Mobile Apps
Assignement 3

Mehdi AZOUZ – 74531

Teacher: BJ Roche

April 2023

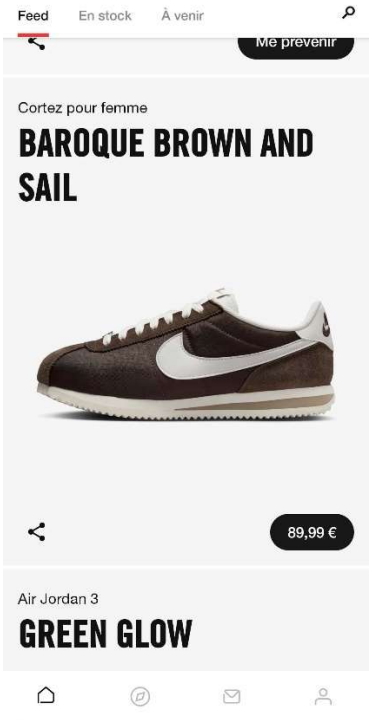
Presentation of the project:

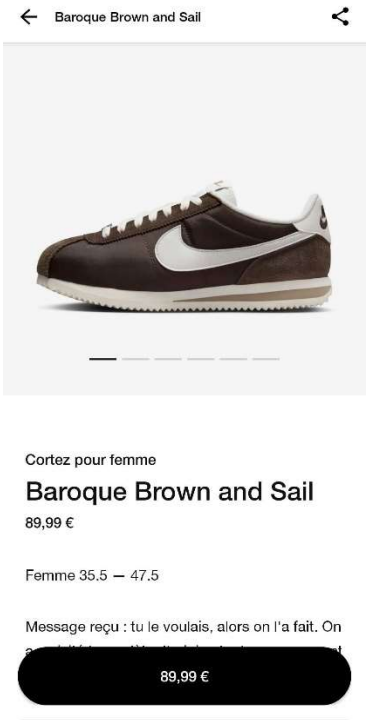
My application is called "Watch and Shoes", which reminds us of "Watch and Choose". As the name suggests, my shop sells only quality watches and shoes. Here's an image of the app's logo :

Logo of *Watch and Shoes*

Wireframe :

For the wireframe, I was inspired by the *SNKRS* Sneakers application developed by Nike. I liked the way their application was made, so I took some inspiration from it. Here are some images from the *SNKRS* site:



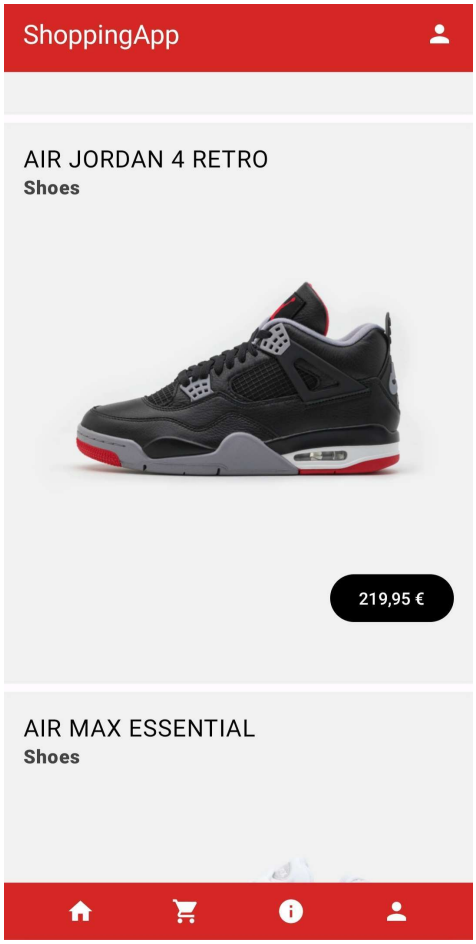


Home Page :

As you can see, we have an app bar with the project name and a button to redirect to the profile page.

Then in the body, we can see that each element is organized in a card with the product name, category, image and price. Clicking on the card or the price button redirects you to the product page with more information.

Finally, we have the NavBar, which redirects to the various pages.



