H4PRO

MINIMUM VIABLE PRODUCT (MVP)

PADEL4PRO

REQUIREMENTS

1. **As a** padel player, **I want to** synchronize my smartwatch to the court session and MOG's API, **so that** I can stay updated with real-time game information and interact with the system.
2. **As a** padel player, **I want to** be able to request a highlight recording of my play from my smartwatch, **so that** I can review my performance and/or save it for later.
3. **As a** padel player, **I want to** request a video assistant referee replay, **so that** I can ensure fairness and accuracy, clarify disputes, and improve user experience
4. **As a** padel player, **I want to** be able to request a replay of the last saved highlight from my smartwatch, **so that** I can review my performance, learn from my mistakes, and improve my gameplay.
5. **As a** padel player, **I want to** update the scoreboard from my smartwatch, **so that** I can ensure accuracy and avoid misunderstandings with my opponents.

ARCHITECTURE AND TECHNOLOGIES

H4PRO is a smartwatch and phone app that is built to help Padel players with their in-game experience. **Important note**: the communication between the smartwatch and phone was not implemented, and the API that was planned was not available. The product created mocks these communications.

System Architecture

The ideal final product, once integrated with MOG's servers, is to have a seamless and efficient user experience for Padel players. For this, all the different components must be perfectly connected so that the interactions occur with no flaws.
The cameras on the Padel courts will communicate directly with MOG's services to send data and receive highlight requests. MOG's streaming servers will also communicate with the servers of the courts to play the requested highlights.
The smartwatch will also have access to MOG's services, so that the highlights may be requested by using the button on it for that purpose.
The phone app the phone also has access to MOG's services and handles the join and disconnection from the court.

Smartwatch application	This app is developed for WearOS smartwatches using Kotlin as the programming language. The use of this language ensures reliability, efficiency, and easy maintenance, while it provides an intuitive experience for the end users.
Phone application	For the phone app, the programming language used is React Native. This language is supported by both iOS and Android smartwatches, which makes it ideal for cross-platform development. This way, the app is expansible to other operating systems.

ARCHITECTURE AND TECHNOLOGIES

API Structure

Besides both applications, we also designed a concept for the API that would go along with our application. The logical progression would be for a user to login, scan the QR code, and consequentially start a match, after that, we would get a match token from which he could do requests like asking for a video referee, updating the score of the game, and much more. In the end, the user could leave the match. The API has the following structure:

Open Endpoint

The following endpoint doesn't require Authentication.

Login

Used to get the Authentication token to access the rest of the API.

URL: /login
Method: POST
Request Body

```
{
    "email": "<string>",
    "password": "<string>"
}
```

Response

```
{
    "authToken": "<string>"
}
```

ARCHITECTURE AND TECHNOLOGIES

Closed Endpoints

The following endpoints require the authToken to be inserted in the header of the request in the Authorization field.

Start a match
Initiate a match and receive an identification for the running match.

URL: /match
Method: POST
Request Body

```
{
    "courtId": "<string>",
    "timestamp": "<number>"
}
```

Response

```
{
    "matchId": "<string>"
}
```

End a match

Terminate a running match.

URL: /match/{matchId}
Method: POST
Request Body

```
{
    "timestamp": "<number>"
}
```

ARCHITECTURE AND TECHNOLOGIES

Closed Endpoints

Get information about a court

Initiate a match and receive an identification for the running match.

URL: /court/{courtId}
Method: GET
Response

```
{
    "isAvailable": "<boolean>",
    "name": "<string>"
}
```

Get the current score

Get the current score of a running match.

URL: /score/{matchId}
Method: GET
Response

```
{
    "pointsA": "<integer>",
    "setsA": "<integer>",
    "pointsB": "<integer>",
    "setsB": "<integer>"
}
```

ARCHITECTURE AND TECHNOLOGIES

Closed Endpoints

Update Score

Update the score of a running match.

URL: /score/{matchId}
Method: POST
Request Body

```
{
    "pointsA": "<integer>",
    "setsA": "<integer>",
    "pointsB": "<integer>",
    "setsB": "<integer>"
}
```

Request referee

Request referee.

URL: /requestRef
Method: POST
Response

```
{
    "matchId": "<string>"
}
```

ARCHITECTURE AND TECHNOLOGIES

Closed Endpoints

Request highlight

Request a to make a highlight.

