iOS Final Project

公車即時動態查詢APP

START

資工三 108590452 林峻霆

CONTENTS

目錄

01

START

動機與參考

[1] 動機與參考

- App　Name: 公車即時動態查詢APP
- Motivation: 不管是上班族通勤或是學生上學，公車都是很多人會選擇的大眾交通工具。於是我們想要設計一款公車即時動態查詢的APP，讓使用者無論在哪都可以即時查詢公車動態、附近站牌、到站時刻、等資訊！(目前是以台北市區為主)
- Reference : Bus+、台北等公車、等APP

02

START

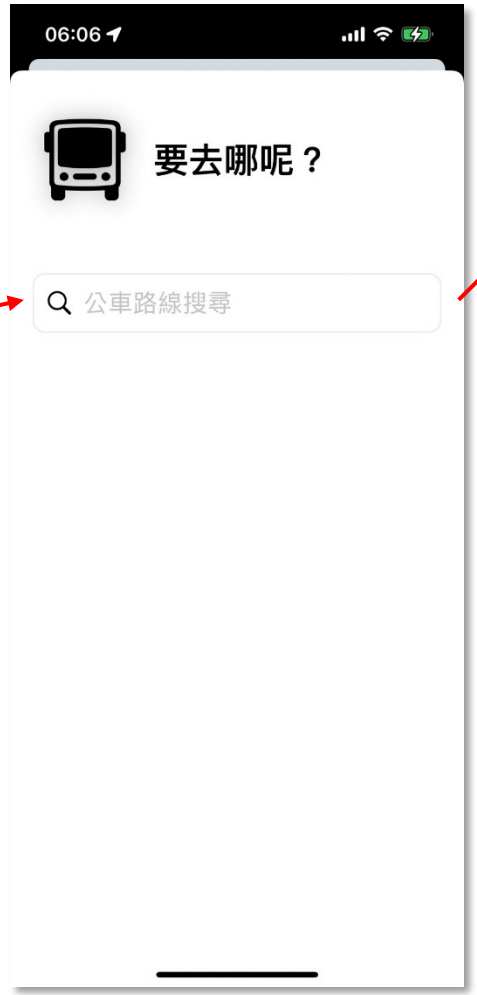
UI簡要描述

[2] UI簡要描述

主頁面包括：首頁、地圖、個人、設定等的TabView

[2] UI簡要描述

➢ 點選“搜尋公車路線”按鈕 ➢ 輸入公車路線 ➢ 即可查看該公車路線



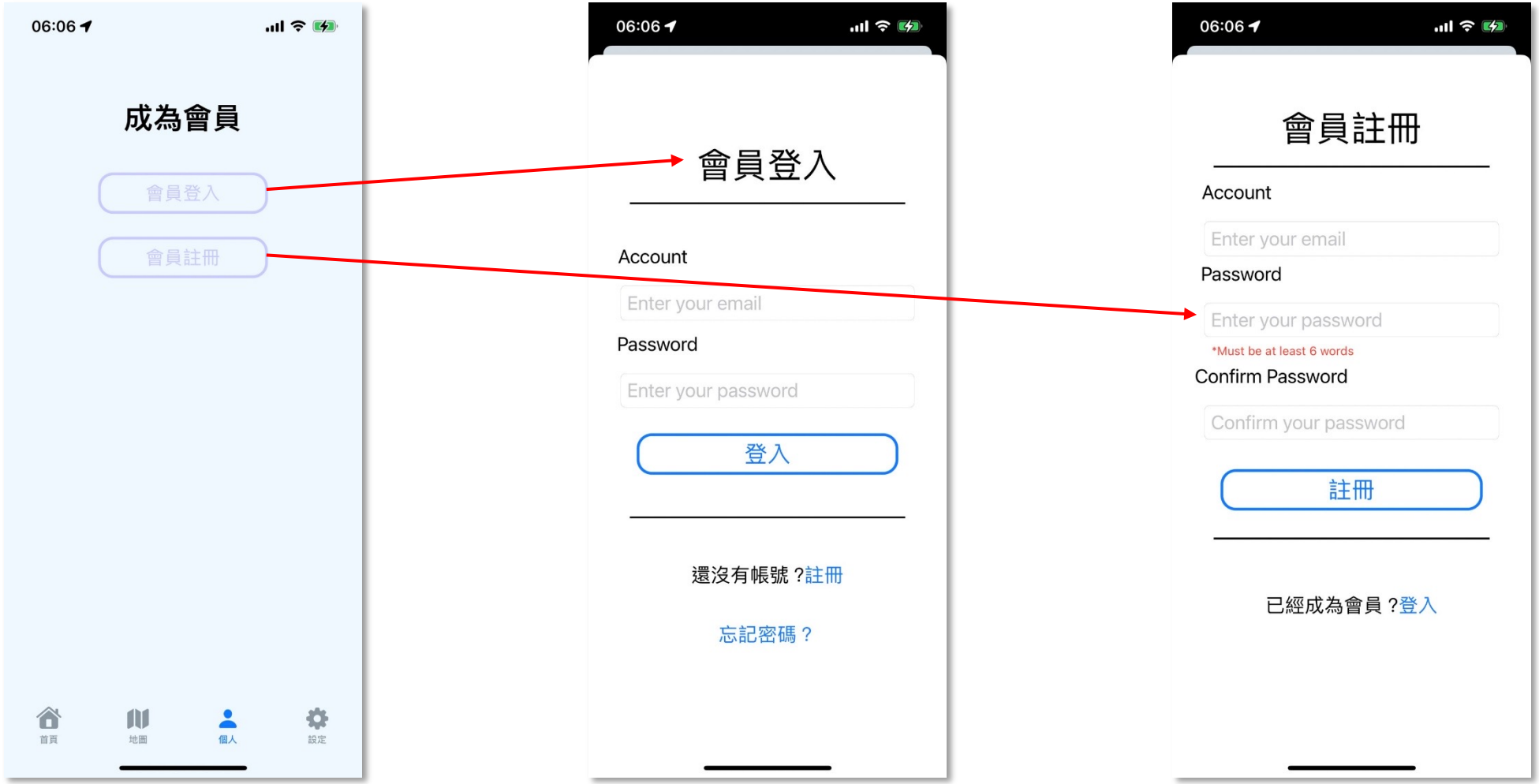
[2] UI簡要描述

- 可查看定位附近的公車站牌

- 關於開發此App的資訊

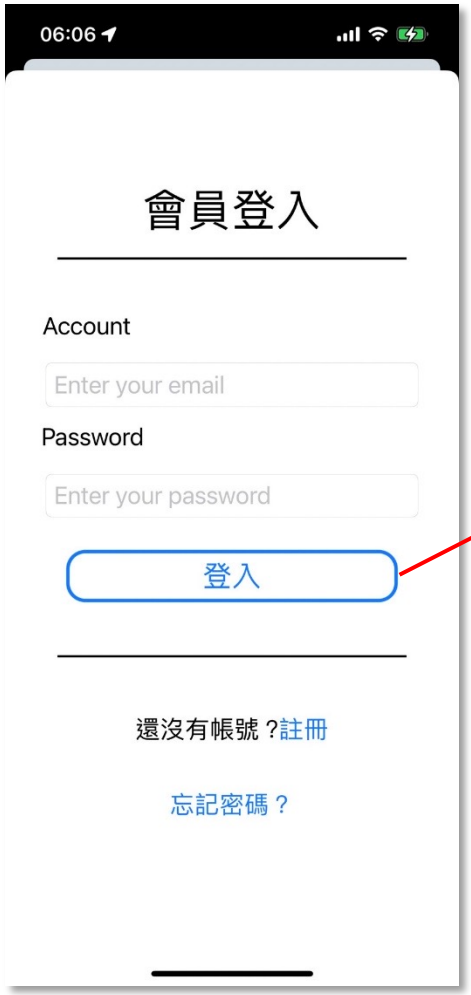
[2] UI簡要描述

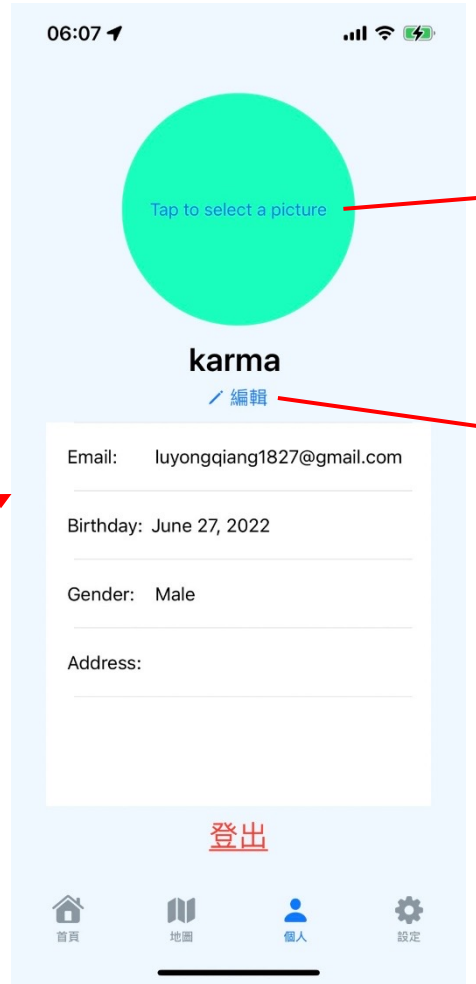
➢ 點選“會員登入”/ “會員註冊”按鈕 ➢ 會員登入 / 會員註冊

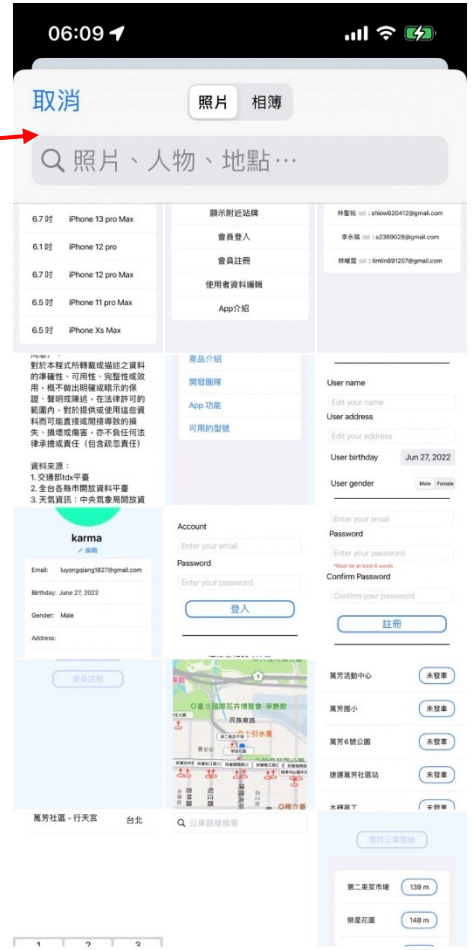


2 UI簡要描述

➢ 點選“搜尋公車路線”按鈕 ➢ 輸入公車路線 ➢ 即可查看該公車路線







03

START

使用iOS的部分和技術

3 使用iOS的部分和技術

- 串接到中央氣象局開放資料平臺的天氣預報API、抓取和呈現資料

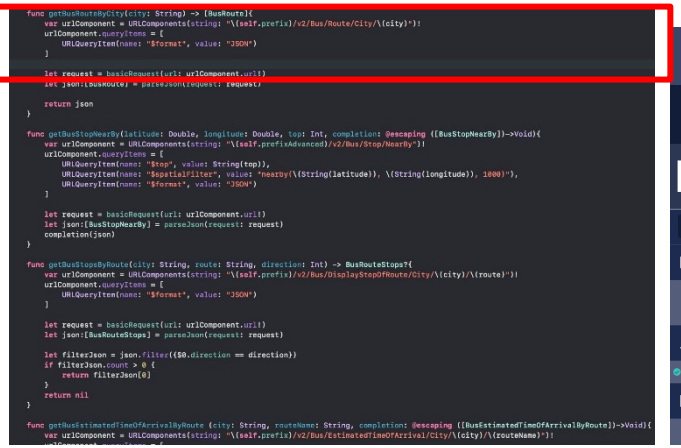
```
import Foundation

struct WeatherData: Codable {
    let success: String
    let records: Record

    struct Record: Codable{
        let datasetDescription: String
        let location: [Location]

        struct Location: Codable{
            let locationName: String
            let weatherElement: [WeatherElement]
        }
