2020년 1학기

모바일앱프로그래밍2

팀프로젝트 기술문서

과목명	모바일앱프로그래밍2
팀원	1조(강민주, 김미주, 김수민, 이수진)
프로젝트 이름	PHP(Pharmacy Hospital Planner)
제출일	2020.06.19

1. 개요

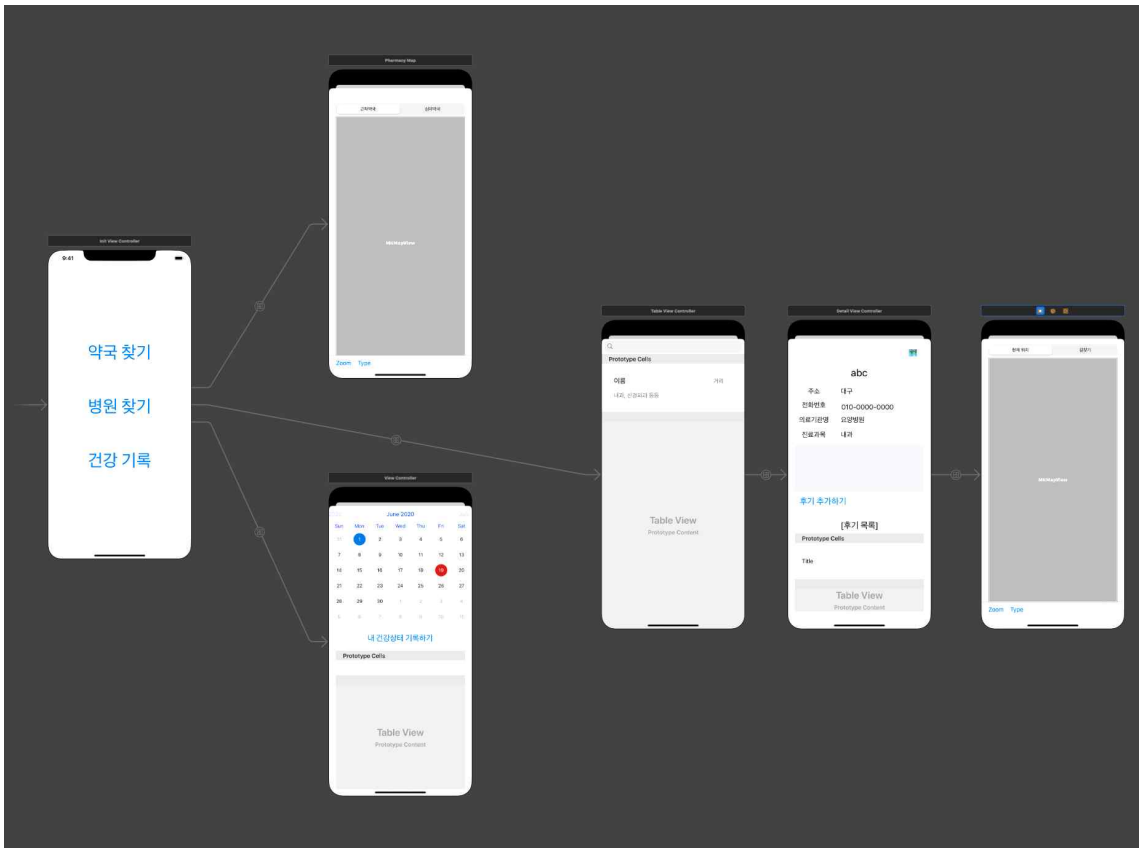


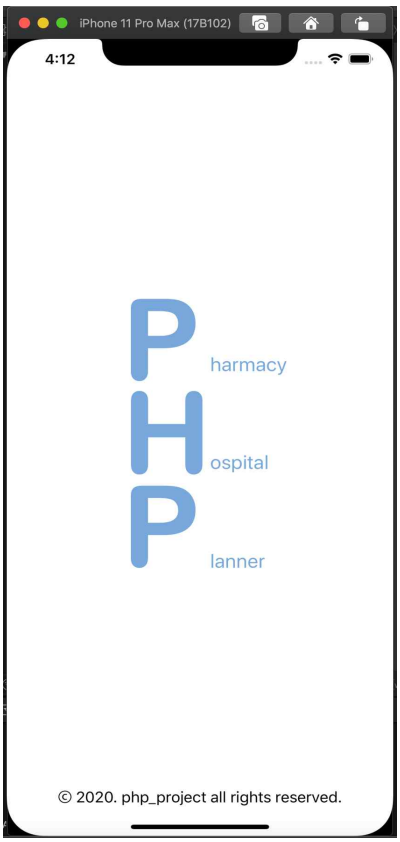
그림 1. 메인 스토리 보드

전체적인 화면 구성은 위의 그림 1과 같다. 첫 화면에서 〈약국 찾기〉, 〈병원 찾기〉, 〈건강 기록〉 이라는 버튼을 추가하였으며, 각 항목들의 화면은 segue를 통해 전환된다.

- 약국 찾기는 사용자 근방 3km에 있는 약국과 심야 약국의 위치를 제공한다.
- 병원 찾기는 경기도 내의 병원 목록을 보인다. 또한 각 병원들마다 후기를 작성할 수 있으며 병원 상세페이지에서 지도 이모티콘을 누르면 해당 병원의 위치를 핀으로 보여주며 길찾기 기능을 제공한다.
- 건강 기록은 캘린더를 제공하여 선택한 날짜에 사용자의 건강 상태를 추가할 수 있습니다. 삭제 또한 가능하다.

2. LaunchScreen.storyboard

- storyboard UI



- 세부사항: PHP(Pharmacy Hospital Planner)를 앱이 시작될 때의 화면에서 보여준다.

3. Main.storyboard(1) - 메뉴 화면

- storyboard UI

- **요약:** 각 항목으로 들어갈 수 있는 초기 화면이다.
- **세부 사항:**
  ① 각 버튼들은 stack view 안에 존재한다.
  ② 각각의 버튼마다 segue로 다음 화면과 연결되어 있다.

- **기타 사항**
  이미지를 사용하여 UI 디자인에 조금 더 신경을 쓰려고 했지만, Constraints 지정에 어려움이 있어 단순한 버튼으로 구현하였다.

4. PharmacyMap.swift

- **요약:** 경기도 약국 공공 데이터로부터 현재 내 위치 기준 반경 3km이내의 약국들의 위치에 pin을 띄어주고 24시간 운영하는 약국의 위치에 pin을 띄어준다.

- **세부 사항:**
