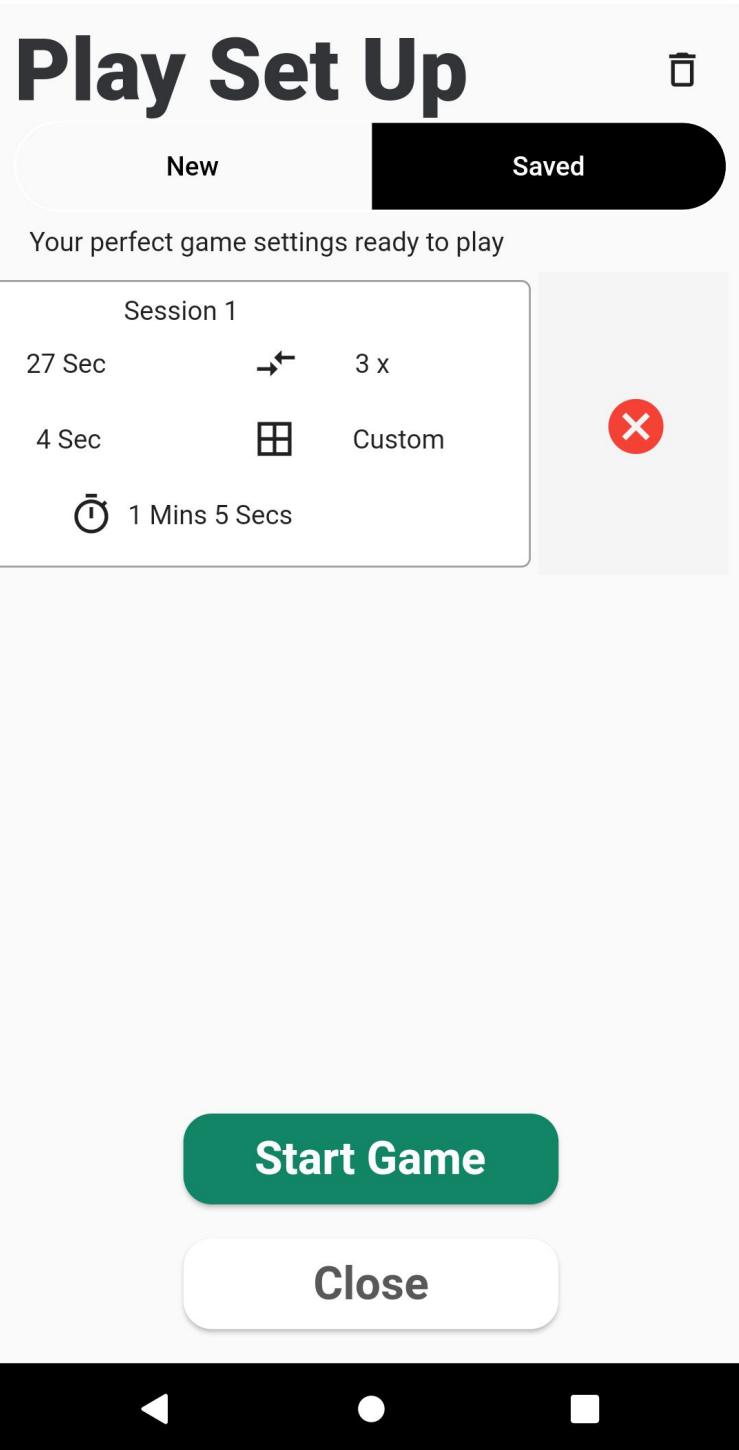
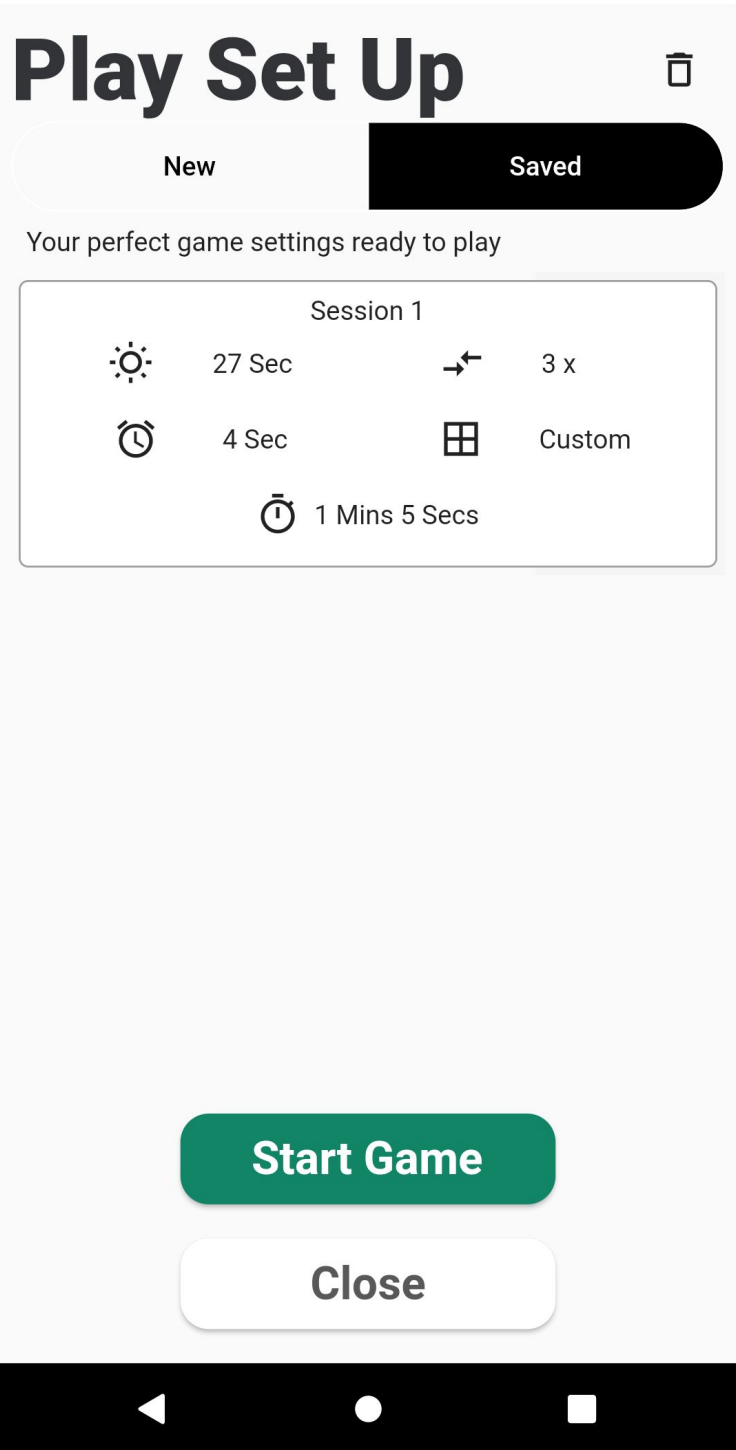
