DAM - Trabalho Prático

Licenciatura em Engenharia Informática e Multimédia

Hoops Dynasty

Turma 61D

Pedro Azevedo A47094

Engenheiro: Pedro Fazenda

Instituto Superior de Engenharia de Lisboa, July 5, 2023

Índice

List of Figures

1 Introduction

In this semester in `Mobile Application Development`, we have learn how to develop Android applications with Android Studio and Kotlin. For the project it was propose to develop an Android app following a certain methodology with the UX Design in the center of the process. It is suppose to go throw four phases, the Concept, Pre-Production, Production and the Post-Production.

2 Objectives

To develop the application, firstly there is a need to learn some contents to use, like Firebase, Room Database and Jetpack Compose for the UI. After that, and following the methodology, there will be an app in the end.

3 Development

3.1 Concept Phase

3.1.1 General Concept and Name

The application is a NBA Manager simulation game, that allows users to step into the role of a NBA team manager. Users will be able to build and manage their own team, making strategic decisions, and play against other teams in simulated games.

The objective of the app is to create a realistic an engaging experience that is to be a manager in a NBA team, providing users with the opportunity to come up with strategies in the team-building and games tactics. By offering a simulation engine that is influence by the players performance in the real NBA league. With a final goal to incorporating an online multiplayer feature, with head-to-head matches and leader boards. Being the target users basketball fans.

Because the purpose of the app is the user to build a team and manage it creating their legacy, the name of the app is **Hoops Dynasty**. In the figure 1 is the logo of the app build with the Bing AI.

Figure 1: *Logo*

3.1.2 Main Characteristics and Functionality

The app is composed by a *Manager* that will manage a NBA *Team*, each *Team* made up of 10 *Players*. There will be a *Season* where each team will play against each other.

The *Manager* is able to set up his starter players and the bench players. Besides that, there will be a list of players that the *Manger* is able to purchase, with virtual money, that is earned after playing a match, or selling players.

Each game will be simulated based on the players on the field, and to make precise, each player as his position in the court and the statistics that have been taken from the NBA API. So to simulate a play, it will see the statistics of each player and the stats that are more relevant of each position.

The app will adopt a free-to-play model with optional in-app purchases, which users can buy virtual currency to buy players.

For future updates, the app will include multiplayer features, where users can play against each other. Improving of the simulation, updating the statistics of the players and a training camp for the players to improve theirs skills.

3.1.3 Wire-Frame

To star the design phase, it is good to start with wire-frames, in the figure 2 and 2 is the initial wire-frame on the app.

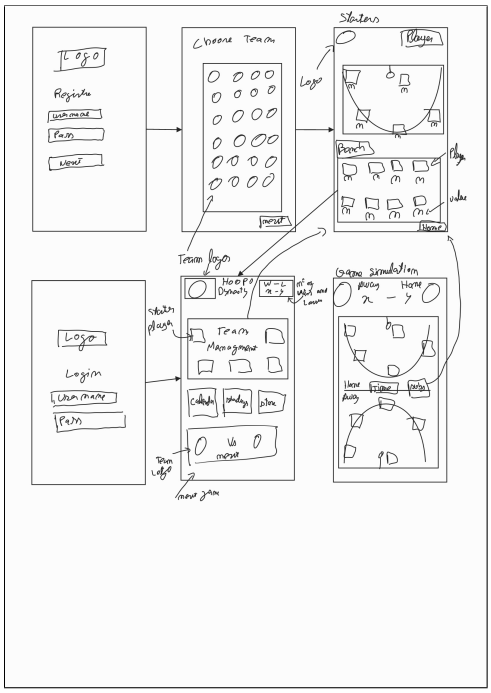

Figure 2: wire-frame 1

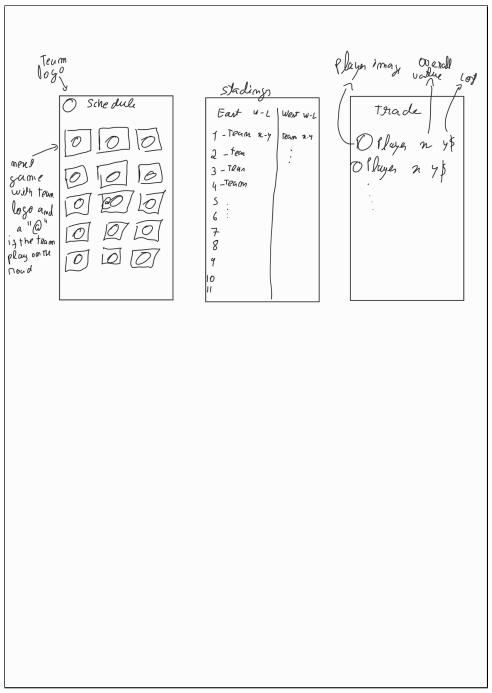

Figure 3: wire-frame 2

3.1.4 Entity-Association Diagram

In the figure 4 is possible to see the Entity-Association Diagram for the database to use, where there is a Manager that has a Team, each Team has Players and Games to play, each season has Players, Teams and Games.

