Tampereen ammattikorkeakoulu

Mobile App Development 1

iOS Final Project

Chou-Ping Ding

Contents

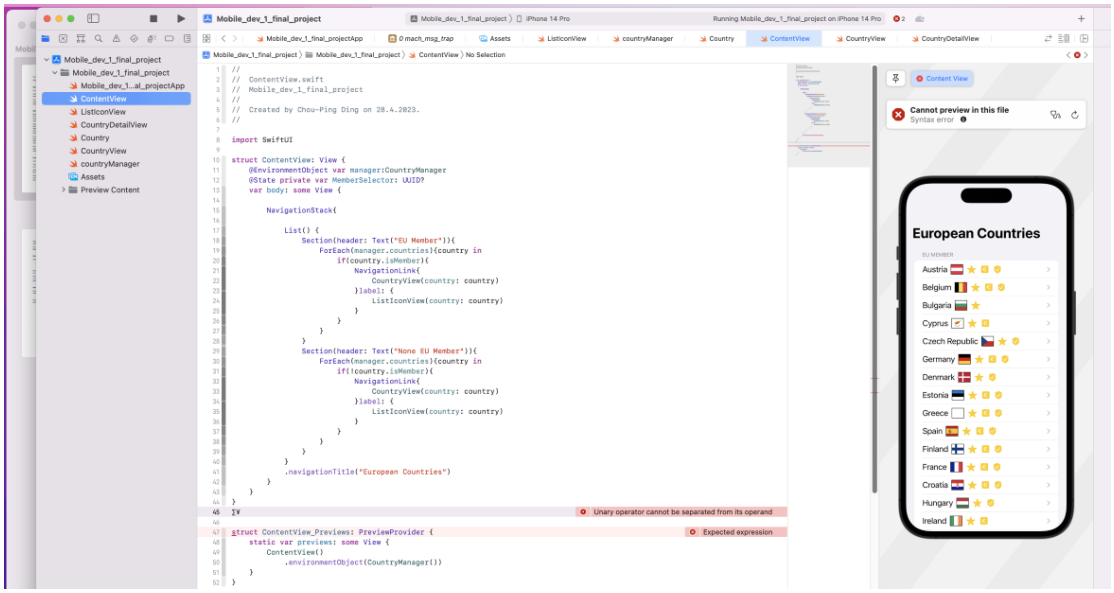
1 iOS Final Project – European Countries App

1.1 Development Process

Since this app's basic structure is based on the lecture demonstrations, the overall picture was not hard to determine. The challenge lay in the part of adding icons to the navigation link list as a nested view, separate the link list by EU membership status, and adding a link to the Wikipedia page of mobile version to the country information page.

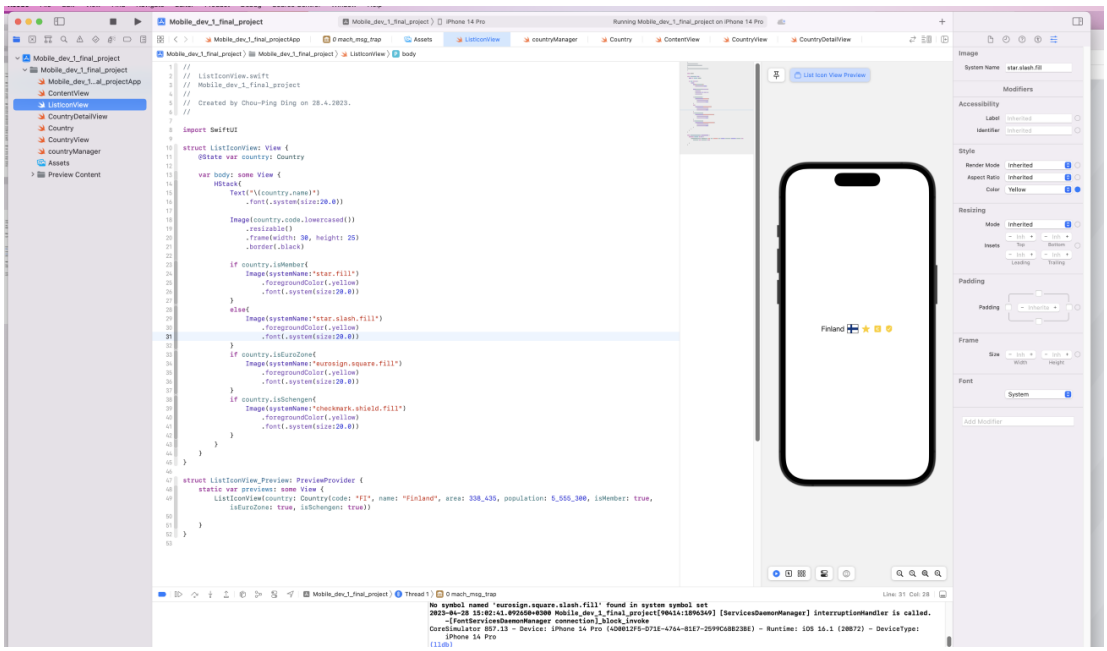
First, the part of making the separation of the list. I used the suggested method of adding sections to the list. While doing this, I've encountered a challenge of how the layer should be and where to implement the variables and if loop for list sorting. I've used a if loop with "isMember" to determine which section of the list the country will be. And with the original structure demonstrated in class, I first came up with each country in their own separate section. But after a few try, I finally get it to work as I wished.

```
struct ContentView: View {
    //list is seperated in to sections depanding on is EU member or not
    @EnvironmentObject var manager:CountryManager
    @State private var MemberSelector: UUID?
    var body: some View {

        NavigationStack{

            List() {
                Section(header: Text("EU Member")){
                    ForEach(manager.countries){country in
                        if(country.isMember){
                            NavigationLink{
                                CountryView(country: country)
                            }label: {
                                ListIconView(country: country)
                            }
                        }
                    }
                }
                Section(header: Text("None EU Member")){
                    ForEach(manager.countries){country in
                        if(!country.isMember){
                            NavigationLink{
                                CountryView(country: country)
                            }label: {
                                ListIconView(country: country)
                            }
                        }
                    }
                }
            }
            .navigationTitle("European Countries")
        }
    }
```



