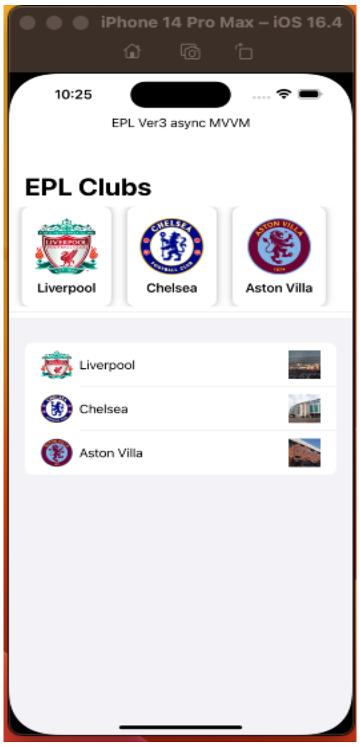
Week4 Tutorial – Key terms CoreLocation, MapKit, MVVM

The tutorial this week continues from last week – where a location address was converted to CLLocationCoordinate2D, and a map region (MKCoordinateRegion) was created based on these coordinates.

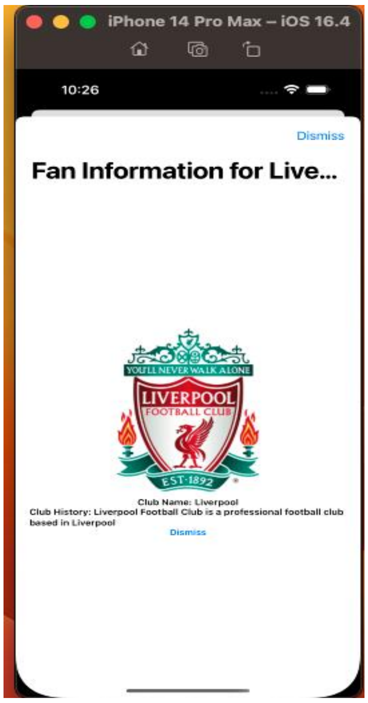
You have been provided with details of some EPL Clubs – clubName, logo, clubLocation, stadium and history. The logo and stadium are image files that must be saved in the *assets* folder of the app – download clubResources.zip and extract the relevant details.
Download clubResources.zip and do as above.

The completed app for this week will function as follows:

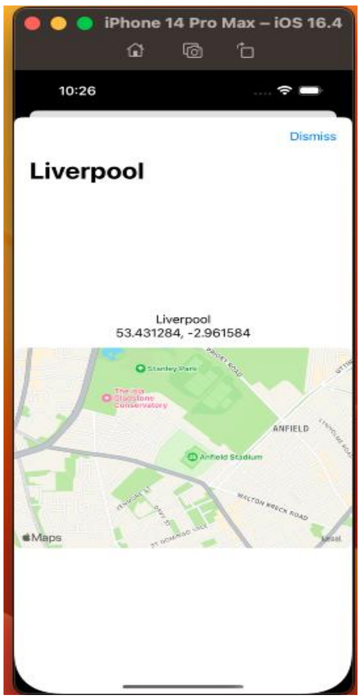
App launch screen:



When the user **taps any** club logo in the top "frame", a sheet view with club details will be presented as shown below with a dismiss button to go back to launch screen:



Club Name: Liverpool
Club History: Liverpool Football Club is a professional football club based in Liverpool
Dismiss

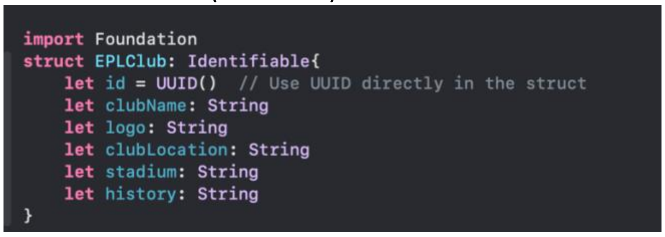


When the user **taps any** row in the bottom "frame", another sheet view with club location on a map will be presented with a dismiss button to go back to launch screen as shown above.

The launch screen design is quite complex in that the club logos are rendered in a horizontal scroll view – to allow many more clubs to be added with a complex tap gesture interaction, revealing a sheet view that shows information about the club that has been tapped. Similarly, clubs in list view in the bottom half too have tap gesture as described in the screen shots.

