Interação Pessoa-Máquina

2024/2025

SportMeet

Stage 4:   Computational Prototype

Authors:

60739, Yaroslav Hayduk
69898, Tomás Filipe Canelas Martins
70703, Tiago Miguel Brito Caeiro
69900, Duarte Ramos Coelho

Lab class Nº P4

Group Nº 17

Professor:
Teresa Romão

Month 11, 2024

Prototype URL
https://github.com/wiirijofct/SportMeet.git

Startup Instructions

Pre-Requisites:
1. *Download Git:* You can download Git here: https://git-scm.com/downloads.
2. *Download the Prototype:* You can download the prototype by git cloning it into your machine, using the following command on you command line: git clone https://github.com/wiirijofct/SportMeet.git.
3. *Download Node.js:* You can download Node.js here: https://nodejs.org/pt. After you downloaded Node.js you will have to run the following command on your command line: **npm install -g json-server** .
4. Create the (false) REST API: Once you downloaded Git, the Prototype and Node.js, open a command line on the mockDB folder, and run the following command: **json-server --watch db.json** .
5. Run flutter commands: In order to run the app, you will need to run a few flutter commands before:
   a. **flutter get pub**;
   b. **flutter upgrade**;

You are now ready to run the app. You can do so by running the **flutter run** command.

Briefing

This project aims to create an app that makes it easy for amateur athletes to find sports partners and book venues for individual and small-team activities. Many sports enthusiasts struggle to coordinate schedules, find partners, and book spaces for activities like tennis, soccer, or basketball, especially those with busy routines, newcomers to a city, or those without established sports groups. These challenges often discourage regular participation. The app provides a unified platform where users can connect with others based on shared sports interests, compatible schedules, and location. It enables users to book public or private venues directly in the app and includes scheduling tools with reminders to ensure attendance. For added convenience, payment and attendance commitment options will also be integrated. Designed for young adults, university students, professionals, and newcomers, the app aims to build an active, engaged sports community by making it simple to join and organize activities regularly.

Scenarios:

Scenario 1:

Host Registering and Managing Court Availability.

Scenario:

You own a sports field and want to get more people to use it. Register it in the app. Keep in mind that the field is only available for bookings from Monday to Saturday, from 9AM to 8PM, and it will be closed during December.

Scenario 2:

Athlete Checking Availability and Booking a Game at 3 PM, next Saturday.

Scenario:

You're free to play at 3 PM on the next Saturday. Check if anyone's willing to play at that time. If not, make a reservation yourself. Be ready for others to join.

Scenario 3:

Athlete Looking for a Weekend Sports Partner.

Scenario:

You're looking for someone to play basketball with on the weekends. Try to find someone with the same interests as you. If you find someone, try to exchange contacts and arrange a meeting.

Project URL:

https://github.com/wiirijofct/SportMeet

Features not yet implemented:
- Photo upload for sports venues is missing;
- A manager is still unable to manage their spaces;
- The favorites page is missing;
- The option to send a message from the meet page is not yet connected to the chats page in the app bar;
- Displaying a host's fields on their homepage is missing;
- The profile page is not complete.

About the app:

Our application was built in Flutter, and we use a JSON file to mock the database, where we store information about users, fields, bookings, etc.

Apart from the issues mentioned as not implemented, there are some flaws in the application, such as:
- When sending a message, the chat is designed for a mobile application, meaning the only way to send a message is by pressing the send button.
- We attempted to implement both dark and light themes but faced difficulties selecting the colors to use. This implementation is something to be added in a future version.