meeTeam

App Objective

meeTeam is a social networking app with a twist. If Instagram, Facebook, TikTok, or BeReal are content-based, meeTeam connects people through sports.

**The app's main goal** is to make team sports more accessible by connecting users with other possible teammates and related business owners (football or tennis fields, basketball or volleyball courts, or even water polo swimming pools). Therefore, meeTeam **promotes team-based sports** amongst different age groups, all whilst **offering a platform for businesses**.

Short Description

All these are done by enabling people to create customizable sports events. The user is presented with the option of choosing the desired sport, based on which, a plethora of possible locations and available timeslots will be shown. Moreover, the event creator is in charge of selecting the number of teammates and opponents they’d like to play with/against. If needed more customization can be added to ensure that the event satisfies all the user’s needs and desires. After being published, an event can be joined by all other users who satisfy the criteria, given that there is still room for them.

Once an event is full, a reservation or full payment can be made for the location, given that it is still available. If the latter has already been booked for that date, alternative dates and locations would be presented.

Domain Details

meeTeam will have 3 main entities which will make up the majority of the app, those being **users**, **locations**, and **events**.

Event

* **Title**
* **Type of sport** that the event implies
* **Author** – the user that created the event
* **The date and location** that it takes place at
* **The price** of the location/player
* **Maximum number of participants**
* **Participants** and their associated profiles
* **Custom messages/requirements** if added by the organizer

User

* **Name**
* **Date of birth**
* **Favorite sports**
* **Ratings of different skills** based on previous team players’/opponents’ reviews

Location

* **Name**
* **Location**
* **Available time slots** in the next week and month
* **Reviews** grated based on previous users’ experiences

CRUD Operations

The most relevant such operations would be the ones involving **events** as that is the main focus of the app.

Create

* An event is created by a user and it must have the aforementioned details, including the sport, the date and the location if decided from the beginning, the maximum number of participants, and any custom requirements/messages if desired
* The price/person will be automatically computed once the location and the maximum number of people are chosen
* At this stage, the participants list is composed of only the creator itself

Read

* The read operation constitutes the action taken by all the other users when interested in an event as they can view all its associated details.
* On the main page, a list of upcoming events will be shown, segregated based on their date (This Week, This Month)
* A filter functionality will also be present, making it possible for users to see the events related to a specific sport for example

Update

* Events will be updated as users join, considering that the participants list needs to be kept up to date to ensure a smooth experience with the app
* An event can have the author changed in case they decide not to attend anymore but the rest still want to carry on with the event
* Once such an event is full, a reservation or payment should be made. In the event that the location is no longer available for the desired timeslot, the latter needs to be updated to an available one

Delete

* In case an event cannot go further due to a lack of participants, failure to find a location or simply due to a team agreement, it can be deleted by the author

Persistence Details

The **Create**, **Update** and **Delete** should all be persisted on the server as this will ensure a smooth UX for all users who then would be interested in viewing the changes. It wouldn't really make sense to have these operations persisted in a local DB as when synced with the server, multiple errors might occur (like the same location being booked for 2 events at the same time).

One operation which might be a good fit for having its persistence on the local DB would be the option of **Bookmarking an event** (saving the events that the user is interested in attending). This wouldn’t impact the other users as saving an event, does not mean they would also attend it. Therefore, such an operation could be persisted on the local DB and updated if needed on the server.

Offline Mode

Due to the nature of the app, which is to interconnect people and enable them to practice team-based sports, it **wouldn’t really make sense for meeTeam to be able to perform in an offline mode**. This opinion is based on prioritizing the UX but in order to exemplify an unfavorable scenario, the main operations should be discussed in great detail:

* It might happen that the location of an event does not have great service, hence why **viewing the events** happening that day/week (reading an event) should be possible when offline.
* When **creating an event**, most fields do not need to be synced, therefore they don’t need the internet. However, when choosing a date and a location, the app needs to present the user with valid choices which cannot be shown in an offline environment as the location could have already been booked. The only way this feature could work is if the event would be created without a location, saved on the local DB and when back online, the user would be prompted with choosing a valid location. After doing so, the event would be finally saved on the server.
* **Updating an event** means changing important information regarding an event that might affect others. If the author doesn’t get back online in a timely manner, all other players will be unaware of the changes brought to the event.
* **Deleting an event** implies thesame behavior as the previous operation which means it should only be performed online.