MobileApps: Assignment 2

Gabriel DEZON
ID: 74533

Teacher: BJ Roche

April 2023

1 Screens

1.1 The Wireframe

The first thing I did after reading the guidelines was create a wireframe using the Figma application. Here are the wireframes I created for the two pages.

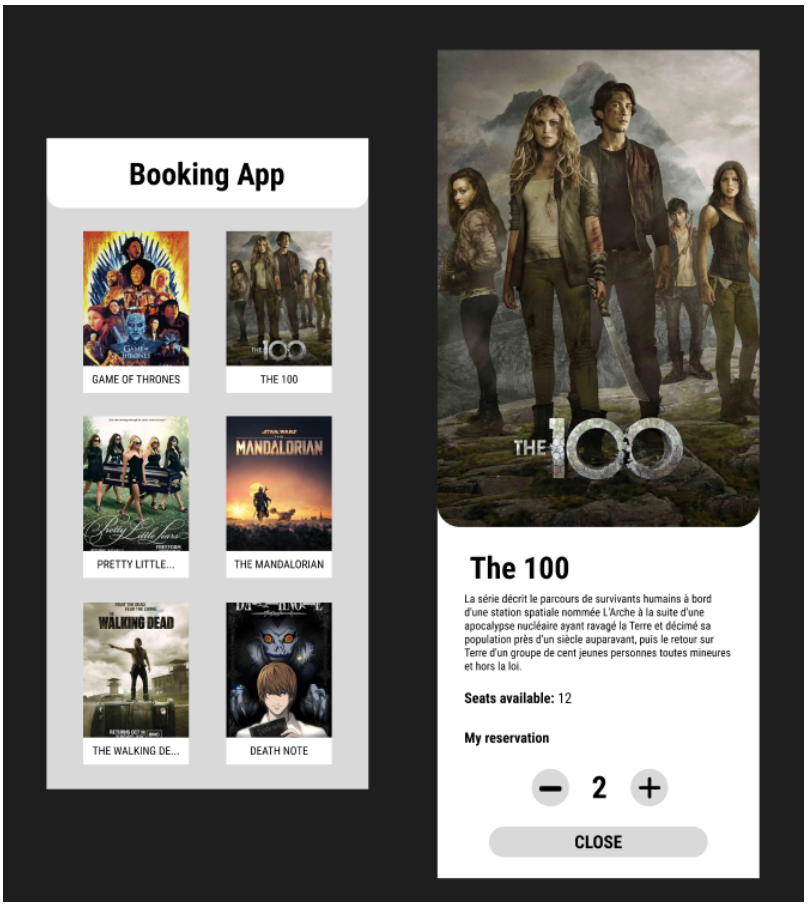

Figure 1: Figma Wireframe (Medium Fidelity)

1.2 Main Screen

As you can see, in the first screen we have an **AppBar**, with the title of the app displayed on it.

The body is made by a grid of two columns, where i display some cards representing films. These cards are made of *Boxes*, where we can see the poster images of the movies and their titles.

All the boxes are clickable and are made to redirect to the details of the movie (in another screen).

