Project – Mobile App

Lucile Pelou

Student n°74526

Git repository: https://github.com/Clarisse78/project_74526

Video Link:
https://www.youtube.com/watch?v=96zmcuvnyYQ

Table of contents

I. Introduction

The project instructions were to make an online mobile application with the following requirements:

1. Authentication
    a. Allow User to Signup ✓
    b. Log In using username and password ✓
    c. Store userID once logged in to keep the user logged in (even after restarting the app) ✓
2. Product Listing
    a. List Product Categories ✓
    b. On clicking a Category, list Products in that Category ✓
    c. On clicking a Product, show Product description, show buy button and controls to change quantity. ✓
3. Cart
    a. Show cart summary ✓
    b. Show total amount ✓
    c. Purchase button to place an order, show order notification ✓
4. Show order history
    a. List users orders ✓
    b. On clicking an Order, show Order details and Products ordered ✓
    c. On clicking a Product, take them to Product description page created for 2.c ✓
5. Show User details
    a. Use the stored userID to show user details ✓
    b. Show a random circular profile image ✓
    c. Show Logout button, on click take back to Signup / Log In page (Restart should not auto login after logout) ✓
6. UI/Implementational Requirements
    a. Lazy lists to be used for all Lists: Categories, Products, Orders ✓
    b. Add a small "About this app" button in the profile page, that shows a page on click with your copyright details and credits ✓

To make this application and achieve all the requested objectives I decided to use Firebase. It's easy to use and everything is already well set up, such as Firebase authentication which allows the user to sign up, login and logout. I had to simulate the sale of products to make the application, to do this I took existing products that I liked and I asked Chat GPT to generate a description for them or I took the one already existing on Amazon.

To write this report I used Google Translate to translate from French to English as needed.

II. Application Design and Architecture

Before diving into anything in terms of coding or creating a Firebase project, I made a low-quality wireframe and noted down the different collections with the fields I will need to complete the project.

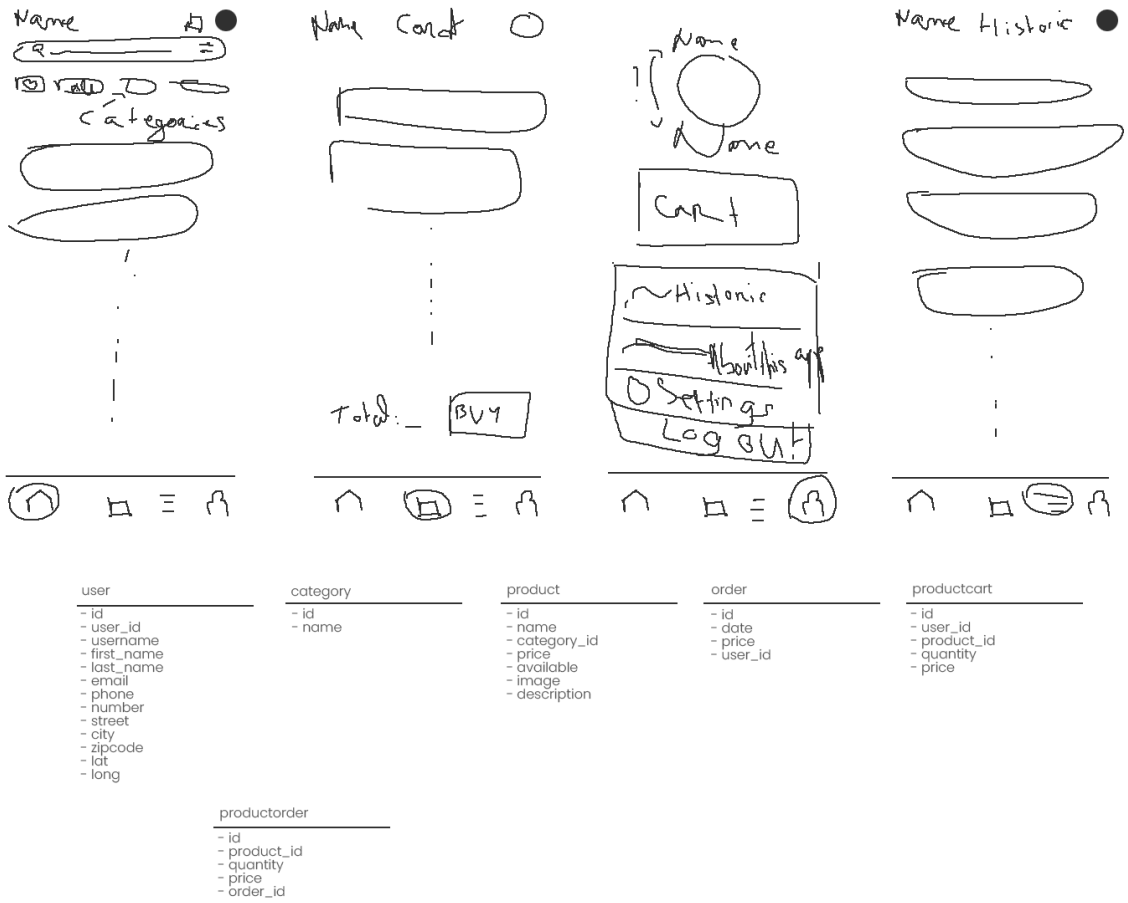

Figure 1: Wireframe & Collections

This was of course only a draft of the final result and things would have to be changed. I wanted to see how to arrange things to be able to meet all the expectations asked for.

In the end here is the UML of my different collections:

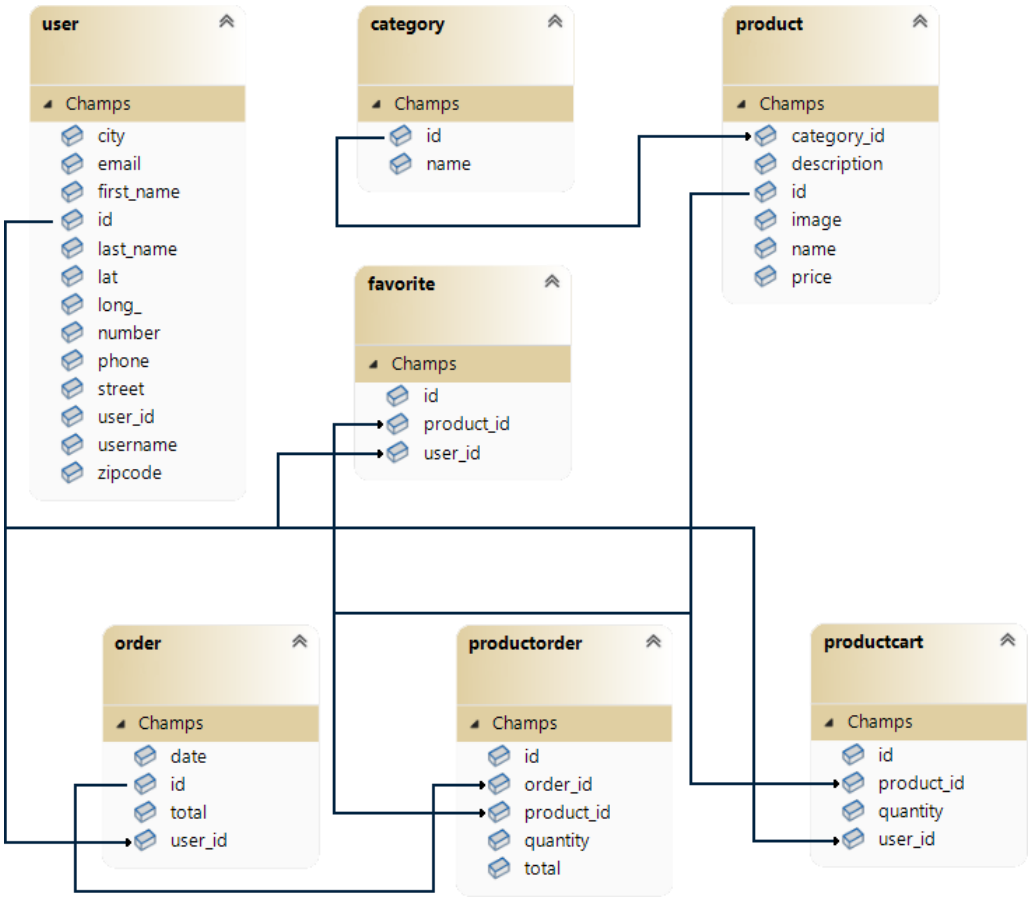

Figure 2: UML of the different collections

I made 7 collections to make this mobile application. The first is users which store all their data. The password is not managed by the user collection but by Firebase authentication, which also stores their email. Then a category collection which will allow you to store the different names of the categories and to be able to link each product to one of them. The product collection has a category_id field to do this. There is also the order collection which has the user_id field because each order is linked to a user. We have the productorder collection which will allow you to retrieve a product with a defined quantity, all attached to an order with the order_id field. There is the productcart collection which has a user_id and product_id field as well as a quantity, this will allow you to store the products which are in the user's basket at the moment. And finally, we have the favorite collection which will allow us to know whether or not a user has a favorite product with the user_id and product_id fields.

With all these collections and fields, the application is now realizable.

III. Implementation of Features

Before starting the project on Android Studio, I created a Firebase project. I then linked it to a project on Android Studio. For this I used the jetpack compose documentation: https://firebase.google.com/docs/android/setup?hl=en.

1. Authentication: Login and Register Pages

I started the application with the login and register pages. For this I took the pages made in assignment 1. I then added the Firebase authentication logic using the jetpack compose documentation: https://firebase.google.com/docs/auth/android/start?hl=en. I also implemented navigation to be able to navigate between the login page and the register page.

Here is a diagram concerning the layout for the login page (it is essentially the same as for the Register page, so I did not find it necessary to make one for this second page):

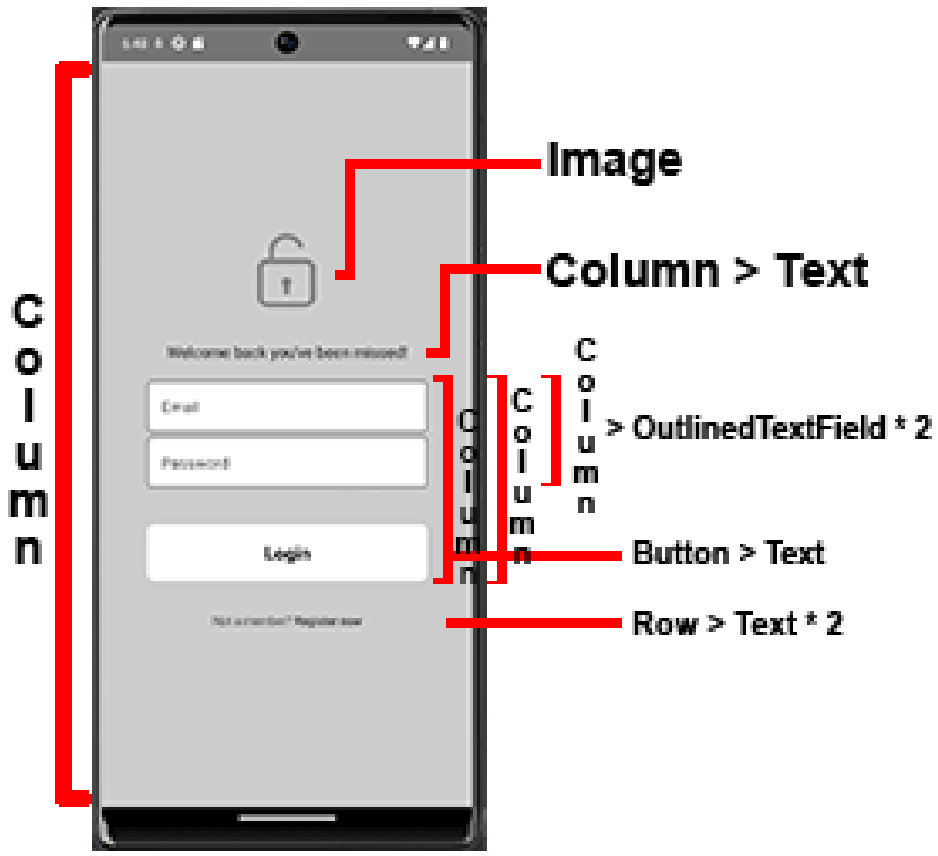

Figure 3: Login page diagram

And here are the final two pages:

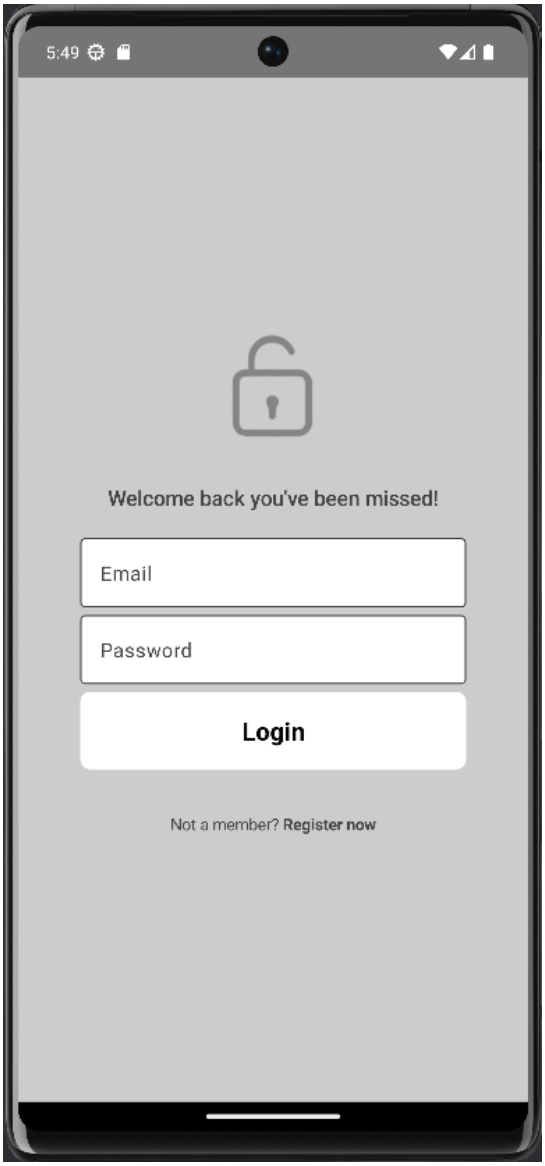

Figure 4: Login Page

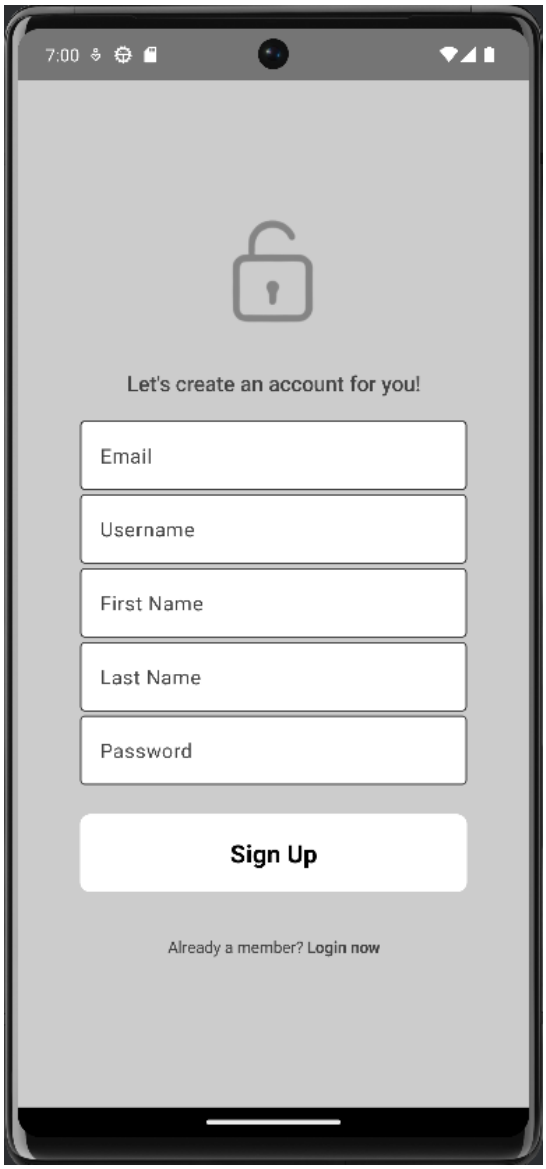

Figure 5: Register Page

I didn't change the design from assignment 1 because I found it fit well with my application.

When registering, users cannot yet enter all their information, they will be able to add it later.

2. List of categories and products (Home Page)

After the Login & Register page, I tackled the home page where the different categories and products are listed. At the very beginning I was thinking of making a search bar but I abandoned this idea because I found that it didn't look very good. To list the different categories, I used a lazy row and I retrieved them with the methods given by Firebase Android Studio. For products, it is done in the same way but a lazy grid is used. I had a lot of trouble getting the data “at the right time”. The major problem I encountered is recomposing the affected components after recovering ALL the data. It sometimes happened that it did not display the different categories or the different products because it did not recompose once the data recovery was completed. I finally succeeded using the LaunchedEffect method, suspend methods to retrieve the data and a mutable boolean that lets me know when all the data has been retrieved.

I used this documentation for data recovery: https://cloud.google.com/firestore/docs/query-data/get-data?hl=en as well as this one for lazy lists: https://developer.android.com/develop/ui/compose/lists.

Here is my result during creation:

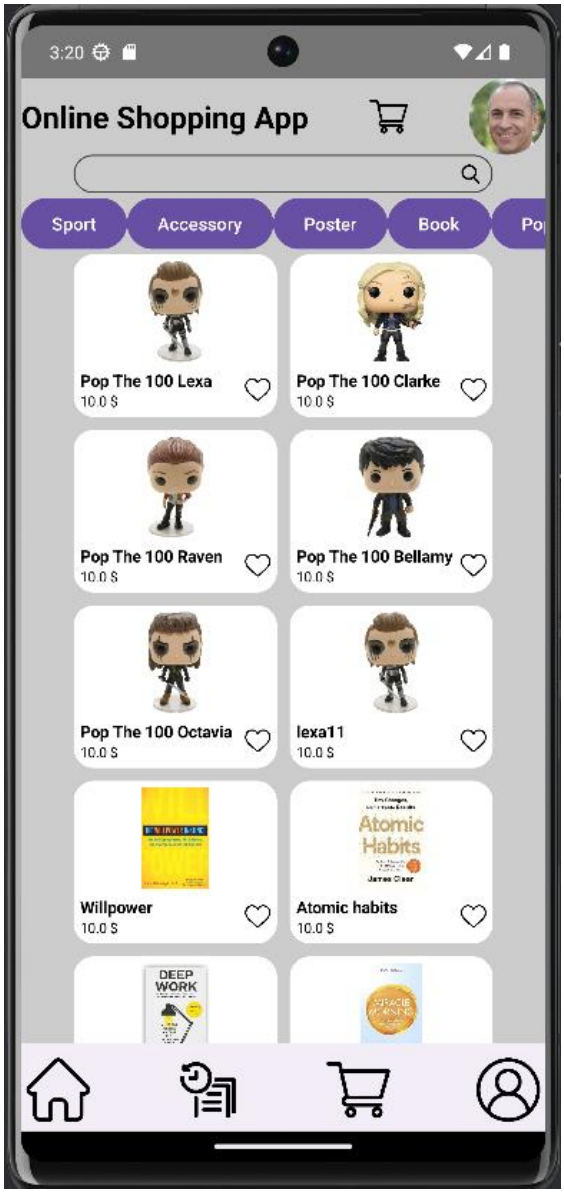

Figure 6: Home Page during creation

I didn't yet know how to arrange the colors, whether at the category or product level. I tried several approaches like the one seen on the previous screen and in the end, I agreed that a white background was the best thing. I just had to sort out the rest to go with this background. Most apps keep a solid dark or light background and I think that's a good thing so I did the same.

Here is the final page:

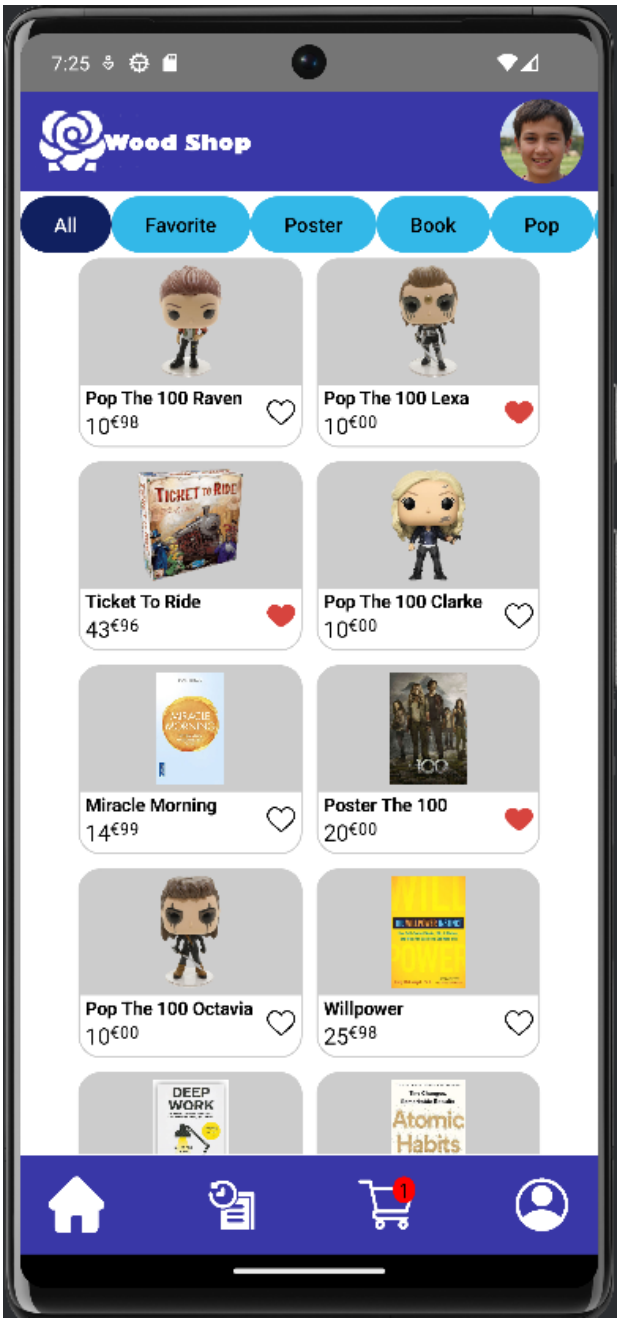

Figure 7: Home Page

Here is a diagram for the layout:

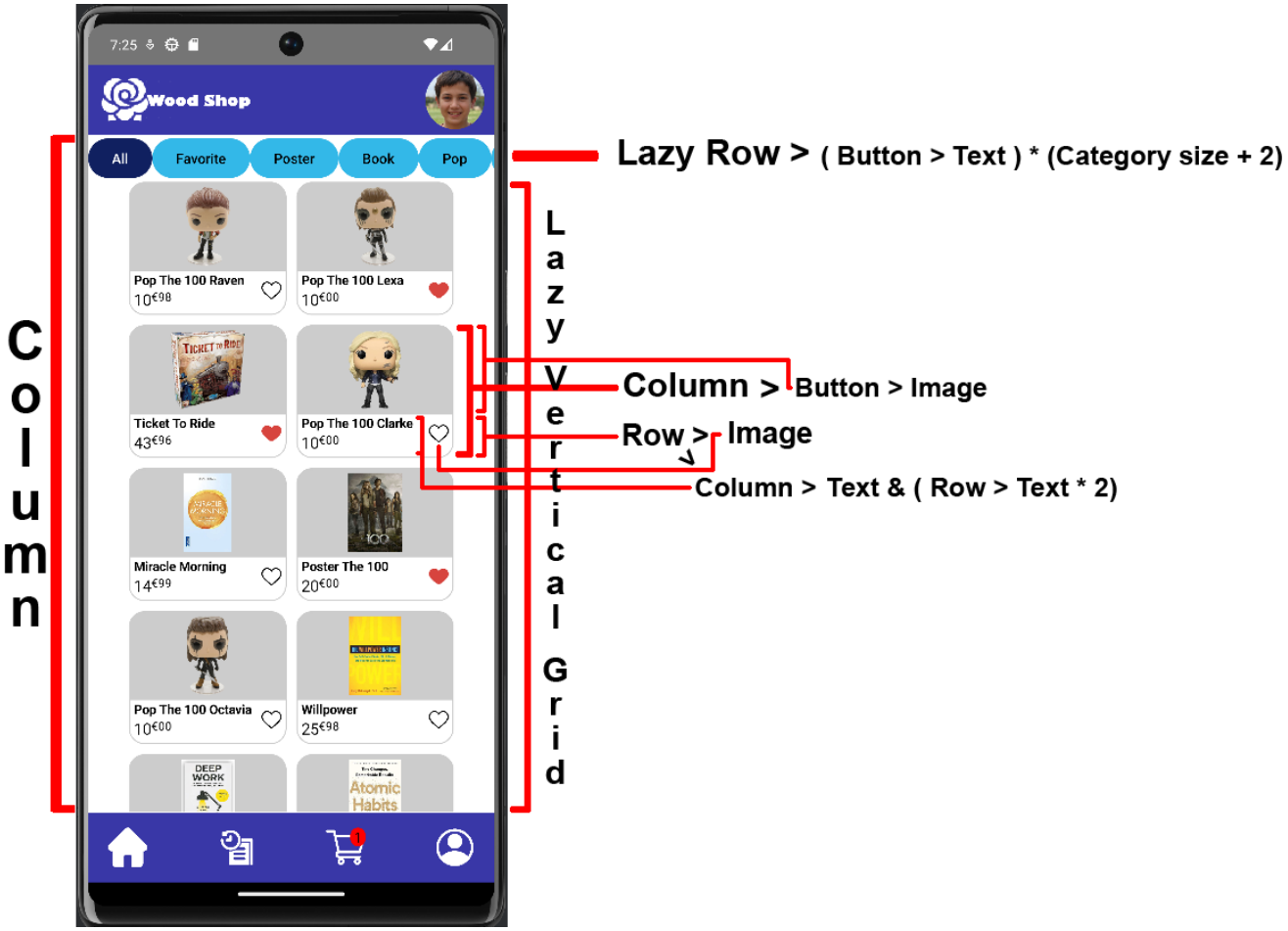

Figure 8: Home Page diagram

The user can click on the different categories to display only the products in that category. Here are two examples by clicking first on the favorites and on the pop figurines.

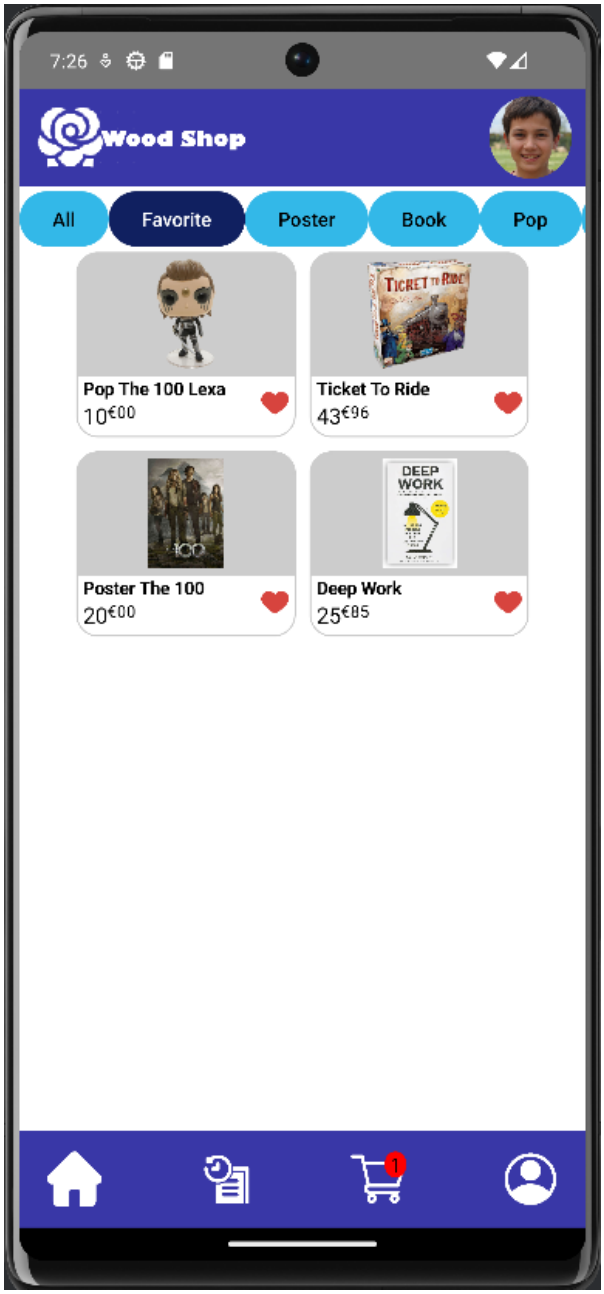

Figure 9: Home Page (Favorite)