    }
}

struct WeatherElement: Codable{
    let elementName: String
    let time: [Time]
}

struct Time: Codable{
    let startTime: String
    let endTime: String
    let parameter: Parameter

    struct Parameter: Codable{
        let parameterName: String
        let parameterUnit: String?
        let parameterValue: String?
    }
}
```


```
func loadData(){
    let url = URL(string:
        "https://opendata.cwb.gov
        .tw/api/v1/rest/datastore/F-C0032-001?Authorization=CWB-97F69598-88F8-4EF1
        -A7D6-F8D68CB225DA&format=JSON&locationName=%E8%87%BA%E5%8C%97%E5%B8%82&el
        ementName=")!
    var request = URLRequest(url: url)
    request.httpMethod = "GET"
    request.setValue("application/json", forHTTPHeaderField: "Accept")

    URLSession.shared.dataTask(with: request) { data, response, error in
        let decoder = JSONDecoder()

        if let data = data {
            do {
                let weatherResponse = try decoder.decode(WeatherData.self, from:
                    data)
                DispatchQueue.main.async {
                    self.weather = weatherResponse
                    self.weatherWx =
                        weatherResponse.records.location[0].weatherElement[0]
                    self.weatherPop =
                        weatherResponse.records.location[0].weatherElement[1]
                    self.weatherMinT =
                        weatherResponse.records.location[0].weatherElement[2]
                    self.weatherCI =
                        weatherResponse.records.location[0].weatherElement[3]
                    self.weatherMaxT =
                        weatherResponse.records.location[0].weatherElement[4]
                }
                print(weatherResponse.success)
            } catch  {
                print(error)
            }
        }
    }.resume()
}
```

```
var body: some View{
    VStack(){
        Text("公車即時動態查詢APP")
            .font(.system(size: 25))
            .bold()
            .padding(.top, 50)

        if let weather = weather { 2 ⚠ Value 'weather' was defined but never used; con...
            let parameterWx = weatherWx!.time[0].parameter
            let parameterPop = weatherPop!.time[0].parameter
            let parameterMinT = weatherMinT!.time[0].parameter
            let parameterMaxT = weatherMaxT!.time[0].parameter
            let photoWT = String(parameterWx.parameterValue!)
            let degree = String((Int(parameterMinT.parameterName)! +
                Int(parameterMaxT.parameterName)!)/2)
            HStack{
                VStack {
                    Text("\(degree)°").font(.system(size: 30)).bold()
                    Text("\(parameterMinT.parameterName)°/\(parameterMaxT
                        .parameterName)°")
                        .font(.system(size: 15)).bold()
                        .fixedSize()
                }
                .frame(width: 100, height: 130)
                .padding(.leading,10)

                Divider().frame(width: 2, height: 70).overlay(.gray)

                Image(photoWT)
                    .resizable()
                    .frame(width: 70, height: 70)
                    .padding(.leading,20)

                VStack{
                    Text(parameterWx.parameterName)
                        .font(.system(size: 20))
                        .bold()
                        .multilineTextAlignment(.center)
                    Text("降雨率:\(parameterPop.parameterName)%")
                        .font(.system(size: 12))
                }
```