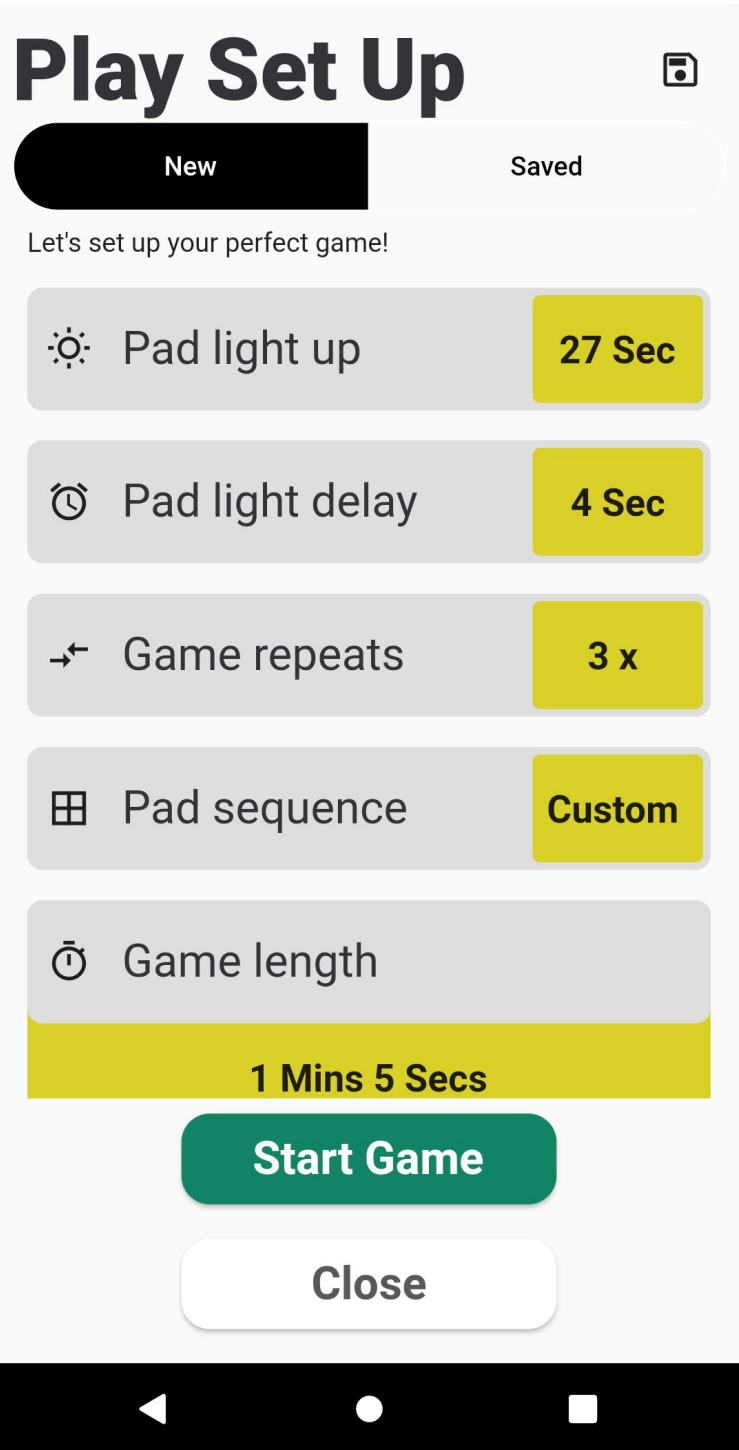
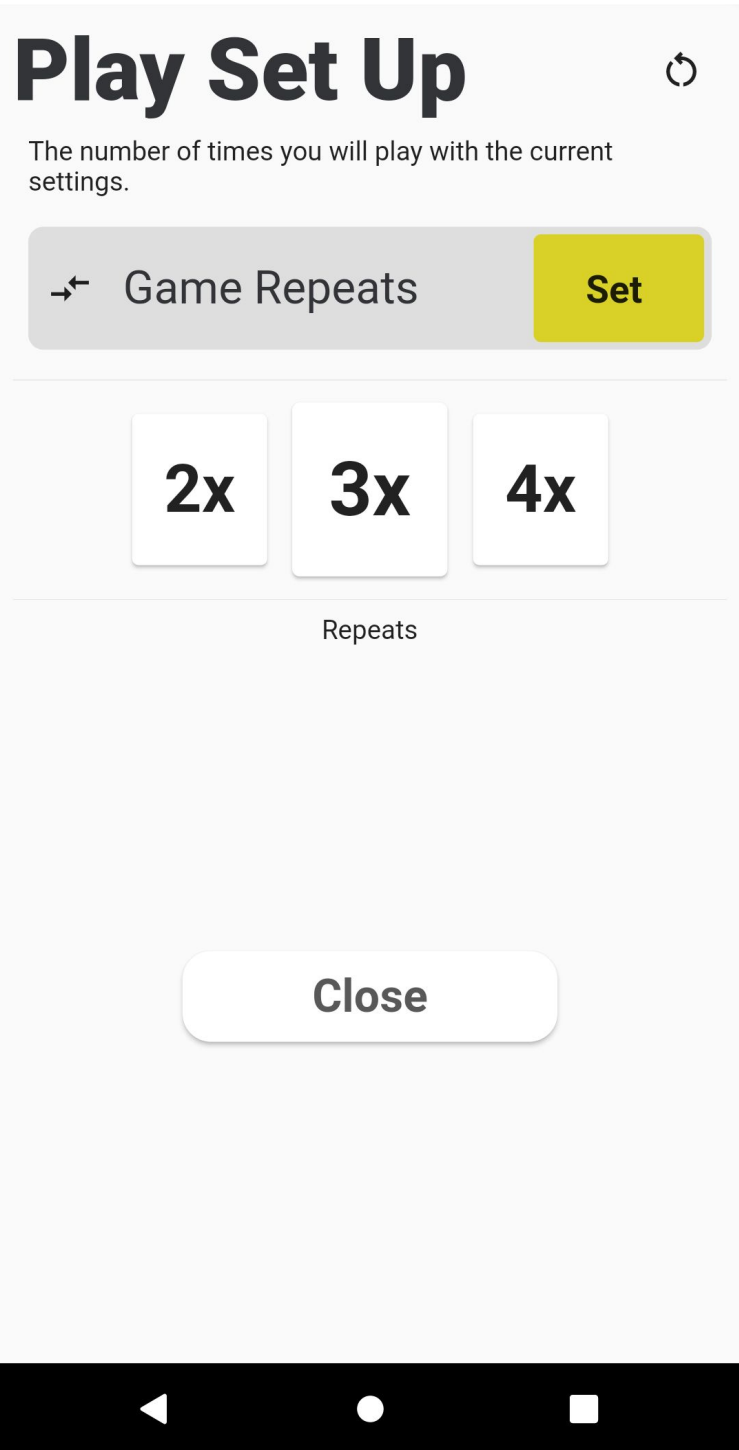