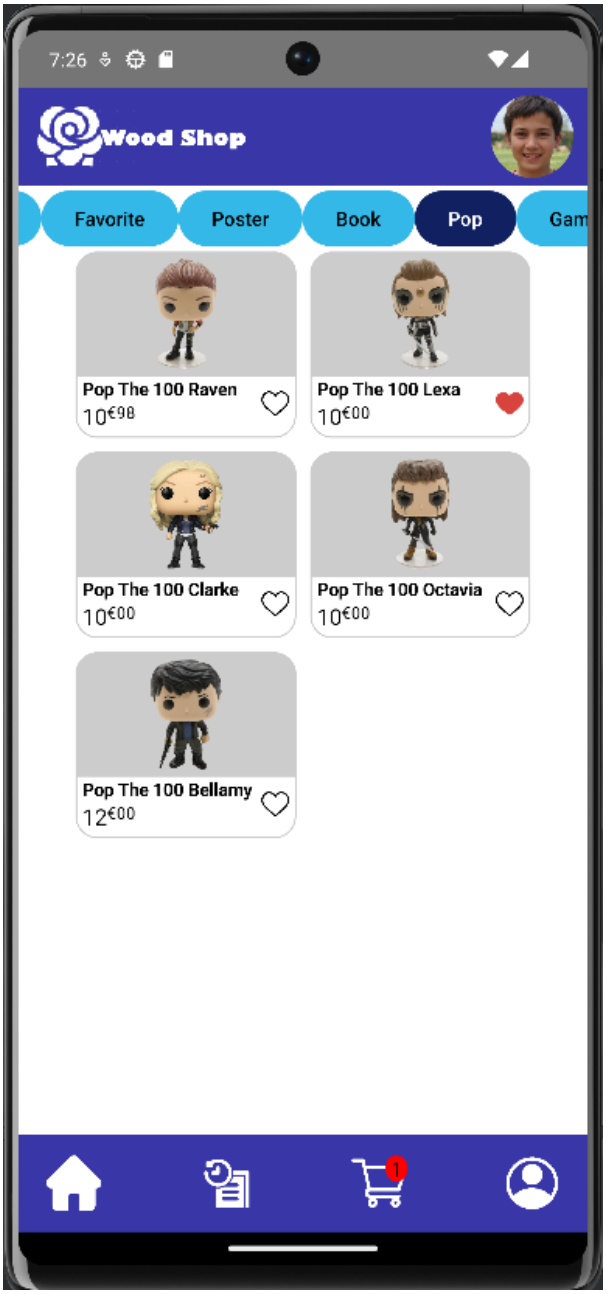

Figure 10: Home Page (Pop Category)

The user can also click on the products to view more information about them or to add them to their cart. By clicking on the description, the user will be able to display the rest of it.

Figure 11: Product Page

Figure 12: Product Page (all description display)

Once the entire description is displayed, you can scroll down to display the rest of the page as in the screen below:

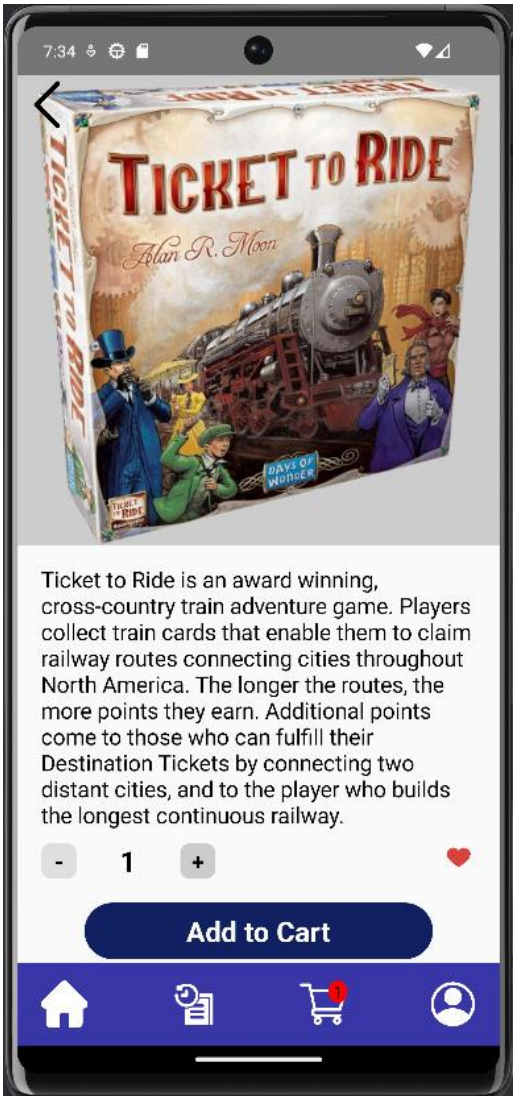

Figure 13: Product Page (description & scroll)

Figure 14: Product Page (Quantity: 4 selected)

The user can also add their favorite products to favorites by clicking on the heart in the product list page or by clicking on the heart in the relevant product page. He can also add or decrease the quantity he wants to add to the card by clicking on the plus and minus buttons. When the quantity is 1, the minus button is deactivated, otherwise it is activated as in the screen above.

3. Cart

The user can therefore add items to his basket, so he must be able to see the contents of his basket. First of all, I added a small red icon to indicate the number of products added in this one. This counts the number of different products and not the number of products with the quantity. If for example I add to the 4 tickets to ride game card:

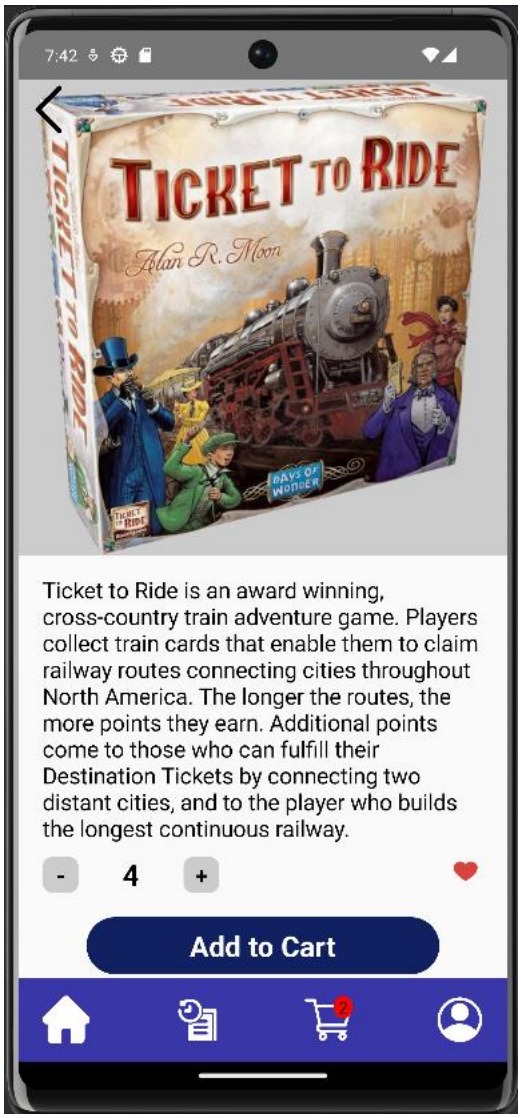

Figure 125: After adding this product to the cart

The number went from 1 to 2.

If we look in the basket, we find our products:

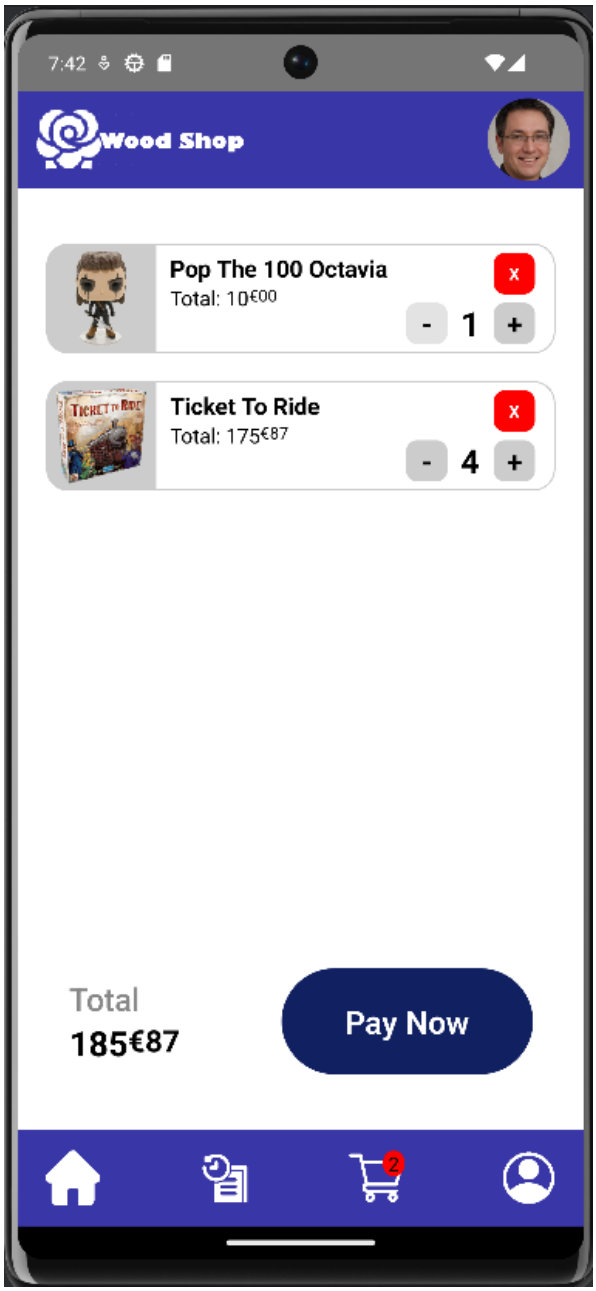

Figure 1613: Cart Page

From this page the user will be able to control their basket. First of all, it can delete and modify the desired quantity of products. For example, if I only want three ticket to ride games because I find it too expensive, I can click on the - button. And if I ultimately no longer want a Pop Octavia figurine, I can delete it by clicking on the red “X” button.

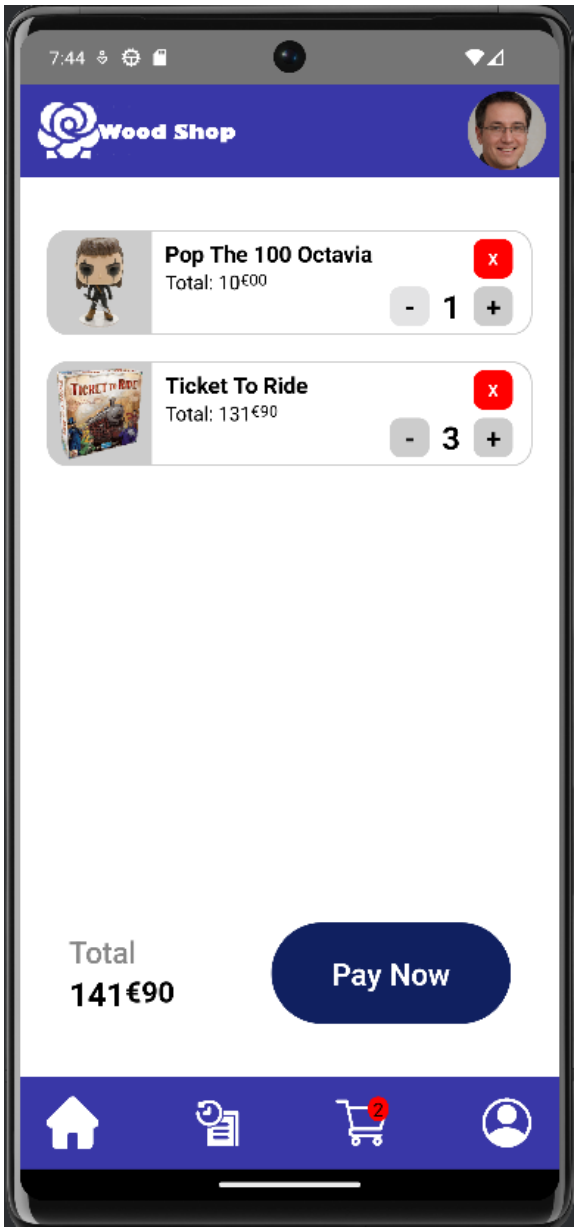

Figure 147: Ater deleting one item

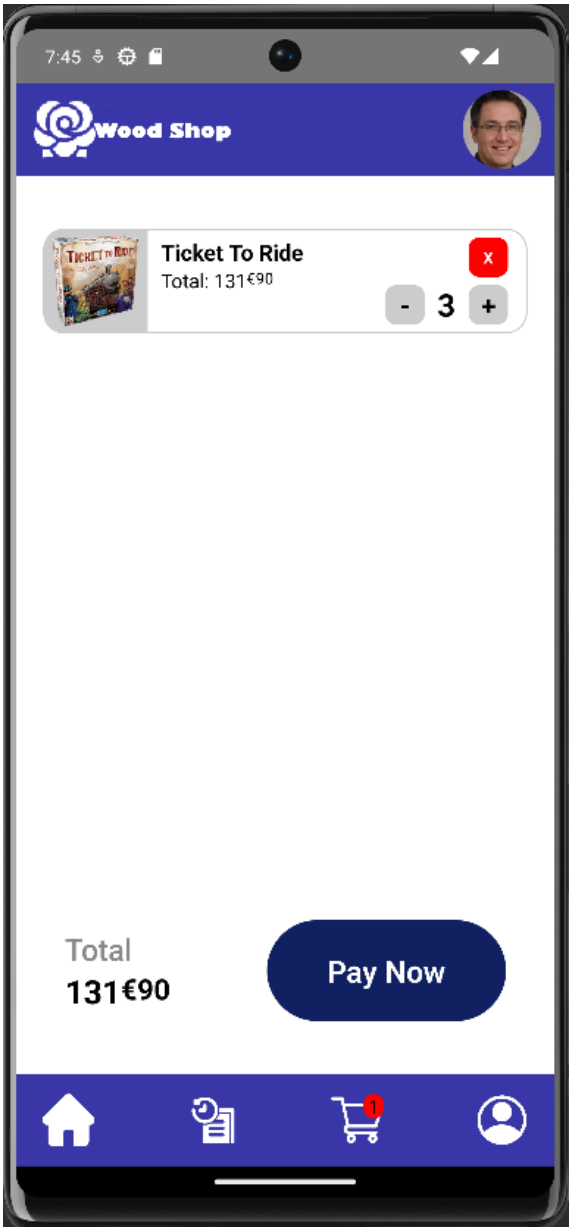

Figure 18: After deleting one product

Now if the order suits me I will be able to order by pressing the "Pay Now" button.

