

# 모바일앱프로그래밍2(COMP0420001)

## 1팀 기술문서

지금 갈까?

떠나자 지금!

컴퓨터학부	2019117921	우경훈
컴퓨터학부	2016115827	윤정균
컴퓨터학부	2017112527	김범준
컴퓨터학부	2016110861	권동영



**경북대학교**  
KYUNGPOOK NATIONAL UNIVERSITY

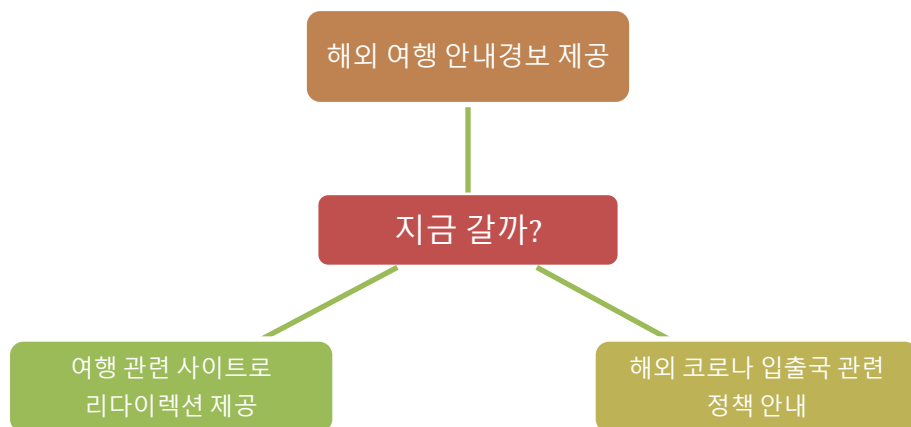
## 프로젝트 개요

전세계 코로나 입국 정책 현황을 한눈에 알아보자!

코로나19의 도래 이후 2년이 지난 현재, 우리나라는 '코로나 엔데믹' 관련 기사가 언론에 지속적으로 등장할 정도로 조금씩 일상으로의 복귀를 준비하고 있습니다. 이에 발맞추어 여행 업계 또한 항공편 예약이 급증하는 등 긍정적인 움직임들이 나타나고 있습니다. 하지만 아직 코로나의 위협이 끝나지 않은 국가들도 많습니다. 일부 국가는 아직 외국인 입국 시 격리정책을 유지하거나 심지어 외국인의 관광목적 입국을 금지하는 경우도 여전히 존재합니다.

저희 팀은 해외여행 계획 이전에 이러한 정보를 쉽게 파악할 수 있는 어플리케이션을 개발하고자 합니다. 외교부에서 제공하는 공공데이터를 활용해서 각 국가의 코로나 정책 및 상황, 대응단계 등을 안내하고, 이를 바탕으로 여행계획을 수립할 수 있도록 관광 및 여행 관련 웹 사이트로의 연결을 지원하고자 합니다.

## 프로젝트 목표



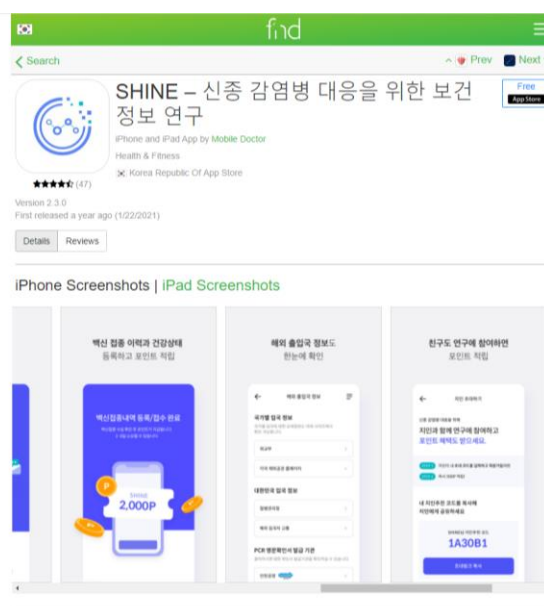
엔데믹에 맞추어, 해외여행을 떠나려는 사용자에게 해외 각국의 코로나 관련 출입국 정책 등을 편리하게 찾아볼 수 있도록 서비스를 제공하는 것이 주요 목표입니다.

코로나 사태 완전 종식 이후에도 각국의 정세에 대한 정보를 지속적으로 제공하여 해외 여행을 계획하는 사람들에게 도움을 제공할 수 있도록 개발하고자 합니다.


## 시장조사



현재 코로나 관련 어플리케이션은 코로나19 현황이나 전자예방접종증명서, 코로나 비대면 진료 등과 같은 서비스를 제공하는 어플리케이션 위주로 제공되고 있습니다.



그 중에서, SHINE이라는 코로나 연구용 어플리케이션에서 해외 출입국 정보를 제공하지만 실제로는 외교부 사이트로 연결되는 링크를 제공하는 수준이었습니다.



**외교부  
해외안전여행**

▶

검색어 + -

---

해외안전정보
여행정보
연결여행서비스
영사서비스/비지
미디어센터
영사초록 발급정보
☰

## 해외안전정보

- > 공지사항
- > 최신안전소식
 

안전공지
- 여행정보 조정
- 한시 상제 소식
- > 국가/지역별 정보
- > 위급상황별 대처방법

국

### 안전공지

출국 전 우리 국민이 꼭 참고해야 할 최신 안전정보를 제공합니다.