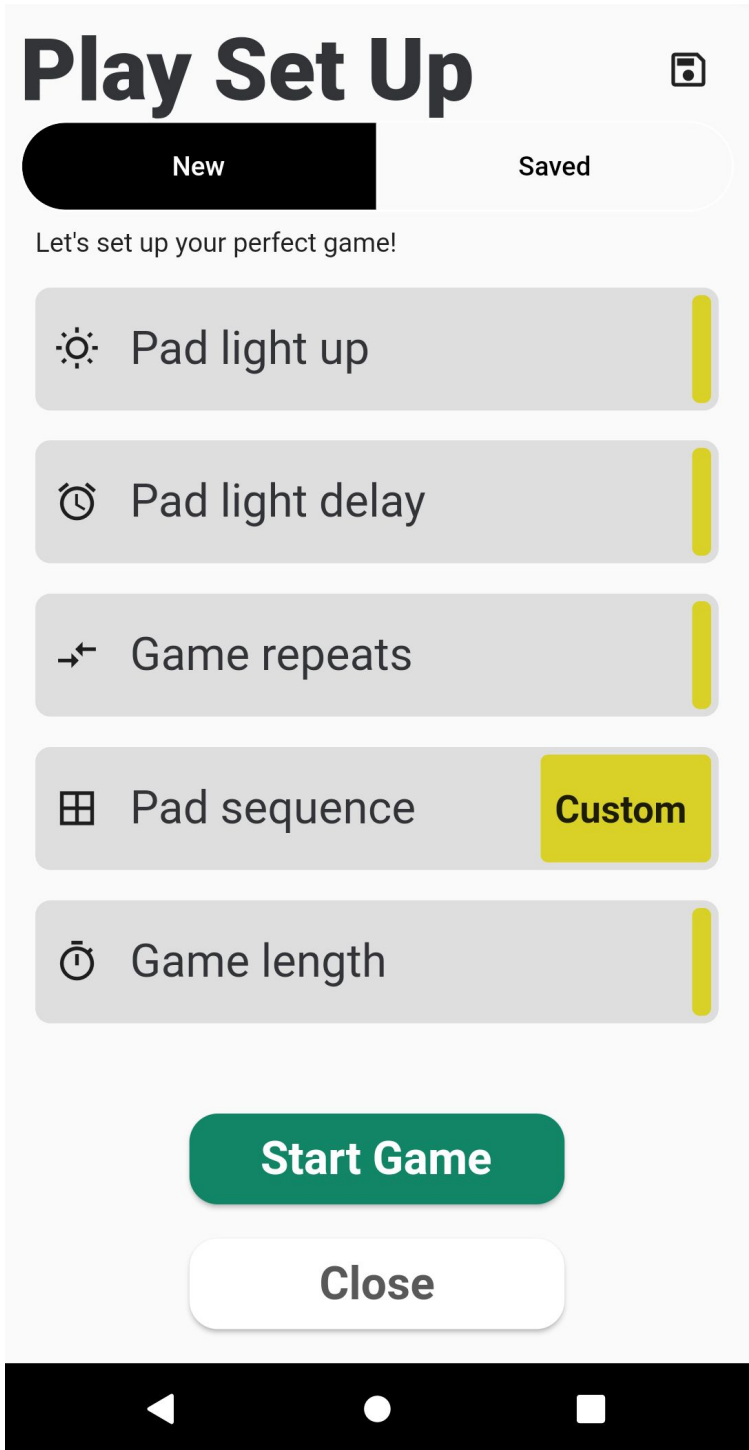
IMPLEMENTATION