  ① 경기도 약국 공공 데이터로부터 현재 내 위치 기준 반경 3km이내의 약국들의 위치에 pin을 띄어준다.
  ② 하단의 Zoom을 누르면 현재 내 위치 기준 남/북 2000미터 공간을 span한다.
  ③ UISegmentedControl에서 심야 약국을 선택하면 경기도 지역의 24시간 운영하는 약국들의 위체 pin을 띄어준다.
  ④ 하단의 Type를 누르면 기본지도/위성지도 변환한다.

- **주요 코드 설명:**

```
import MapKit
import CoreLocation
class PharmacyMap: UIViewController, MKMapViewDelegate {

    let data = DataLoader().pharmacyData
    let data1 = DataLoader().emergencyData
```

:AppleMap을 사용하기 위해 Mapkit, CoreLocation을 import한다.
경기도 약국 공공데이터 정보와 24시간 운영 약국 공공데이터를 load해온다.

```
@IBAction func zoomIn(_ sender: Any) {

let userLocation = mapView.userLocation//현재 위치 좌표, 남/북 2000미터 스팬

let region = MKCoordinateRegion(center: userLocation.location!.coordinate,
    latitudinalMeters: latitude, longitudinalMeters: longitude)
    mapView.setRegion(region, animated: true)
}
```

:Zoom 누르면 현재 위치가 지도에 포시된다. Zoom을 누르면 현재 위치기준으로 지도가 확대된다.

```
@IBAction func changeMapType(_ sender: Any) {


    if mapView.mapType == MKMapType.standard {
        mapView.mapType = MKMapType.satellite
    }
    else {
        mapView.mapType = MKMapType.standard
    }

}
```