국가	전세계		
제목	<b>[안내] 코로나19 확산 관련 각국의 해외입국자에 대한 입국제한 조치 실시 국가(지역) (4.8. 17:00)</b>		
등록일	2022-04-08	조회	547356
첨부파일	<a href="#">220408_코로나19 확산 관련 각국의 해외입국자에 대한 조치 현황_1700.pdf</a> <a href="#">사용면허발령 장장 공지 국가 목록.pdf</a>		

★ 발령의 초기 현황은 국민들이 편을 위하여 정보 제공 차원에서 참고로 작성된 것임을 알려드립니다. 보다 상세한 정보 확정을 위해서는 방문목적에 따라 국가/지역 구분의 공식 홈페이지 또는 해당 국가 관할 우리 대사관 홈페이지 등을 필수적으로 참고하시기 바랍니다. ★

○ 코로나19가 전 세계로 확산됨에 따라 대부분의 국가들이 외국인에 대상으로 입국제한 조치를 실시하고 있습니다. 또한 다수 항공 노선이 두절됨에 따라 해외에서 격리되거나 고립되는 등 큰 불편을 겪을 가능성이 있습니다. 코로나19와 관련하여 해외입국자에게 대한 입국제한 조치를 실시하고 있는 국가(지역)를 붙임과 같이 안내합니다.

※ 입국이 허용되는 조건을 갖추더라도 **각국의 입국(국경)을 관리하는 실무자의 판단에 따라 입국이 거절**되는 경우가 생길 수 있으며, 현재 전 세계적으로 특별여행주의보가 내려진 참을 감안하여 비필수적인 여행은 가능한 한 연기하기 바랍니다.

※ 붙임 : 코로나 19 확산 관련 각국의 해외입국자에 대한 조치 현황 (**4월 8일 17:00 현재**)

○ 코로나19 대응을 위한 우리 정부의 사용면허발령 장장 정치 4개 국가(뉴질랜드, 페루, 그레나다, 파나마)를 단계별로 목적용 방문을 희망하는 우리 국민은 활용 해당국가에 미국금지를 해제하여 입국이 가능하도록 하고 협정이 체결될 때까지는 **출국 전 해당국 사용(Visa)를 취득**하여야 하니 유의하시기 바랍니다.

※ 4.1일부터 우리 정부의 사용면허발령 장장 정치 국가 4개국(뉴질랜드, 페루, 그레나다, 파나마)으로 축소 됨.

※ 우리 정부의 사용면허발령 장장 정치 조처에도 불구하고 일부 국가는 우리 국민에 대한 단기방문 무사통 입국을 허용하고 있음에, 코로나19의 전 세계적 확산으로 **각국의 입국조치가 자주 변동**되고 있으나, **출국 전 입국 예정 국가 주한공관 또는 해당지역 주재 취체처**에 미리 문의하시기 바랍니다.

해외 코로나 입국 관련 정보를 찾기 위해선, 외교부에서 제공하는 해외안전여행 사이트를 통해 확인할 수 있습니다. 하지만 관련 정보를 확인하기 위한 별도의 기능은 존재하지 않았고, 공지사항에 게시된 글에 포함된 첨부파일을 통해 확인할 수 있었습니다.

저희 팀은 다가오는 엔데믹 환경에 대응하고, 기존의 불편한 방식으로 접근해서 확인할 수 있는 정보들을 손쉽게 확인하고, 이를 바탕으로 다양한 방향으로 행동할 수 있도록 해외여행에 필요한 코로나19 관련 정보 제공 어플리케이션을 제작하고자 합니다.