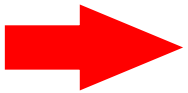
3 使用iOS的部分和技術

- 串接到運輸資料流通服務平臺的公車API、抓取資料和呈現搜尋功能



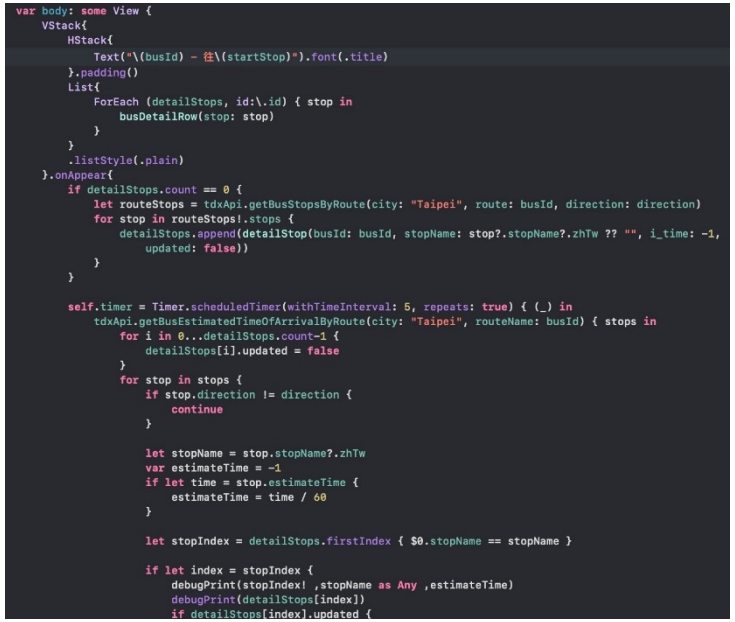

```
import SwiftUI
struct BusStop:Identifiable {
    var id:String{busId}
    var busId:String
    var startStop:String
    var endStop:String
    var city:String
}
extension BusStop{
    static let demo = BusStop(busId: "307", startStop: "板橋", endStop: "撫遠街", city:
        "台北")
}

struct SearchResultRowView: View {
    @State var stop:BusStop;
    @State var detail = false

    var body: some View {
        HStack{
            VStack(alignment: .leading){
                Text(stop.busId)
                Text("\(stop.startStop) - \(stop.endStop)")
            }
            Spacer()
            VStack(spacing:10){
                Button {
                    detail = true
                } label: {
                    Image(systemName: "star")
                        .resizable()
                        .frame(width: 30, height: 30)
                }
                .sheet(isPresented: $detail) {
                    busDetail(search: stop.busId, startStop: stop.startStop, endStop:
                        stop.endStop)
                }
                Text(stop.city)
            }
        }.padding()
    }
}
```

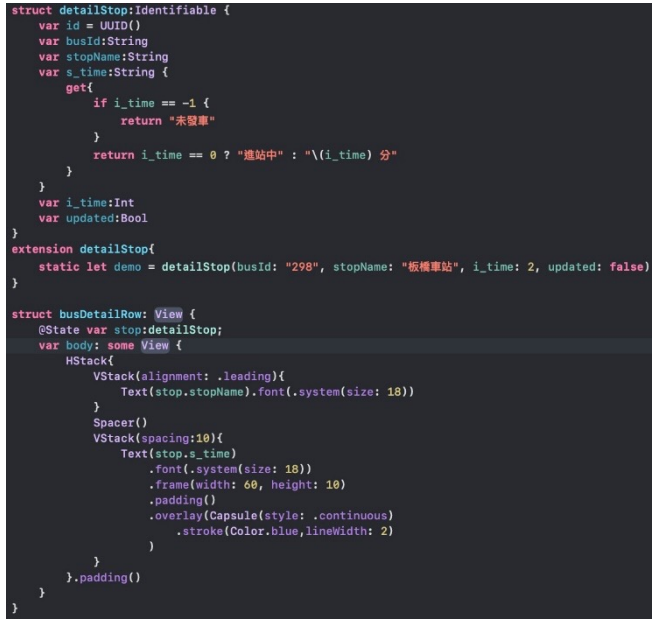
```
struct SearchBusView: View {
    let tdxApi = tdxAPI()
    var taipeiRoutes:[BusRoute] = []
    @State var busStops:[BusStop] = []
    @State var search:String = "";
    @FocusState private var isFocused: Bool
    init(){
        taipeiRoutes = tdxApi.getBusRouteByCity(city: "Taipei")
    }
    func searchChanged(to value:String){
        busStops = []
        if search != "" {
            for route in taipeiRoutes {
                if let routeName = route.routeName {
                    if routeName.zhTw.contains(search){
                        busStops.append(BusStop(busId: routeName.zhTw , startStop:
                            route.departureStopNameZh!, endStop:
                            route.destinationStopNameZh!, city: "台北"))
                    }
                }
            }
        }
    }
    var body: some View {
        VStack{
            HStack {
                Image(systemName: "bus")
                    .resizable()
                    .frame(width: 70, height: 70)
                    .shadow(radius: 10)
                Text("要去哪呢？")
                    .font(.system(size:30))
                    .bold()
                    .padding()
            }
            .frame(width: 350, height: 100)
            .padding(.trailing, 70)


            HStack{
                HStack{
                    Image(systemName: "magnifyingglass")
                        .padding(.leading, 10)
```