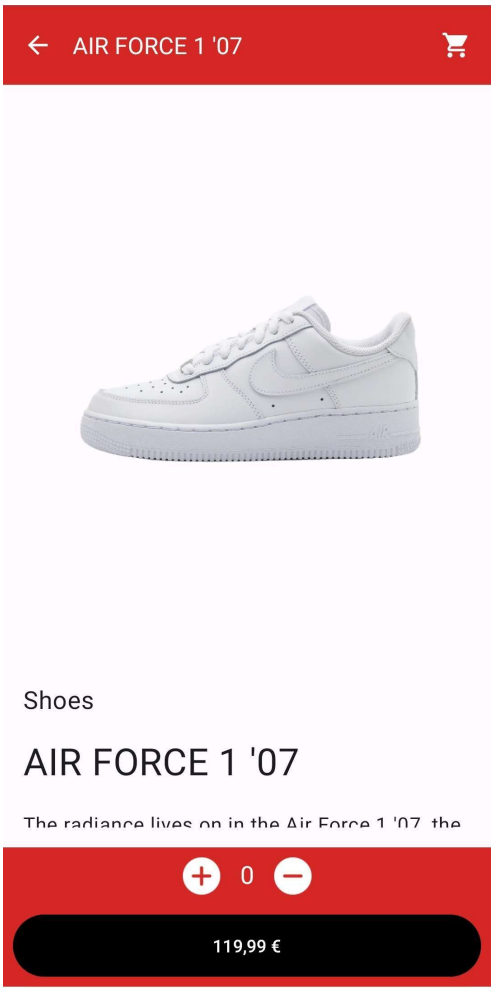
Product page:

In this page, the user can the product detail and buy it.

The App Bar consists of 3 elements: a back arrow to return to the home page, a text with the product name and finally an icon button to access the Cart page.

The body is made up of the product image, its category, its name and a short description of the product.

The Bottom Bar consists of 2 buttons + and - to adjust the number of products the customer wishes to purchase, and a button to validate.



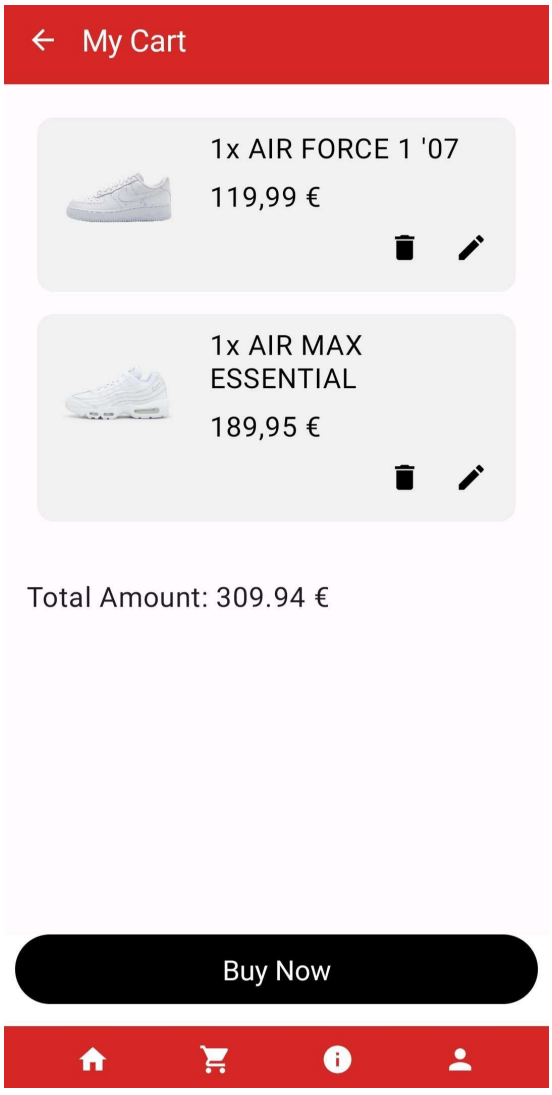
Cart Page:

On this page, the user can see the details of what he wants to buy before paying.

The App Bar consists of 2 elements: a back arrow to return to the home page, and a text.