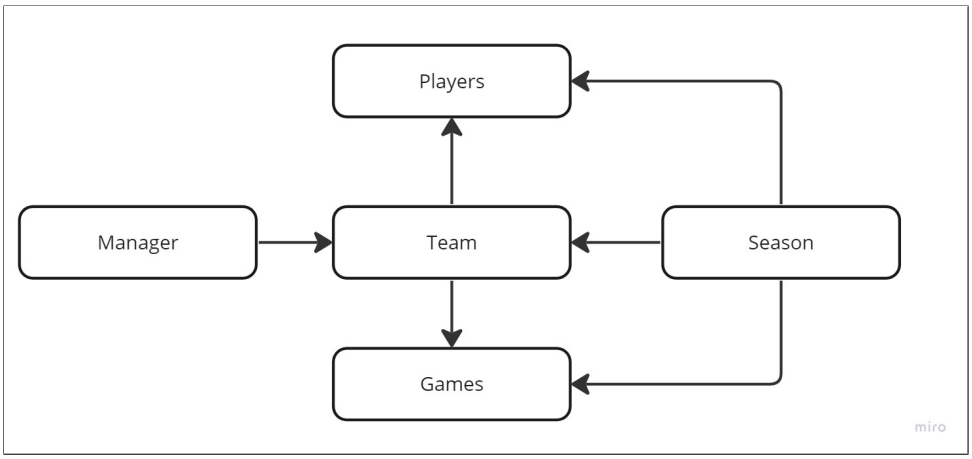

Figure 4: *Entity-Association diagram*

3.1.5 Firebase

To store and manage data and user it was created a Firebase project with Authentication, where is possible to register users, and Realtime Database, to store the state of the game, here is store in a Json format. Applying the dependencies to the gradle, is possible to register and log in to the user within the app, and sabe the data throw code, where to write and read it is necessary to serialize the data.

3.2 Pre-Production Phase

3.2.1 Mockups

With a concept created and an idea of how the app will be design, it is essential to create mockups, therefore I created 3 similar mockups with changes in fonts, and colors, as it is possible to see in 5.

Figure 5: *Mockups*

It was ask to some users which one they prefer, its more comfortable, more appealing, and this was the result.