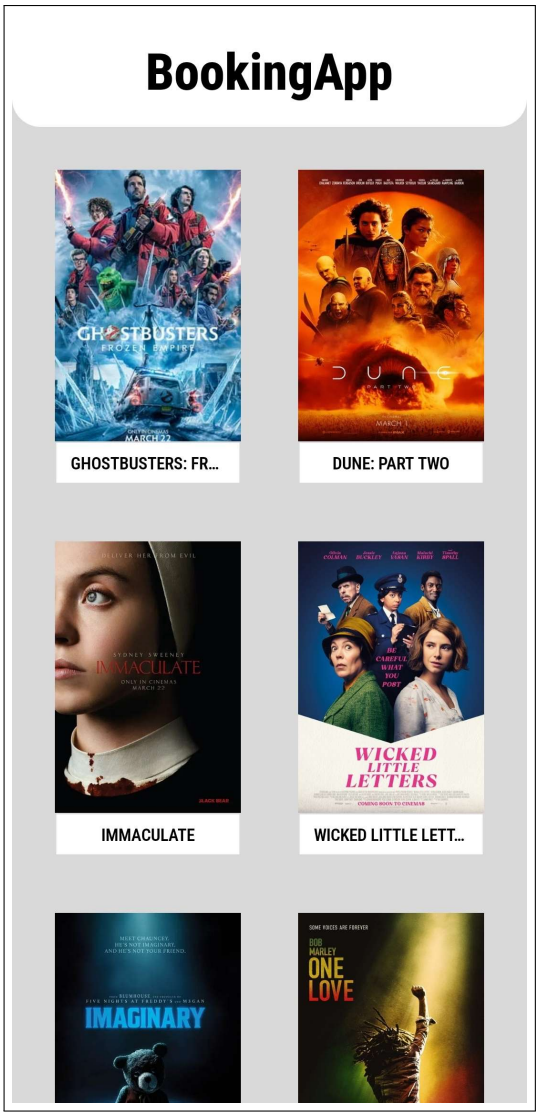

Figure 2: Main Screen

1.3 Details Screen

On the details screen, we have all the details of the movie asked in the guidelines:

- the poster image
- the title
- the description
- the certification
- the actors
- the number of seats remaining
- the number of seats reserved

We have a buttons to change our reservation (changed the number of reserved seats), displayed in a row.

At the end there is a button which allow us to return to the main screen.

Figure 3: Details Screen

2 Navigation

To make easier the navigation, I used a NavController. To do that I have two classes, the class *Screen* with the routes (two routes) of all my screens. The second class, *Navigation* handle the NavHost and the NavController. By the way, it's in this class that i load my films.

Using my route, I pass a parameter to my details screen to know which movie I shoud display. The parameter is the index of the chosen movie in the movie List that I initialised in my nav.

3 Features & Problems

3.1 Mutable Text

To allow the screen to change the text of the remaining seats and the selected seats when I change them, I had to use a *MutableState¡Int¿* variable. It allows me to reload the screen when the variable change and so helps me to show to the users every changees.

3.2 Images display Problem

I can't choose the initial width and height of the images I use. The problem is that when I want to modify the image it keep the proportions and don't display correctly.

It's even more visible when I try to expend the image.

3.3 Font Family: Roboto Condensed

As asked in the guidelines, I changed the font to Roboto Condensed. In order to do that, I download all the *ttf* files that I needed from Google Fonts.

After that, I created a FontFamily element to use the new font I got from the *ttf* files.
To do that I had to fix an error, because to import the *ttf* file in the code, they must contain no uppercase and no dash, which where present when I