The body is made up of several cards with the image of the product, how much he has bought, its price and 2 buttons to modify or delete the item.

The bottom bar is made up of a buy button and a navBar to direct visitors to different pages.



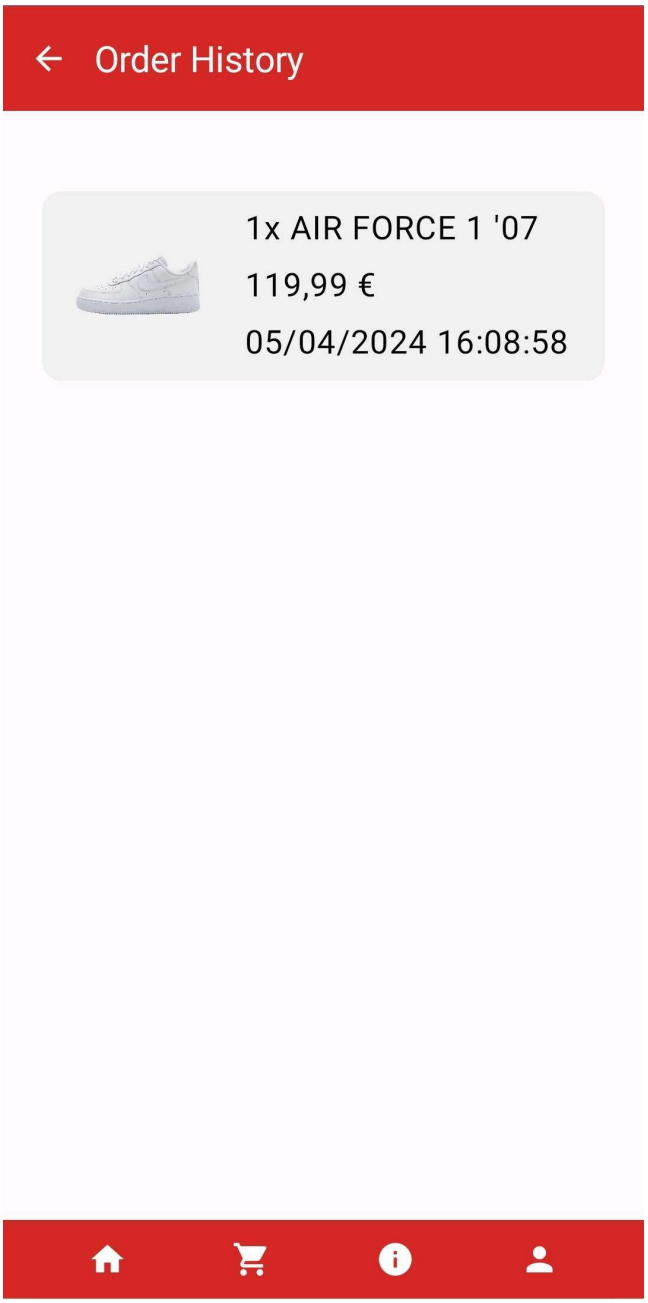
Order history page:

This page is used to view all the user's past purchases.

The App Bar consists of 2 elements: a back arrow to return to the home page, and a text.

In the Body we have cards with the image of the product purchased, how much, its name, price and date of purchase.

