

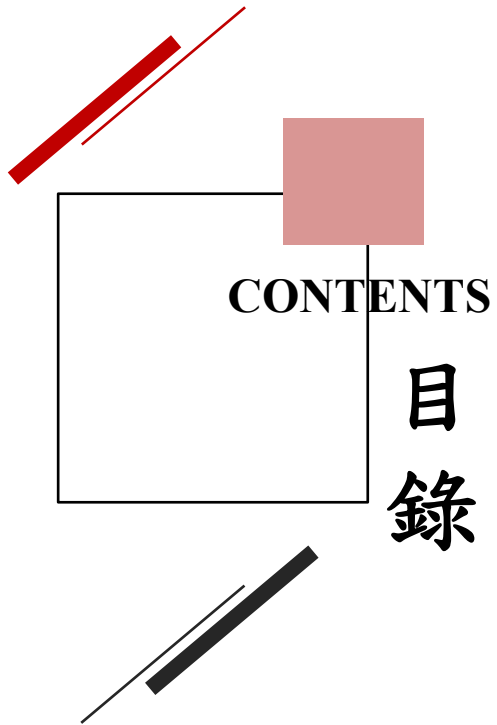


iOS Final Project

# 公車即時動態查詢APP

START

資工三 108590452 林峻霆



[1] 動機與參照

[2] UI簡要描述

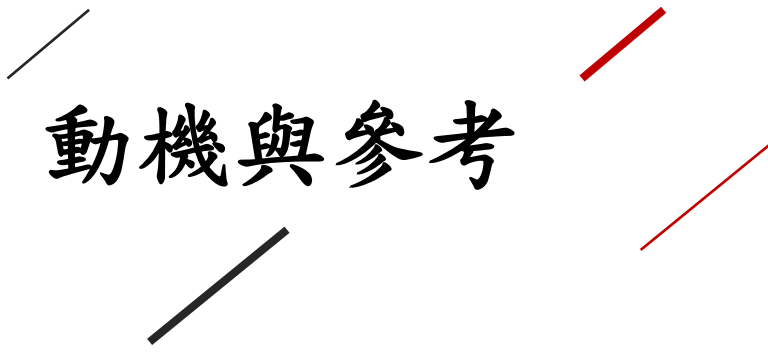
[3] 使用iOS的部分和技術

[4] App Demo時間



# 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

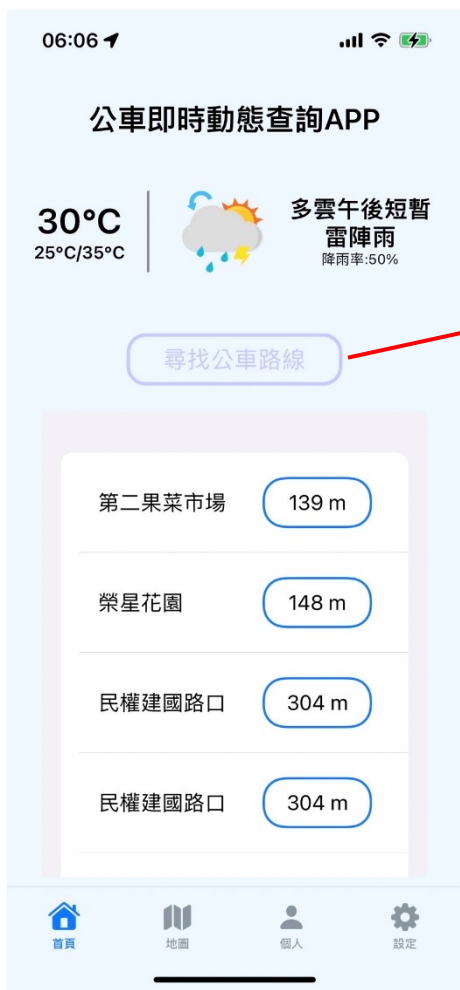
## 2 UI簡要描述

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



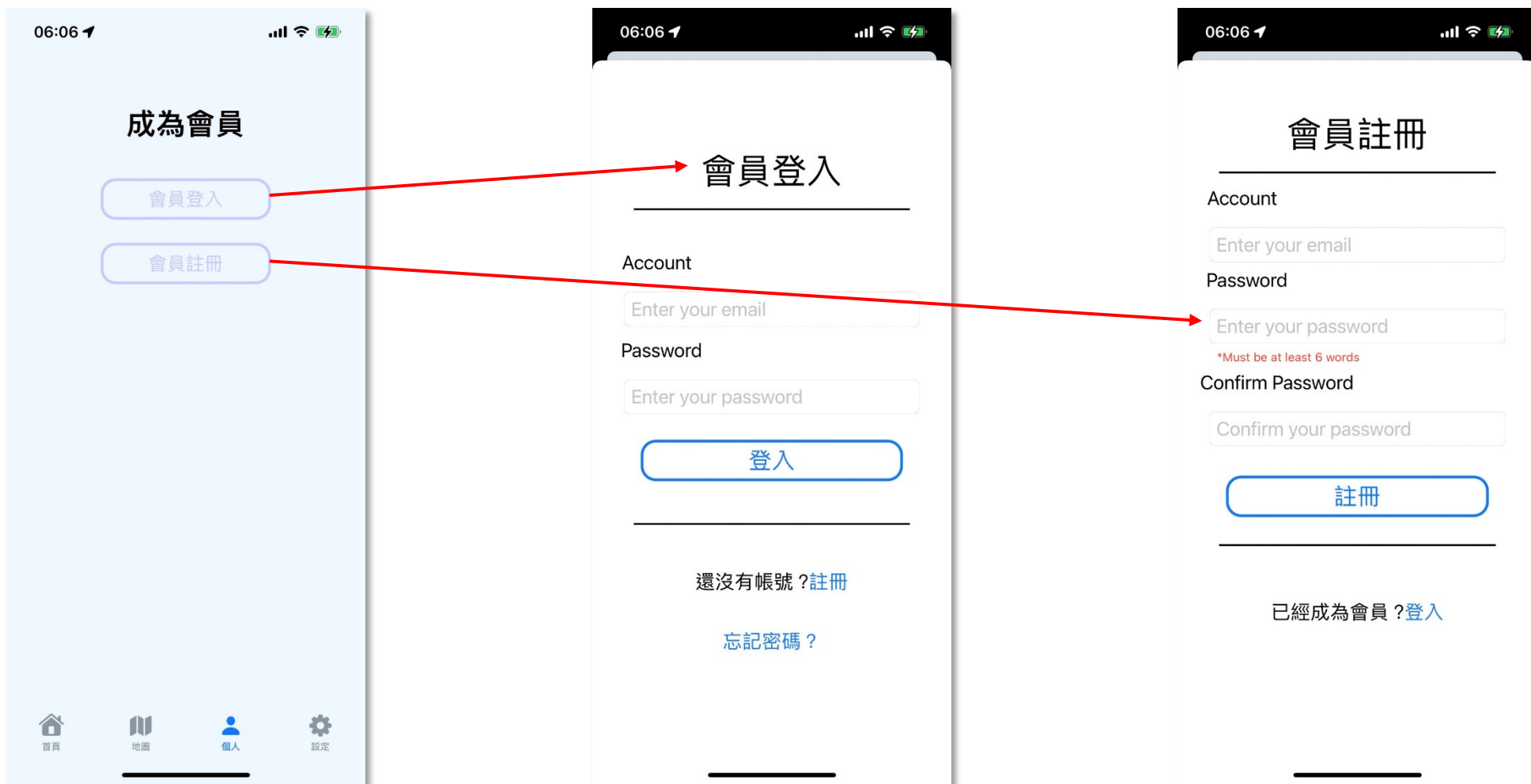
- 關於開發此App的資訊





## [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

06:06

### 會員登入

Account

Enter your email

Password

Enter your password

登入

還沒有帳號? [註冊](#)

[忘記密碼?](#)

06:07

Tap to select a picture

karma

編輯

Email: luyongqiang1827@gmail.com

Birthday: June 27, 2022

Gender: Male

Address:

登出

首頁 地圖 個人 設定

06:07

### 編輯使用者資料

User name

Edit your name

User address

Edit your address

User birthday

Jun 27, 2022

User gender

Male Female

Submit

06:09

取消 照片 相簿

Q 照片、人物、地點...