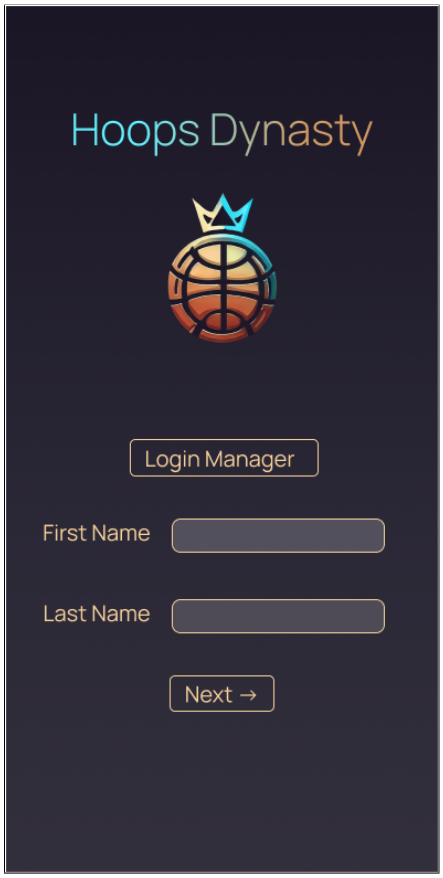

Figure 6: Login

Figure 7: Select Team

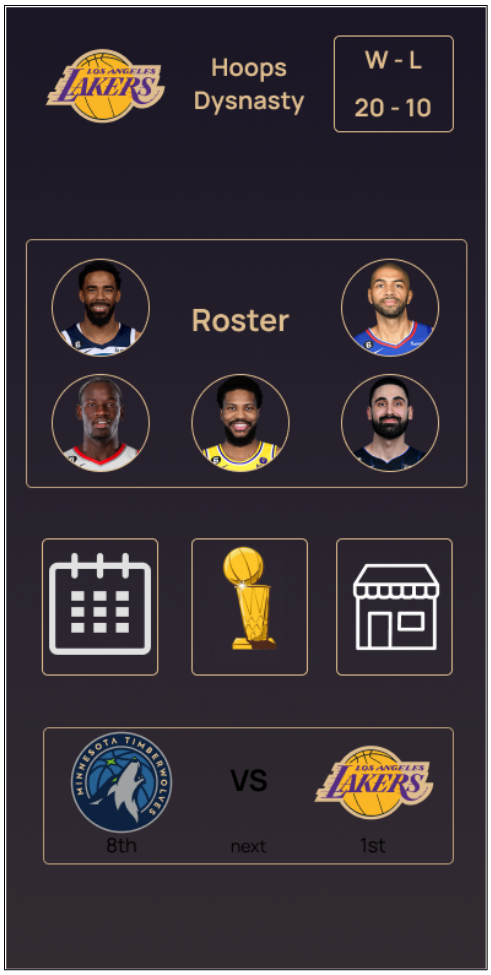

Figure 8: Home Page

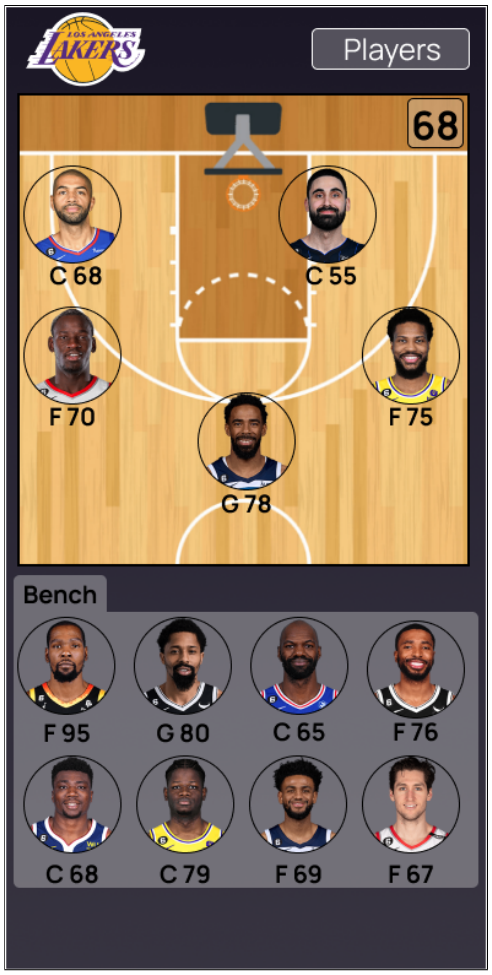

Figure 9: Roster

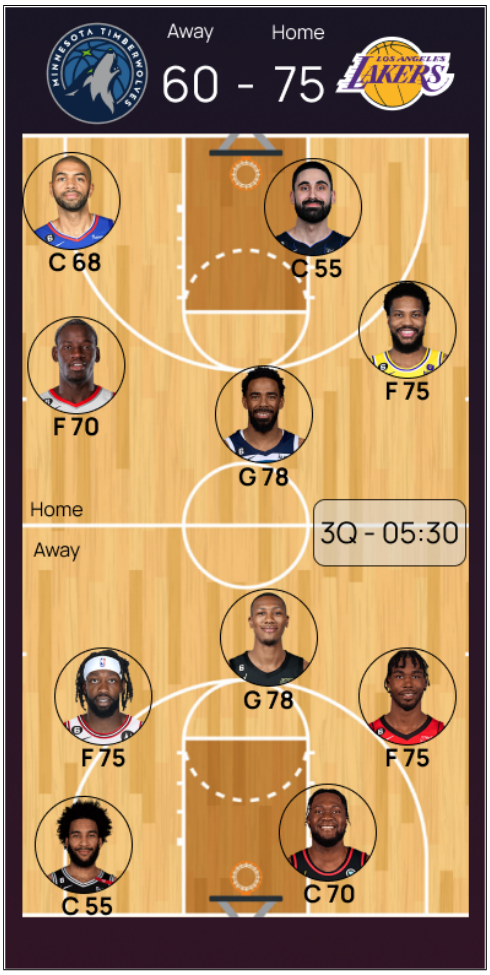

Figure 10: *Game*

3.2.2 Full Entity-Association Diagram

Knowing more about the concept and how the application will work, its possible to define the attributes of the Entities, in the figure 11, its possible to see it all.