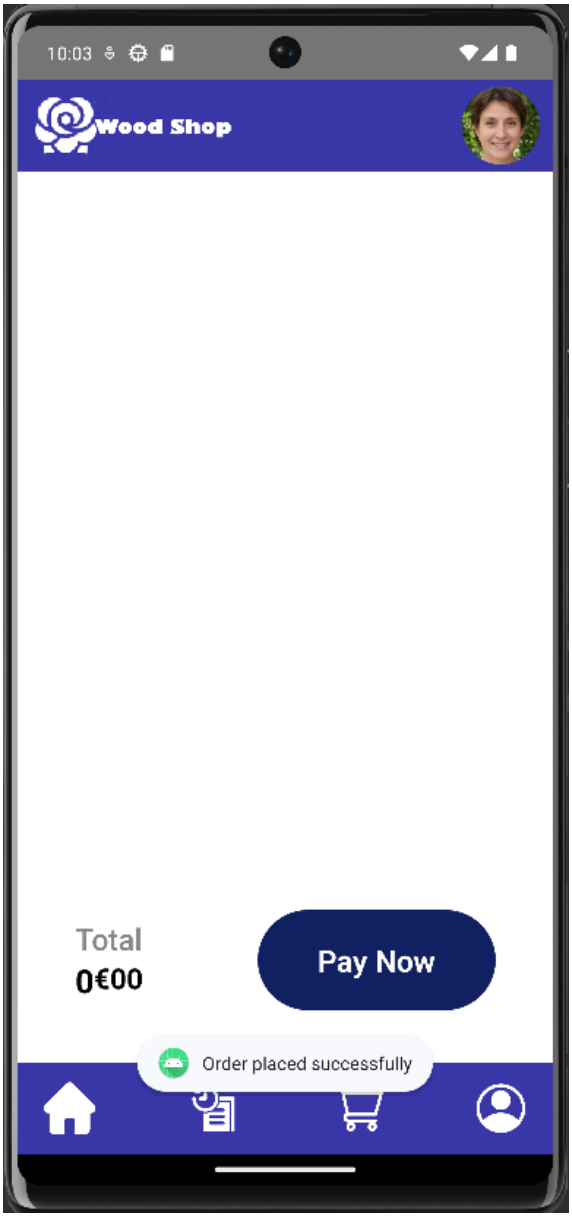

Figure 19: After order passed

The order has been successfully placed and we can see a notification that tells us this.

4. Order History

We can now go to orders to find our past orders.

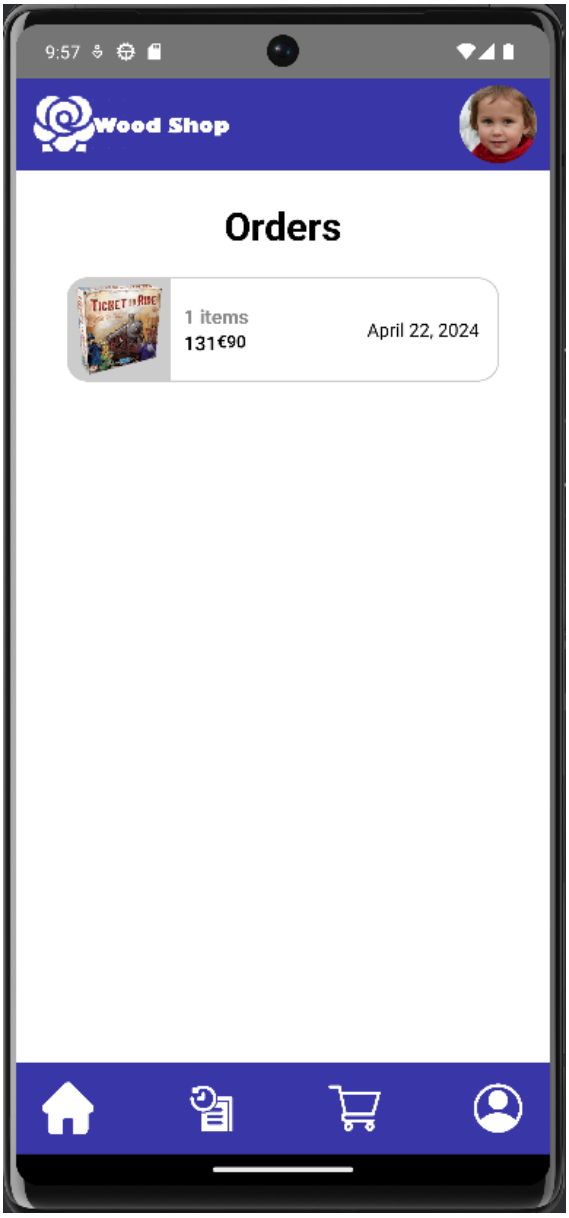

Figure 20: Orders

Here we can see the order that we just placed. The image shown in the preview is that of one of the items in the order. If we click on the order, we access its details. Here is an example with the order placed previously and another with more items.

Figure 21: Order information 1

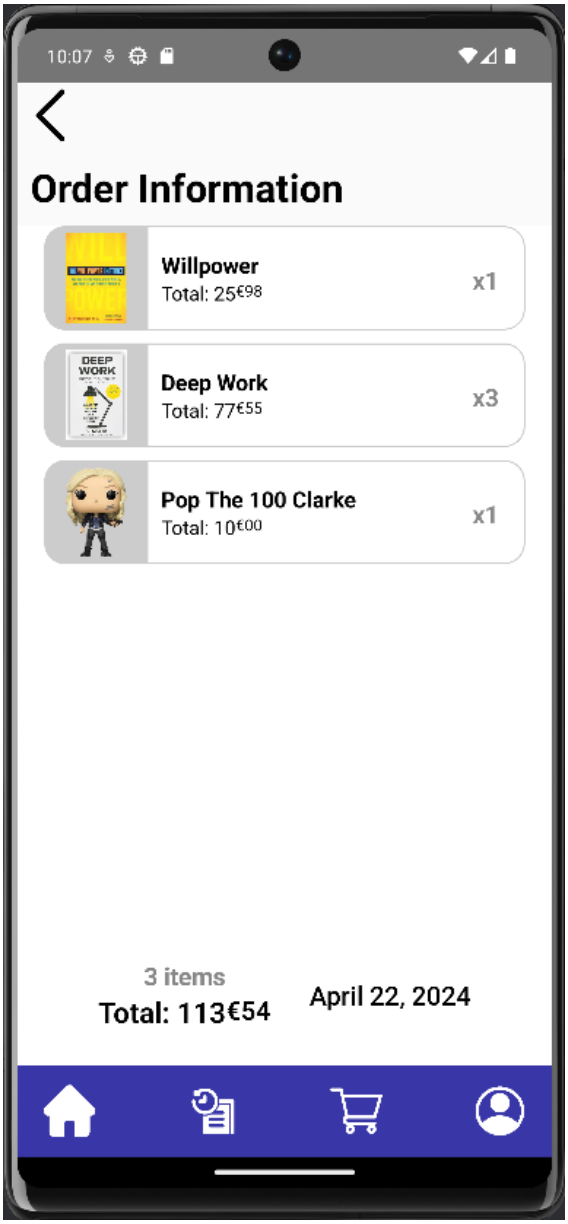

Figure 22: Order information 2

By clicking on the products, we return to their details page already shown previously.

5. User Details

The user can also access their information on their profile page.