For the string of icons, I created another sub-view in another file called ListIcon-View, in this view, the icons and the country name are in a horizontal stack. Then this view is applied to the link list in the parameter of “labels”. I had encountered some problem when trying to apply this sub-view initially, since the demonstration in class was applying the label at the navigationLink with brackets, and this section don’t take sub-view. This was solved after in class consultation with the teacher and the solution was to use the parameter of “Labels”. I had kept the style simple and didn’t add any other styles to the list, so it remained simple and clear.

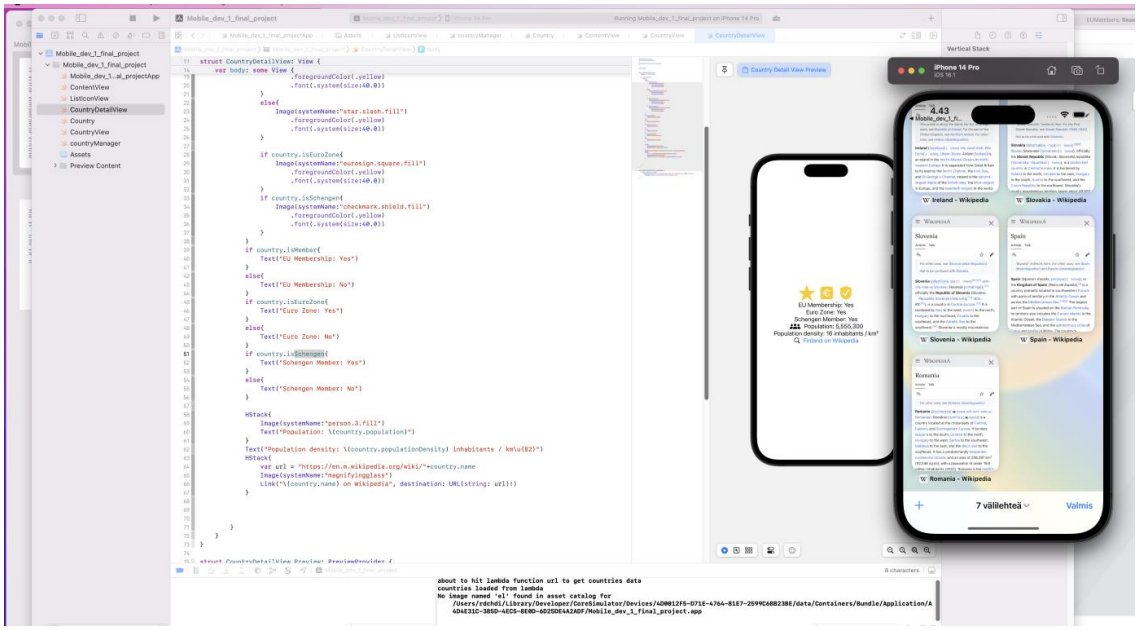
```
struct ListIconView: View {
    @State var country: Country
    //horizontal line up of icons showing the status of the countries  in the country navigation link list
    var body: some View {
        HStack{
            //country name
            Text("\(country.name)")
                .font(.system(size:20.0))
            //country flag
            Image(country.code.lowercased())
                .resizable()
                .frame(width: 30, height: 25)
                .border(.black)
            //star representing EU membership (crossed out if not)
            if country.isMember{
                Image(systemName:"star.fill")
                    .foregroundColor(.yellow)
                    .font(.system(size:20.0))
            }
            else{
                Image(systemName:"star.slash.fill")
                    .foregroundColor(.yellow)
                    .font(.system(size:20.0))
            }
            //Euro icon for the indication of Euro zone
            if country.isEuroZone{
                Image(systemName:"eurosign.square.fill")
                    .foregroundColor(.yellow)
                    .font(.system(size:20.0))
            }
            //shield with checkmark indicating the country belongs to the Schengen area
            if country.isSchengen{
                Image(systemName:"checkmark.shield.fill")
                    .foregroundColor(.yellow)
                    .font(.system(size:20.0))
            }
        }
    }
}
```