:Type 누르면 기본지도/위성지도로 전환된다.

```
override func viewDidLoad() {

    super.viewDidLoad()
    mapView.delegate = self
    locationManager.desiredAccuracy = kCLLocationAccuracyBest//정확도 최고로 설정
    locationManager.requestWhenInUseAuthorization()//위치데이터 추적위해 사용자에게 승인요구
    locationManager.startUpdatingLocation()//위치업데이트 시작
    mapView.showsUserLocation = true//위치 보기

    let coor = locationManager.location?.coordinate

    let myLocation = CLLocation(latitude: coor?.latitude ?? 37.2839294 ,
        longitude: coor?.longitude ?? 127.076259)

    for i in 0..<data.count {
            let insertData = Pharmacy()
                insertData.name = data[i].BIZPLC_NM
                insertData.address = data[i].REFINE_ROADNM_ADDR
                insertData.latitude = (data[i].REFINE_WGS84_LAT as
                    NSString).doubleValue
                insertData.longitude = (data[i].REFINE_WGS84_LOGT as
                    NSString).doubleValue
              let coor = locationManager.location?.coordinate
          let myLocation = CLLocation(latitude: coor!.latitude , longitude:
!.longitude)
            let pharmacy = CLLocation(latitude: insertData.latitude, longitude:
                insertData.longitude)
            let distanceInMeters =  myLocation.distance(from: pharmacy)
            if(distanceInMeters <= 3000){
                setAnnotation(latitudeValue: insertData.latitude ,
                    longitudeValue: insertData.longitude, delta: 1, title:
                    insertData.name, subtitile: insertData.address)
            }
    }
    // Do any additional setup after loading the view.
```

:처음에 현재 위치 기준으로 3000미터 이내의 약국들의 위치에 pin을 띄운다.

```
@IBAction func sgChangeLocation(_ sender: UISegmentedControl) {
    if sender.selectedSegmentIndex == 0{
        mapView.removeAnnotations(mapView.annotations)
        let coor = locationManager.location?.coordinate
        let myLocation = CLLocation(latitude: coor?.latitude ?? 37.2839294 , longitude:
            coor?.longitude ?? 127.076259)
            for i in 0..<data.count {
                let insertData = Pharmacy()
                    insertData.name = data[i].BIZPLC_NM
                    insertData.address = data[i].REFINE_ROADNM_ADDR
                    insertData.latitude = (data[i].REFINE_WGS84_LAT as NSString).doubleValue
                    insertData.longitude = (data[i].REFINE_WGS84_LOGT as NSString).doubleValue
                  let coor = locationManager.location?.coordinate
                  let myLocation = CLLocation(latitude: coor!.latitude , longitude: coor!.longitude)
                let pharmacy = CLLocation(latitude: insertData.latitude, longitude:
                    insertData.longitude)
                let distanceInMeters =  myLocation.distance(from: pharmacy)
                if(distanceInMeters <= 3000){
                    setAnnotation(latitudeValue: insertData.latitude , longitudeValue:
                        insertData.longitude, delta: 1, title: insertData.name, subtitile:
                        insertData.address)
                }
        }

    }
    else if sender.selectedSegmentIndex == 1 {
         mapView.removeAnnotations(mapView.annotations)
        for i in 0..<data1.count {
                let insertData1 = Emergency()
                    insertData1.name = data1[i].BIZPLC_NM
                    insertData1.address = data1[i].REFINE_ROADNM_ADDR
                    insertData1.latitude = (data1[i].REFINE_WGS84_LAT as NSString).doubleValue
                    insertData1.longitude = (data1[i].REFINE_WGS84_LOGT as NSString).doubleValue

                    setAnnotation(latitudeValue: insertData1.latitude , longitudeValue:
                        insertData1.longitude, delta: 1, title: insertData1.name, subtitile:
                        insertData1.address)
                }

        }
```

:Segmented Control이 0이면 현재 위치 기준으로 반경 3000미터 이내에 있는 약국 위치의 핀을 띄우고 1이면 현재위치와 24시 운영하는 약국 위치에 핀을 띄운다.