The bottom bar is a navBar to direct visitors to different pages

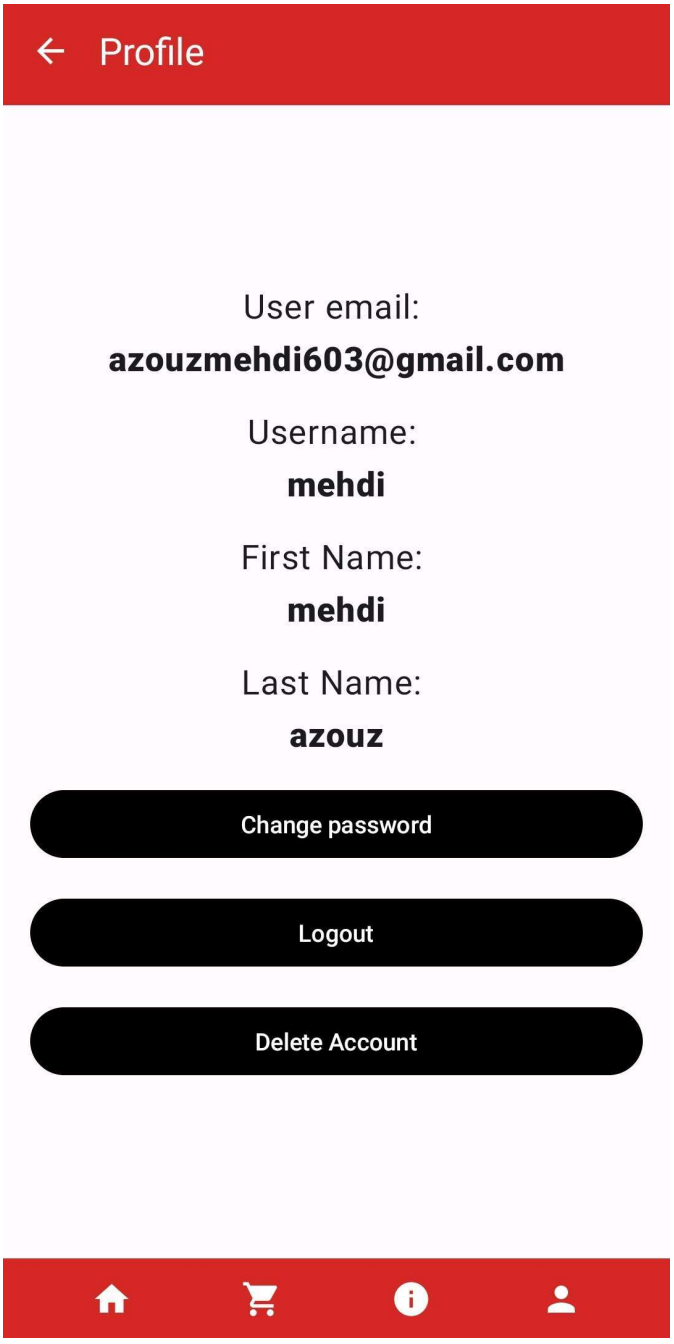


Profile page:

The App Bar consists of 2 elements: a back arrow to return to the home page, and a text.

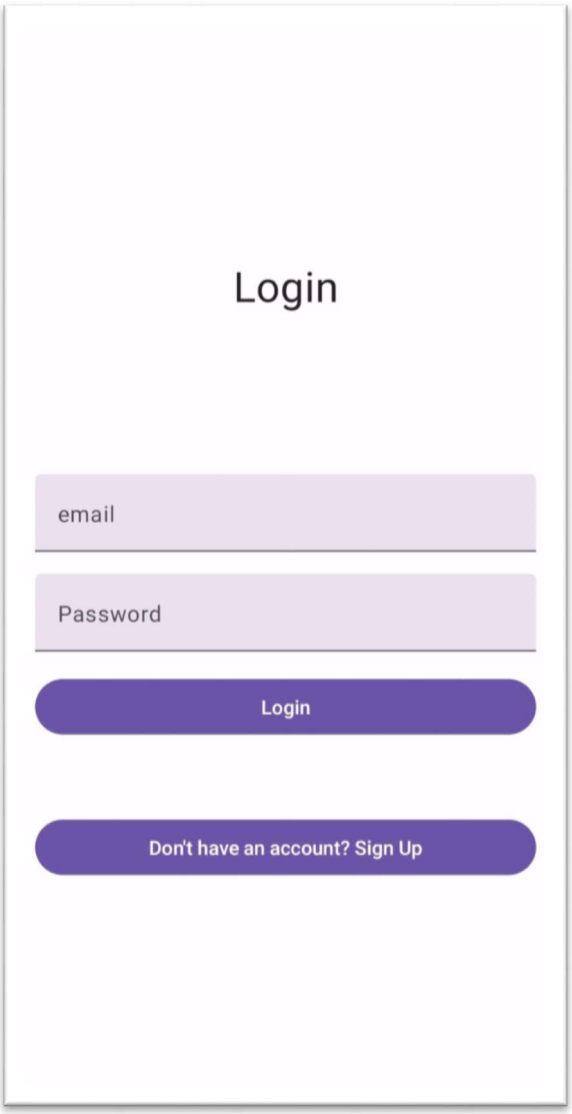
In this page we can see various information about the user, such as email, user name, first name and last name. The user can change his password, log out and delete his account, which will remove all this data from the database.