6.7 吋 iPhone 13 pro Max  
6.1 吋 iPhone 12 pro  
6.7 吋 iPhone 12 pro Max  
6.5 吋 iPhone 11 pro Max  
6.5 吋 iPhone Xs Max

顯示附近站牌

會員登入  
會員註冊  
使用者資料編輯  
App 介紹

聯繫我們: a18w620412@gmail.com  
李永強: a238902@gmail.com  
陳耀宗: 891-89102@gmail.com

產品介紹  
開發團隊  
App 功能  
可用的型號

資料來源:  
1. 交通部hda平臺  
2. 全台各縣市開放資料平臺  
3. 天氣資訊: 中央氣象局開放資

Account

Enter your email

Enter your password

登入

註冊

確認密碼

確認密碼

首頁活動中心  
萬芳國小  
萬芳6號公園  
捷運萬芳社區站  
大興國中

萬芳社區 - 行天宮 台北

Q 公車路線搜尋

第二萬里市場 139 m  
樂華花園 148 m



03

START

使用iOS的部分和技术

### 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%B8%97%E5%B8%92&e
        lementName=")!
    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")
        Text("\(degree)°")
        .font(.system(size: 26))
        .bold()
        .padding(.top, 50)

        if let weather = weather {
            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)
            VStack{
                VStack {
                    Text("\(degree)°")
                    Text("\(parameterMinT.parameterName)/\(parameterMaxT
                        .parameterName)")
                    .font(.system(size: 15)).bold()
                    .fixedSize()
                }
                .frame(width: 100, height: 130)
                .padding(.leading, 20)

                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

```
func getBusRouteByCity(city: String) -> [BusRoute]{
    var urlComponent = URLComponents(string: "\(self.prefix)/v2/Bus/Route/City/\(city)")!
    urlComponent.queryItems = [
        URLQueryItem(name: "format", value: "JSON")
    ]
    let request = URLRequest(url: urlComponent.url!)
    let json([BusRoute]) = parseJson(request: request)
    return json
}

func getBusStopNearby(latitude: Double, longitude: Double, top: Int, completion: @escaping ([BusStopNearby]) -> Void){
    var urlComponent = URLComponents(string: "\(self.prefix)/v2/Bus/Stop/Nearby")!
    urlComponent.queryItems = [
        URLQueryItem(name: "stop", value: String(top)),
        URLQueryItem(name: "latitude", value: "\(String(latitude)), \(String(longitude)), 1000"),
        URLQueryItem(name: "format", value: "JSON")
    ]
    let request = URLRequest(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 = URLRequest(url: urlComponent.url!)
    let json([BusRouteStops]) = 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)")!
    let request = URLRequest(url: urlComponent.url!)
    let json([BusEstimatedTimeOfArrivalByRoute]) = parseJson(request: request)
    completion(json)
}
```



```
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: "捷運街", 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")
                }
                .sheet(isPresented: $detail) {
                    BusDetail(search: stop, startStop: stop.startStop, endStop: stop.endStop)
                }
            }
            Text(stop.city)
        }.padding()
    }
}

struct BusDetail: View {
    @State var search: String
    @State var detail = false

    var body: some View {
        HStack {
            VStack(alignment: .leading) {
                Text(search)
                Text(detail ? "詳細資訊" : "搜尋結果")
            }
            Spacer()
            VStack(spacing: 10) {
                Button {
                    detail = true
                } label: {
                    Image(systemName: "star")
                }
                .sheet(isPresented: $detail) {
                    BusDetail(search: search, startStop: search.startStop, endStop: search.endStop)
                }
            }
            Text(search.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.departureStopNameZh1, endStop: route.destinationStopNameZh1, 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

```
func getBusStopNearby(latitude: Double, longitude: Double, top: Int, completion: @escaping ([BusStopNearby]) -> Void) {
    var urlComponent = URLComponents(string: "\({self.prefix.advanced}/v2/Bus/Stop/Nearby")!
    urlComponent.queryItems = [
        URLQueryItem(name: "stop", value: String(top)),
        URLQueryItem(name: "spatialFilter", value: "nearby"),
        URLQueryItem(name: "format", value: "JSON")
    ]
    let request = URLRequest(url: urlComponent.url!)
    let json: [BusStopNearby] = parseJson(request: request)
    completion(json)
}

func getBusStopsByRoute(city: String, route: String, direction: Int) -> [BusRouteStop] {
    var urlComponent = URLComponents(string: "\({self.prefix}/v2/Bus/DisplayStopOfRoute/City/{city}/{route}")!
    urlComponent.queryItems = [
        URLQueryItem(name: "format", value: "JSON")
    ]
    let request = URLRequest(url: urlComponent.url!)
    let json: [BusRouteStop] = 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 = URLRequest(url: urlComponent.url!)
    let json: [BusEstimatedTimeOfArrivalByRoute] = parseJson(request: request)
    completion(json)
}
```