## 서비스 구현

## 주요기능

## ● 여행 경보 정보 제공

시각적으로 각 국가의 여행경보 정보를 외교부 제공 Data를 통해 출력

## ● 코로나 입국 정보 제공

외교부 제공 Data를 통해 코로나 입국 정보를 제공

## ● 예약을 위한 리다이렉트

입국 정보 확인 후 항공권 예매 등을 위해 웹사이트 리다이렉트 기능 제공

## ● 플래너 기능

여행정보 확인 후, 여행 계획을 간략하게 구성

## 화면 디자인

### ● 로딩 화면

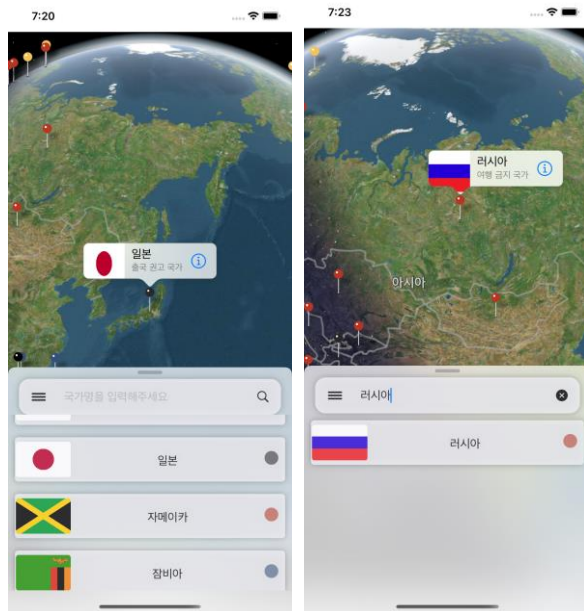


### ● 초기 화면



앱 시작 시 각 나라의 국가의 여행 경보를 Pin색깔 별로 표기

## ● 검색 화면



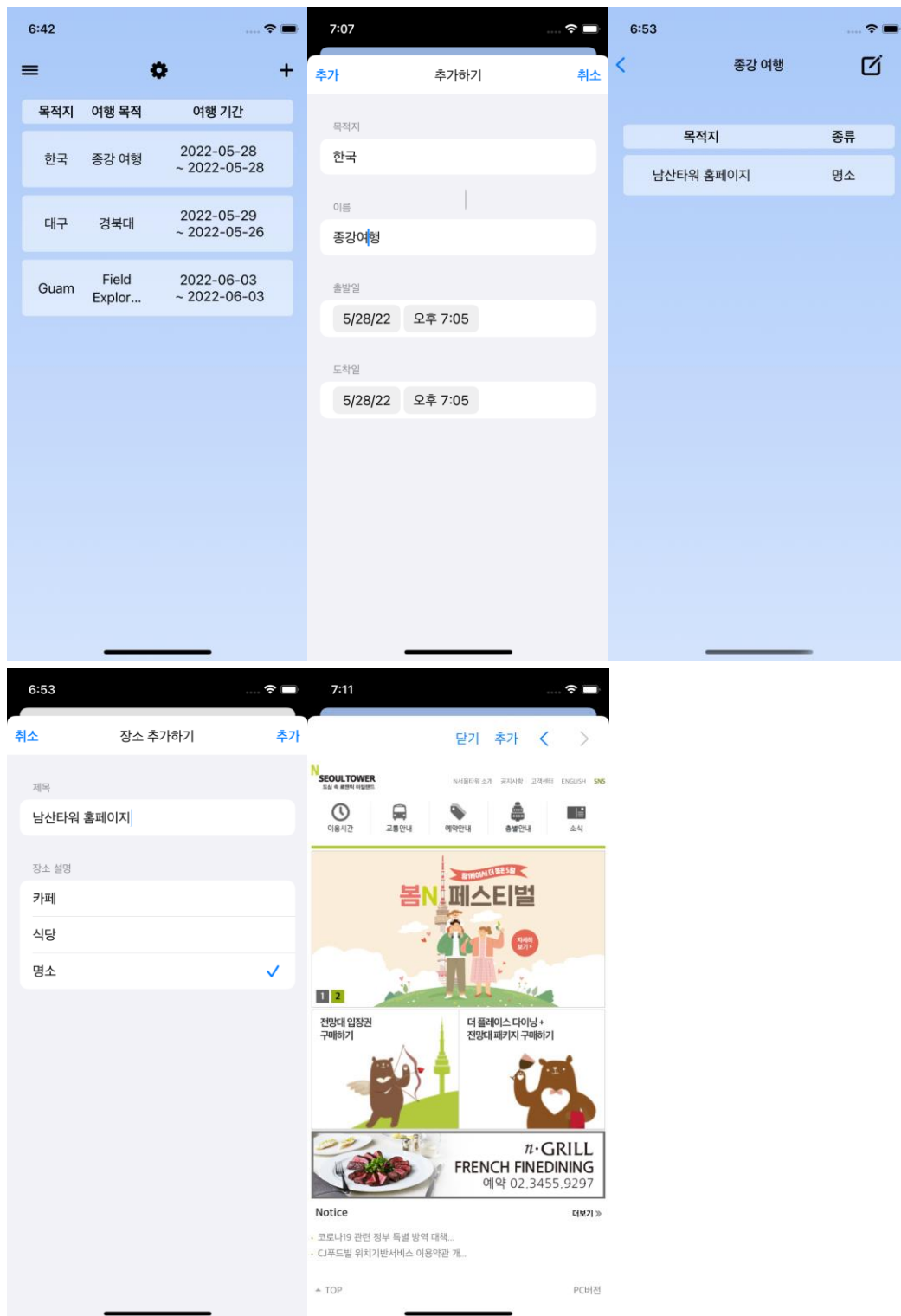
검색창을 터치하면 국가리스트 Sheet의 위치가 Middle 상태로 변경됨  
 검색창에서 국가를 클릭하면 Map의 카메라 위치가 해당 국가로 이동하며 핀이  
 자동으로 선택  
 핀 선택 시 해당국가의 국기와 간략한 여행 경보를 표기