The process was relatively smooth since we already counted with a Figma design which we could follow. The adversities I faced in the implementation face involved researching and understanding the Flutter widgets needed to bring to life the vision of the app.

REFERENCES

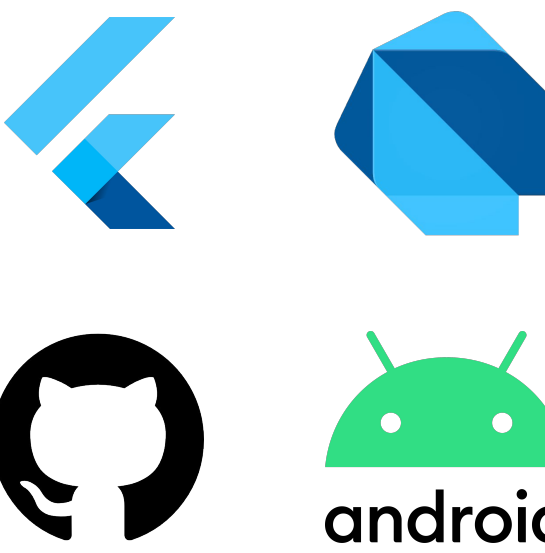
Flutter:

- <https://docs.flutter.dev/cookbook>
- Bailey, T., & Biessek, A. (2023). Flutter for beginners (3rd ed.). Packt Publishing.
- <https://en.wikipedia.org/wiki/WebSocket>
- https://pub.dev/packages/web_socket_channel
- <https://networkinterview.com/what-is-mdns-multicast-dns/>.
- <https://medium.com/mqtt-buddy/mqtt-vs-http-which-one-is-the-best-for-iot-c868169b3105>
- <https://nitin-sharma.medium.com/getting-started-with-mqtt-part-1-a3c365e3a488>



REQUIREMENTS

Flutter
Dart
GitHub
Android Studio
Android VM
Android Tool Chain



ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by Gummi Oxndal. We/I thank Gummi Traustason and students from previous semesters for their assistance and mentorship that we received throughout the senior design project.

FLORIDA INTERNATIONAL UNIVERSITY

Skillcourt

Students: Luis Chavez

Mentor: Gummi Oxndal, Product Owner

Instructor/Faculty: Masoud Sadjadi,, Florida International University

PROBLEM

The SkillCourt mobile app grappled with challenges arising from incomplete development and UI implementation errors, including redundant code and unexpected behavior. Routing issues from buttons to pages also appeared but more notably, the UI element code demanded abstraction to enhance codebase usability. Addressing navigation methods was crucial for an improved user experience and overall performance. Key features that needed implementation included a color picker screen on both the home and settings pages, an FAQ page, a brightness page, and the ability to send signals to software connecting user input to the LED pad's code.

SUMMARY

My role in this project was pivotal in making the Skillcourt mobile application seem less vacant in its User Experience features. Through assistance with the team captain who had a proficient understanding of Google's Flutter / the programming language Dart, I was able to learn on the process to build working features such as the color picker, FAQ page, and brightness page and route them successfully via the home page and settings page. Both of these were built from the ground up as the groundwork laid by previous teams was purely design and so I believe we made good progress given our time frame. The user now has a better means of interaction with the purpose of the Skillcourt Hardware which is to train their soccer skills via LED pads.

VERIFICATION

Verifying the process involved referring to the Figma website that had the designs which we based the app's new UI feature's on. I was able to implement the Color Picker feature following the request of the designer where the user selects the color of the grid with the overlay of a soccer ball, the chosen color produces a useable color value. Also, I was able to implement the Brightness Page feature where users can slide to their preferred % brightness of their LED pads.