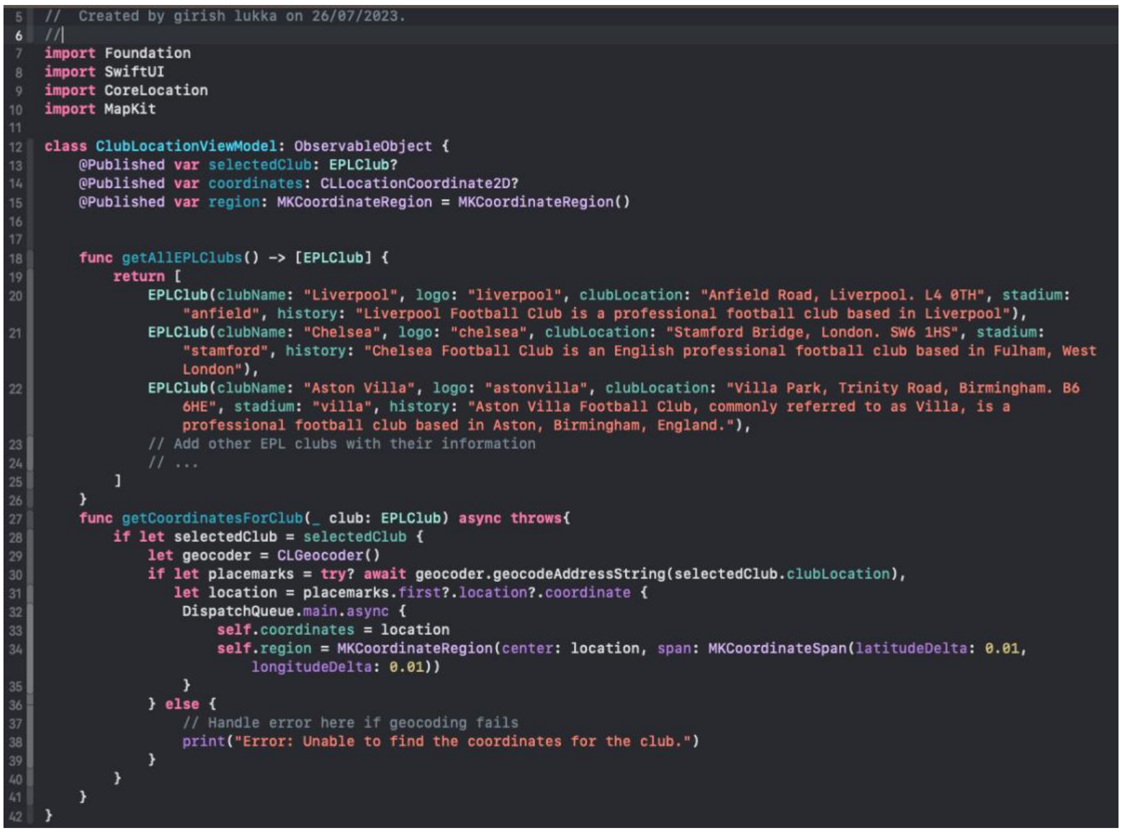
Guidance on building this app:

1. Create a new project, name it EPLClub.
2. Create a model (swift file) – EPLClub that is a struct to represent EPLClub data.



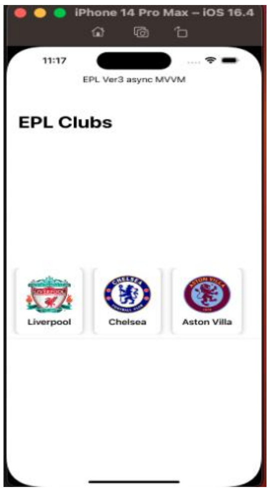
```swift
import Foundation
struct EPLClub: Identifiable{
    let id = UUID()  // Use UUID directly in the struct
    let clubName: String
    let logo: String
    let clubLocation: String
    let stadium: String
    let history: String
}
```

