SportCenter Android Application

by Han Leonard UVT FMI, IE2, Gr1, Sg2

Programming for Mobile Devices

Phd. Lecturer Liviu Octavian Mafteiu-Scai

12.05.2021

Abstract

The application is a mobile app written in C# and XAML using the Xamarin platform, that has the purpose of filling the gap between sport centers and it's clients when a tournament for a specific sport is running.

Index

1 Introduction

1.1 What it is?

The project is a multi-platform mobile application (primarily Android).

1.2 What it does?

The application is a tool for organizing sport tournaments, generating brackets, following the results of them, and distributing this information to participants.

1.3 Who is it for?

It is intended for sport centers to help manage tournaments, primarily while also being suitable for any king of competition, people using it ranging from young people to old people from a broad range of professions, studies, and even languages.

2 General concepts

2.1 Usage

The application is intended to be used in **two different versions**:

1. the public version for the app store (allows just viewing)
2. the admin version (allows modification)

Either version allows access to the tournaments page, and all the pages corresponding to a tournament.

2.2 Architecture

The application was created with the Xamarin Forms app platform, using C# and XAML (for back-end, respectively front-end), with software architectural pattern MVVM.

2.3 Structure

The user application is structured into multiple Views, that represent the pages of the app that the user interacts with, and multiple ViewModels that handle the user input, interaction and information loading.

The user application communicates with a private API which runs on a server and handles information storing and serving.

2.4 Diagram

The application screens, and access to them is as presented in Fig.1.

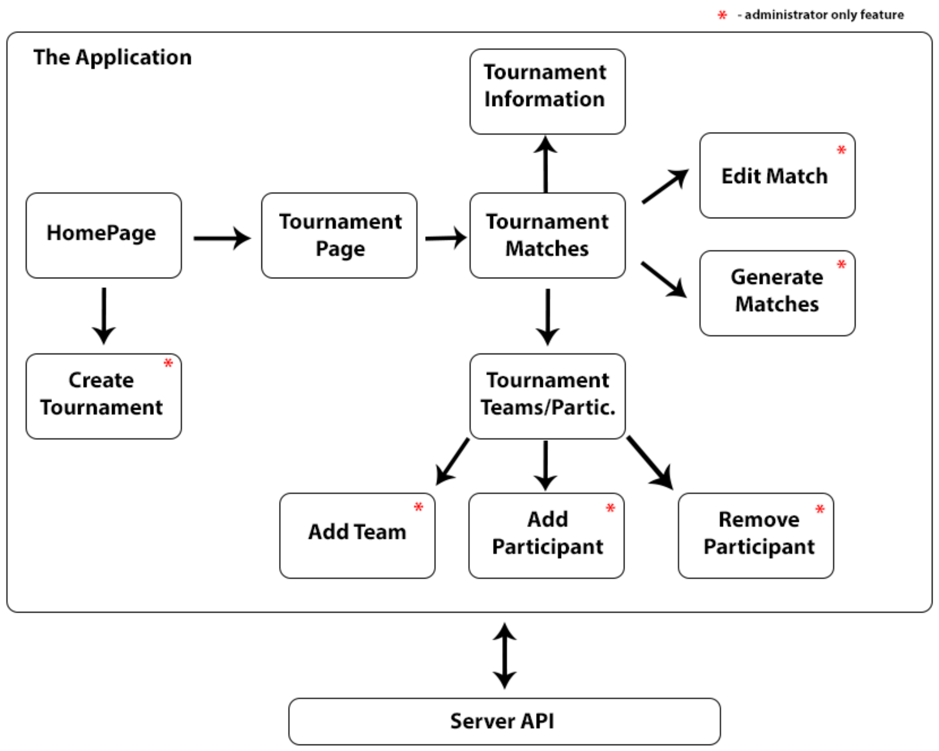


Fig. 1: Flow of access between pages

2.5 Originality

In order to stand out, the application covers a market sector that is almost empty, and helps solve it's main problem using modern means.

3 Pages

3.1 The home page

The homepage is the first page a user sees, on which all the tournaments are listed, with some representative information (see Fig.2).

Here administrators can use the + button to create new tournaments (see Fig.3).