```
var body: some View {
    VStack{
        HStack{
            Text("\(busId) - 往 \(startStop)").font(.title)
        }.padding()
        List{
            ForEach (detailStops, id:\.id) { stop in
                busDetailRow(stop: stop)
            }
        }.listStyle(.plain)
    }.onAppear{
        if detailStops.count == 0 {
            let routeStops = tdxApi.getBusStopsByRoute(city: "Taipei", route: busId, direction: direction)
            for stop in routeStops.stops {
                detailStops.append(detailStop(busId: busId, stopName: stop?.stopName?.zhTw ?? "", i_time: -1, updated: false))
            }
        }
        self.timer = Timer.scheduledTimer(withTimeInterval: 5, repeats: true) { (_) in
            tdxApi.getBusEstimatedTimeOfArrivalByRoute(city: "Taipei", routeName: busId) { stops in
                for i in 0...detailStops.count-1 {
                    detailStops[i].updated = false
                }
                for stop in stops {
                    if stop.direction != direction {
                        continue
                    }
                    let stopName = stop.stopName?.zhTw
                    var estimateTime = -1
                    if let time = stop.estimateTime {
                        estimateTime = time / 60
                    }
                    let stopIndex = detailStops.firstIndex { $0.stopName == stopName }
                    if let index = stopIndex {
                        debugPrint(stopIndex, stopName as Any, estimateTime)
                        debugPrint(detailStops[index])
                        if detailStops[index].updated {

```

```
struct detailStop: Identifiable {
    var id = UUID()
    var busId: String
    var stopName: String
    var s_time: String {
        get {
            if i_time == -1 {
                return "未發車"
            }
            return i_time == 0 ? "進站中" : "\(i_time) 分"
        }
    }
    var i_time: Int
    var updated: Bool
}

extension detailStop {
    static let demo = detailStop(busId: "298", stopName: "板橋車站", i_time: 2, updated: false)
}

struct busDetailRow: View {
    @State var stop: detailStop
    var body: some View {
        HStack{
            VStack(alignment: .leading){
                Text(stop.stopName).font(.system(size: 18))
            }
            Spacer()
            VStack(spacing: 10){
                Text(stop.s_time)
                    .font(.system(size: 18))
                    .frame(width: 60, height: 10)
                    .padding()
                    .overlay(Capsule(style: .continuous)
                        .stroke(Color.blue, lineWidth: 2)
                    )
            }.padding()
        }
    }
}
```





## 3

- 

```
func getBusStopNearby(latitude Double, longitude Double, top Int, completion: @escaping ((BusStopNearby)) -> Void) {
    var urlComponent = URLComponents(string: "http://preproddevapi.v2/bus/stop/nearby")!
    urlComponent.queryItems = [
        URLQueryItem(name: "stop", value: String(top)),
        URLQueryItem(name: "SpatialFilter", value: "nearby:\\(String(latitude)), \\(String(longitude)), 1000"),
        URLQueryItem(name: "Format", value: "JSON")
    ]

    let request = URLRequest(url: urlComponent.url!)
    let json: BusStopNearby? = parseJson(request: request)
    completion(json)
}

func getBusStopOnRoute(city: String, route: String, direction: Int) -> BusRouteSteps {
    var urlComponent = URLComponents(string: "http://preproddevapi.v2/bus/display/stopOnRoute/City/(city)/(route)")!
    urlComponent.queryItems = [
        URLQueryItem(name: "Format", value: "JSON")
    ]

    let request = URLRequest(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 getEstimatedTimeOfArrivalByRoute(city: String, routeName: String, completion: @escaping ((BusEstimatedTimeOfArrivalByRoute)) -> Void) {
    var urlComponent = URLComponents(string: "http://preproddevapi.v2/bus/estimatedimeofarrival/City/(city)/(routeName)")!
    urlComponent.queryItems = [
        URLQueryItem(name: "Format", value: "JSON")
    ]