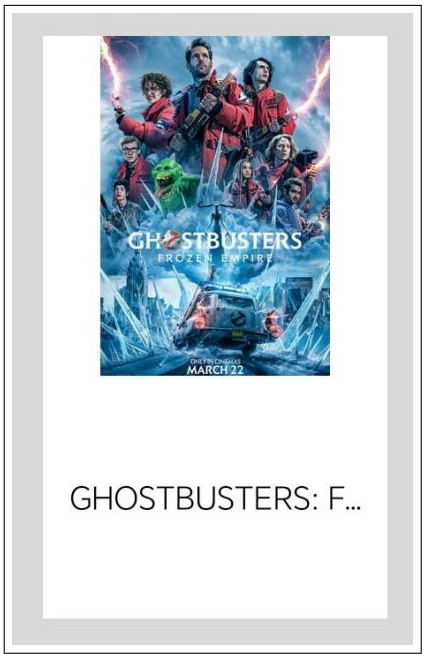

Figure 4: Image display problem 1

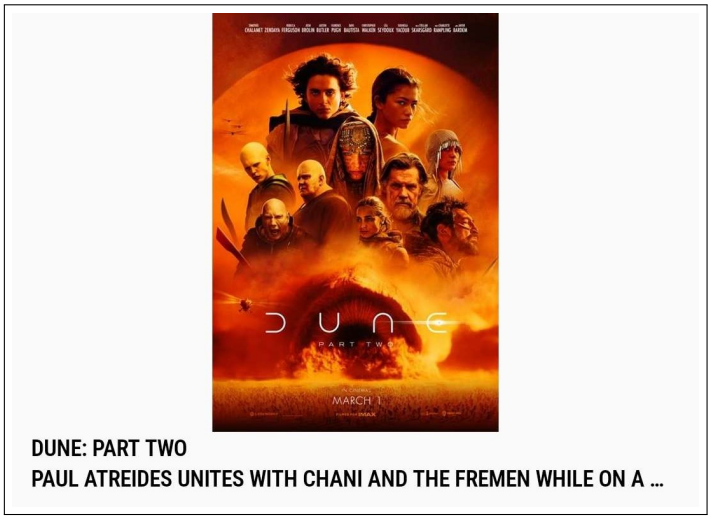

Figure 5: Image display problem 2

downloaded them.

3.4 Images From URL

At the begin, I used images I had downloaded before. But to comply with the instructions I had to use the images from Internet via an URL, so it's what I need. But even with the good code for importing image via URL (using *AsyncImage*) didn't worked.

I had to have made a mistake, and it was the case. I forgot to allow my application to access to Internet. So I add that line in my *AndroidManifest.xml* file:

```
<uses-permission android:name="android.permission
    .INTERNET" />
```

I had these imports in my *build.gradle* file.

```
implementation("io.coil-kt:coil-compose:2.6.0")
```

3.5 Saving my class in JSON

At the end, I manage to save my movies into a json file and load it from the same file. It is really useful because it allows me to keep my data even when I close the app. The second advantage is that if i need to change the data, the only thing I need to do is to provide another json file.

If the file doesn't exist, I create it with y own data.

I imported this module in the *build.gradle* file.

```
implementation("com.google.code.gson:gson:2.8.8")
```

The JSON file follow the model of the guidelines, like that:

```
{
    "name": "String",
    "image": "String",
    "certification": "String",
    "description": "String",
    "starring": "String Array",
    "running_time_mins": "Int",
    "seats_remaining": "Int",
    "seats_selected": "Int"
}
```

3.6 Selected seats in the Main Screen

I display the number of selected seats in the MainScreen, but only when it's different of 0. Else it displays the remaining seats.

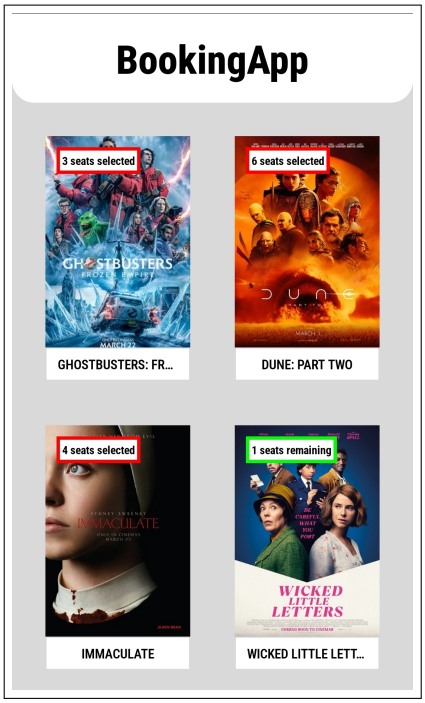

Figure 6: Selected seats on MainScreen

List of Figures

Sources

- Google Fonts
- Android: Coil
- Github: Coil
- Android: Connecting to a Network
- Android: Navigation
- Philipp Lackner: Styling Text
- Wingineers: JSON Data Parsing
- DeepL
- MyVue

PS: All items are clickable