

Project1_Milestone2 Partially Functional Prototype:

Check Project1_1.0

Journal documentation & Roadblocks:

I chose to build the app based on my lab2, since it already gave the basic structure of the whole application. Right now, it has the basic structure of the whole app and just need to apply the encrypt and decrypt method successfully(which is the main point). Also, still need to add more flow and change the details of whole app.

Flow and Interactivity:

It will have the table view which lists the types of what password used for, such as work, school, life and such on. The user should be able to add their own types to classified their passwords. If the user taps the 'work,' user should able to see all the password info(has been encrypted) in 'work' type. The title will be the place the password used for and the subtitle will show the hint of the password(user decide). If user hit the '+' in the page of list of encrypted password, user will be direct to the add new password info page, which ask user to enter the basic information of new password (include what type is the password, password, where the password used for, hint for the password, and key value for the password.) Hit button Encrypt and then hit save. The new password info will show on the previous page. If the user clicks on one password info, it will direct the user to the details of password info page, which shows the encrypted password and where the password used for. The user could click the Decrypt button to go to the Decrypt Password page. It will automatically show the encrypted password and hint for the password. The user only need to file out the 'Key' of the password (the one they enter when they add the password) and hit the Decrypt button. The password will show below.

Logic and Algorithm:

https://developer.apple.com/documentation/security/keychain_services

To set a custom service name identifier or access group, you may now create your own instance of the keychain wrapper as follows:

```
let uniqueServiceName = "customServiceName"
let uniqueAccessGroup = "sharedAccessGroupName"
let customKeychainWrapperInstance = KeychainWrapper(serviceName: uniqueServiceName, accessGroup: uniqueAc
```

The custom instance can then be used in place of the shared instance or static accessors:

```
let saveSuccessful: Bool = customKeychainWrapperInstance.set("Some String", forKey: "myKey")

let retrievedString: String? = customKeychainWrapperInstance.string(forKey: "myKey")

let removeSuccessful: Bool = customKeychainWrapperInstance.removeObject(forKey: "myKey")
```