REFERENCES

Flutter:

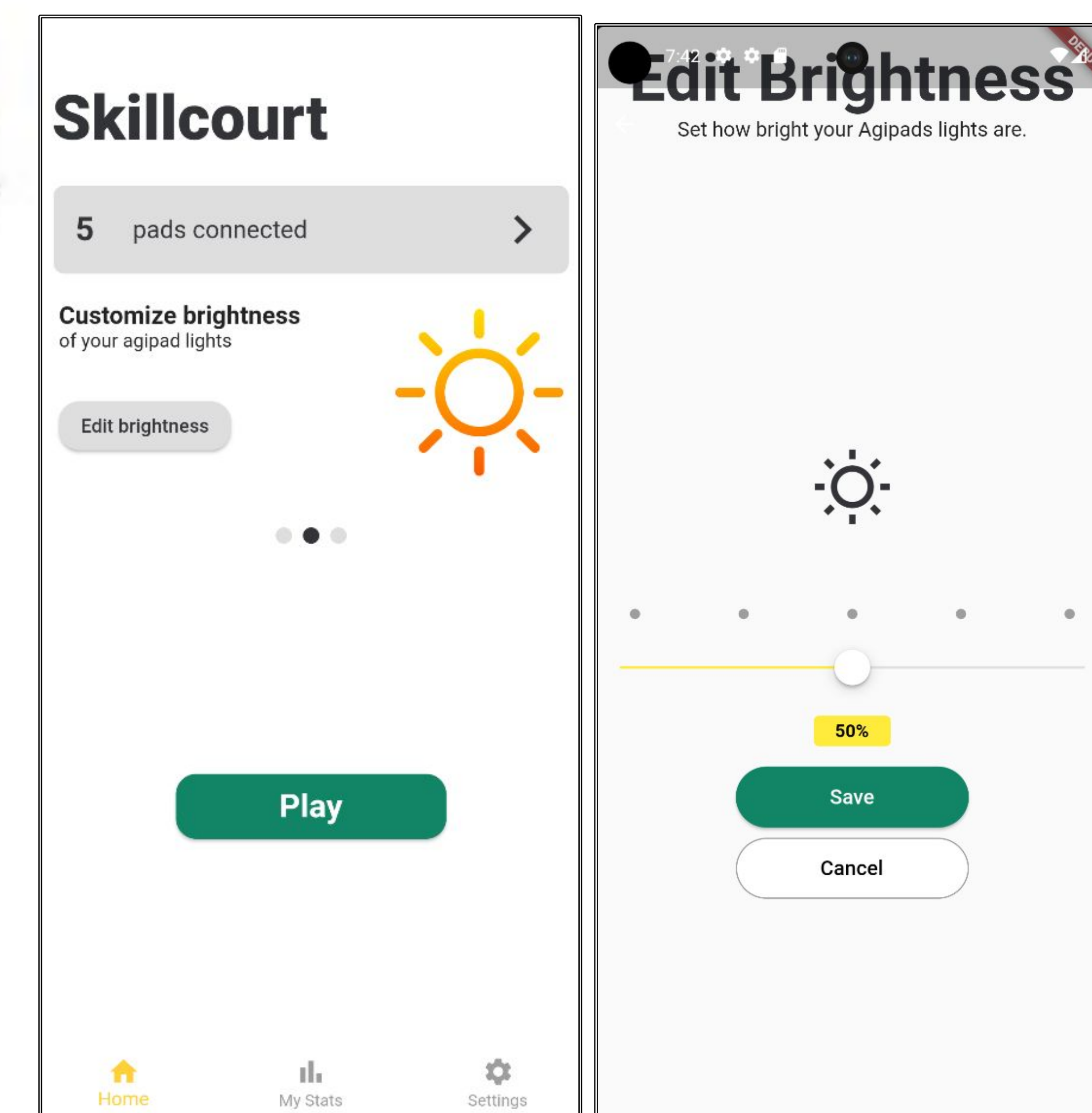
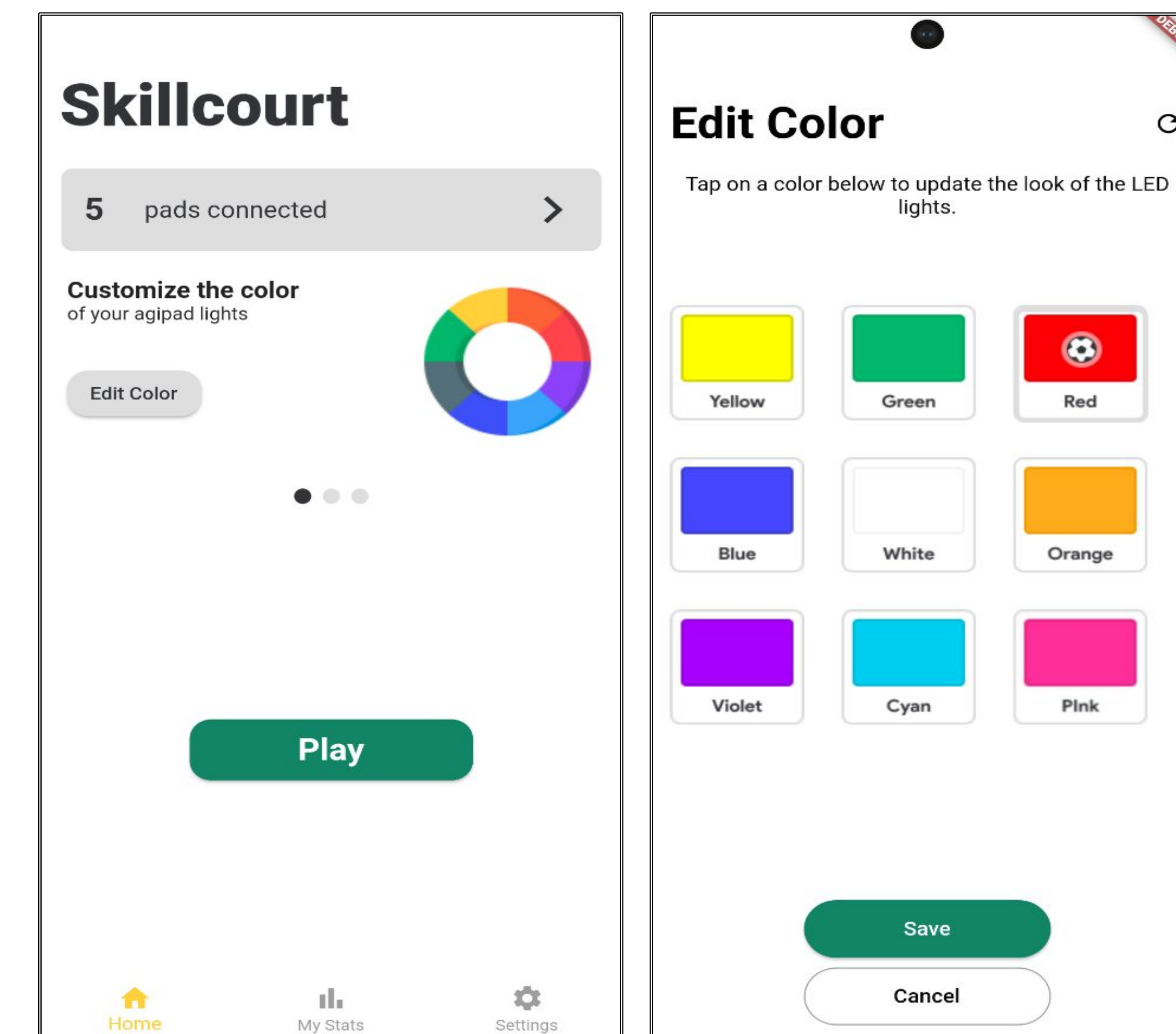
- <https://api.flutter.dev/flutter/cupertino/CupertinoSlider-class.html>
- <https://docs.flutter.dev/cookbook>
- [The pubspec file](https://pubspec.io) | Dart,
- <https://dart.dev/tools/pub/pubspec>
- [Run apps on the Android Emulator](https://developer.android.com/studio/run/emulator) | Android Studio | Android Developers
- <https://developer.android.com/studio/run/emulator>