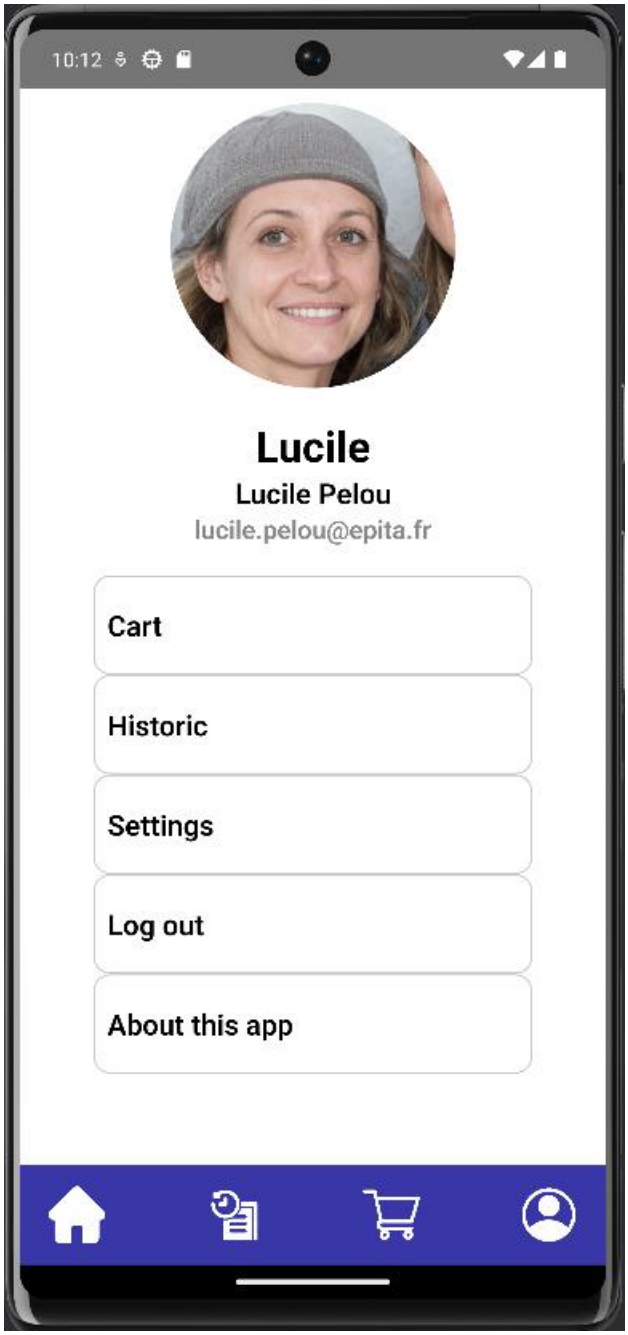

Figure 23: Profile page

By clicking on “Cart” or “Historic” you will return to the pages previously described. By clicking on “Settings” he can update his information.

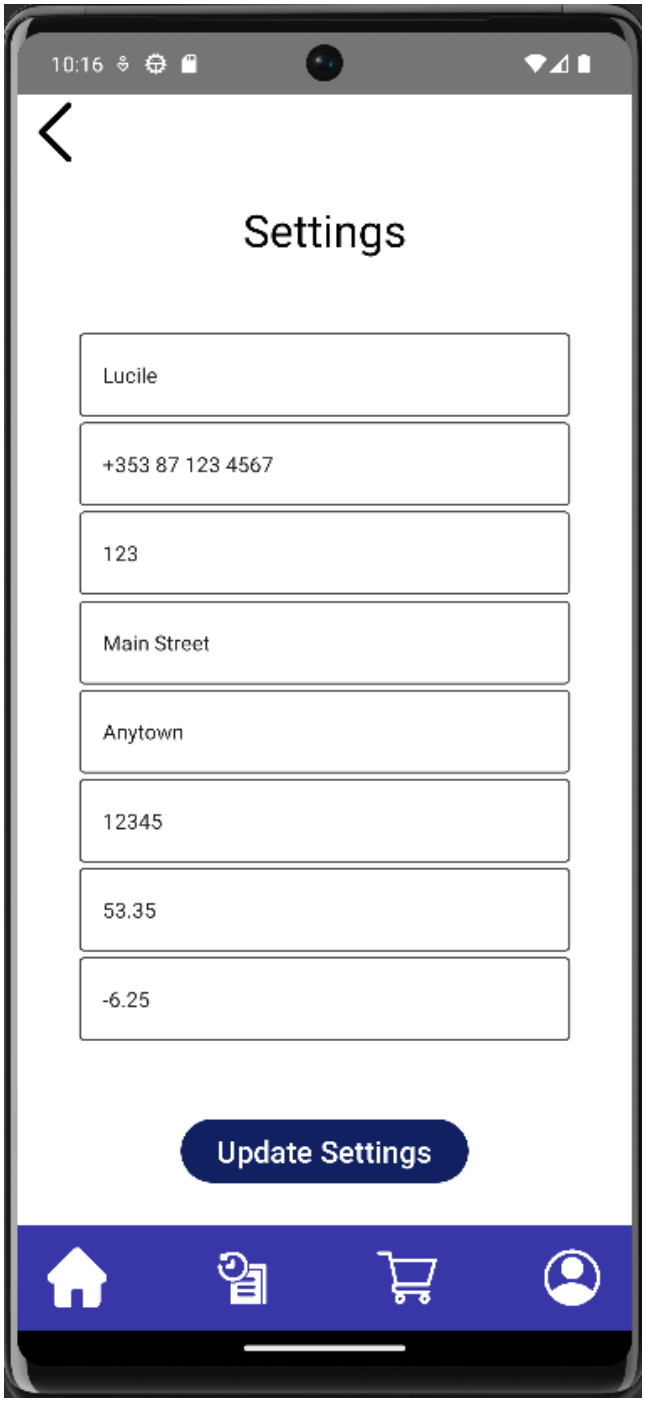

Figure 15: Settings Page

By pressing “Update Settings” this data will be updated. The profile photo is generated by this link: https://thispersondoesnotexist.com/. The user can also log out by tapping “Log out” or see more information about the app by tapping “About this app”.

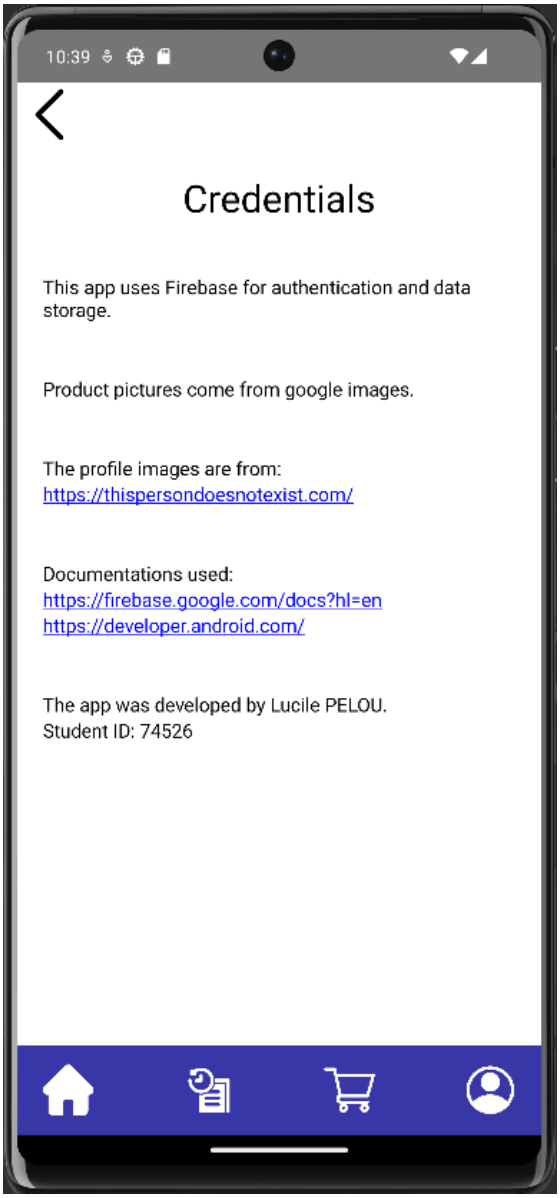

Figure 25: About this App

6. App bar & Bottom navigation bar

The application also has an app bar and a bottom navigation bar as seen in the previous screens. Like many of the other clickable components in my application, each object has a clickable modifier with an action.

IV. Conclusion

Finally, this online shopping application project has been a rewarding experience that allowed me to put into practice my mobile development skills and utilize Firebase effectively. I successfully implemented all the requested features, from authentication to cart management and order history. Throughout the process, I encountered challenges such as handling asynchronous data or update the price, but I was able to overcome them by looking for solutions and persevering

While the application is functional, there are always areas that could be improved with more time, such as adding additional features like a product recommendation system or the search bar that I wanted to do at the start.

Ultimately, this project has allowed me to solidify my skills in mobile app development and has given me valuable experience in creating a comprehensive application from conception to implementation. I am confident that this experience will serve me well in my future !