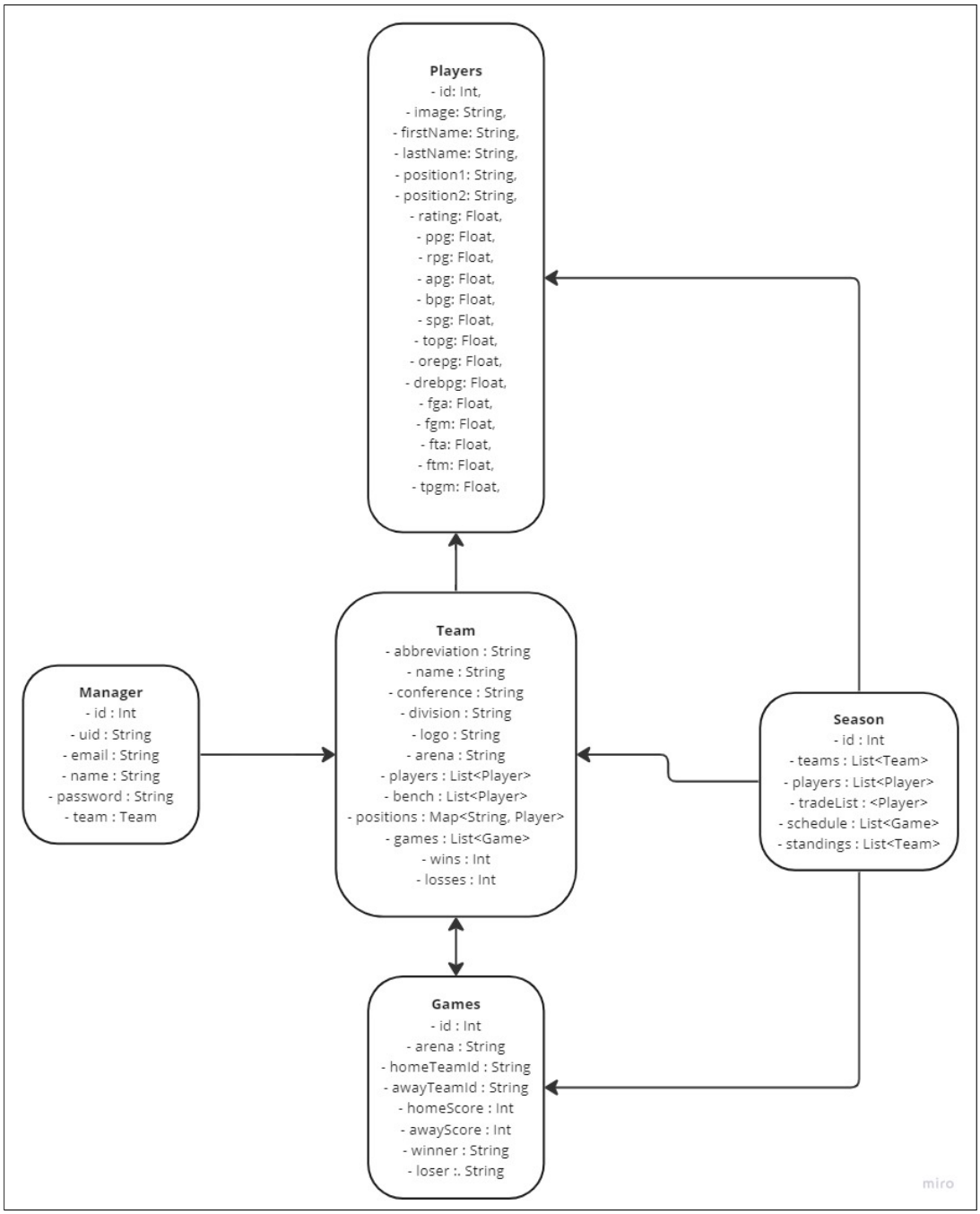

Figure 11: *Entity-Association Diagram with Attributes*

3.3 Production

3.3.1 Database

To start developing the App's database, it was decided to use Room, is a persistence library provided by Google as part of the Android Jetpack set of libraries. Room simplifies the process of working with databases by providing compile-time checks for SQL queries and mapping database tables to Kotlin objects.

With the objective of having a clean architecture and organization in the code, it was created a directory for the data as you can see in the figure 12, and to build a Room database it is need the following:

- Entity - for each object to create a class representing a table in the database;
- DAO (Data Access Object) - to define database operations like insert, updating, deleting and querying;
- Database - an abstract class that represents the database itself.

To access each DAO it is created a class Repository where the functions dealing with the logic of the app will access, avoiding contact directly with Room.