IMPLEMENTATION

Implementing the features assigned to me was relatively easy given that we had the Figma designs to borrow our buttons and text from and maintain that simple yet professional design language. Challenges may have been coming up with the idea to use an array to store the slider brightness images as the user increases their % brightness the sun rays expand. As well as use of the Cupertino slider included in the flutter UX library. Little details of this type create larger impressions of quality for the product.

CURRENT SYSTEM

Our current system is a Dart application using flutter to facilitate easy responsiveness across platforms as well as efficient building of UI components. Testing was done using mobile phone emulators that used Android OS.



REQUIREMENTS

Flutter
Dart
GitHub
Android Studio
Android VM
Android Tool Chain



Students: Johan Balceiro 
Mentor: Gummi Oxndal
Product Owner: Gummi Oxndal
Instructor: Instructor/Faculty: Masoud Sadjadi, FIU



Introducing SkillCourt: The Ultimate Soccer Training App!

SkillCourt is a cutting-edge application designed to revolutionize the way soccer athletes hone their skills. This innovative platform provides a controlled environment for players to engage in dynamic drills, personalize training regimens, and collaborate seamlessly with coaches. The app communicates with interactive pads that respond to hits and light up with LED feedback, offering real-time insights into performance.

Key Features:

- 🌟 Elevate Your Skills: Engage in dynamic drills for precision and mastery.
- 📋 Personalized Regimens: Craft customized routines targeting specific plays and habits.
- 👨‍🏫 Coach Collaboration: Seamlessly interact with coaches for expert guidance.
- 💡 Interactive Feedback: LED-enhanced pads provide real-time insights.
- 🌐 Controlled Environment: Practice in a dynamic yet controlled space.

SkillCourt – Precision, Collaboration, Innovation, Excellence. Elevate your soccer game to new heights!

Server Side (Pad) Implementation

In the server-side implementation for the interactive pads, SkillCourt leverages the Python programming language. The server is responsible for handling communication with the application, processing hits, and triggering LED feedback. Here's a concise overview of the implementation:

- **WebSocket Integration:** Utilized Python's WebSocket libraries, such as websockets or socket.io, to establish a bidirectional communication channel with the Flutter application.
- **Event Handling:** Implemented event-driven mechanisms to handle incoming hit data from the application. This involves processing hit patterns and triggering corresponding LED responses.
- **LED Control Logic:** Developed logic to control the LED illumination patterns on the interactive pads based on real-time data received through the WebSocket connection.
- **Concurrency and Threading:** Implemented concurrency mechanisms, such as threading, to manage simultaneous communication with multiple pads, ensuring efficient and responsive handling of hits.
- **Error Handling:** Implemented robust error-handling mechanisms to gracefully manage communication disruptions or unexpected events, ensuring the stability of the server-side logic.