The bottom bar is a navBar to direct visitors to different pages.



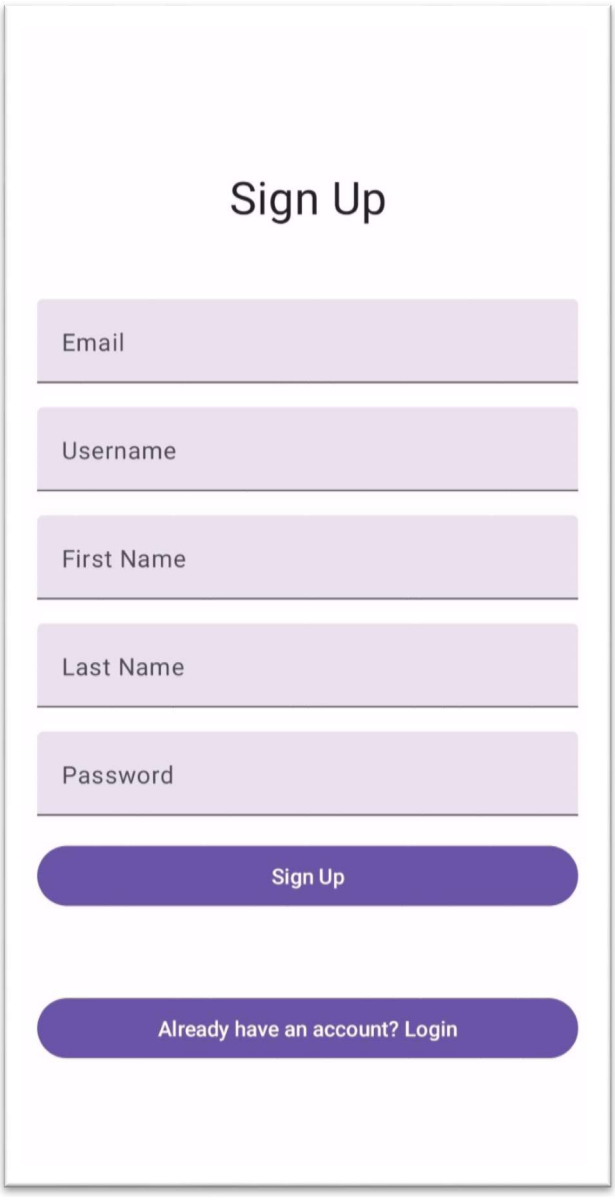
Login page:

In the login page, we can see 2 text fields to put the email and password if it is good the user has access to the other pages showed before. There are also 2 buttons one to try to login and the other to go to the sign up page.



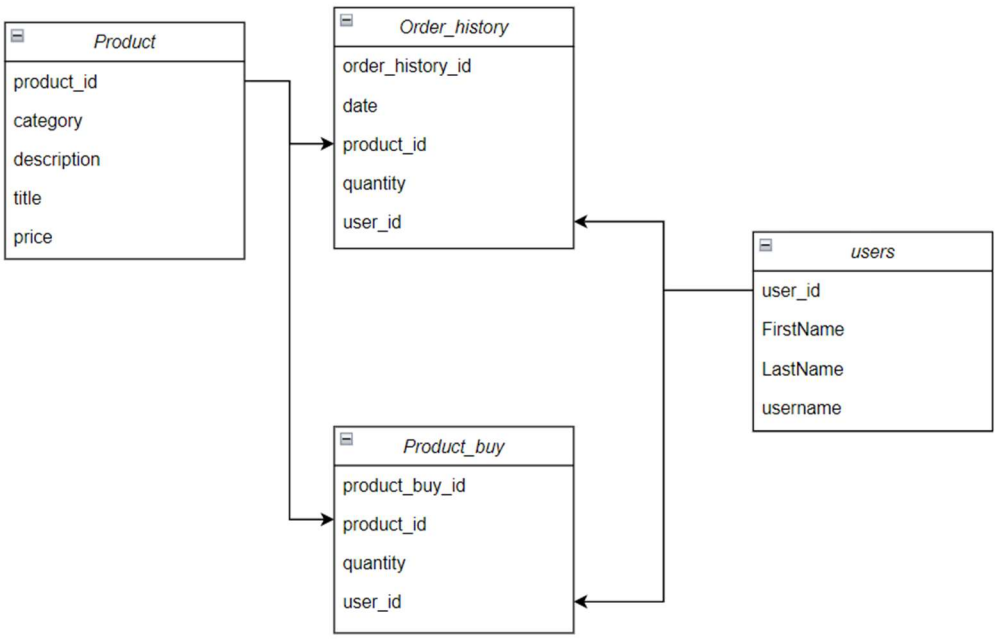
Sign up page:

The Sign-up page is composed of 5 text fields that allow the user to input his personal information to complete the registration for the app. There are also 2 buttons: one to confirm the sign-up and the other to go to the Login page.



Database:

For the database, I decided to use Firebase because it is useful for storing images with Storage and data with Firestore Database. Additionally, it is easy to use this tool. Here is a representation of my database: I created 4 collections, each connected to the others with different IDs:



Problems:

Api problem:

Initially, I intended to utilize the API (https://fakestoreapi.com), but I quickly encountered several issues. Firstly, there were inconsistencies in the images; some were horizontal while others were vertical, and they were not all of the same size. This inconsistency posed a significant challenge in correctly displaying each image. Additionally, the API did not provide functionality for creating user accounts, despite giving the impression that it did. Similarly, while it appeared that purchases could be made through the API, these actions did not affect the API's functionality.

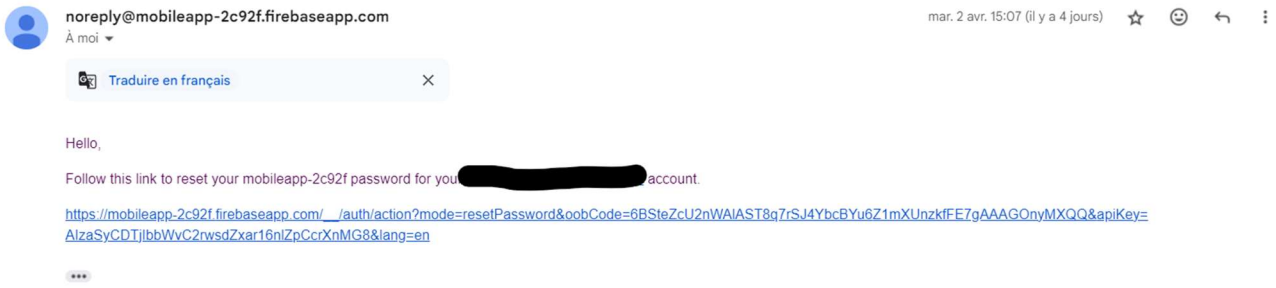
To address these challenges, I opted to use the Firebase database and generate my own articles. This approach provided greater control and flexibility, allowing me to ensure uniformity in image sizes and accurately manage user accounts and transactions.

Card problem:

Initially, I encountered difficulties arranging the cards for the articles properly. I needed to rearrange the images, shorten some of the text due to excessively long names, convert the prices to euros, and adjust the layout of the cards since having two per line didn't fit properly.

Change Password Problem :

I've been facing challenges with Firebase, specifically regarding changing my password due to lacking the necessary authorizations. To resolve this issue, I opted to activate the Firebase function that sends an email to the user for password reset.



Categories Problem:

I got a problem when I wanted to show only products with one particular category the button to show this category didn't work. The problem I didn't create a new composable for the navController which parse the right category and after select only products with the specific category.

Storage Problem:

I got problem to connect my Firebase with my project. I didn't know in Android the app didn't have the default access to the Wi-Fi so I modified the Manifest to allow the app to have access to the Wi-Fi.

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