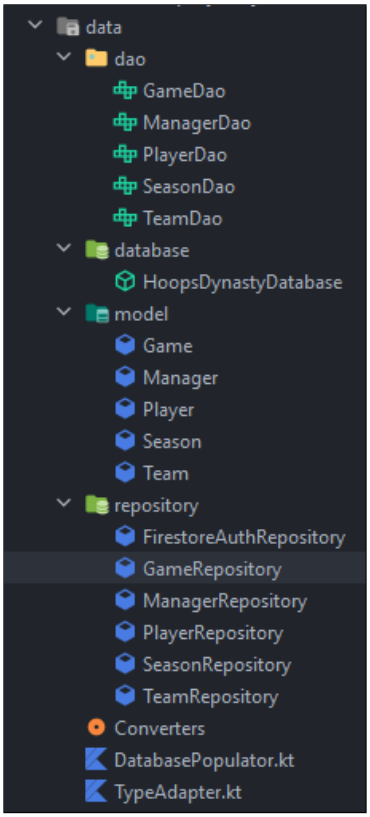

Figure 12: Data Directory

3.3.2 Model-View-ViewModel

As already refered to, its a good practice to have the code and organized and well structure, and for that using a Model-View-ViewModel (MVVM) architecture pattern is a good way to achieve that. Here the user interface (View) is separated from the logic and data (Model), using a intermediate layer, ViewModel that exposes the data and operations required by the View, and provides user actions. Whit this, the ViewModel retrieves data from the Model, processes it and presents to the View.

For the Model in this project it is used the Room database talked in above. As for the View, Jetpack Compose.

3.3.3 View - Jetpack Compose

The Jetpack Compose is a UI toolkit for building native Android apps, it simplifies and accelerate the process of creating user interfaces with a declarative programming model, instead of XML.

3.4 Final Result

Following the flow of the application:

When the user opens the app, the name "Hoops Dynasty" appears in the middle of the screen and slides to the top showing the `Login` page (figure 13), if the user does not have an account, it can create on (figure 14).