URL: /highlight
Method: POST
Request Body

```
{
    "currentMatch": "<string>",
    "currentCourt": "<string>",
    "timestamp": "<dateTime>"
}
```

Response

```
{
    "highlight": "<integer>"
}
```

Request a highlight to be shown

Request a highlight to be shown in the court.

URL: /highlight/{highlightId}
Method: GET

ARCHITECTURE AND TECHNOLOGIES

Features

Smartwatch application

- **Request Highlight**: The user can either request a highlight by making a thumbs up to the camera or by pressing a button on the smartwatch, sending an API request;
- **Request Highlight** Visualization: The user presses a button on the smartwatch, sending an API request to show the highlight menu on the court's screen;
- **Request Video Referee**: The user presses a button on the smartwatch, sending an API request to show the last moment on the court's screen;
- **Score Tracking**: The smartwatch app has a built-in score tracker, which the user can increment, and the smartwatch updates the smartwatch with the score information.

Phone application

- **Login:** The user should log in with the MOG's platform credentials;
- **Join Court**: The user opens the app's camera and scans a QR code, communicates with the API, receives a game token, and sends it to the smartwatch;
- **Leave Court**: The user clicks a button, the app sends an API request and communicates to the smartwatch that the player left the game.

Extra

- **QR code Generator**: The project maintainor can generate court QR-Codes for the user to scan and join the court session.

ARCHITECTURE AND TECHNOLOGIES

Development Constraints

Constraint	Description
WatchOS Development	The development of both WearOS and WatchOS applications didn't add value to the MVP to be delivered. This way, the company decided to only do the development of the WearOS and the phone apps.
Bluetooth Connection	The phone app is developed in React. After long research, we agreed that even though it can be emulated, React doesn't have a feasible way to do it.
Hardware	Between all team members, there was no smartwatch with WearOS so that the smartwatch app could be tested outside the emulator.

PADEL4PRO

HANDOVER

INSTALLATION GUIDE

SMARTPHONE

To install the smartphone app on an Android device, follow these steps:

1. Ensure that the device is running Android OS version 5.0 (Lollipop) or higher.
2. Generate the APK file:
   a. in the folder **./padel4pro-phone/** execute the command **expo build:android**;
   b. if you haven't logged in to Expo, it will prompt you to authenticate and log in using your Expo account credentials;
   c. after the build process finishes, Expo will provide a download link for the generated APK file.
3. Enable installation from unknown sources in the device settings.
4. Connect the phone to the PC and transfer the APK file.
5. Locate the downloaded APK file and tap on it to initiate the installation process.
6. Follow the on-screen instructions to complete the installation.
7. Once installed, the app icon will appear on the device's home screen.

PADEL4PRO

HANDOVER

INSTALLATION GUIDE

SMARTWATCH

To install the smartwatch app on a WearOS device you will need a computer. Then, follow these steps:

1. Ensure that the device is running WearOS version 11.0 or higher.
2. Generate the APK file:
    a. In the folder **/PADEL4PROAndroid/wear/** select Build -> Build Bundle(s)/APK(s) -> Build APK(s).
    b. Wait until the .apk file is generated.
3. Setup ADB:
    a. Download the ADB file based on your computer operating system.
    b. Extract the file to a chosen location. On Windows, hold the Shift key and right-click on the extracted file, and select 'Open command window here.' Press Enter.
    c. To enable ADB on the smartwatch, open Settings. Scroll to the System button and open it. Tap About. Scroll again until the Build Number is found. Tap 7 times on top of it to unlock the developer's options on the smartwatch.
    d. Open again the Settings app. There will be a Developer Options option. Open it.
    e. Enable the toggle next to ADB debugging, and confirm. You should also enable the toggle next to the Debug over Wi-Fi. If a message shows up, saying Unavailable, it will soon be replaced by the IP address.

