NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

**Human-Computer Interaction**

**2023/2024**

# SyncShop

## Stage 4: Functional Prototype

**Authors:**

67208, Francisco Barreiras
67209, Tomás Carvalho
67210, Miguel Palma
67211, Alexandre Madeira

**Lab Class Nº** P2

**Group Nº** 14

**Professor:**
Teresa Romão

November, 2023

# Startup Instructions

This project is currently under development and is hosted on the GitHub repository: https://github.com/tomascarvalho7/ipm-group14-2324

For additional details regarding the project's development process, please visit our website: https://tomascarvalho7.github.io/

The URL of the computational prototype is the following:
https://github.com/tomascarvalho7/ipm-group14-2324/releases/tag/Computational_Prototype .
The user must then download the APK corresponding to their device's CPU architecture. Keep in mind the most common architecture in Android devices is arm64.

It is important to note that the application is exclusively compatible with Android devices, whether physical or emulated. To run the application, kindly visit the "Releases" section on the GitHub repository page and download the APK file. Once downloaded, installation can be accomplished by simply executing the APK. Please ensure that a minimum of two users have the app installed to execute the tasks outlined in the scenarios described later.

# Briefing

We aim to develop a mobile application that works as a shared shopping list, in which all members of a household or group can edit the list collaboratively, allowing them to add and remove items.  Additionally, the list can be filtered and organized by categories, making the shopping process more efficient and organized. Our goal is to simplify the synchronization and organization of purchases for households or housemates.

# Scenarios

**Scenario 0:** Setup

a. Start by creating a new account in the application;
b. Create a new shopping list.

**Scenario 1:** Adding a new user to a shopping list

a. Begin by logging into SyncShop, where you already have an account and an existing shopping list;
b. Open the settings of the specific shopping list you want to share;

c. Copy the unique invite code associated with that list;
d. Share the invite code with the new user;
e. The new user, upon opening the app for the first time, proceeds to sign up;
f. After successfully signing up and logging in, the new user taps to join a new shopping list;
g. Enter the invite code received from the already existing user to join the shared shopping list.

**Scenario 2:** Adding items to an existing shopping list

a. Log into SyncShop with your existing account credentials;
b. Open the SyncShop app and select the desired shopping list;
c. Tap to add a product and proceed to create a new item from scratch
d. Input the details for the new item:
    i. Product name: Banana
    ii. Category: Fruit & Vegetables
    iii. Priority level: Highest
e. Finish product creation.

**Scenario 3:** Shopping for items in the list

a. Log into SyncShop using your account credentials;
b. Open the SyncShop app and select the shopping list you plan to shop from;
c. Utilize the filtering option to display only bakery products within the list;
d. Simulate the act of purchasing by marking each bakery product as "bought" after acquiring them during your shopping trip.

# Important Notes

As described in the Briefing section, our mobile application consists of a shared shopping list, in which all members of a household or group can edit the list collaboratively, allowing them to add and remove items.

To develop SyncShop, we utilized Flutter, an open-source framework created by Google, designed for building multi-platform applications. This decision was influenced by our team's prior positive experiences with the tool, recognizing its efficiency in crafting well-designed interfaces within a condensed time frame.

In a strategic choice, we opted not to incorporate a server, given the relatively straightforward nature of the application's business logic. Additionally, considering the curricular unit's primary focus is not on server-side operations, this decision aligns with our streamlined development approach. For the backend, we integrated Supabase, an open-source alternative to Firebase, that utilizes PostgreSQL as its relational database and offers real-time features.

Regarding the prototype's incompleteness, there are some points to consider. We can then separate the missing features as usability and visual features.

The missing usability feature is the logout. Implementation wise it is a pretty straight forward operation, as it’s already implemented in, but since we didn’t feel the need to create a profile page namely because the presentable information about a user is scarce, there was no good place to put this operation in the user interface, so we just decided to not implement it, since it’s not crucial for the application. It’s also worth noting that there may be a small quantity of choosable categories, so the user can feel like there is no fitting category for the inserted item. This was solved by adding a “General Goods” category, to fit all the products which don’t have a more specific category.

The missing visual feature is the insertion and updating of a list’s image. This operation is not crucial usability-wise, it’s just a visual feature to better identify a list, so we opted to not make it functional and just present an icon instead. A good implementation of this feature would require using a BLOB storage service, a new component, which would require attention from the group and since it’s only usable in one operation, we opted out of it for the computational prototype because it’s time consuming and irrelevant for the prototype.