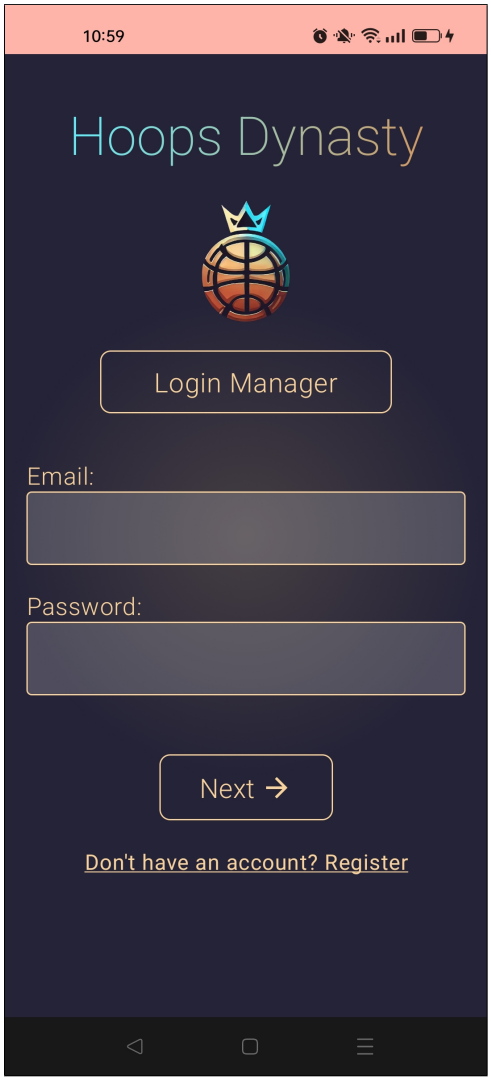

Figure 13: Login Page

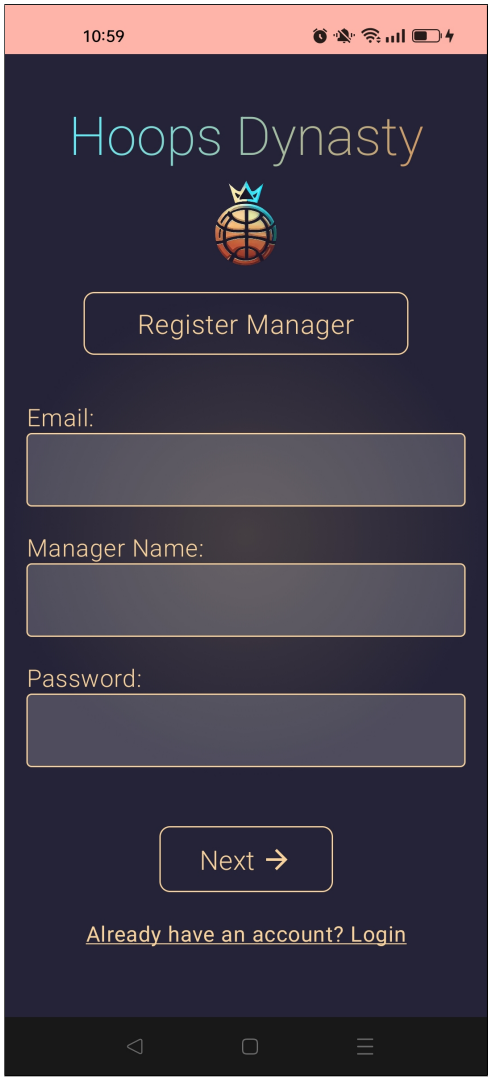

Figure 14: Register Page

After the user inserts the data to register he as to choose a team to be the manager (figure 15) from the 30 NBA teams, by sliding horizontally its possible to browse throw the teams, by choosing one, the account is created and saved in the firebase. following that, his roster will be shown (figure 16), here the user can, by dragging and dropping the players switching the position. Each player has a rating number associated depending on his performing, and the circle color of the image is according to the value of the rating.

Figure 15: Choose Team Page

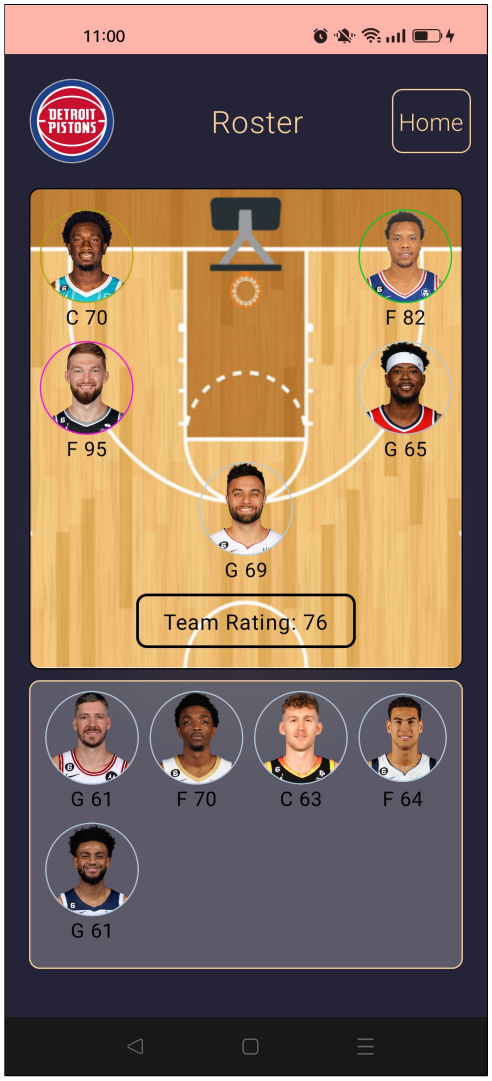

Figure 16: Roster Page

By logging in or finishing the register, the user is navigates to the Home page (figure 17). In the Home page the user can see information and navigate throw the app. Fist the user is able to see the logo of the Team he chose, the team numbers of wins and losses, the starting five and the next game. By clicking in the Roster, the user can edit the roster as it was shown in the register,(figure 16. Going to the Calendar, it will be presented the team's next games 18.