When creating a tournament, a format for it must be selected, which will change how the matches will be created for it (see Fig.3).

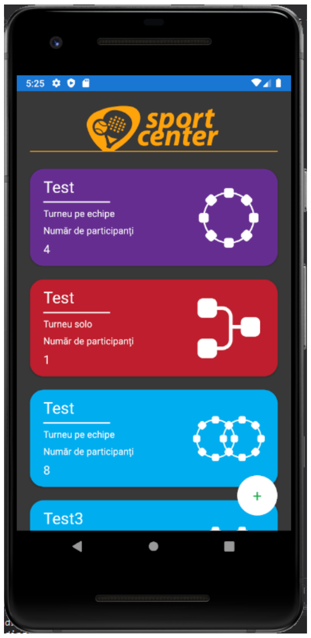
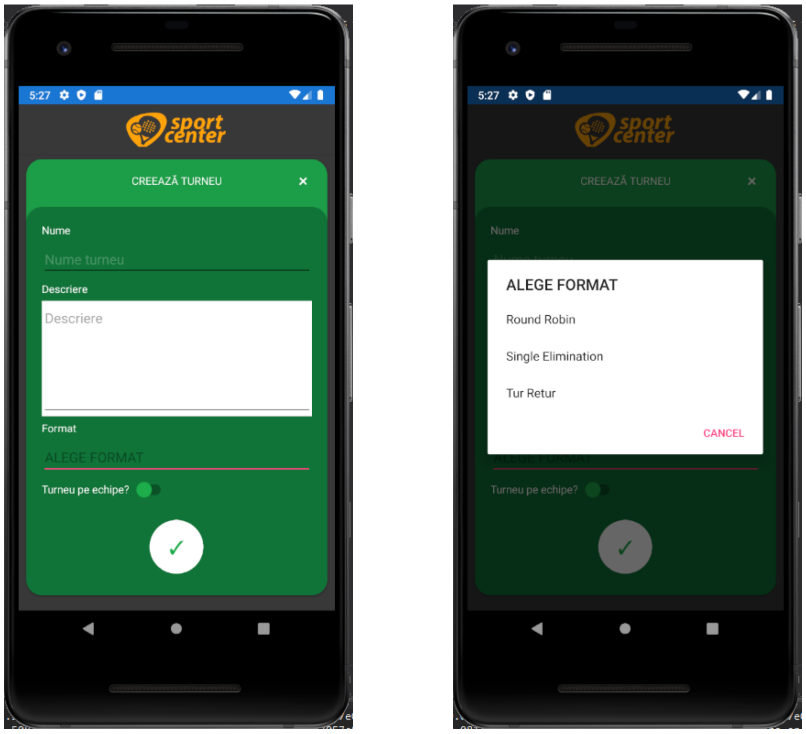