For the country detailed view section, I added the link to Wikipedia with a magnifying glass in front of the link button to indicate “research” for more information. Since I used the structure of "https://en.m.wikipedia.org/wiki/"+country.name, the problem occurred when there were spaces in the country name, such as “Great Britain” or “Czech Republic”, as they contain space in their names. The solution I’ve used here to solve the problem is simply create a function to replace the “ “ (space) with “_” (underscore). This is more of a primitive way of dealing with the problem, as when consulted the teacher, he did suggest by creating a hash table and used indexes would be a better practice. But after researching a bit into this method, I felt it’s a bit of an overkill for the current project and the database we had. Of course, there’s also another way to solve this problem, such as by defining the special cases of each country independently, but then it would be a bit messier than I preferred, thus I stuck with the method of using a condition to check and modify the space.

Mobile_dev_1_final_project 〉 Mobile_dev_1_final_project 〉 CountryDetailView 〉 No Selection

```swift
11  struct CountryDetailView: View {
14      var body: some View {
58
59              HStack{
60                  Image(systemName:"person.3.fill")
61                  Text("Population: \(country.population)")
62              }
63              Text("Population density: \(country.populationDensity) inhabitants / km\u{B2}")
64
65              //link to wikipedia
66              HStack{
67                  var url =
                        "https://en.m.wikipedia.org/wiki/"+countryNameWikiForm(country.name)
68                  Image(systemName:"magnifyingglass")
69                  Link("\(country.name) on Wikipedia", destination: URL(string: url)!)
70              }
71          }
72      }
73  }
74
75  //function to replace " " in the country name to "_" so that wikipedia will take it
76  func countryNameWikiForm(_ CountryName: String)->String{
77      let formatedName = CountryName.replacingOccurrences(of: " ", with: "_")
78      return formatedName
79  }
80
```



1.2 End results

1.3 Reflects

This is the first time I work with Swift and Xcode, it was an interesting experience, since recently I was working more with C++ and Java in other courses. It was an interesting experience to use high-level programming language, as it’s in a way quite convenient at the same time incorporating graphical design features into the functional code. Although over all the final project was a simple application, with not so much design to the GUI, the existing features and functions already made it easy to apply a simple sleek appearance, such as creating a navigational link list and sectioning the list.

What I found a bit challenging was the part related to passing variable across different swift files, for example the difference between @State and @Binding which I came across when making the sub-view of icons, I did some research but still need to think it through a bit more thoroughly. I think the code structure is something that I do need to strengthen in the near future.

Sources used

1. Mastering NavigationStack in SwiftUI. Navigator Pattern. https://swiftwithmajid.com/2022/06/15/mastering-navigationstack-in-swiftui-navigator-pattern/
2. How to open web links in Safari. https://www.hackingwithswift.com/quick-start/swiftui/how-to-open-web-links-in-safari
3. How to create a tappable button https://www.hackingwithswift.com/quick-start/swiftui/how-to-create-a-tappable-button
4. How to implement a HashTable using Swift. https://alcivanio.medium.com/how-to-implement-a-hashtable-using-swift-aea632ee75ac
5. Understanding Hash Tables, Dictionaries & Sets with Swift. https://medium.com/swift-algorithms-extras/understanding-hash-tables-dictionaries-sets-with-swift-4605e905973e