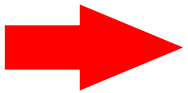


3 使用iOS的部分和技術

- 串接到運輸資料流通服務平臺的公車API、抓取資料和呈現公車到站時間







3 使用iOS的部分和技術

- 串接到運輸資料流通服務平臺的公車API、抓取資料、在首頁和地圖上呈現附近站牌

```
func getBusStopNearBy(latitude: Double, longitude: Double, top: Int, completion: @escaping ([BusStopNearBy])->Void){
    var urlComponent = URLComponents(string: "\(self.prefixAdvanced)/v2/Bus/Stop/NearBy")!
    urlComponent.queryItems = [
        URLQueryItem(name: "$top", value: String(top)),
        URLQueryItem(name: "$spatialFilter", value: "nearby(\(String(latitude)), \(String(longitude)), 1000)"),
        URLQueryItem(name: "$format", value: "JSON")
    ]

    let request = basicRequest(url: urlComponent.url!)
    let json:[BusStopNearBy] = parseJson(request: request)
    completion(json)
}

func getBusStopsByRoute(city: String, route: String, direction: Int) -> BusRouteStops?{
    var urlComponent = URLComponents(string: "\(self.prefix)/v2/Bus/DisplayStopOfRoute/City/\(city)/\(route)")!
    urlComponent.queryItems = [
        URLQueryItem(name: "$format", value: "JSON")
    ]

    let request = basicRequest(url: urlComponent.url!)
    let json:[BusRouteSteps] = parseJson(request: request)

    let filterJson = json.filter({$0.direction == direction})
    if filterJson.count > 0 {
        return filterJson[0]
    }
    return nil
}

func getBusEstimatedTimeOfArrivalByRoute (city: String, routeName: String, completion: @escaping ([BusEstimatedTimeOfArrivalByRoute])->Void){
    var urlComponent = URLComponents(string: "\(self.prefix)/v2/Bus/EstimatedTimeOfArrival/City/\(city)/\(routeName)")!
    urlComponent.queryItems = [

        URLQueryItem(name: "$format", value: "JSON")
    ]

    let request = basicRequest(url: urlComponent.url!)
    let json:[BusEstimatedTimeOfArrivalByRoute] = parseJson(request: request)
    completion(json)
}
```