HANDOVER

INSTALLATION GUIDE

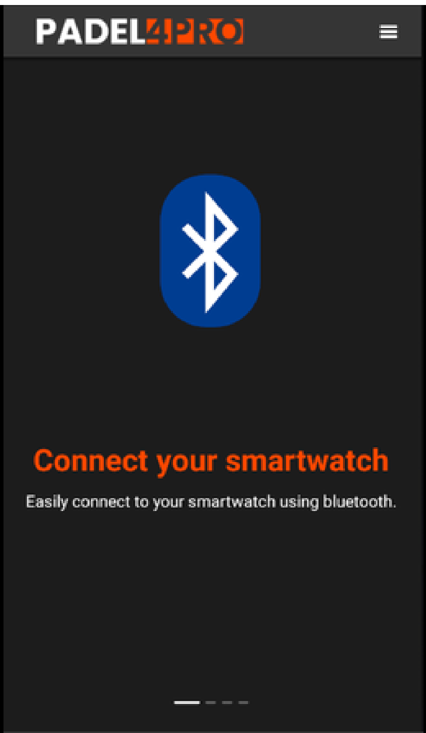
SMARTWATCH

4. Install the APK via ADB:
    a. Copy the file **.apk** to the folder **platform-tools** where ADB is installed.
    b. In the previously opened terminal, type **./adb** connect IP_address displayed before on the screen of your smartwatch.
    c. In the smartwatch, a new prompt must be shown. Select OK to Allow Debugging.
    d. The terminal window must show that is connected to the smartwatch now.
    e. In the same terminal, execute the command **./adb push <filename.apk>/sdcard/**
    f. The file must have been pushed to the smartwatch.
    g. Now, install the app by running **./adb -e install <filename.apk>**
    h. A Success message should be displayed in the terminal.
    i. For efficiency improvement, go back and disable the ADB debugging toggle.
5. Open the app that should now appear in the app drawer of the smartwatch.

PADEL4PRO

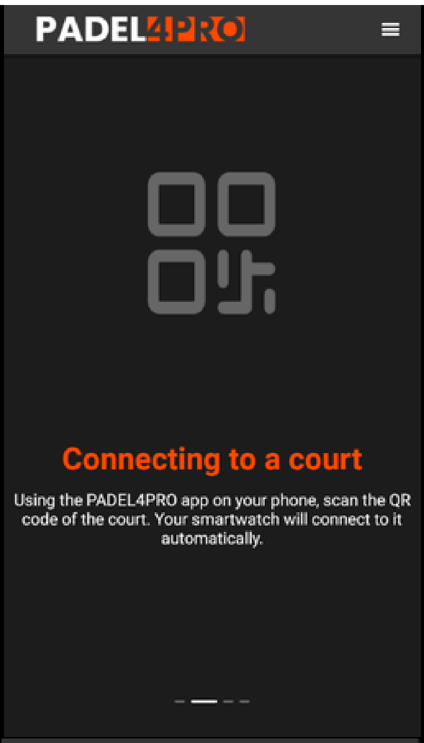
USER MANUALS

When the phone application is installed and initialized, the user is provided with a tutorial on how to use the app, with illustrations to help understand everything. This tutorial is available at all times, by entering the help page. This tutorial has information for both the phone and smartwatch apps.



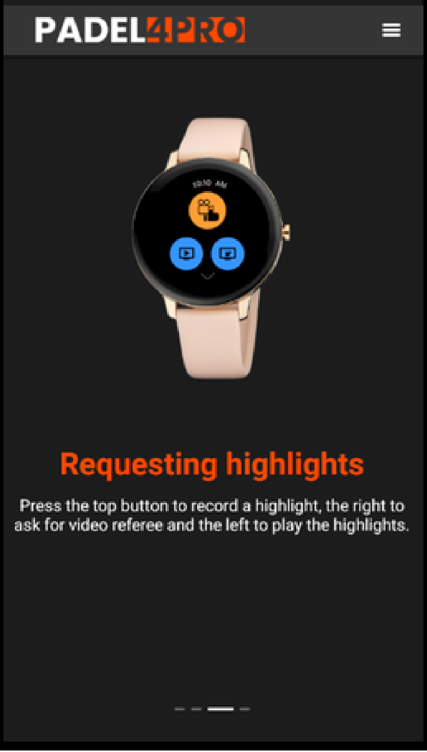
Connect your smartwatch

Easily connect to your smartwatch using bluetooth.



