Objective Task 11 – "All_in_One"

The target of this objective task is to develop a Jetpack Compose Android APP **using MVVM architecture, asynchronous navigation, repositories and authentication**. Later, it will be showed to classmates and teachers in a presentation's session.

Architecture and good practices

The project should be done following a **Model-View-ViewModel architecture** where the different files are organized into packages depending on what they implement.

All configuration files should contain only the necessary dependencies and imports and always using components from the Material3 library.

Variables and methods naming must have a meaning. **Bad coding practices will be penalized**.

Proper use of resources is a must. Resources must be stored in their corresponding file/folder.

Theme and Content

The APP to develop will be a **purchasing application**. The theme/content is open: supermarket, clothes, electronics, ticketing …

The used elements, the architecture, the navigation, the layout, the logic ... must be as the ones explained in class. At the slightest hint of **plagiarism** or use of chatGPT type tools the activity **will be graded with a '0'.**

Visual design of the APP must be **attractive** (use of images, icons, borders, colors...) and the layout of the components on the screens **must facilitate its use** (positioning, size, no overlapping ...)

A representative icon of the APP must be showed when it is installed on a device.

The MINIMUM content is as follows:

- Welcome screen
- Login/Register screen
- Shopping cart screen
- Add product screen
- Profile screen
- Exit screen

Layout and components

All screens should contain:

1. A **Top Bar** showing the screen title
2. The Top Bar should also include the name of the registered user together with an icon (except on welcome and login screens)
3. A ***NavigationDrawer*** component on the side with icons and text
4. A **Bottom Bar** with the same icons as the drawer (no text)

Bottom bar and drawer will allow navigation between Login, Shopping cart, Profile and Exit screens as long as the requirements are met, e.g. the user is authenticated. Icons in navigation drawer and bottom bar should be highlighted according to the active screen.

Password components must include the option to display or not display characters.

The **content** displayed on the screens **cannot be lost due to re-compositions or application state changes**.

When using *TextFields* (or similar), the content of *TextFields* must be limited to a single line and must always use placeholders.

Keyboard must be closed when navigating from one screen to another.

Error management and user messages

In case of errors or incorrectness that limit the operability of the APP (e.g. incorrect logging, empty fields …) **an error message should be showed** in the form of any of the explained components in class (*Toast*, *Dialog* or *Snackbar*).

When user successfully completes an action (logging, registering, adding a product to the shopping cart, logs out …), **a message should be also showed**.

When users receive an email from *Firebase* related with authentication, **the emails must include your name as sender and the subject must be of the form**:

"[*user email*], you've to [*action*] your [*password/email*] for the [*APP name*] APP"

Example:

> Sender: "Monica Puig"
> Subject: "pepe@fake_email.com, you've to reset your password for the TeleChurros APP"
> Subject: "pepe@fake_email.com, you've to verify your email for the TeleChurros APP"

Screens details

1- Welcome screen

It will show some **APP characteristics** (image, company info …) **and a loading progress bar**. After some seconds it will jump to the login screen.

2- Login/Register screen

A user login/registration screen to **authenticate the user based on an email/password pair**. It will allow the user to access to the shopping part. Screen specs to meet are:

1. **At least** it must have **3 input fields**: name, email address and password
2. Must include **logging** components, user/email **register** capabilities and their corresponding action components (buttons, clickable elements …)
3. **If user is not properly logged, he/she cannot access to the other screens**, not even using the navigation buttons on the bottom bar/navigation drawer. An error message should be showed

3- Shopping cart screen

Characteristics to accomplish are:

1. Screen must contain the **content of the shopping cart**, a **button to submit the order** and a **FAB to add more products** to the cart
2. The shopping cart must initially be empty
3. Pressing the **FAB will show the add product screen**
4. The **submit button must be disabled when cart is empty** and show the text “Nothing to buy”
5. After adding products, the button text should change accordantly: “Buy 1 product”, “Buy 2 products”, “Buy 3 products” …
6. Each time the shopping cart screen is displayed, the cart content and the button text must be updated
7. **When pressing the submit button, a purchase confirmation Dialog must be showed**. Depending on the user action on the Dialog, an information message should be showed

4- Add product screen

To enter a product, you must implement at least:

1. A **category selector with a minimum of 4 categories** (e.g. Computers, TV, mobile phones and household appliances) by means of a selector component such as a drop-down list, radio buttons ...
2. A field to enter the **product name**
3. A field to enter the **quantity** to buy. This field should **only allow numbers** as input

Once the fields are filled in, the user will have a **button to upload the product** to the shopping cart and then close the screen.

The units of each product must be stored in the DB as a number, not as a string.

5- Profile screen

This screen **will show the user data** (at least name, email and password) **and a reset password utility**. Password will be hidden by default and it should be showed by clicking on an icon.

6- Exit screen

The characteristics of the last screen should be:

- It will show **an exit message and some decorations**
- It must also include the **APP credits** (IES, module, author, date …)
- When moving to Exit screen, user will be logged out from *Firebase*
- Once in the Exit screen, **it is allowed to navigate back to other screens**, but only to those that do not require to be logged.

Project presentation

You've to defend your project in a presentations session, including the following:

- Overall project introduction
- Screens and navigation explanation
- Authentication and DB access
- Demonstration (live or using embedded videos/gif)
- Summary and improvements proposal

Bonus

Extra points in the 2nd term evaluation will be given if:

1. Implementation of user Google authentication instead of email/password
2. Creation of a Signed APK of your APP