**POWERPOINT**

**SLIDE 1: Title**

**SLIDE 2: Team**

**SLIDE 3: Introduction**

The original idea of myAPTracker project has been given by BendingSpoons. The project consists in an application that let users to start tracking an Amazon product and receiving updates on price changes.

Our idea was to create an application in a social network style where all the users can access products added by other users, see all the price history and place comments on products they follow or not.

For this purpose, we also create an ad-hoc backend to keep application data and user accounts. Even if the application is accessible without an account, to track a product and receive notification on price changes, an account is required.

Thanks to the backend, data are synchronized in all the devices accessed with the same account.

**SLIDE 4: Technologies**

In this slide all the technologies used in the project from the application to the backend are reported. Now we just focus on SwiftUI framework used to create the iOS application and Firebase (FCM) to manage per-user push notification.

We also provide three social authentication, Google, Facebook and Apple in addition to the classic username and password.

Just two words on the backend, API and business logic part is written in PHP while the Amazon scraper is written in Swift and called by shell execution from the PHP script.

**SLIDE 5: Device & Capabilities**

myAPTracker supports iOS/iPadOS 14, watchOS 7 and later versions, the interfaces support iPhone in portrait mode and iPad in portrait, landscape and all the size combination of the split view.

The localization has been activated in the application and now it is translated in English and Italian, other languages can be easily added.

Finally, we also created a bunch of widgets for iPhone and iPad to have a quick access from the home screen to the latest discount.

**SLIDE 6: Functionalities**

**In this slide we can see the main functionalities supported by myAPTracker.**

**The application will keep track of the prices of all the products the user is interested in. Then, we will have a proper section in the application (the Explore one) where the user can explore some relevant products between all the products tracked by the myAPTracker community.**

**Of crucial importance is the Notification functionality that enable the application to notify the user whenever one of his products is in sale.**

**Lastly, we have the synchronization functionality, that thanks to the different services of authentication implemented, let the user access to the same tracked products between multiple devices.**

**SLIDE 7: Testing**

**In this slide we can see the testing campaign that we have done. In the unit tests we have tested the initialization of some viewModel and some async calls to the API, all the remaining logic was in the backend and for that reason has not been tested.**

**Since different UI have been created for different devices, we have been done multiple tests for the supported devices and their correspondent supported orientation. In general, the UI tests are regarding the presence of a specific view, the flows of our application, the logic to enable different buttons and so on so fore….**

**Finally, we have also created a test campaign through Apple TestFlight platform to have some people testing the application in a real scenario.**

**SLIDE 8: Further implementations**

**We thought about different possible further implementations, like**

* **The internationalization of Amazon website: let the user access the Amazon website, corresponding to the user language.**
* **The currencies internationalization backend side: for the conversion of the currencies based on the location of the user.**
* **The improvement of user suggestion: adding a ML back-end engine to suggest the user products in which they could be interested in.**

**DEMO**