Code snippets description

I included some of the files necessary to empower one screen in the app to give an idea of how I write and architecture my code. Not all files to empower the screen are included for non-disclosure reasons.

In blue are the included files in the portfolio.

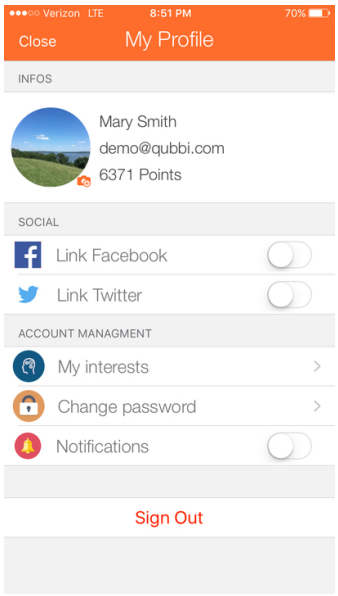
MyProfileTableViewController is a controller for a static UITableView, it's associated with the screen below in the app.
This controller has many responsibilities including:

1) Be aware of the state of iOS remote notifications authorization status, and update the Notifications switch in concordance as well as register/unregister for remote notifications and updating the User singleton class in consequence. This will trigger a unique device token being generated and retrieved via the appDelegate, this token is securely send to our backend. Firebase is used to empower the notifications.

2) Allow the user to change his profile picture via a created reusable Controller called UserImagePickerController (included in the portfolio) . The new picture is compressed, saved in Ream (internal database) and uploaded to our backend.

3) It also allows the user to login his Facebook or Twitter account, check if it's already linked

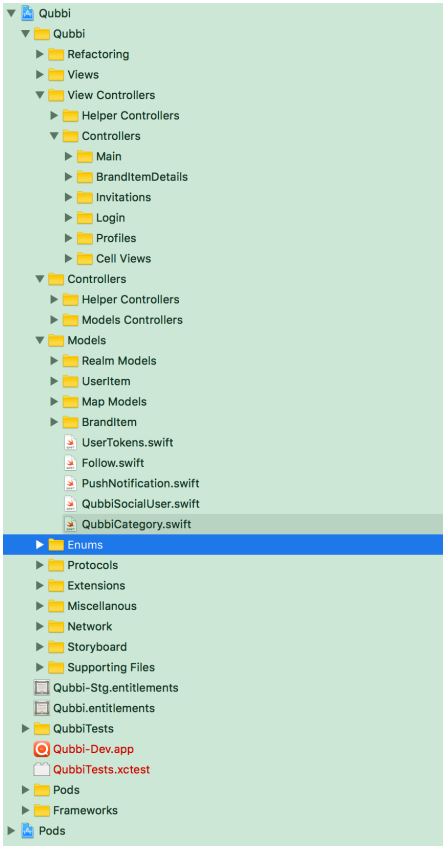
4) The sign out method will confirm with the user and then delete all his infos in the app using QubbiLoginController (included in the portfolio) (clean all the DB, singleton, unregister for notifs, logout FB/Twitter, delete API user token using UserTokensController (included in the portfolio) and go back smoothly to the login screen allowing for a completely different user to login without issues.



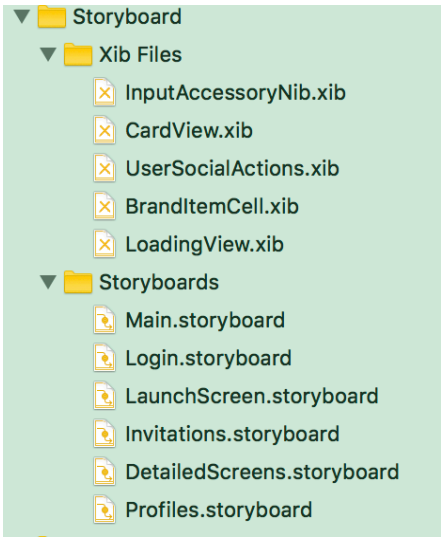
5) This screen can take you to 2 other screens that are Interest screen and change password screen

The most challenging part for this controller was related to the push notifications using Firebase, it was challenging to make everything works even when the user closes the app and changes his iOS settings (register/unregister notifs). It was harder because we use 2 environment (dev and prod), but at this day everything is working properly.
To display the user infos, I created a Realm model of the user that is accessible via a singleton class. It was a better choice than using NSUserdefaults because the user model is complex and needs to be accessed a lot in the app. See QubbiUserRealm (included in the portfolio).

Screenshots



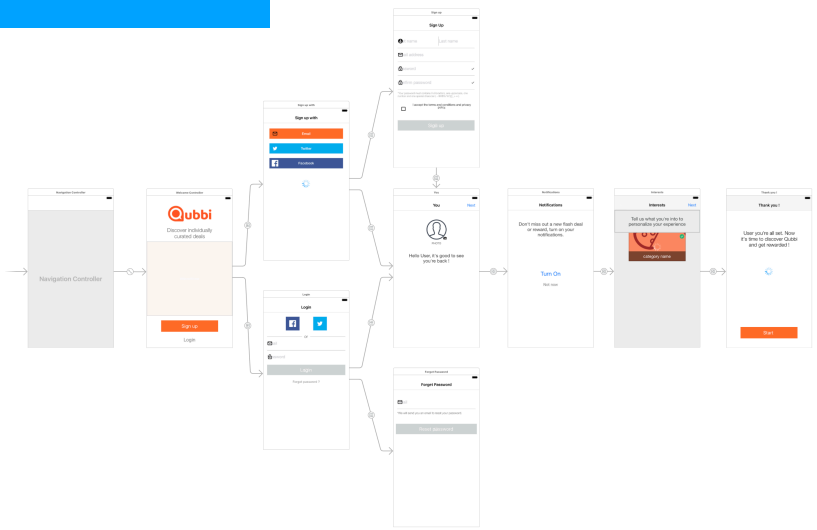
App Folders Architecture



Storyboards and XIB files

- Forget Password Scene
- Welcome Controller Scene
- Sign up with Scene
- Login Scene
- You Scene
- Navigation Controller Scene
- Sign up Scene
- Notifications Scene
- Interests Scene
- Categories Scene
- Thank you ! Scene

Login Storyboard



- ItemDetailsContainer Scene
- BrandProfileViewController Scene
- Live Feed Scene
- Map View Controller Scene
- Flash Deals Scene
- Reveal View Controller Scene
- Slide menu Scene
- My Profile Scene
- Payment Details Controller Scene
- Navigation Controller Scene
- My Wallet Scene
- User Wallet Scene
- Redeem Item View Controller Scene
- ItemDetailsContainer Scene
- Feeds Screen Scene
- Navigation Controller Scene
- Navigation Controller Scene
- About Scene
- Web Scene
- Change Password Scene
- CategoriesCollection Scene
- Navigation Controller Scene
- Navigation Controller Scene

Main Storyboard