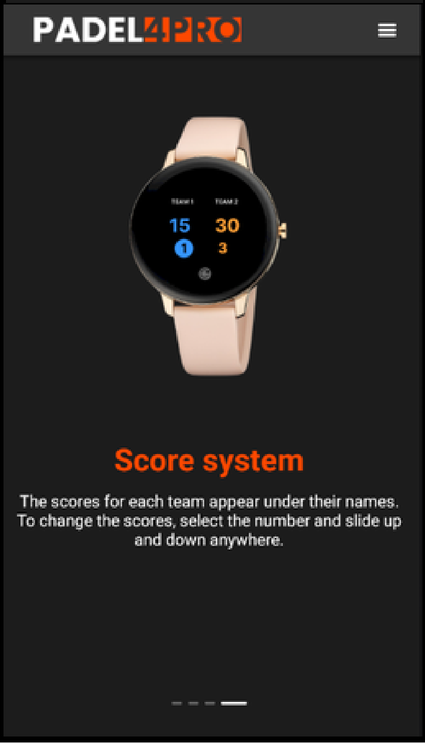
Connecting to a court

Using the PADEL4PRO app on your phone, scan the QR code of the court. Your smartwatch will connect to it automatically.



Requesting highlights

Press the top button to record a highlight, the right to ask for video referee and the left to play the highlights.

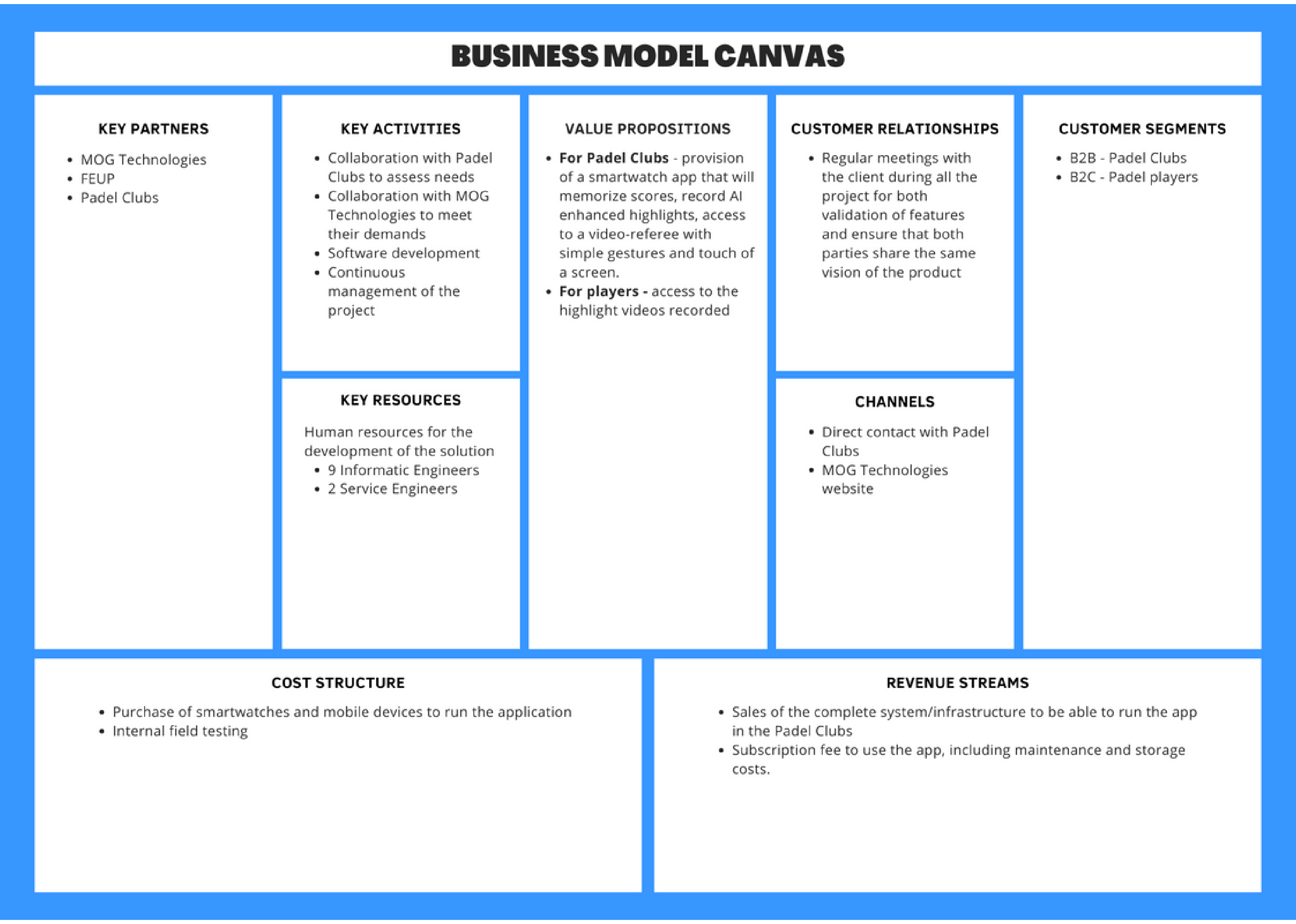


Score system

The scores for each team appear under their names. To change the scores, select the number and slide up and down anywhere.