Client Side (App) Implementation

WebSocket Integration: Leveraged Flutter packages like web_socket_channel to establish and manage WebSocket connections with the Python server.

- **User Interface (UI):** Designed an intuitive and user-friendly interface to enable athletes to interact with the application effortlessly. This includes features for selecting drills, creating personalized regimens, and initiating training sessions.
- **Data Serialization:** Implemented mechanisms for serializing and deserializing data to be exchanged between the application and the server, ensuring a standardized communication protocol.
- **Real-Time Feedback Display:** Integrated components to visually represent real-time feedback from the interactive pads, allowing users to track their performance during training sessions.
- **Error Handling and Resilience:** Incorporated error-handling strategies to gracefully manage communication disruptions or errors, providing a resilient user experience.
- **Cross-Platform Compatibility:** Ensured compatibility with multiple platforms (iOS and Android) by utilizing Flutter's inherent cross-platform capabilities

Florida International University

Fall 2023 Capstone II Project

SkillCourt

Communication Evolution

SkillCourt initially employed the mDNS (Multicast Domain Name System) as its communication protocol with the interactive pads. However, in our pursuit of providing users with an even more seamless and robust experience, we have made a strategic shift to the WebSocket protocol.

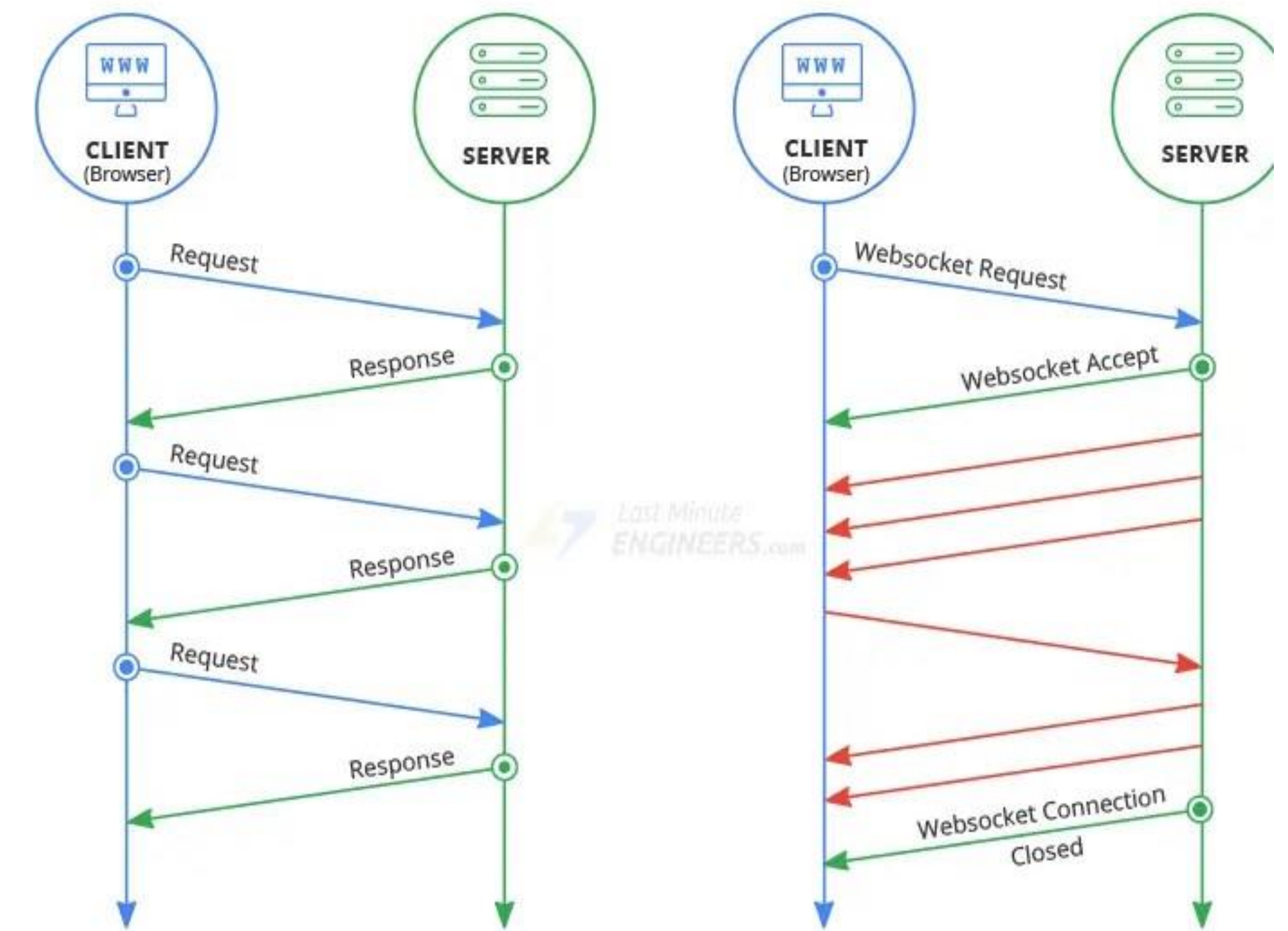
Why the Change?