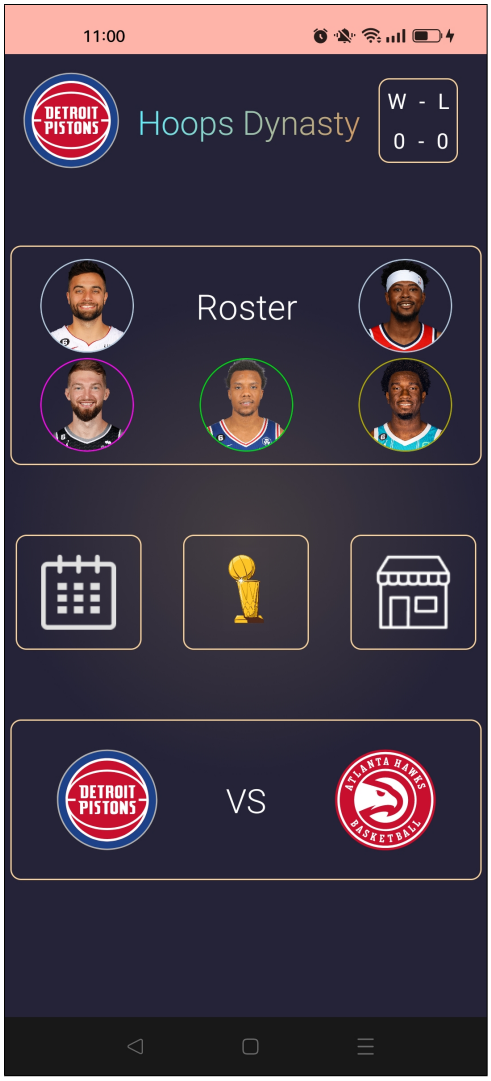

Figure 17: Home Page

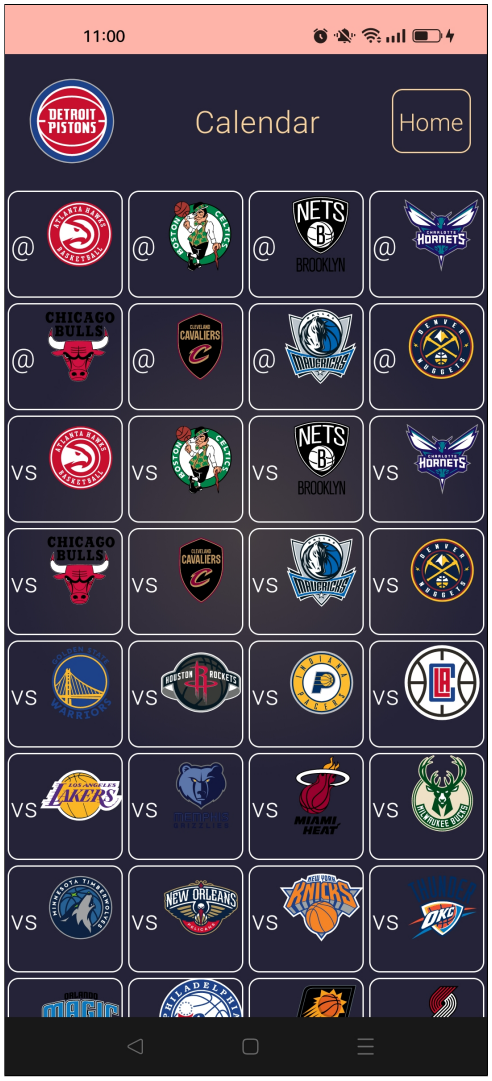

Figure 18: Calendar Page

By clicking the Championship[1] the user access the standings (figure 19). Clicking in the store icon, the user navigates to the Marketplace, where there is a list of players that are in neither of the teams, and the user can buy them (in this version, this feature is not implemented). The price of each player varies depending on the the rating value (figure 20.

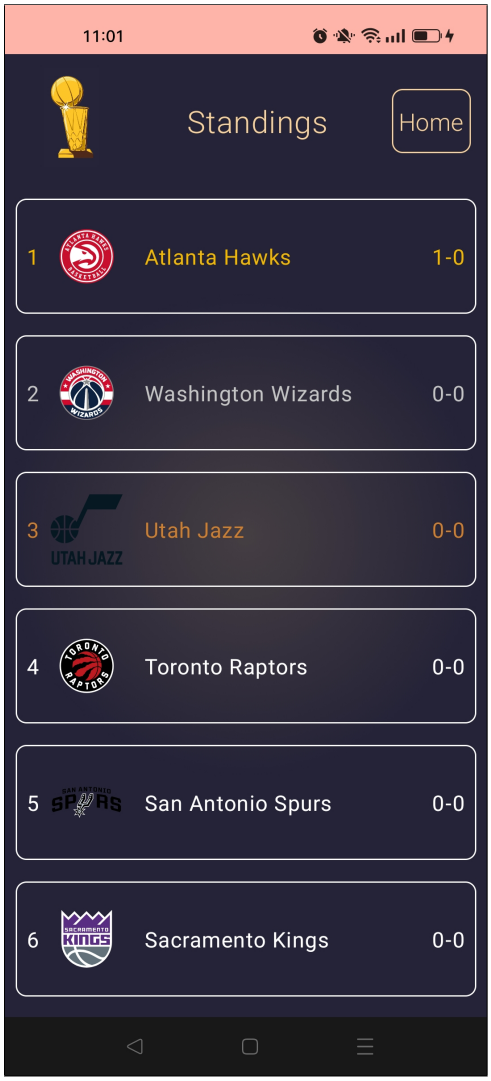

Figure 19: Standings Page

Figure 20: Marketplace Page

[1]The icon in the app is the Larry O'Brien trophy of the NBA

Finally, the user can simulate the next game, by pressing the button it will go to the Game Page (figure 21), it shows the result in real time of the simulation, as well as the time passing, additionally shows the starting 5 of both teams. When the game is over, it will show the winner and the final result as it shows in the figure 22.