PADEL4PRO

BUSINESS MODEL



BUSINESS MODEL CANVAS

KEY PARTNERS

- MOG Technologies
- FEUP
- Padel Clubs

KEY ACTIVITIES

- Collaboration with Padel Clubs to assess needs
- Collaboration with MOG Technologies to meet their demands
- Software development
- Continuous management of the project

KEY RESOURCES

Human resources for the development of the solution
- 9 Informatic Engineers
- 2 Service Engineers

VALUE PROPOSITIONS

- **For Padel Clubs** - provision of a smartwatch app that will memorize scores, record AI enhanced highlights, access to a video-referee with simple gestures and touch of a screen.
- **For players -** access to the highlight videos recorded

CUSTOMER RELATIONSHIPS

- Regular meetings with the client during all the project for both validation of features and ensure that both parties share the same vision of the product

CHANNELS

- Direct contact with Padel Clubs
- MOG Technologies website

CUSTOMER SEGMENTS

- B2B - Padel Clubs
- B2C - Padel players

COST STRUCTURE

- Purchase of smartwatches and mobile devices to run the application
- Internal field testing

REVENUE STREAMS

- Sales of the complete system/infrastructure to be able to run the app in the Padel Clubs
- Subscription fee to use the app, including maintenance and storage costs.

PRODUCT VISION STATEMENT

For	Padel Clubs who want to enhance their court experience and consequently to the Padel players.		
Who	Those who are looking for innovative and personalized sports watch applications and game-highlighting solutions that strengthen and celebrate sports.		
The	Padel4Pro	**is a**	Sports Watch App and Game Highlighting Solution
What	Provides powerful tools for game optimization, such as video referee, highlights recording and memorization of scores.		
Unlike	Other sports viewing apps and game highlighting solutions on the market, such as the ones from the competitors PlaySight and Padel-Tenis.		
Our product	A commitment to capture the essence of the game and make it an enjoyable experience and set new standards for the sports industry in meeting the evolving needs of customers.		

PRODUCT VISION BOARD

Vision	Padel4PRO's vision is to revolutionize how players interact with sports and enhance their experience while playing. We aim to provide a leading and innovative solution that will revolutionize how Padel is played worldwide.
Target group	• **Users:** Padel players • **Customers:** Padel clubs/courts
Needs	• Provide users with real-time highlights, statistics and video-referee. • Enable coaches and athletes to review and analyze game footage for performance improvement. • Have a smoother game experience, without having to memorize scores or keep track of the game.

PRODUCT VISION BOARD

Product	Our application differentiates itself from other sports viewing and game highlighting solutions on the market with advanced features like real-time player and team statistics and real-time game highlights. In addition to providing users with basic scores to help them keep track of the game, the highlights capture solution provides an additional asset. This way, our app captures the essence of the game, setting new standards in sports. Overall, the app is designed to meet the needs of Padel players, optimizing MOG services and supporting decision-making.
Business Case	• To be the market leader in innovative and personalized smartwatch apps for Padel. • To increase awareness and reach and elevate the Padel game to the next level. • To continuously improve and develop our product in line with the changing needs of our customers and the sports industry.

MINIMUM VALUE PRODUCT

Base Goal	• Smarthwatch APP for Android and ios • Allow users to send and receive information to **MOG's PADEL4PRO**
Base Mechanism	• Session ID received VIA QR or Link • APP receives and sends information to **MOG API**
Base actions	• Receive, display and edit **Scoreboard** • Buttons for request **highlight**, **instant replay** and **video referee**

We want the MVP to be a stable version where the user has all the features that define and distinguish our product. The MVP will contain the smartwatch application, for both iOS and Android systems, and the phone application. These pieces of software will be able to communicate with the MOG's API. Furthermore, our product will be able to receive session IDs through links and QR codes, receive, display, and edit the scoreboard, and request replays, highlights, and the video referee.