Fig. 2: Home Page



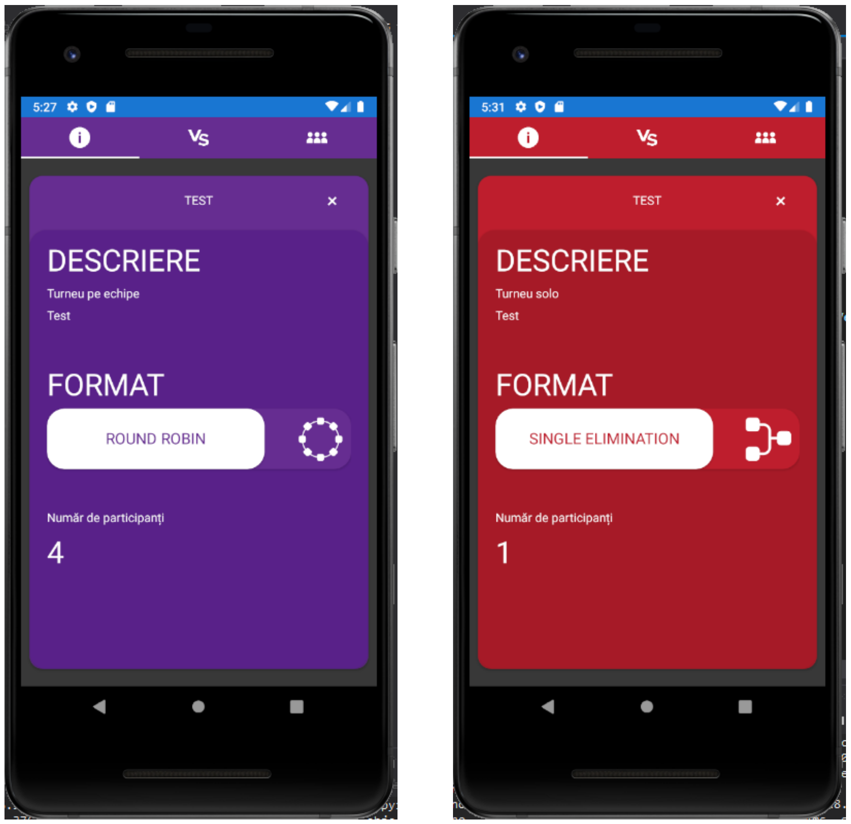
(a) Standard View (b) Format selection

Fig. 3: Create tournaments page

3.2 The tournament page

A tournament's page contains information about the the tournament, the list of matches, and the list of teams/participants (see Fig.4 a). Depending on the tournament type, this page can be colored differently, and is made with ease of use in mind (see the alternative Fig.4 b).

They are organized in three tabs, which can be changed by dragging on the screen, or tapping the tab.

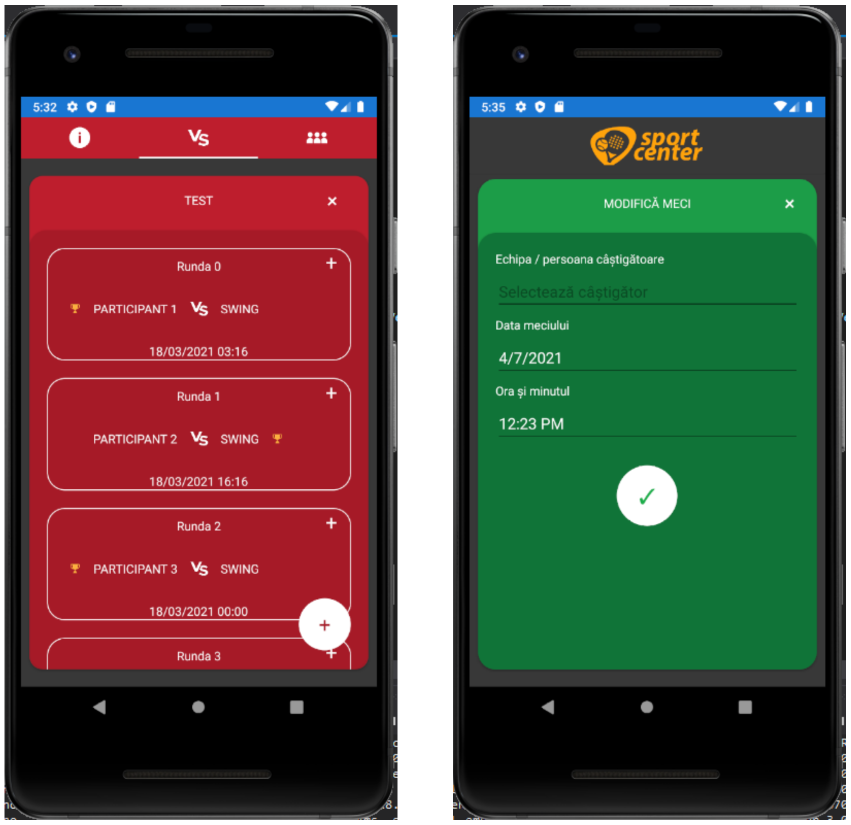


(a) Tournament Page for teams (b) Tournament Page solo

Fig. 4: Tournament Page

3.3 The tournament matches page

On this page (Fig.5) all the matches in the tournament are displayed, they can be generated, and be modified (to select a winner or change information).