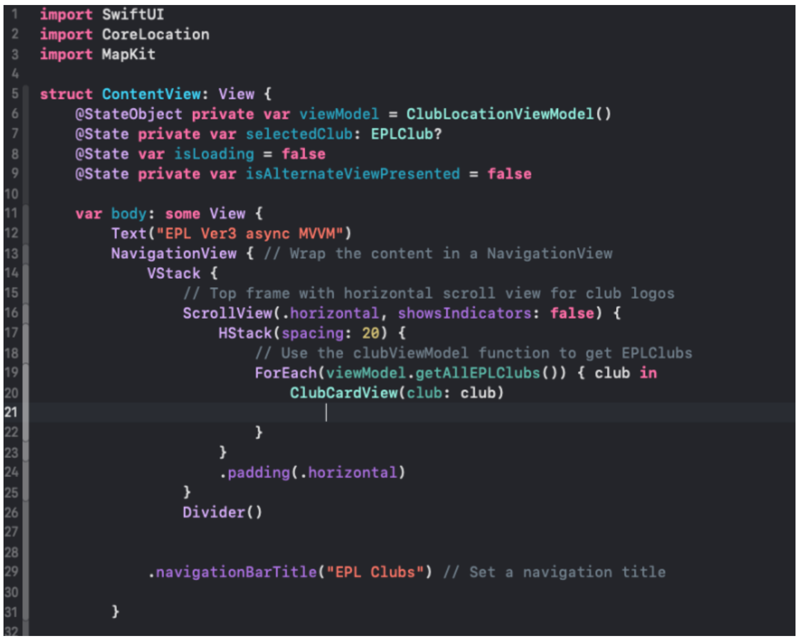
3. Create a class (swift file) – ClubLocationViewModel that is an ObservableObject as shown below:



```swift
//  Created by girish lukka on 26/07/2023.
//
import Foundation
import SwiftUI
import CoreLocation
import MapKit

class ClubLocationViewModel: ObservableObject {
    @Published var selectedClub: EPLClub?
    @Published var coordinates: CLLocationCoordinate2D?
    @Published var region: MKCoordinateRegion = MKCoordinateRegion()


    func getAllEPLClubs() -> [EPLClub] {
        return [
            EPLClub(clubName: "Liverpool", logo: "liverpool", clubLocation: "Anfield Road, Liverpool. L4 0TH", stadium:
                "anfield", history: "Liverpool Football Club is a professional football club based in Liverpool"),
            EPLClub(clubName: "Chelsea", logo: "chelsea", clubLocation: "Stamford Bridge, London. SW6 1HS", stadium:
                "stamford", history: "Chelsea Football Club is an English professional football club based in Fulham, West
                London"),
            EPLClub(clubName: "Aston Villa", logo: "astonvilla", clubLocation: "Villa Park, Trinity Road, Birmingham. B6
                6HE", stadium: "villa", history: "Aston Villa Football Club, commonly referred to as Villa, is a
                professional football club based in Aston, Birmingham, England."),
            // Add other EPL clubs with their information
            // ...
        ]
    }
    func getCoordinatesForClub(_ club: EPLClub) async throws{
        if let selectedClub = selectedClub {
            let geocoder = CLGeocoder()
            if let placemarks = try? await geocoder.geocodeAddressString(selectedClub.clubLocation),
               let location = placemarks.first?.location?.coordinate {
                DispatchQueue.main.async {
                    self.coordinates = location
                    self.region = MKCoordinateRegion(center: location, span: MKCoordinateSpan(latitudeDelta: 0.01,
                        longitudeDelta: 0.01))
                }
            } else {
                // Handle error here if geocoding fails
                print("Error: Unable to find the coordinates for the club.")
            }
        }
    }
}
```

4. Modify ContentView so that it shows a horizontal scrollbar that has each club logo with a name under the logo – this can be thought of as displaying a set of **playing cards,** with each card generated with club logo and name.