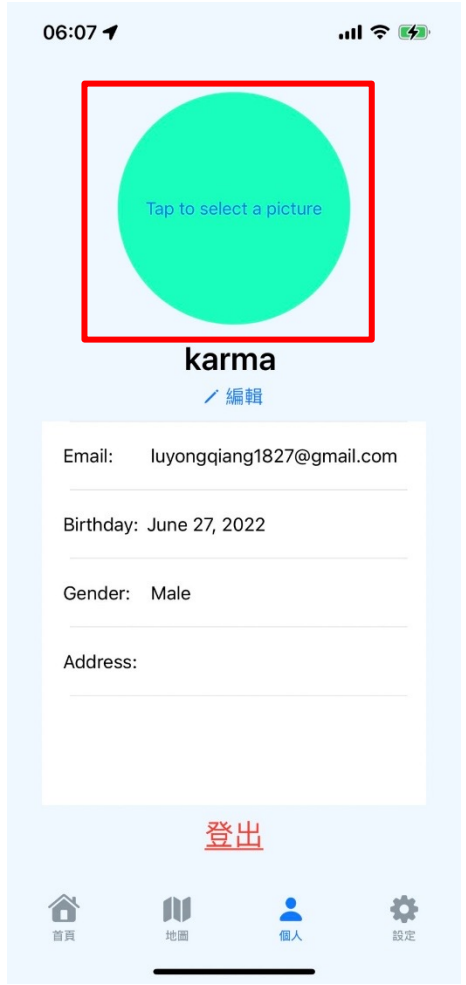
```
struct nearStop:Identifiable {
    var id:String{busId}
    var busId:String
    var name :String
    var distance :Double
}
extension nearStop{
    static let demo = nearStop(busId: "307", name: "板橋車站", distance: 200)
}

struct nearStopRow: View {
    @State var stop:nearStop;
    var body: some View {
        HStack{
            VStack(alignment: .leading){
                Text(stop.name).font(.system(size: 18))
            }
            Spacer()
            VStack(spacing:10){
                Text("\(String(format: "%.f", stop.distance)) m")
                    .font(.system(size: 18))
                    .frame(width: 60, height: 10)
                    .padding()
                    .overlay(Capsule(style: .continuous)
                        .stroke(Color.blue,lineWidth: 2)
                    )
            }
        }.padding()
    }
}
```

```
struct MapPage: View {
    @EnvironmentObject var viewModel: ContentViewModel

    var body: some View{
        VStack{
            Text("附近站牌")
                .font(.system(.title))
                .bold()
                .padding(.top,50)
            VStack{
                Map(coordinateRegion: $viewModel.region,
                    showsUserLocation: true,
                    annotationItems: viewModel.MapLocations,
                    annotationContent: { location in
                        MapAnnotation(coordinate: location.coordinate){
                            VStack {
                                Text(location.stopName?.Zh_tw ?? "Stop")
                                    .font(.system(size: 10))
                                    .frame(width: 80, height: 20)
                                    .background(Color.white.opacity(0.7))
                                    .fixedSize()
                                    .overlay(RoundedRectangle(cornerRadius: 6)
                                        .stroke(Color.gray, lineWidth: 1)
                                    )
                                Image(systemName: "mappin.and.ellipse")
                                    .foregroundColor(.red)

                            }.padding(10)
                        }
                    })
                    .ignoresSafeArea()
            }
            .frame(width: 350, height: 550)
            .cornerRadius(50)
            .padding(.bottom,20)
        }
        .frame(width: 400, height: 750)
    }
```