(a) Standard Page (b) Page for modifying a match

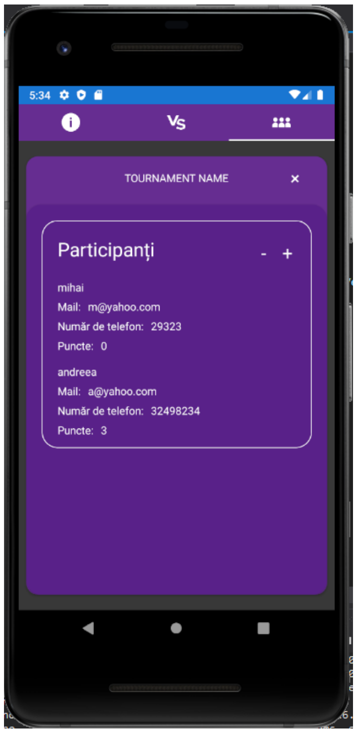
Fig. 5: Tournament matches page

3.4 The tournament teams/participants page

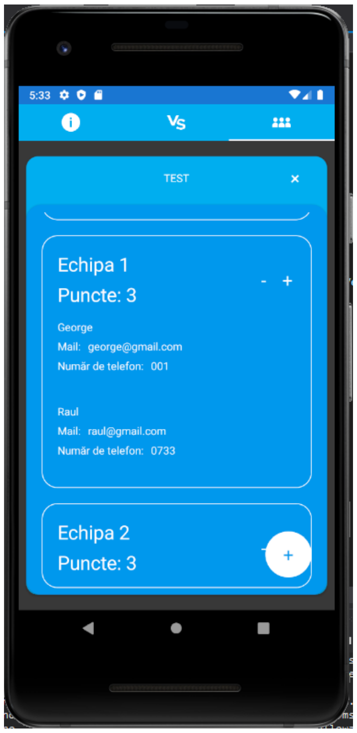
This page shows depending on the tournament type, either the teams and their participants, or the participants (for a solo tournament).

For solo tournaments, participants can be added or removed, while for team based ones, teams can be added with participants, the View presenting the information differently based on the tournament type (see Fig.6).

Adding a participant can be done using the + button (see Fig.7).


5:34
TOURNAMENT NAME
Participanţi
mihai
Mail: m@yahoo.com
Număr de telefon: 29323
Puncte: 0
andreea
Mail: a@yahoo.com
Număr de telefon: 32498234
Puncte: 3

(a) Page showing the participants of the tournament


5:33
TEST
Echipa 1
Puncte: 3
George
Mail: george@gmail.com
Număr de telefon: 001
Raul
Mail: raul@gmail.com
Număr de telefon: 0733
Echipa 2
Puncte: 3

(b) Page showing the teams in the tournament

Fig. 6: Tournament matches page

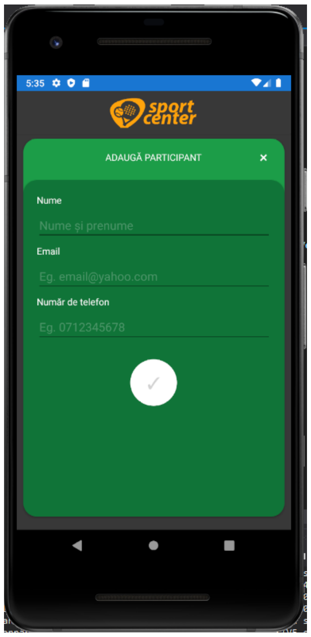
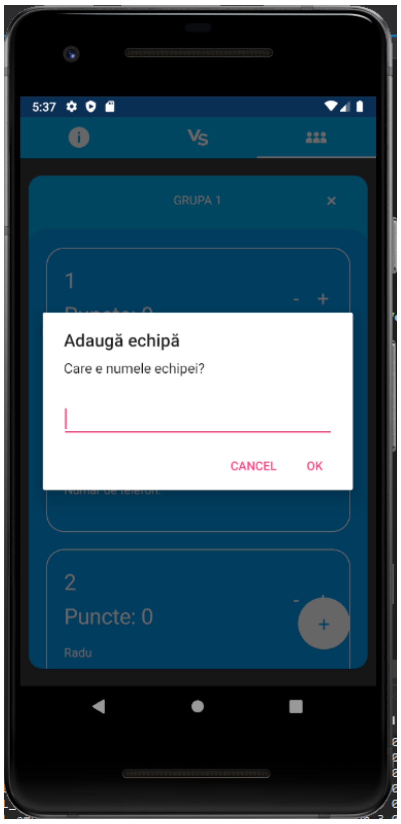