## ● 국가별 코로나 입국 정보 화면



각 국가별 여행 경보와 입국정보를 보기 편한 형태로 제공  
 필요한 경우, 하단의 예약 버튼을 클릭하여 예약 관련 사이트로 리다이렉션 제공

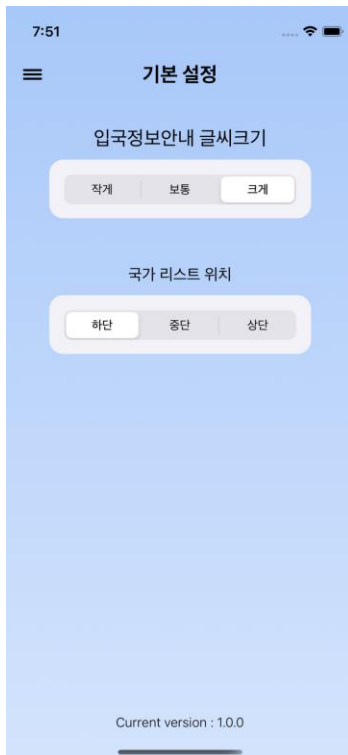
## ● 플래너 화면



여행 계획 저장 가능 및 온라인 검색을 통해 여행 관련 정보가 담긴 웹 주소를 세부적으로 저장 가능

## ● 기타

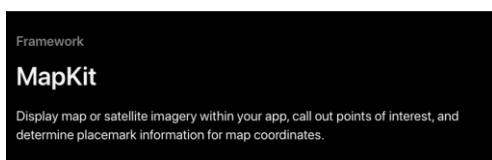
### Setting 화면



설명화면에서 글씨크기와 초기 화면의 국가리스트Sheet의 위치를 설정

## 코드 설명

### ● MapView



설명화면에서 글씨크기와 초기 화면의 국가리스트Sheet의 위치를 설정

Map View를 구현하기 위해 apple에서 기본으로 제공하는 Framework인 MapKit를 사용

Map에 각 국가에 대한 Pin을 생성하기 위한 위치 정보를 받아오기 위해 Mapbox API 활용



```

5 struct MapView: UIViewRepresentable {
6     @EnvironmentObject var mapModelView: MapModelView
7
8     func makeUIView(context: Context) -> MKMapView {
9         let mapView = mapModelView.model.mapView
10        let coords = CLLocationCoordinate2D(latitude: 53.062640, longitude: -2.968900)
11        let span = MKCoordinateSpan(latitudeDelta: 180, longitudeDelta: 180)
12        let region = MKCoordinateRegion(center: coords, span: span)
13
14        mapView.mapType = .hybridFlyover
15        mapView.setRegion(region, animated: true)
16        return mapView
17    }
18
19    func makeCoordinator() -> MapViewCoordinator {
20        MapViewCoordinator(self, mapModelView)
21    }
22
23    func updateUIView(_ view: MKMapView, context: Context) {
24        view.delegate = context.coordinator
25    }
26

```

UIViewRepresentable Protocol를 적용시켜 MapView를 호출하면 MapView가 생성  
makeUIView()에는 초기 MapView에 대한 카메라 위치와 Zoom Level, 지도의  
Type을 설정  
updateUIView()에는 Map의 UI가 변경될 때마다 실행되는 코드로 해당 코드에서는  
delegate으로 처리

```

27 public class MapViewCoordinator: NSObject, MKMapViewDelegate {
28     var parentMapView: MapView
29     var mapModelView: MapModelView
30     var selectedCountry: GoRightNowModel.Country?
31
32     init(_ control: MapView, _ modelView: MapModelView) {
33         self.parentMapView = control
34         self.mapModelView = modelView
35     }
36
37     //Pin이 선택되었을 때 선택된 country를 model에 넘김
38     public func mapView(_ mapView: MKMapView, didSelect view: MKAnnotationView) {
39         let model = view.annotation as? CustomAnnotationModel
40         mapModelView.model.selectedCountry = model?.countryInfo
41     }
42
43     //Pin의 버튼이 선택되었을 때 Sheet을 Show하도록 값을 변경
44     public func mapView(_ mapView: MKMapView, annotationView view: MKAnnotationView, calloutAccessoryControlTapped: UIControl?) {
45         mapModelView.model.isDetailSheet = true
46     }
47
48     //MKAnnotation Pin 설정
49     public func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView? {
50         let annotationView = MKPinAnnotationView(annotation: annotation, reuseIdentifier: "")
51         let model = annotation as? CustomAnnotationModel
52         annotationView.isEnabled = true
53         annotationView.canShowCallout = true
54
55         //Pin 양 버튼 이미지 설정
56         let btn = UIButton(type: .detailDisclosure)
57         let image = UIButton(type: .custom)
58         if UIImage(named: (model?.countryInfo!.iso_alp2!)) != nil {
59             image.setImage(UIImage(named: (model?.countryInfo!.iso_alp2!)), for: UIControl.State.disabled)
60         } else {
61             let data = urlToImage(strUrl: model!.countryInfo!.flagImageUrl)
62             image.setImage(data, for: UIControl.State.normal)
63         }
64
65         image.frame = CGRect(x: 0, y: 0, width: 60, height: 60)
66         image.isEnabled = false
67
68         //Pin 색 변경하기
69         annotationView.pinTintColor = getColorInfoFromAlarm(model?.countryInfo?.alarmLevel)
70
71         annotationView.rightCalloutAccessoryView = btn
72         annotationView.leftCalloutAccessoryView = image
73         return annotationView
74     }
75
76

```

UIViewRepresentable Protocol를 적용시켜 MapView를 호출하면 MapView가 생성  
makeUIView()에는 초기 MapView에 대한 카메라 위치와 Zoom Level, 지도의  
Type을 설정

updateUIView()에는 Map의 UI가 변경될 때마다 실행되는 코드로 해당 코드에서는  
delegate으로 처리함해당 코드에서 Pin을 선택했을 때 국가 정보를 넘겨줄 수  
있도록 기본 Pin 대신 CustomAnnotationModel을 생성

```

8      class CustomAnnotationModel: MKPointAnnotation {
9          var countryInfo: GoRightNowModel.Country?
10     }

```

CustomAnnotationModel은 MKPointAnnotation을 상속받고 countryInfo Property를 추가

Pin 선택 시 선택된 Pin의 국가 정보를 MapModel에 넘겨줌과 동시에 국가에 대한 상세 정보(sheet)가 보일 수 있도록 isDetailSheet Property를 true로 변경  
mainApp.swift파일에서 각 국가에 대한 정보와 위치 정보, Pin까지 설정했을 때 isGotData(true) 클로저를 실행함으로 로딩화면에서 MapView로 전환

```

58 func getLocationInfoAndMakeLocationPin(modelView: GoRightNowModelView, mapModelView: MapModelView, completion: @escaping ()
59 isAllSearchingDone(countries: modelView.model.countries) { isSuccess, countries in
60     if (isSuccess) {
61         modelView.model.countries = countries
62         mapModelView.makePin(countries: countries)
63         completion()
64     }
65 }
66
67
68
69 func getSearchResponse(text: String) async -> CLPlacemark? {
70     let geocoder = CLGeocoder()
71     let gecInfo: CLPlacemark? = try? await geocoder.geocodeAddressString(text).first
72     return gecInfo
73 }
74