Enhanced Real-Time Responsiveness: WebSocket offers a more efficient and low-latency communication channel compared to mDNS, ensuring that the feedback from the interactive pads is even more instantaneous.

Reliability and Stability: WebSocket's persistent connection architecture enhances reliability, reducing the chances of communication disruptions during training sessions. This change ensures a more stable and consistent user experience.

Scalability: WebSocket's support for bidirectional communication makes it inherently scalable, allowing SkillCourt to accommodate future enhancements and features seamlessly.

Cross-Platform Compatibility: WebSocket is widely supported across various platforms and devices, ensuring a consistent and uniform experience for SkillCourt users, regardless of their chosen device.



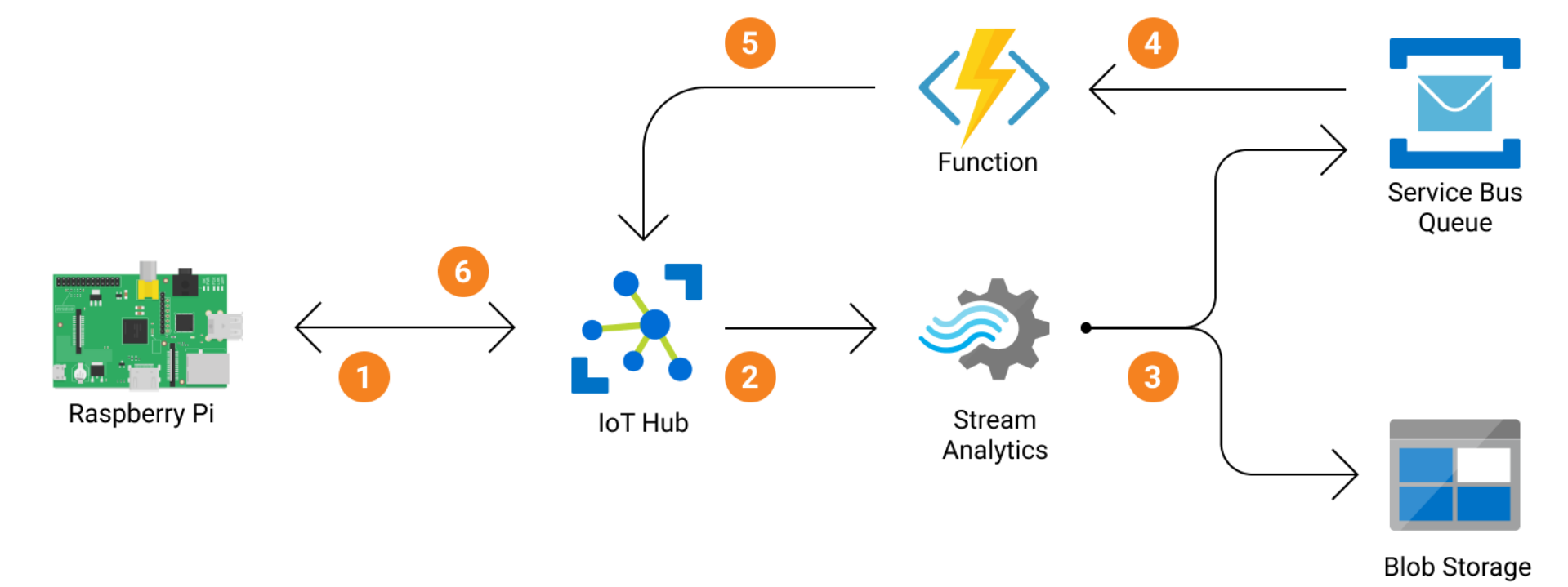
Summary

As a pivotal member of the SkillCourt project, my primary role revolved around orchestrating the transition between communication protocols, a critical evolution that significantly impacted the app's performance. I actively contributed to the initial research and decision-making process that led to the strategic migration from mDNS to the WebSocket protocol.

This involved delving into the intricacies of each protocol, analyzing their strengths and weaknesses, and ultimately steering the team towards the adoption of WebSocket for its superior bidirectional communication capabilities.

In addition to this, I took the initiative to create a robust and testable emulator in IoT Hub Azure. This emulator serves as a valuable asset for future teams, enabling them to simulate real-world scenarios and thoroughly test the communication system without the need for physical pads. The emulator not only streamlines the development process but also enhances the overall reliability and scalability of SkillCourt.

Being at the forefront of these technical decisions and implementations has been an enriching experience, and I'm proud to have played a key role in enhancing the communication infrastructure of SkillCourt for a more responsive and innovative soccer training platform



References

Mozilla Developer Network. (n.d.). Writing WebSocket servers. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API/Writing_WebSocket_servers

Microsoft. (n.d.). Understand Different Connection Strings in Azure IoT Hub. Retrieved from <https://devblogs.microsoft.com/iotdev/understand-different-connection-strings-in-azure-iot-hub/>

Microsoft. (n.d.). IoT Hub live data visualization in web apps. Retrieved from <https://learn.microsoft.com/en-us/azure/iot-hub/iot-hub-live-data-visualization-in-web-apps>

REQUIREMENTS

Flutter
Dart
GitHub
Android Studio
Android VM
Python
PyCharm
Android Tool Chain

