ITP 342 iOS Development

SwiftUI Homework Proposal

Zihan Qi

January 17, 2022

Contents

Project goals

What We Are Building

We are building a suite of SwiftUI homework material to help students in ITP 342 use SwiftUI in a real-world setting.

In each homework, we will guide the students to build a fully functional app. We will show them what to build, and step them through homework goals and requirements.

What We Are NOT Building

We are building this suite of homework:

- NOT to replace or complement existing homework, since they were designed with different goals and contexts in mind, and has served ITP 342 well for several semesters
- NOT to TEACH how to use SwiftUI, since the lectures are more suited to teach new materials, and the homework should serve more as a hands-on practice session
- NOT to cover all of the possible skills relevant to SwiftUI, since there are more than we could cover, and the APIs are expanding each year

Goals: Minimum

We want each of the homework to be **at least**:

1. Delightful for students to follow through
    - Gives instructions that are simple to follow, avoids unnecessary strictness and vagueness
    - Encourages students to try out different layouts, color schemes, and contents
2. Suitable for students with varying levels of interest in the class
    - Gives general, adaptable requirements, avoids too much detail and strictness
    - Graded requirements should be simple, general, and adaptable
    - Encourages students to expand the app's functionality when possible
3. Easy to debug
    - Broken down into separate components that students can build and debug separately
4. Easy to grade
    - Asks students to include a README, listing requirements they have and have not satisfied
    - Asks students to record graded requirements on the simulator, and upload to Blackboard
    - Asks students to record extra-credit requirements separately, and upload to Blackboard
5. Decoupled from other homework in the suite, so that instructors can select any subsets of homework as they see fit
    - Includes a list summarizing topics instructors should cover before assigning each homework

Goals: Good to Have

It would be great if we could make each of the homework:

1. Fun and useful as a real app that most students want to use for themselves
2. Including game-like experiences in homework, by:
   - List graded requirements not as requirements, but as real-world problems and user needs
   - Encourage students to try out their favorite layouts, color schemes, and contents
   - Drastically reducing the time and commitment a student need to put in to complete homework for full-mark
   - Encourage a few interested students to explore further on their own, by pointing at areas they can improve for their app and directing them to real-world Developer Documentations and WWDC talks
3. Easy to modify, if instructors want to adapt homework to incorporate new materials
   - By developing each homework with a close reference WWDC sessions or Apple tutorials on SwiftUI, so it's easy to reference back to the source and make updates; and, a removal or update of any referenced material will serve as a timely reminder to update the current suite of homework

Tentative Homework Content

SwiftUI Basics (2-3 homework)

Following the structure and content of "Introduction to SwiftUI" (https://developer.apple.com/wwdc20/10119), and SwiftUI Tutorial (https://developer.apple.com/tutorials/swiftui/handling-user-input), but we tell the students that they choose any content! (Lipsticks, Cars, Your favorite characters, etc), as long as the content involves a dynamic list/stack, can be favorited, and can show some details for each item!

- Breaking down the SwiftUI Basics session into 2-3 homework total (static stacks + dynamic stacks, lists and navigation, handling user input like favoriting)
- Structure the homework so that it's easy for students to customize for their own content, color schemes, and layouts.
- Do things in code, but tell students that you can also do things in inspector! "If you want to, check out the WWDC talk."
- Make the graded requirements general, simple, adaptive. Make it come from real-world user problem and user need.
- Don't introduce anything that needs lots of formatting yet.
- Optional ideas:
  - filtering list by favorite (but perhaps the favoriting part still required)
  - Adding new item using a modal.
- Keep in mind, we are not here to TEACH best practices or guide students towards debugging and learning. We are here to develop homework.

Sectioned Data, Complex UI

Following the structure and content of SwiftUI Tutorial (https://developer.apple.com/tutorials/swiftui/composing-complex-interfaces), but we tell the students that they choose any content! (Lipsticks, Cars, Your favorite characters, etc), as long as the content is sectioned + tabbed.

- Don't require List, students can do anything if they make it sectioned!
- Don't require the layout, students can do anything if they make it sectioned!
- Don't require what the other tab shows, students can show anything in the other tab.
- Optional ideas:
  - Point students to add searching capability by using .searchable()
  - Point them to accessibility and voiceover
  - Point them to run on iPad! And add a Text() in the Navigation View.

GitHub Tutorial? (Refer to 435)