```

mainApp.swift에서 getLocationInfoAndMakeLoactionPin()을 호출 시 내부에서는 파라미터에 국가 Array를 받아 모든 국가 위치 정보를 검색하는 isAllSearchingDone()을 실행

```

9     let geocoder = Geocoder.shared
10
11 func isAllSearchingDone(countries: [GoRightNowModel.Country], completion: @escaping (Bool, [GoRightNowModel.Country]) -> Vo
12     print("Start All Searching")
13     var newCountries: [GoRightNowModel.Country] = []
14     var searchedCount = 0;
15     var maxCount = countries.count
16     for country in countries {
17         isSearchingDone(country: country) { isResult, country in
18             if (isResult) {
19                 print("\(country.name) Searching Done")
20                 searchedCount += 1
21                 newCountries.append(country)
22                 if(maxCount == searchedCount){
23                     print("All Searching Done")
24                     completion(true, newCountries)
25                 }
26             } else {
27                 searchedCount += 1
28                 print("[All]Cannot Found \(country.name)")
29             }
30         }
31     }
32 }
33

```

isAllSearchingDone()이 수행된 후 클로저를 통해 model에 있는 국가Array를 위치정보가 담겨있는 새로운 국가 Array로 변경하며 모든 국가에 대한 pin을 생성  
isAllSearchingDone()함수를 실행시키면 for문을 통해 각각의 국가를 국가정보를 파라미터로 받는 isSearchingDone() 함수를 통해 국가위치를 전달받음

```

54 func isSearchingDone(country: GoRightNowModel.Country, completion: @escaping (Bool, GoRightNowModel.Country) -> Void) {
55     let options = ForwardGeocodeOptions(query: country.iso_alp2)
56     var newCountry: GoRightNowModel.Country = country
57     options.allowedISOCountryCodes = [country.iso_alp2]
58     options.allowedScopes = [.country]
59     geocoder.geocode(options) { (placemarks, attribution, error) in
60         guard let coordinate = placemarks?.first?.location?.coordinate else {
61             print("[MapBox] \(country.name) No Found")
62             searchingToLocalSearch(country) { b, location in
63                 if (b) {
64                     newCountry.location = location
65                     completion(true, newCountry)
66                 } else {
67                     completion(false, newCountry)
68                 }
69             }
70             return
71         }
72         newCountry.location = Location(title: country.name, latitude: coordinate.latitude, longitude: coordinate.longitude)
73         completion(true, newCountry)
74     }
75 }
76

```

isSearchingDone()이 수행된 후 국가에 대한 위치 정보가 존재하면 새로운 국가배열에 추가

파라미터를 통해 입력받은 나라의 개수와 검색이 완료된 나라의 개수를 비교하여 모든 나라가 검색되었을 때 클로저를 수행

isSearchingDone()함수를 수행시키면 기본적으로 MapBox API로 국가를 검색

검색query는 국가 코드를 이용하며 검색에 대한 Scope은 country로 제한

검색 성공 시, 위치정보에 대한 값을 추가하여 클로저를 통해 새로운 국가 정보 전달

검색 실패 시, searchingToLoadSearch함수를 통해 2차 검색을 실시

2차 검색 성공 시 위치정보를 추가하여 클로저를 통해 새로운 국가 정보를 전달하며 실패 시 클로저에 false값을 추가하여 정보를 전달

```

76 func searchingToLocalSearch(_ country: GoRightNowModel.Country, completion: @escaping (Bool, Location?) -> Void) {
77     let searchRequest = MKLocalSearch.Request()
78     searchRequest.naturalLanguageQuery = countryNameHandler(country.name)
79     let search = MKLocalSearch(request: searchRequest)
80     search.start { response, error in
81         guard let response=response?.mapItems.first?.placemark.coordinate else {
82             print("[MK Search] \(country.name) No Found")
83             completion(false, nil)
84             return
85         }
86         let location = Location(title: country.name, latitude: response.latitude, longitude: response.longitude)
87         completion(true, location)
88     }
89 }

```

searchingToLocalSearch()는 Apple의 지도 검색 API를 사용해 국가 위치를 얻어옴

검색query는 국가 이름을 사용하며 클로저를 통해 검색 성공시 검색 결과중 첫번째 위치 정보를 넘기며, 실패시 nil을 넘김

```

10 struct CountryList: View {
11     @EnvironmentObject var mapModelView: MapModelView
12     @EnvironmentObject var bottomSheetModelView: BottomSheetModelView;
13     var countries: [GoRightNowModel.Country]
14     //Tap한 값을 Modal에 넘겨주기 위한 Property
15     @State private var showModal: Bool = false
16
17
18     var body: some View {
19         ForEach(countries, id: \.self) { data in
20             ListTile(name: data.name, immg: data.immigInfo, imgUrl: data.flagImageUrl, isocode: data.iso_alp2,
21                     alarm: data.alarmLevel)
22                 .onTapGesture {
23                     mapModelView.goToRegion(data)
24                     mapModelView.selectPinToCountry(country: data)
25                     bottomSheetModelView.position = .bottom
26                 }
27             .padding(EdgeInsets(top: 10, leading: 10, bottom: 0, trailing: 10))
28         }
29     }
30 }
31