5:35
sport center
ADAUGĂ PARTICIPANT
Nume
Nume şi prenume
Email
Eg. email@yahoo.com
Număr de telefon
Eg. 0712345678

Fig. 7: Page for adding a praticipant

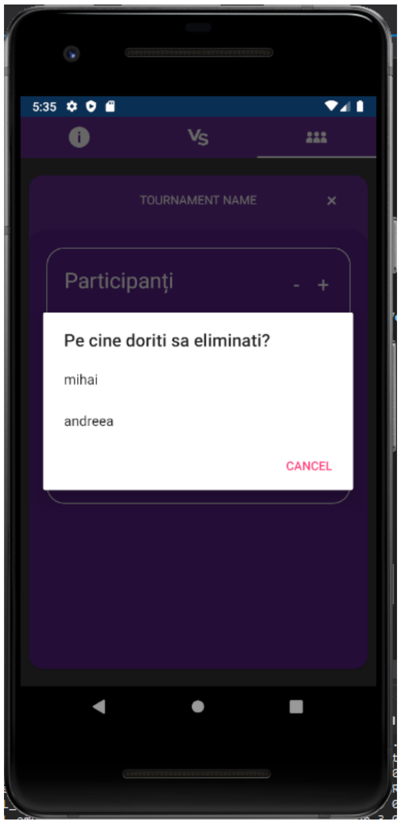
For removing a participant, you can use the - button adjacent to the team box, while for adding a team, you can use the + button at the bottom (see Fig.8).



Adaugă echipă

Care e numele echipei?

(a) Pop-up for adding a team



Pe cine doriti sa eliminati?

mihai

andreea

(b) Participant removal list

Fig. 8

4 Technical concepts

As stated in the previous paragraphs, due to the MVVM design pattern, the application is composed of Views, Viewmodels and Models.

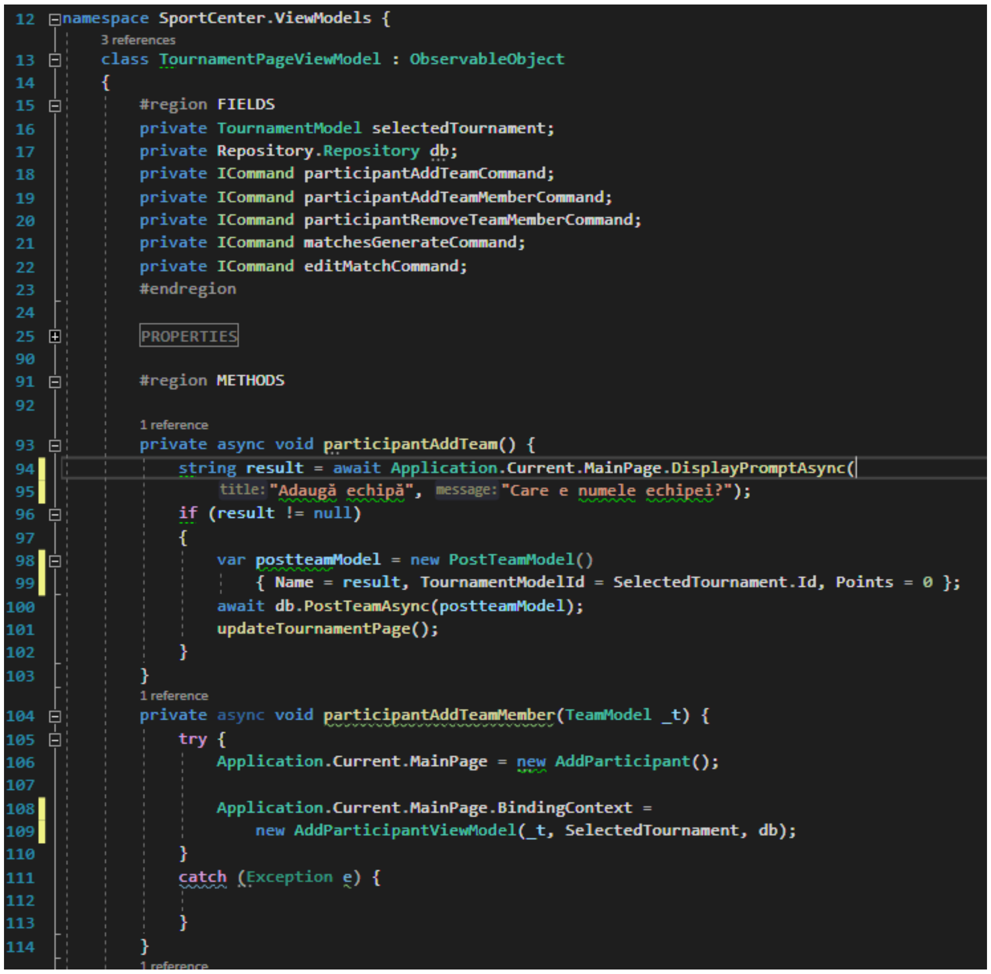
Below is a sample of code from a ViewModel, that handles interaction from all the Views that represent the tournament pages (see Fig.9).

