Group3 Project 1

# Introduction

1. Aims of the project.

Many people may have not even realized that the current calendar system (Gregorian) is unnecessarily complicated and cannot match with weeks very well, since week is the most-used time unit measuring productivities in most industries. In this case, credit to Dr. Donny, we have a convenient and memory-friendly version calendar system – the NexCalendar.

In this project, we aim on soothing the process for a normal person to learn the new calendar system and have fun using it. Reading materials and timely tests are important for memorizing a new thing. Using our application, we made the learning and testing more interesting and customized. Besides, we also have the display for both calendar system for comparison. It helps people to see the difference and the advantages of the NexCalendar system.

The tutorial system has two modes of displaying, list view and the card view. By scrolling and previewing the tutorial cards, the user can directly enter the content. By clicking the explore button, user can view the list of tutorials. For each tutorial we also provide the star/unstar feature for user to conditionally select those tutorials that they are interested in.

The quiz system has 25 questions in total. Each time user will get 5 random questions and test their skills. The instant scoring system will tell user the number of correct answers. After that user may choose to see the answers.

The whole project was designed and developed in the new swiftUI framework which is an experiment for developer, since it’s new and lack enough documentation, and many resources are already outdated. By the end of the project I manage to come over the difficulties and the outcome deliverables shows that the result of the experiment is quite successful.

1. Functionalities.
   1. Tutorials
      1. Scrollable tutorial cards
      2. Tutorial List with different showing mode
      3. Star/un-star certain tutorials
   2. Quizzes
      1. A detailed question library.
      2. Randomly Choose 5 MCQ each
      3. Alert testing result at the end of the quiz.
      4. Can check the answers after the quiz.
   3. Calendar
      1. Gregorian (normal) calendar displayed.
      2. NexCalendar displayed.
      3. Combined View displayed
      4. Can input exact date with a date picker.
      5. Swipe left/right to different next/previous month.
      6. Double tapped on calendar can go back to today.
   4. General
      1. Localization. Translation in Chinese simplified and Chinese HK.
      2. Theme Color. Four different themes: Tropical, Ocean, Forest and Dark.
2. Project Member
   1. LI Haotian, Ronnie 54780576. Software Engineer, UI UX designer, PPT slides, documentation and screen recording.
   2. Wong Shu Ki 54800543. Go-through video editing.
   3. SHEN Menghan 54780320. Tutorial & quiz data base, translations.

# Screen shots

1. Colorful UI with different themes

A screen shot of a smart phone

Description automatically generatedA close up of electronics

Description automatically generated

1. Calendar system

A screen shot of a smart phone

Description automatically generated A close up of a screen

Description automatically generated Screen of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

1. Tutorial system

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

1. Quiz system

A picture containing screenshot

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

# Personal reflection

=================

LI Haotian 54780576

During the development of group project 1, I obtained a lot personal growth.

First of all, swiftUI framework was brought up by me to use. So the responsibility for coding falls on me. As it’s a quite new framework and was still in beta version during the starting period of development, the setup procedure was quite painful for me. But as I get used to the framework, I speeded up a lot thank to swiftUI. Its structure is very similar to Reactjs with its strong component-based structure. This helps me a lot in catch up my work as the project grows over time. I simply work on a few small components files at a time and combine them later. This not only speeds up the developing procedure, but also increases the readability of the overall project. For a starter, the learning curve can be a little high, but it gets much faster when you get the hang of it.

Secondly, the new canvas feature makes my life much easier in unit testing. Without live/hot reloading, I have to compile millions of times and waste all my time in waiting for compiling, figuring out why it crashed, and so on. But the canvas feature helps me to only preview a little component at a time as I develop. I can clearly mark my progress without worrying the app might crash on somewhere I don’t know. Although, sometimes I still don’t know if it’s the instability of XCode or my codes are actually buggy, the canvas will crash and loads nothing no matter how I change the codes. After a lot research on the Internet, the good old restart saves me from endless trying.

Thirdly, one of the few things to complain about the new framework. The documentation and tutorials are so few compared to UIKit. Besides the few documentations, Apple changed the variable names in latest version and deprecated a lot functions. for example, @ObjectBinding changed into @ObservedObject, and TabbedView is deprecated. These transitions are not very smooth as I need to put a lot time searching again for the new functions and new ways of using property wrappers. Worst case scenarios, the links in those online tutorials which was referred to Apple’s developer webpages are REMOVED! I mean, they can easily redirect to new related features’ pages and that will be much, much more considerate.

Fourthly, I found no libraries for swiftUI when I was developing the APP. Maybe people think the framework is quite complete, I still consider the basic components or animations can be extracted to preventing developers implement them again and again. Due to the small size of group project, this pain spot is not so obvious to me. But I do think people should start contributing in the swiftUI community as soon as possible.

Finally, this trial for swiftUI is considered quite successful and pleasant for me. I explored how swiftUI takes care of reactive programming and its own way of passing information between objects and components. Because of my previous experience in reactjs and flutter, I didn’t take much time getting used to it. Although the implementation of calendar algorithm is annoying, but it feels great when I play with the final product. As for the teamwork, my teammates saved me from a lot of boring tasks such as translation and thinking of questions for the new calendars. I would not come up with the current rich contents without their efforts.

===========================

Wong Shu Ki 54800543

At the beginning of this project, I was pretty confused by the topic as I had never heard of such a new idea. Nowadays, most of the people are used to the current calendar system, and so am I. Therefore, I had never thought of making any changes to these norms. So, it takes me time to understand the aims of this project. Through this project, the division of labor is distinct as it plays an essential role in the success of this project. Since this is my first time to code with Swift, writing a simple function and debug the errors could cost me a day. As the old sayings, practice makes perfect, and now I am more familiar with Xcode. Though, I am always discouraged by the debugging function in Xcode for further coding. As most of the time, I cannot understand the error statements, not even to trace the error. As a result, one of my usual practice is to make comments, which helps me to escape from the trouble of debugging. Meanwhile, I feel grateful for the effort of my groupmates, especially Ronnie, as he is good at coding with Swift and leads us to get the optimized deliverables. Besides my improvement in coding, I acquire a new skill, which is to produce a promotional video for the app. Since I am not used to creating and editing videos, it takes me hours to learn and explore the video-making software.  To my surprise, I find it fun and easy to pick up; therefore, I will put more effort into learning more about this skill.

==========================

SHEN Menghan 54780320

In this report, I will discuss some difficulties that I have met through this project. Firstly, since I was entirely new to designing and developing an app, I always feel overwhelmed by the contents of this course and struggle to catch up with the class. When I tried to design a calendar for this project, I realized that what I learned so far is insufficient to accomplish this task. Then I searched online resources and watched several educational videos. However, I feel frustrated that it is more complex and difficult than I thought. Later, our group leader, Ronnie, took the initiative to be responsible for coding. He contributed much efforts to design this app and finally developed a really cool app. I helped him with the content of the quiz library and course tutorials. I am really thankful for his contributions but also feel guilty that I could not contribute more. For the app, if we have more time, maybe more functionality of the calendar can be added. For example, the calendar can be designed to allow the user to schedule their activities and events.