CS4298 IOS Application Development

Individual Project

Self-Reflection

## APP-Name: My-cardholder

Project Intro: An offline card information organizing application.

During this Project, I gained much experience and knowledge compared to the previous 2 projects.

1. Frontend. This time I tried integrating UIKit with SwiftUI together. For instance, the credit card scanning view and the Image Picker View are integrated using UIViewControllerRepresentable protocol. From this process, I felt that Apple did a decent job in gradually migrating UIKit with the new framework. I tried both way at the beginning: first way is to using storyboard as the project foundation and then import a few SwiftUI Views to simulate the scenario that if the whole project is written in old framework. Secondly, I also tried using SwiftUI as the primary framework and import some viewcontrollers with specific functions such as enabling albums and cameras for importing images. Basically, by applying the UIViewControllerRepresentable protocol, swift translate the UIViewController to a SwiftView for displaying. Thus need to implement an intermediate view, which realizes makeUIViewController and updateViewController two functions. As for UIKit, the experience was not as bad as I thought in the previous project. Probably learning a new framework also improves my understanding of each life cycle during the layout rendering, my programming time in frontend takes a lot shorter time compared to previous.
2. Database. In this project, one of the biggest challenges is to implement a complete database with workable schema and transaction codes. I used Realm swift database inside of the project. One thing I like Realm is about the visibility. Building a schema from scratch is painful. It requires a lot of modification on both code level and structural level. I create classes and test relations on playground first. After testing the logic and some basic functions such as creation, deletion, updating. I move the code into the project. As for the visibility, I can open the database of the simulator and check it in real time. Sometimes I delete the relations between two models only but forgot to remove the unused models. After checking the db, I saw redundant results which reminds me to update the functions. The query of objects is also relatively simple compared to the SQL like db I used before such as postgreql. Initially, I was suggested to use Realm by a very senior programmer. But during that time, I was not familiar with swift programming at all. This time after practicing database on operational level, I feel like my database knowledge improves as well.
3. Sensors. The nature of the app is an electronic wallet; therefore, security level is crucial. I decided to use the native security methods – FaceID, which is the same procedure in unlocking screen. Beside FaceID, I also used other API for scanning credit card for faster inputting information. I compared around 5-6 API this one works relatively stable and easy to implement. I also tried to implement other API for multiple variation of cards such as business cards, debit cards, ID cards… some API like face++ provides online text recognition, which is not very suitable for a simple offline app. It requires the app to send the picture to server and receive answers from server as well. The advantage is that the module is customizable so theoretically I can customize any kinds of cards for recognition. The drawback is that I need to apply for API key and the configuration is rather complicated.
4. Bugs. The developing procedure is not very smooth. Especially for me since I don’t have an iPhone for testing all the sensors and UI. Finally, I used my iPad instead. Some layout issue seems annoying to me. (1) The NavigationView for iPad is buggy, It’s either split or not showing in portrait orientation. (2) some unexpected issue happened when I try to detect user put the app into background (pressed home button etc.). I did a lot of research on these issues and most forum said that it was the OS issues, after Apple split its IOS into IOS13 and IPadOS13, some bugs remain to be solved which means these bugs wouldn’t appear on IOS12 or on an iPhone.
5. Things to Improve. (1) productivity, when doing a project alone, time management is crucial. (2) The code is not currently realizing all function that I scoped at the beginning. Will refine it on Github later.

Project Repo: <https://github.com/blueice930/IOS-my-card-holder>