You can see there the object containing the current tournament information, the database connection object, and the commands that correspond to actions associated to buttons in the UI.

The public properties hidden in the Properties block are public in the ViewModel object and are the ones that the View binds to.

The methods are called from the properties for the commands.

For example, the participantAddTeam method shows a UI prompt to enter a team name to add, and creates after a team object that is sent to the database.



```
namespace SportCenter.ViewModels {
    3 references
    class TournamentPageViewModel : ObservableObject
    {
        #region FIELDS
        private TournamentModel selectedTournament;
        private Repository.Repository db;
        private ICommand participantAddTeamCommand;
        private ICommand participantAddTeamMemberCommand;
        private ICommand participantRemoveTeamMemberCommand;
        private ICommand matchesGenerateCommand;
        private ICommand editMatchCommand;
        #endregion

        PROPERTIES

        #region METHODS

        1 reference
        private async void participantAddTeam() {
            string result = await Application.Current.MainPage.DisplayPromptAsync(
                title: "Adaugă echipă", message: "Care e numele echipei?");
            if (result != null)
            {
                var postteamModel = new PostTeamModel()
                    { Name = result, TournamentModelId = SelectedTournament.Id, Points = 0 };
                await db.PostTeamAsync(postteamModel);
                updateTournamentPage();
            }
        }
        1 reference
        private async void participantAddTeamMember(TeamModel _t) {
            try {
                Application.Current.MainPage = new AddParticipant();

                Application.Current.MainPage.BindingContext =
                    new AddParticipantViewModel(_t, SelectedTournament, db);
            }
            catch (Exception e) {

            }
        }
        1 reference
```

Fig. 9: ViewModel Example Code

5 Application Use Case Diagram (UML)

The application usage can be easily explained using a Use Case Diagram (see Fig.10). This UML diagram expresses the interaction between users and the application, by presenting all the uses of the app that they may come across.