```

국가List에서 국가를 Tap하게 되면 goToRegion()를 통해 해당 국가로 카메라를 이동시키며 selectPinToCountry()를 통해 해당 국가의 Pin을 자동으로 선택  
Pin을 선택했을 때 rightCalloutAccessory를 선택하게 되면 Model의 isDetailSheet 가 true로 변경되며 MainApp.swift의 sheet가 보임

```

17     var body: some View {
18         ZStack {
19             if (modelView.isNotGotData()) {
20                 LoadingView()
21             } else {
22                 ZStack {
23                     MapView()
24                     .bottomSheet(bottomSheetPosition: $bottomSheetModelView.model.bottomSheetPosition,
25                               options: [.appleScrollBehavior],
26                               headerContent: {
27                                   BottomSheetHeader(bottomSheetView: bottomSheetModelView, modelView: modelView)
28                               }) {
29                         CountryList(countries: modelView.model.countryList)
30                     }
31                     .sheet(isPresented: $mapModelView.model.isDetailSheet) {
32                         if let countryInfo = mapModelView.model.selectedCountry {
33                             CountryDetailView(
34                                 countryName: countryInfo.name,
35                                 immigInfo: countryInfo.immigInfo,
36                                 isoCode: countryInfo.iso_alp2,
37                                 imgUrl: countryInfo.flagImageUrl,
38                                 alarmLevel: countryInfo.alarmLevel
39                             )
40                         } else {
41                             Text("[Error] No Data")
42                         }
43                     }
44                 }
45             }
46         }
47     }
48 }
49

```

mapModel에 selectedCountry내용이 존재하면 해당 정보를 이용하여 Sheet에 CountryDetailView를 구성

## ● API 호출



API 호출을 위해 Alamofire 라이브러리를 활용

```
8 import Foundation
9 import Alamofire
10
11 struct ArrivalAPIResponse: Codable {
12     let data: [searchedCountry]
13 }
14
15 struct AlarmAPIResponse: Codable {
16     let data: [searchedAlarm]
17 }
18
19 struct searchedCountry: Codable {
20     let country_nm : String
21     let txt_origin_cn : String
22     let country_iso_alp2: String
23 }
24
25 struct searchedAlarm: Codable {
26     let country_nm : String
27     let country_iso_alp2: String
28     let flag_download_url : String
29     let alarm_lvl: Int?
30 }
```

JSON 형태로 Data를 받아와서 사용하기 위해 Codable로 선언함  
이때, API를 두 번 호출하므로 각 API 별로 구조체를 별도로 선언

```

33 func getCountryInfo(countries: inout [GoRightNowModel.Country]) {
34     // 새로운 값을 받아오기 위한 곳.
35     var getCountries: [GoRightNowModel.Country] = []
36
37     // 동기화 처리를 위한 세마포어 // 값을 순차적으로 받아옴
38     let semaphore = DispatchSemaphore(value: 0)
39     let queue = DispatchQueue.global(qos: .utility)
40
41     // 값을 받아올 URL.
42     let countryOverseasArrivalsServiceURL =
43         "https://apis.data.go
44         .kr/1262000/CountryOverseasArrivalsService/getCountryOverseasArrivalsList?serviceKey=%
45         &returnType=JSON&numOfRows=2000"
46
47     let TravelAlarmService2URL =
48         "https://apis.data.go
49         .kr/1262000/TravelAlarmService2/getTravelAlarmList2?serviceKey=%
50         ON&pageNo=1&numOfRows=2000"

```

func getCountryInfo()를 통해 API를 호출

순차적으로 Data를 받아오기 위해 DispatchSemaphore()를 활용

Alamofire를 사용하여 API를 호출하기 위해 DispatchQueue()를 활용

Parameter의 Key에 특수문자가 들어가는 경우가 존재하여 별도의 Parameter 변수를 선언할 수 없는 관계로 URL 그대로 활용

(API 호출시 필요한 serviceKey부분은 노출되면 안 되므로 모자이크 처리)

```

50 AF.request(TravelAlarmService2URL, method: .get, encoding: JSONEncoding.default).responseDecodable(of:
51     AlarmAPIResponse.self, queue: queue) { json in
52
53     switch json.result {
54     case .success:
55         if let jsonData = json.value {
56             for getData in jsonData.data {
57                 getCountries.append(GoRightNowModel.Country(name: getData.country_nm, iso_alp2:
58                     getData.country_iso_alp2, immigInfo: "", flagImageURL: getData.flag_download_url,
59                     alarmLevel: getData.alarm_lv1 ?? 0))
60             }
61         } else {
62             print("TravelAlarmService2 Error");
63         }
64     case .failure(let err):
65         print(err.localizedDescription)
66     }
67     semaphore.signal()
68 }
69 semaphore.wait()