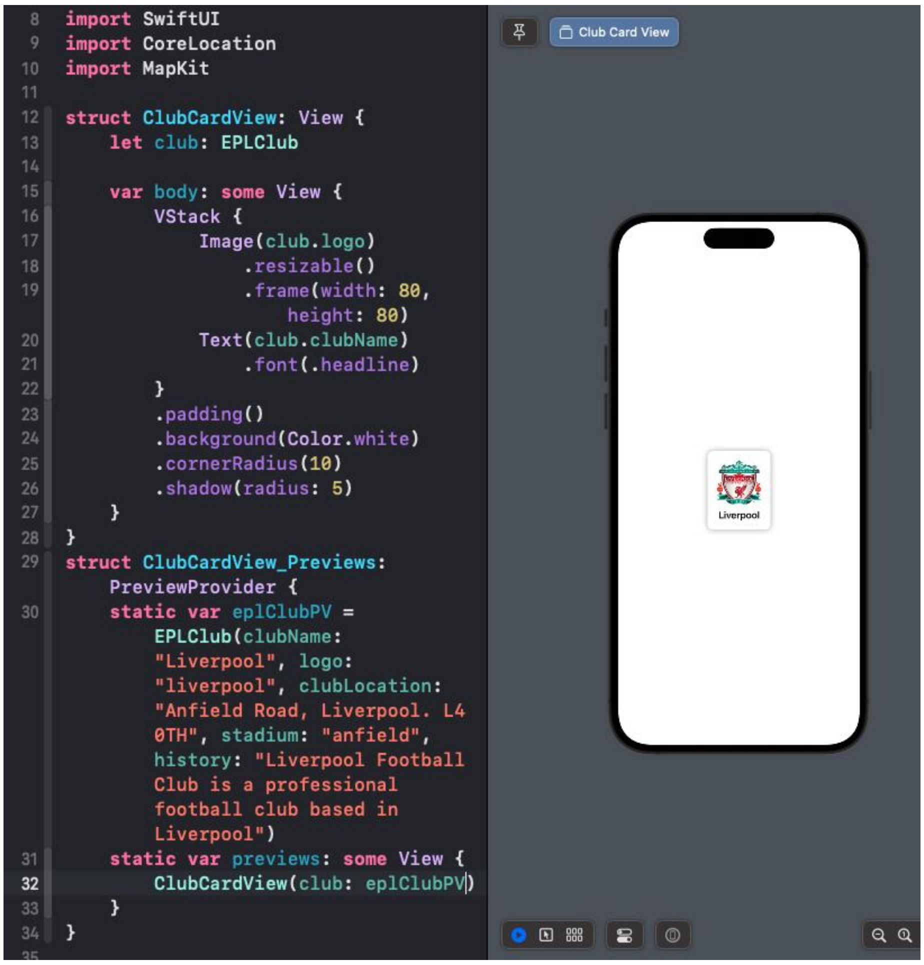




```swift
import SwiftUI
import CoreLocation
import MapKit

struct ContentView: View {
    @StateObject private var viewModel = ClubLocationViewModel()
    @State private var selectedClub: EPLClub?
    @State var isLoading = false
    @State private var isAlternateViewPresented = false

    var body: some View {
        Text("EPL Ver3 async MVVM")
        NavigationView { // Wrap the content in a NavigationView
            VStack {
                // Top frame with horizontal scroll view for club logos
                ScrollView(.horizontal, showsIndicators: false) {
                    HStack(spacing: 20) {
                        // Use the clubViewModel function to get EPLClubs
                        ForEach(viewModel.getAllEPLClubs()) { club in
                            ClubCardView(club: club)

                        }
                    }
                    .padding(.horizontal)
                }
                Divider()


            .navigationBarTitle("EPL Clubs") // Set a navigation title

        }
```

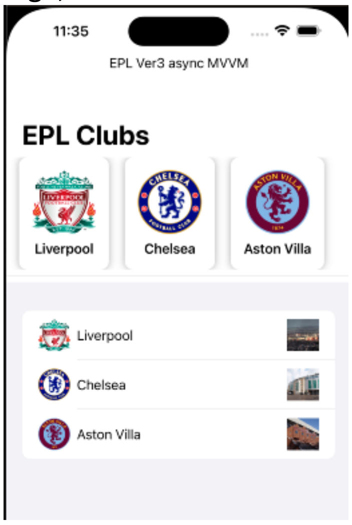
Examine the code snippet and see that “club card” is actually a ClubCardView that takes a parameter - club.

Create ClubCardView (like a playing card) – a SwiftUI View file that generates the card that is shown in the scrollview.