PROTOTYPE

Our smartwatch app will have two screens - (Fig. 1) display the score and allow manual updates with slide gestures, and the other (Fig. 2) with three buttons to save a highlight, watch game footage, and request a video assistant referee (VAR) recap of the game.

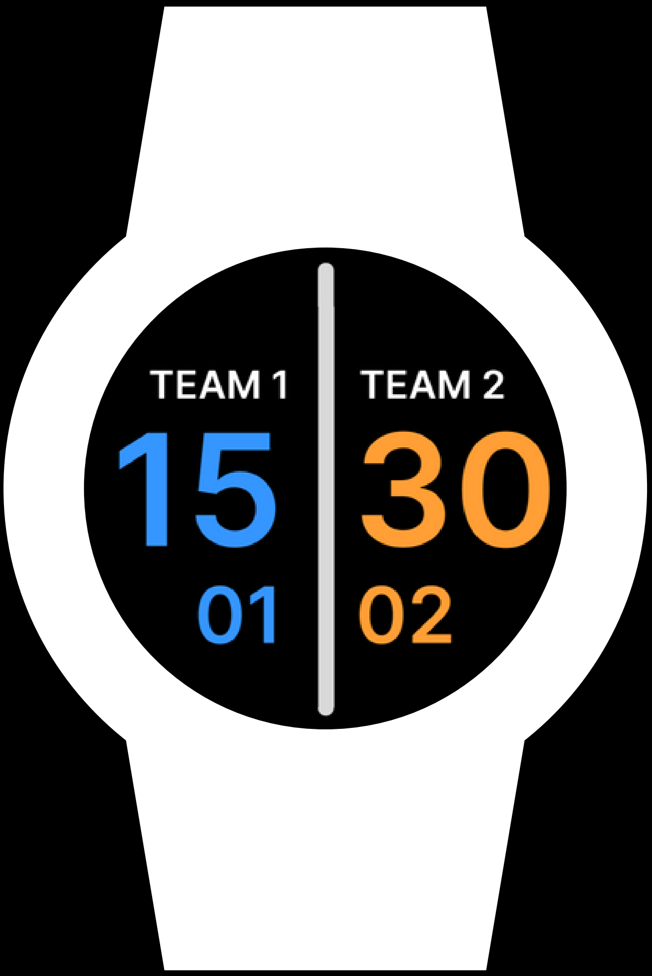


Fig 1. Scoreboard Menu, the larger numbers represent the current points and the smaller the set points.

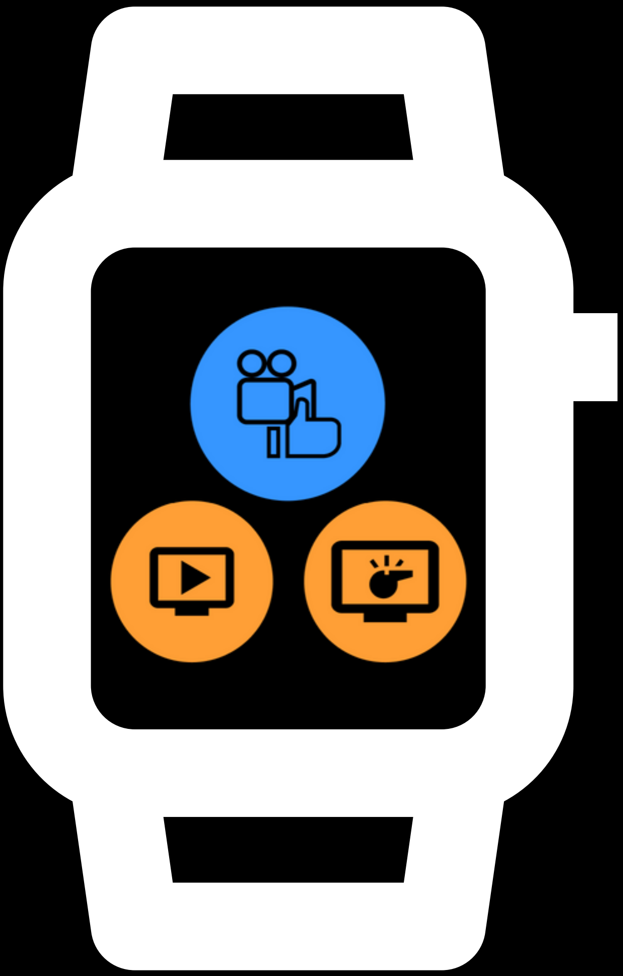


Fig 2. Main Menu, with the blue icon being the save highlight button, the right orange icon the request video referee assistance button, and the left orange icon the watch the game footage button.