    let request = URLRequest(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: SviewModel.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: 88, height: 28)
                                    .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: 480, height: 750)
    }
}

```



## [3] 使用iOS的部分和技术

- 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 }

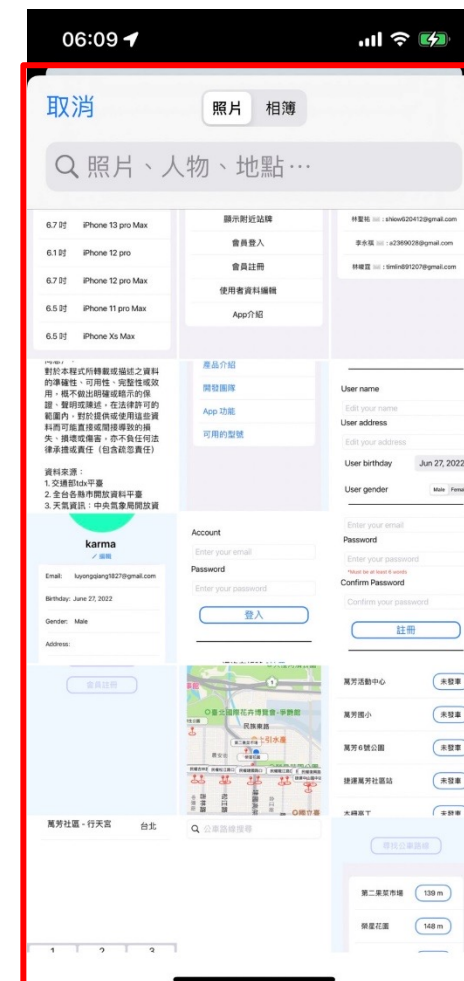
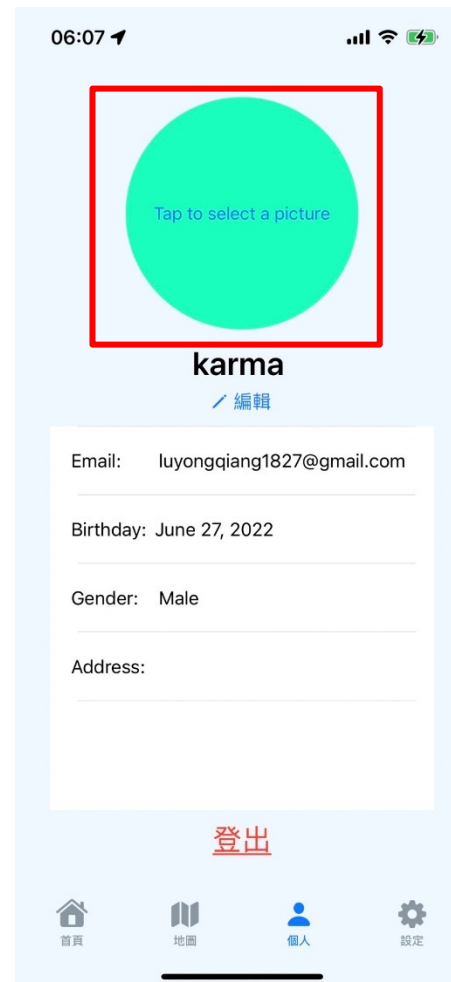
            if provider.canLoadObject(ofClass: UIImage.self) {
                provider.loadObject(ofClass: UIImage.self) { image, _ in
                    self.parent.image = image as? UIImage
                }
            }
        }
    }

    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

iOSFinalProject ▾ 瀏覽說明文件 🔔 👤 ?

### Authentication

[Users](#) [Sign-in method](#) [Templates](#) [Usage](#)

[新增使用者](#)

ID	識別資訊提供者	建立日期 ↓	登入日期	使用者 UID	
luyongqiang1827@gmail.c...	✉	2022年6月2...	2022年6月2...	0WFHZFvUs3aMFqNqL96QXD0H3...	
hihi123@gmail.com	✉	2022年6月2...	2022年6月2...	K0QmL2yBJXcpTqiERwHSBokfQ2...	
789@gmail.com	✉	2022年6月2...	2022年6月2...	lv0wlgdrfxdiOnNAst37PakFIIR2	
timlin891207@gmail.com	✉	2022年6月1...	2022年6月1...	lvEcjnlQdfhPsshq2JBoKwFblES2	
456@gmail.com	✉	2022年6月1...	2022年6月2...	rSDBwH1N8lfcKCjLC62NvlMkldg1	
123@gmail.com	✉	2022年6月1...	2022年6月2...	Q0mqs1dgqne4qtdj7MEQutsGfbg2	

每頁列數: 50 ▾ 1 - 6 of 6 < >



04

START



App Demo時間



iOS Final Project

**Thanks For Your Attention**



END