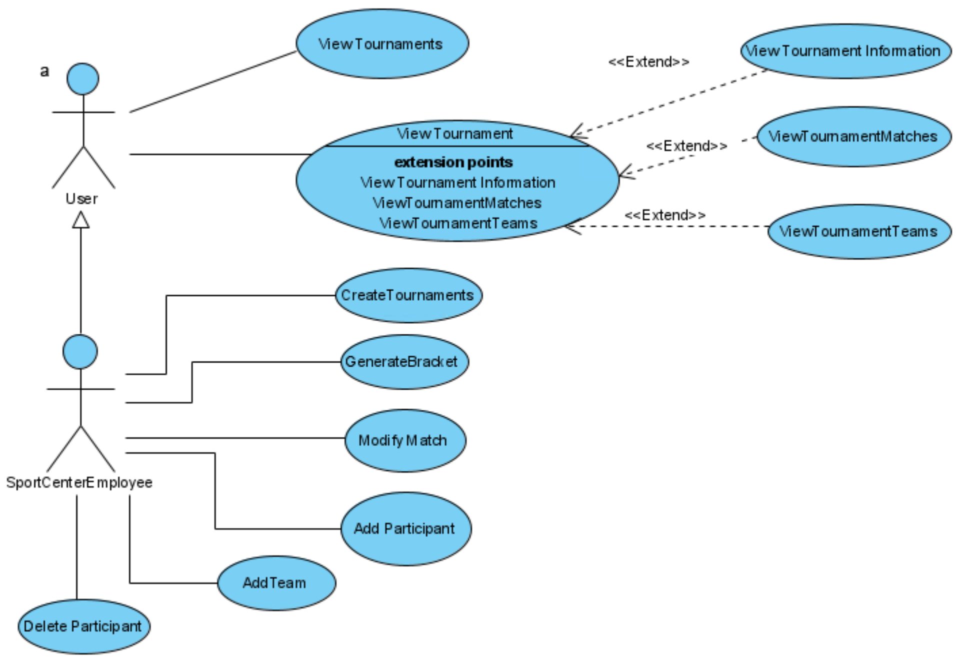


Fig. 10: Use Case Diagram

6 The sector and similar applications

The Fitness field has seen an incredible increase in user revenue and digitalization in recent years.

Although gyms started to adopt an Online presence, some even having mobile applications, their use is pretty low, as features are lacking from most of them.

What I propose is something currently missing from the market. Here are some examples of fitness applications:

6.1 UPfit.today

UPfit.today is a fitness app designed to be used by fitness gym customers in Romania (see Fig.11).

It currently has over 10.000 downloads, and a review average of over 4 stars.

The application can be used to check schedules, make reservations to classes, and check your membership.

The most relevant feature here is the one regarding the class reservations, which is a primitive version of the tournaments feature my application will contain.

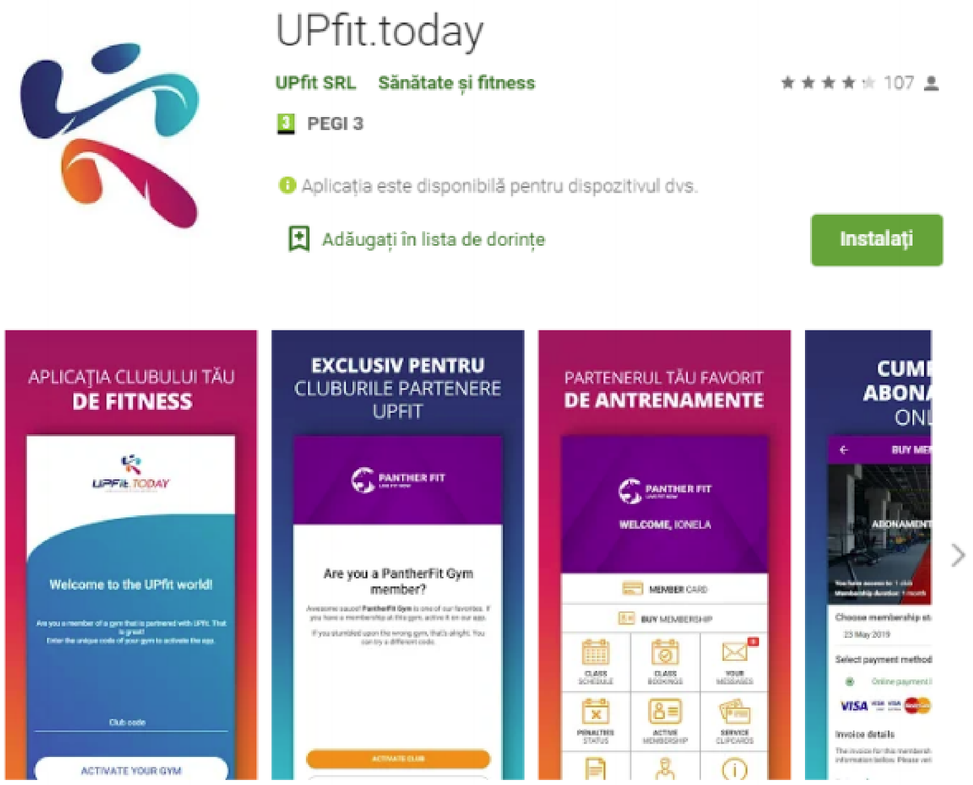

UPfit.today
UPfit SRL
Sănătate și fitness
107
PEGI 3
Aplicația este disponibilă pentru dispozitivul dvs.
Adăugați în lista de dorințe
Instalați
APLICAȚIA CLUBULUI TĂU DE FITNESS
EXCLUSIV PENTRU CLUBURILE PARTENERE UPFIT
PARTENERUL TĂU FAVORIT DE ANTRENAMENTE
Welcome to the UPfit world!
ACTIVATE YOUR GYM
Are you a PantherFit Gym member?
ACTIVATE CLUB
WELCOME, IONELA
MEMBER CARD
BUY MEMBERSHIP