PROTOTYPE

Our phone app has a main screen where all the main actions are made (Fig. 3). Here it is possible to join a court session, by clicking the Join Court button, which opens the camera, and scanning the Court's QR code; to leave the court session, and accessing the setting menu by clicking the top-right button. The project also contains a script to generate the court QR codes.

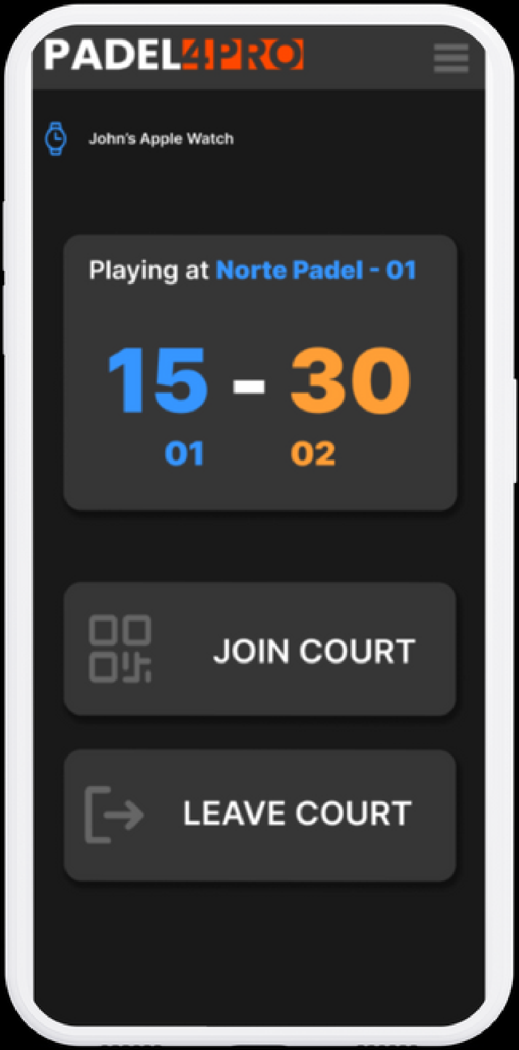


Fig 3. Phone Menu, from here it is possible to join a court, by clicking the respective button and scanning a QR code, and leaving the court.

FINDINGS AND METRICS MEASURED

Metrics

- **Usability and Design Metrics:** Usability metrics were collected through user inquiry sessions, surveys, and feedback. These metrics evaluated the app’s ease of use, intuitiveness, and user satisfaction.

- **Feature Relevance Metrics:** The relevance of the features implemented was assessed through questionnaires and interviews. The sources evaluated and confirmed their appositeness.

FINDINGS AND METRICS MEASURED

Findings - Market

- **Scoreboard:** during our public assessment, the interactive scoreboard was a feature that had great feedback.

- **Selling Strategy:** during our public assessment, the vast majority of the public said they would not pay an extra amount for a padel court that had the PADEL4PRO system installed, however, they said they would prefer a padel court that had this system installed over one that didn't, which give good information on how the selling strategy for the product could be executed.

Findings - Technological

- **Smartwatch cross-platform development:** Before the start of the build-measure-learn phase, while the team was planning the tech stack to be used, they noticed the lack of a cross-platform option for Smartwatch development in the market, forcing developers to create and maintain two distinct repositories for the same app, one for Android smartwatch operating system and other for Apple Watch operating system (WatchOS), to have a multi-platform application.

- **React-Native BLE (Bluetooth Low Energy) integration:** The team encountered issues with Bluetooth Low Energy (BLE) integration in React Native when developing the smartwatch-phone connection. They tested common BLE packages such as react-native-ble-plx and react-native-ble-manager. However, they faced lackluster support for smartwatch BLE communication, running into problems with unreliable connectivity and restricted usefulness, for example, to detect already connected devices to the smartphone. To overcome these challenges, the team may look for updates and community assistance, investigate native modules, or use other libraries or SDKs developed for BLE communication.