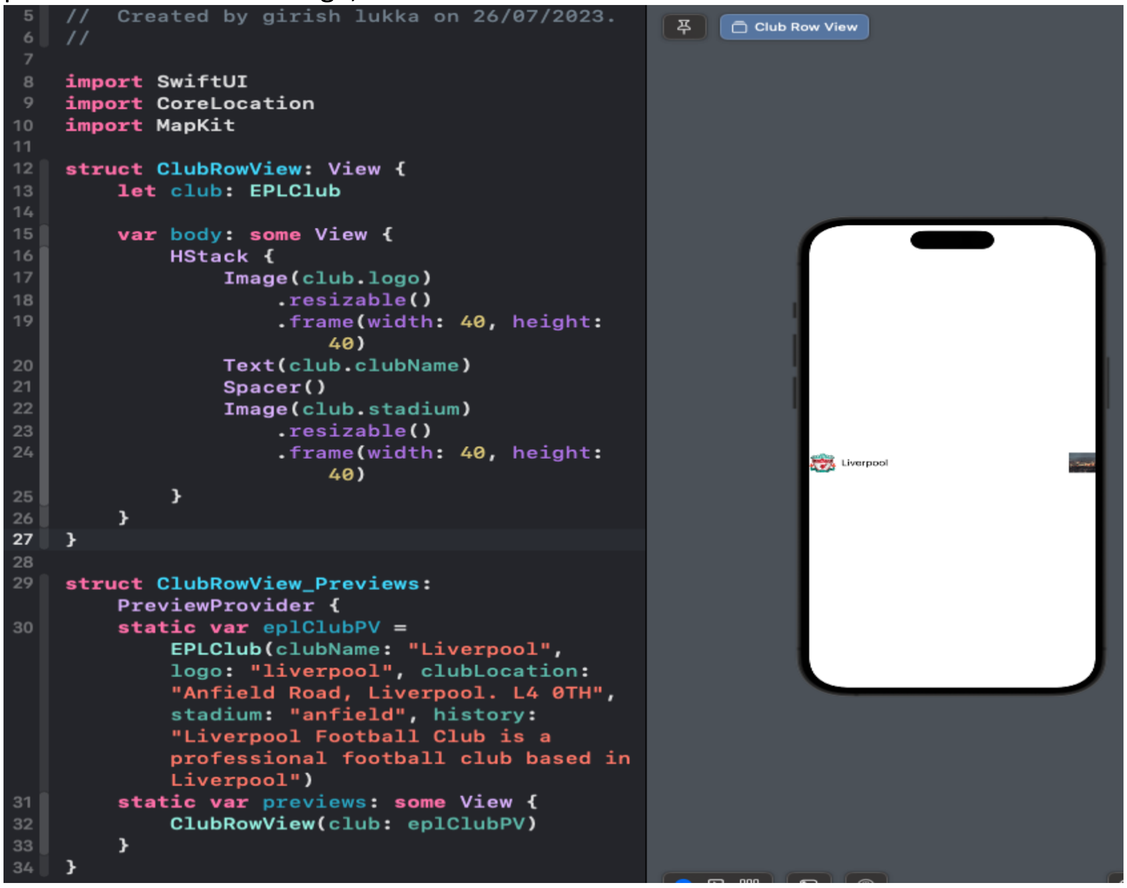


```swift
import SwiftUI
import CoreLocation
import MapKit

struct ClubCardView: View {
    let club: EPLClub

    var body: some View {
        VStack {
            Image(club.logo)
                .resizable()
                .frame(width: 80,
                    height: 80)
            Text(club.clubName)
                .font(.headline)
        }
        .padding()
        .background(Color.white)
        .cornerRadius(10)
        .shadow(radius: 5)
    }
}
struct ClubCardView_Previews:
    PreviewProvider {
    static var eplClubPV =
        EPLClub(clubName:
        "Liverpool", logo:
        "liverpool", clubLocation:
        "Anfield Road, Liverpool. L4
        0TH", stadium: "anfield",
        history: "Liverpool Football
        Club is a professional
        football club based in
        Liverpool")
    static var previews: some View {
        ClubCardView(club: eplClubPV)
    }
}
```

Next, add a list view of clubs, below the horizontal scrollview with each row that shows club logo, club name and club stadium as shown below:

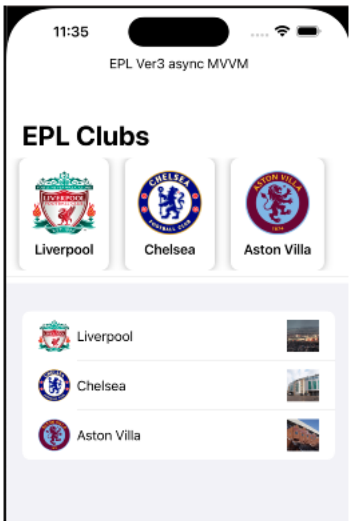


Again, each row is similar to the card that was created earlier, except this is a horizontal profile with an extra image, called ClubRowView



```
//  Created by girish lukka on 26/07/2023.
//

import SwiftUI
import CoreLocation
import MapKit

struct ClubRowView: View {
    let club: EPLClub

    var body: some View {
        HStack {
            Image(club.logo)
                .resizable()
                .frame(width: 40, height:
                    40)
            Text(club.clubName)
            Spacer()
            Image(club.stadium)
                .resizable()
                .frame(width: 40, height:
                    40)
        }
    }
}

struct ClubRowView_Previews:
    PreviewProvider {
    static var eplClubPV =
        EPLClub(clubName: "Liverpool",
        logo: "liverpool", clubLocation:
        "Anfield Road, Liverpool. L4 0TH",
        stadium: "anfield", history:
        "Liverpool Football Club is a
        professional football club based in
        Liverpool")
    static var previews: some View {
        ClubRowView(club: eplClubPV)
    }
}
```

This completes the build-up of the interface for the launch screen. The app should render showing the clubs without any interaction.



The interaction will be covered in the lecture which you should follow in the seminar to complete the app build.