Fig. 11: Upfit.today on the play store

6.2 Kingdom Sports Center

The app provides a lot of functionalities related to the tournaments it contains, such as schedules, contact, and team information (see Fig.12).

It has a decent amount of downloads (only 1000+), if you take into account it's usage, and cumbersome UI.

The app does not contain bracket generation, or a way to add tournaments and people, which is a big minus to ease of use.

7 Conclusions and future work

As a conclusion, the app covers a need in the fitness and sport sector that is pretty much neglected currently.

Future updates to the functionality are not planned, but could be done regarding the team representation in the app, and the bracket generation.

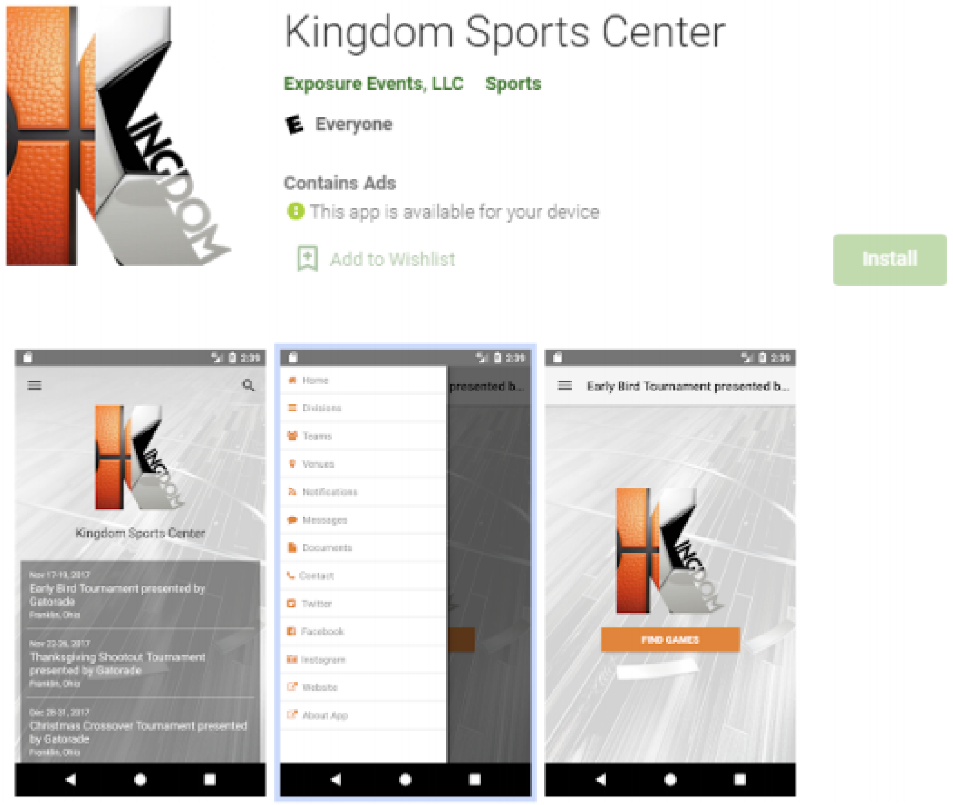


Fig. 12: Kingdom Sports Center on the play store

8 References

- https://play.google.com/store/apps/details?id=today.upfit
- https://play.google.com/store/apps/details?id=com.exposure.kingdomsportscenter&hl=en_US&gl=US
- https://docs.microsoft.com/en-us/xamarin/xamarin-forms/platform/
- https://material.io/design/color/the-color-system.html#tools-for-picking-colors