3 使用iOS的部分和技術

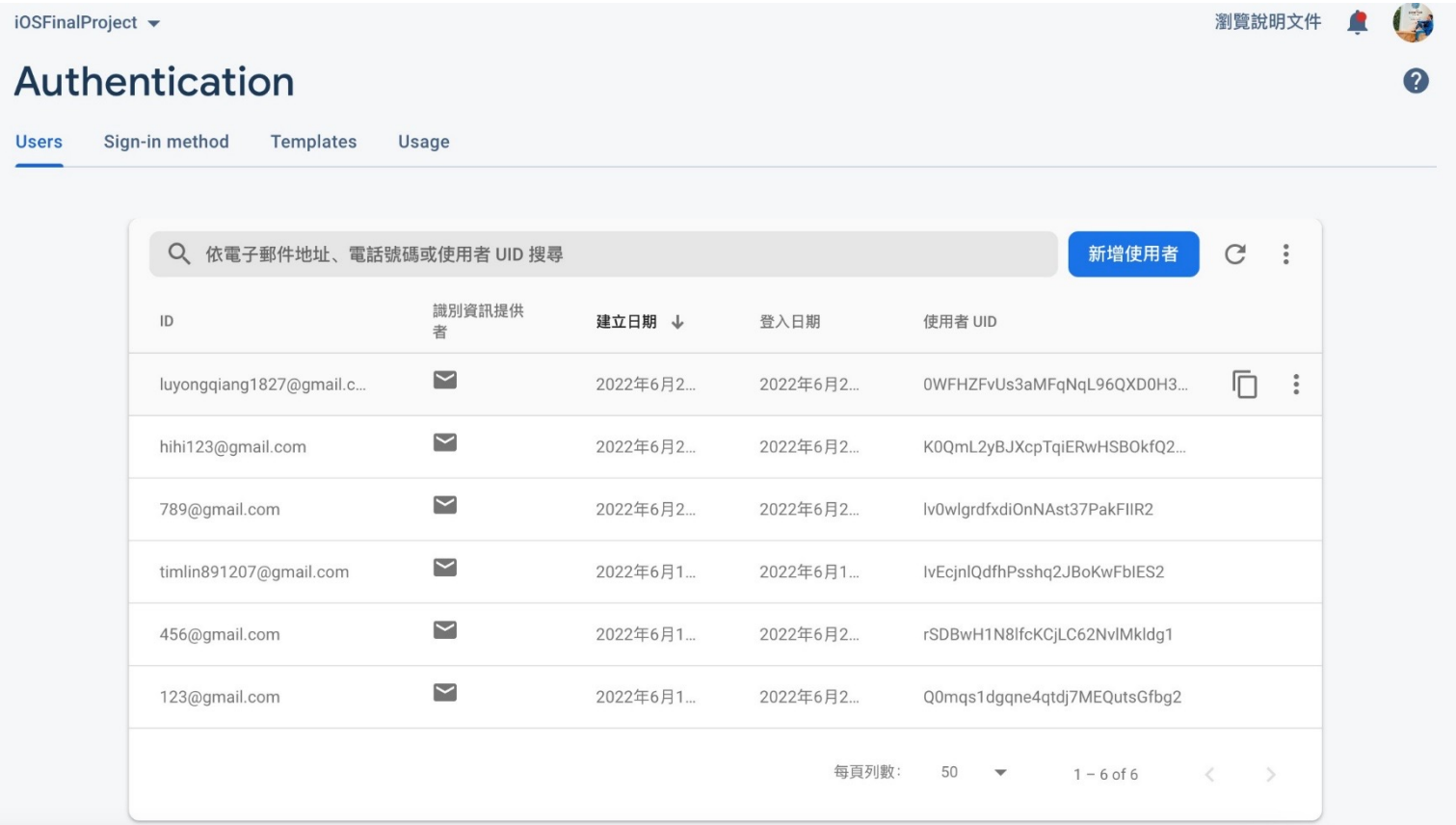
- 使用ImagePicker來選擇手機相簿裡的照片和上傳

```
struct ImagePicker: UIViewControllerRepresentable {
    @Binding var image: UIImage?

    class Coordinator: NSObject, PHPickerViewControllerDelegate {
        var parent: ImagePicker

        init(_ parent: ImagePicker) {
            self.parent = parent
        }

        func picker(_ picker: PHPickerViewController, didFinishPicking results: [PHPickerResult]) {
            picker.dismiss(animated: true)

            guard let provider = results.first?.itemProvider else { return }

            if provider.canLoadObject(ofClass: UIImage.self) {
                provider.loadObject(ofClass: UIImage.self) { image, _ in
                    self.parent.image = image as? UIImage
                }
            }
        }
    }

    func makeUIViewController(context: Context) -> PHPickerViewController {
        var config = PHPickerConfiguration()
        config.filter = .images

        let picker = PHPickerViewController(configuration: config)
        picker.delegate = context.coordinator
        return picker
    }

    func updateUIViewController(_ uiViewController: PHPickerViewController, context: Context) {

    }

    func makeCoordinator() -> Coordinator {
        Coordinator(self)
    }
}
```



[3] 使用iOS的部分和技術

- 使用firebase來存取使用者資料：帳號、密碼、等等



04

START

App Demo時間

iOS Final Project

Thanks For Your Attention

END