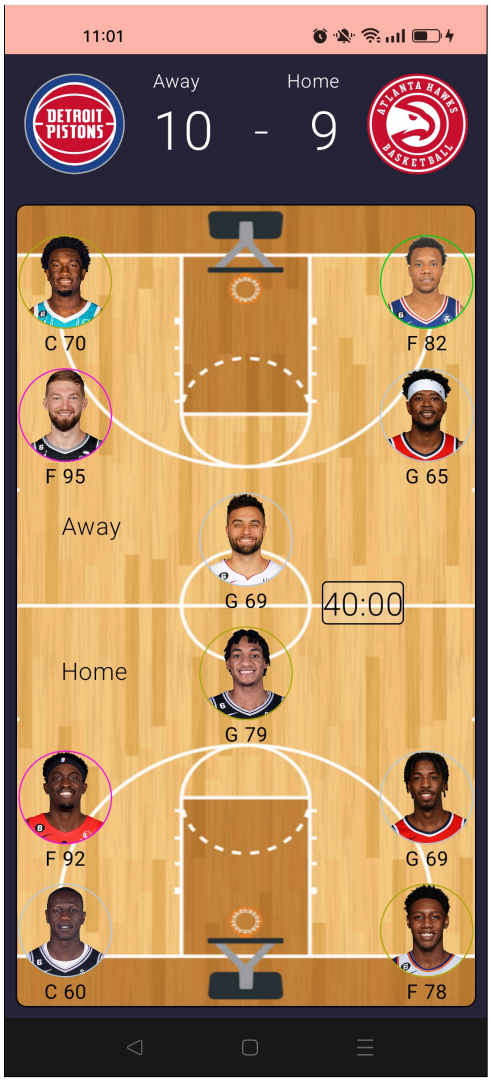

Figure 21: Game Page

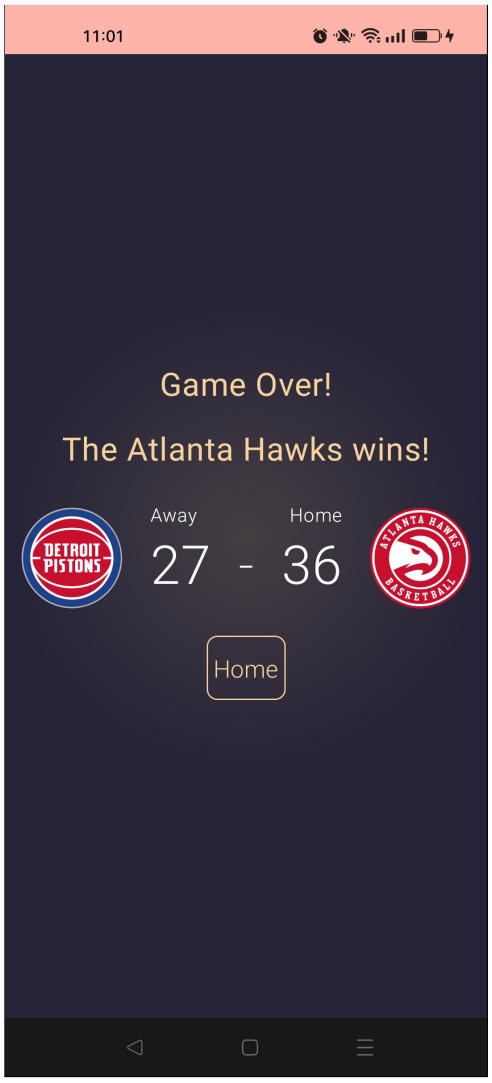

Figure 22: Game Over Page

4 Conclusion

Concluding this semester, I'm able to develop an Android application, using tools that speeds up the process, and gets the code gets better organized like Jetpack Compose, MVVM architecture. About the backend, it gets really easy to use Firebase, however, to interact with data I use Room where is also simple, but also, sometimes, comprehending the current state of Live Data can be complex.

Some difficulties and parts that took more time encountered, was getting all the data of the players, the stats and the images. Also, in the beginning, learning how to use the Room database, and retrieving the data through the ViewModel. Additionally, implementing the drag and drop of the players in the roster, was also complicated, updating the screen as the player is drop.

In the end, I manage to develop an Android App where the user is putted in a NBA Manager shoes and has team with players, being able to build the line-up, see the calendar of the next games, see the standings, see the players in the market and simulate a game.

There were some features that remained unimplemented, notably the earning money when a game is over, the purchase of a player, finishing the season. The simulation of the game is not that precise.