```

AF.request를 통해 Alamofire로 여행경보 API를 호출

GET method로, JSON format으로 Encoding

switch문을 통해 성공, 실패 여부에 따라 동작을 다르게 실시

성공 시, Array 형태로 된 데이터들을 가져와서 어플리케이션에 저장

Semaphore를 활용하여 API에 대한 통신이 종료될 때까지 다음으로 진행하지 않음

```

69 AF.request(countryOverseasArrivalsServiceURL, method: .get, encoding:
    JSONEncoding.default).responseDecodable(of: ArrivalAPIResponse.self, queue: queue) { json in
70     switch json.result {
71     case .success:
72         do {
73             if let jsonData = json.value {
74                 for getData in jsonData.data {
75                     var flag = false // 나라가 있었는지 확인하기 - 현재 단계에서는 데이터 테스트용으로 사용
76                     for countriesIndice in getCountries.indices {
77                         if getCountries[countriesIndice].name == getData.country_nm {
78                             getCountries[countriesIndice].immigInfo = getData.txt_origin_cn
79                             flag = true
80                             break
81                         }
82                     }
83                     if flag == false { // 현재 단계에서는 데이터 테스트 용으로 사용
84                         getCountries.append(GoRightNowModel.Country(name: getData.country_nm, iso_alp2:
                            getData.country_iso_alp2, immigInfo: getData.txt_origin_cn, flagImageUrl: "",
                            alarmLevel: -1))
85                     }
86                 }
87             } else {
88                 print("countryOverseasArrival Error");
89             }
90         }
91     case .failure(let err):
92         print(err.localizedDescription)
93     }
94     semaphore.signal()
95 }
96
97 // 데이터를 다 받아올때까지 대기함.
98 semaphore.wait()
99 // 받아온 국가들의 정보를 모델에 적용.
100 countries = getCountries
101
102 print("Got data")
103 }

```

AF.request를 통해 Alamofire로 코로나 입국 정보 API를 호출

이전에 호출한 방식에서, URL을 제외하고 동일하게 호출

이전 API 호출을 통해 저장했던 Data와 이번에 API를 호출해서 받아온 Data를  
비교해서 어플리케이션에 필요한 정보들을 저장

## ● 플래너 화면

```

fileprivate enum Mode {
    case list // 일반 모드
    case edit // 수정 모드
    case delete // 삭제 모드
}

```

Enum의 적당한 활용으로 화면의 상태를 조절

```

.sheet(isPresented: $addSheet) {
    AddSheet(dismiss: $addSheet, alert: $addAlert)
}

```

추가내용은 sheet를 이용해 구현



```
Form {
  Section(header: Text("목적지")) {
    TextField("\($countryName)", text: $countryName)
  }
  Section(header: Text("이름")) {
    TextField("\($planName)", text: $planName)
  }
  Section(header: Text("출발일")) {
    DatePicker(selection: $departDate, label: {})
      .labelsHidden()
  }
  Section(header: Text("도착일")) {
    DatePicker(selection: $returnDate, label: {})
      .labelsHidden()
  }
}
```

이외 값을 추가하는 부분은 Form과 Section을 적당히 이용하였으며  
날짜는 DatePicker를 이용하여 받아오도록 구현

## ● 플래너 저장

```
struct PlannerModel: Codable {

    enum Place: Codable {
        case Landmark
        case Food
        case Cafe
        // etc
    }

    struct Plan : Identifiable, Codable {
        var countryName: String // 나라 이름
        var planName: String // 계획 이름
        var departDate: Date // 출발 일자
        var returnDate: Date // 도착 일자

        var id: String {
            countryName + planName + departDate.description + returnDate.description
        }

        var places: [Landmarks] // 가볼곳들
    }

    struct Landmarks: Hashable, Equatable, Codable {
        var url: URL // 인터넷 주소
        var title: String // 표시할 이름
        var place: Place // 장소에 대한 간단한 설명
    }

    // Document에 데이터 저장하기 위한 메서드 & initializer
    func json() throws -> Data {
        try JSONEncoder().encode(self)
    }

    init(json: Data) throws {
        self = try JSONDecoder().decode(PlannerModel.self, from: json)
    }

    init(url: URL) throws {
        let data = try Data(contentsOf: url)
        self = try PlannerModel(json: data)
    }

    init() { }
}
```

플래너 모델 전체를 Codable 프로토콜에 Conform하게 만들어 JSONEncoder로 다룰 수 있도록 만들어주었고 데이터 인코딩, 디코딩에 필요한 function과 initializer를 정의

```

@Published var model: PlannerModel {
    didSet {
        save() //여기서 데이터 저장
    }
}

// 사용자 Document 경로
private struct PlannerChanged {
    static let filename = "Saved.planner"
    static var url: URL? {
        let docDirectory = FileManager.default.urls(for: .documentDirectory, in: .userDomainMask).first
        return docDirectory?.appendingPathComponent(filename)
    }
}

// 플래너 데이터를 저장하기 위한 메서드
private func save() {
    let thisFunction = "\(String(describing: self)).\(#function)"
    if let url = PlannerChanged.url {
        do {
            let data: Data = try model.json()
            print("\(thisFunction) json = \(String(data: data, encoding: .utf8) ?? "nil")")
            try data.write(to: url)
            print("\(thisFunction) success!")
        } catch let encodingError where encodingError is EncodingError {
            print("\(thisFunction) couldn't encode Planner as JSON
                    because\(encodingError.localizedDescription)")
        } catch {
            print("\(thisFunction) error = \(error)")
        }
    }
}

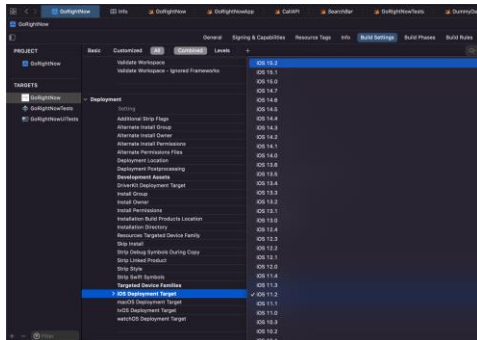
// 플래너 modelView 초기 생성 시 사용자 Document의 데이터로 생성
init() {
    if let url = PlannerChanged.url, let savedModel = try? PlannerModel(url: url) {
        model = savedModel
    } else {
        model = PlannerModel()
    }
}

