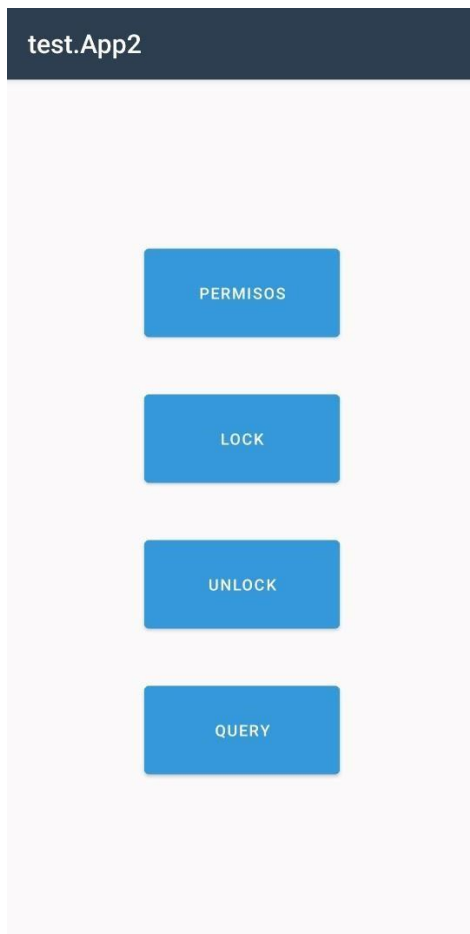


Objective:

The objective of the test is to demonstrate the ability to carry out a small project in Xamarin, which is cross-platform and integrates native libraries.

Xamarin.Forms project will be created for iOS and Android, implementing native Android and iOS libraries. That works on Android 12 (Minimum Android version 7.1, Target Android version 12) and iOS 15.5.

The interface should be similar to this:



You can take 2 paths. Directly implement the provided SDKs (with their corresponding sample native application) anticipating that it will be necessary to create intermediate libraries since the SDK cannot be implemented directly. Or implement the native libraries already created.

The behavior of the application will be as follows:

At startup, all buttons except the permission button must be disabled.

When pressed, you must request permissions to access the state of the network, geolocation (all) and bluetooth (all). And once granted, request the Api Key and the Secret Key, call an initialization method to pass that data, request a token (FetchAccessToken) and enable the buttons if the call has been successful.

The Buttons must call the Lock, UnLock and Query functions, which will request when pressed a Mac that the user must enter.

It must show messages corresponding to the different Callbacks that are generated.

Any added aesthetic and functional improvements will be valued.

Ways to Implement It

If you choose to deploy the SDK:

FetchAccessKey, Query, Lock, and UnLock functions must be implemented (as in the provided proxy libraries) it is not necessary to implement the entire SDK

These are located in the Original SDK folder.

If you choose to deploy the proxies already created:

These are located in the Proxies folder. For Android, an explanatory file of the necessary references is attached.

Both the time in which it is carried out and the option chosen will be valued, valuing more the implementation of the SDK.

Attached as a help and/or reference are the ILinka.cs and ILinkaCallbacks interfaces.cs and the implementations Linka_Android.cs on Android and Linka_iOS.cs on iOS.

Bindings have to be made in separate projects, the minimum project structure should be:

Test.APP

Test.APP.iOS

Test.APP.Android

BindingLibrary.Android

BindingLibrary.iOS

Note, if the provided proxy library is used for Android, in theory it is complicated to have two .aar in the same bindinglibrary, so it would be accepted to create a BindingLibrary.Android.EasyPrefs. If it is necessary to create more, there would be no problem, but this structure must be respected at all.