```

플래너 ViewModel에서는 Published된 Model이 변경되었을 경우 호출하는 didSet  
 옵저버에서 Model을 저장  
 앱을 실행시킬 때 호출되는 ModeView initializer에서 저장된 Model이 존재하는  
 경우 이를 Decode한 뒤 모델로 사용할 수 있도록 구현

# 이슈사항

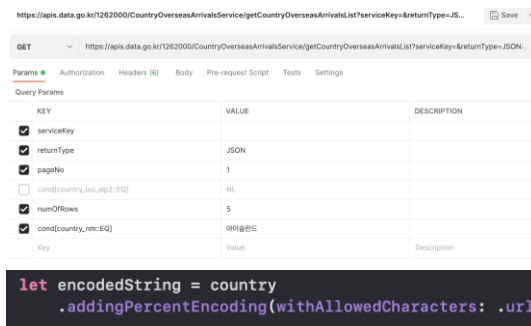
## ● 프로젝트 연동 문제



팀원 일부 중 Github를 통해 공유한 Project를 열면 Simulator가 동작하지 않는 문제 발생

⇒ 프로젝트 설정에서 iOS 버전 변경을 통해 해결

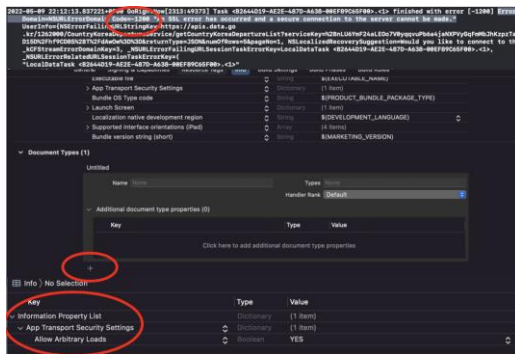
## ● API 호출 관련 이슈



iOS에서 URL에 한글 포함이 불가능함

⇒ 한글 String을 UTF-8 Code로 변환하여 사용

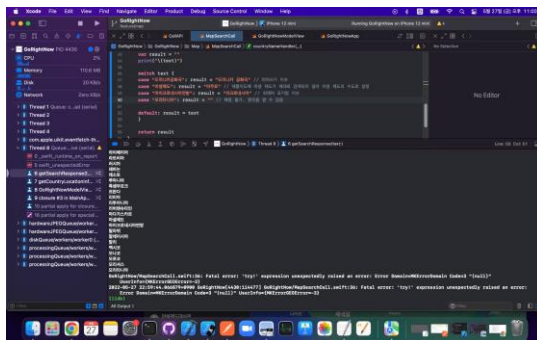
## ● Alamofire 통신 에러(http 호출) 문제



API 호출 URL이 SSL 보안이 안되어 있어(http호출) 오류 발생

⇒ http 호출을 허용하도록 Info.plist 생성 후 allow arbitrary load를 true로 변경

## ● MKLocal Search API 호출 관련 이슈



Apple의 기본 지도 검색 API인 MKLocal Search API를 통해 국가의 위치 검색 시

일정 검색 수가 넘어가면 Throttle/Limit Prevention 발생

⇒ Apple API는 일정 시간안에 정해진 횟수만 호출가능함

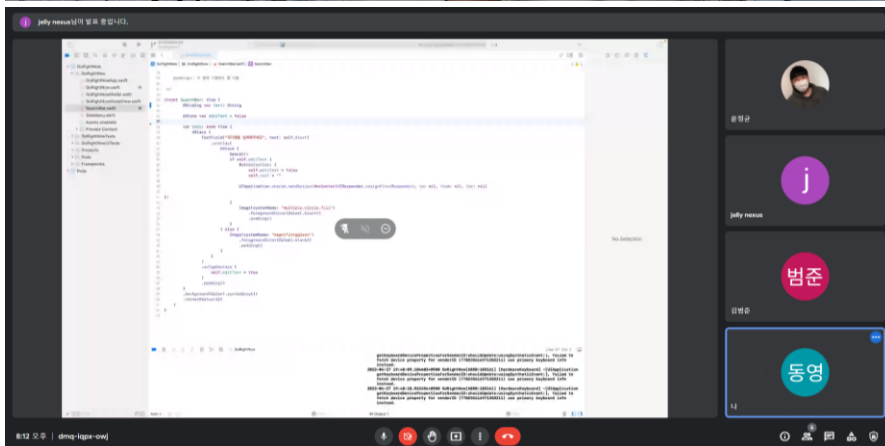
⇒ MapBox API를 통해 국가 위치를 검색하고 MapBox를 통해 검색되지 않는 국가만 MKLocalSearch를 통해 검색

## 기대효과

- 해외여행 증가 기조에 맞추어 코로나 관련 각국 입국정보를 편리하게 제공
- 차후 타 전염병(ex: 원숭이두창) 관련 입국 정보 제공으로 확대
- 국민들의 안전한 해외여행을 위해 기여
- 깔끔한 디자인 활용을 통한 쉬운 사용

## 협업내용

- 매주 1, 2회 이상 비대면 및 대면 회의 진행



- Github를 활용한 협업 진행

Github 주소: <https://github.com/nexusbusim/MobileAppProgramming2>



## 시연영상

[https://www.youtube.com/watch?v=2y\\_ySRi-GHE](https://www.youtube.com/watch?v=2y_ySRi-GHE)