● 참고 문서

① distance함수를 사용하여 현재 내 위치와 약국간의 거리 계산하는데 사용했으며, 현재 내 위치로부터 3000미터 이내의 약국만 지도에 pin이 표시되도록 하였다.

https://developer.apple.com/documentation/corelocation/cllocation/1423689-distance
https://stackoverflow.com/questions/33304340/how-to-find-out-distance-between-coordinates

② CLLLocationManager 사용해서 현재 위치의 위도, 경도 가져오기

https://sanghuiiiiii.tistory.com/entry/SWIFT-%ED%98%84%EC%9E%AC-%EC%9C%84%EC%B9%98-%EC%A3%BC%EC%86%8C-%EA%B0%80%EC%A0%B8%EC%98%A4%EA%B8%B0-%EB%AF%B8%EC%84%B8%EB%A8%BC%EC%A7%80%EC%95%B1-1-Day

③ MapView, Segmented Control 구현하는데 활용하였고, 위도/경도를 숫자로 직접 넣은 것이 아니라 함수를 사용하여 parameter로 위도/경도를 넘김.

https://books.google.co.kr/books?id=0p2CDwAAQBAJ&pg=PA234&dq=swift+%EA%B5%AC%EA%B8%80+%EB%A7%B5+%ED%98%84%EC%9E%AC+%EB%82%B4+%EC%9C%84%EC%B9%98&hl=ko&sa=X&ved=0ahUKEwim5L7Gj_3pAhUJPnAKHRk_Di0Q6AEIJzAA#v=onepage&q=swift%20%EA%B5%AC%EA%B8%80%20%EB%A7%B5%20%ED%98%84%EC%9E%AC%20%EB%82%B4%20%EC%9C%84%EC%B9%98&f=false
https://moonibot.tistory.com/27

④ Zoom, Type 활용
https://m.blog.naver.com/go4693/221223628491

5. TableViewController.swift

● **요약:** 경기도 병원 공공 데이터로부터 파싱한 데이터를 사용하여 현재 위치를 기준으로 가까운 거리부터 정렬한 후, 병원이름, 거리, 진료과목을 테이블에 보여준다. 병원 셀을 선택할 시 병원에 대한 세부 정보 페이지로 넘어간다.

● **세부 사항:**

① 현재 위치를 기준으로 가까운 병원부터 리스트에 보여준다.
② 병원 이름, 진료 과목으로 검색이 가능하다.
③ 병원 셀을 선택하면 병원에 대한 세부 정보 페이지로 넘어간다.

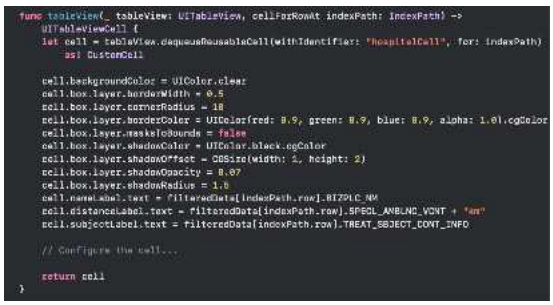
● **주요 코드 설명:**

```
class TableViewController: UIViewController, UITableViewDataSource, UISearchBarDelegate {
```

: 테이블 뷰와 검색바를 사용하기 위해 UITableViewDataSource와 UISearchBarDelegate를 상속받는다.

```
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    // #warning Incomplete implementation, return the number of rows
    return filteredData.count
}
```

: 데이터 수만큼 cell을 만든다.



: cell 모양을 커스텀해서 병원 이름, 거리, 진료 과목이 cell에 보이도록 한다.

```
func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
    self.filteredData = searchText.isEmpty ? allHospital : allHospital.filter({(hospital:
        HospitalData) -> Bool in
        var categoryMatch = hospital.TREAT_SBJECT_CONT_INFO.contains(searchText)  Variable '...
        var stringMatch = hospital.BIZPLC_NM.contains(searchText)  Variable 'stringMatch' was ne...
        return categoryMatch || stringMatch
    })

    tableView.reloadData()
}
```

: searchBar에 병원이나 진료과목을 검색하면 그 문자열을 포함하는 cell들만 띄워주는 함수이다.

```swift
func searchBarTextDidBeginEditing(_ searchBar: UISearchBar) {
    searchBar.showsCancelButton = true
}

func searchBarCancelButtonClicked(_ searchBar: UISearchBar) {
    searchBar.showsCancelButton = false
    searchBar.text = ""
    searchBar.resignFirstResponder()
    filteredData = allHospital
    tableView.reloadData()
}
```

:searchBar에 검색하기 시작하면 오른쪽에 전체 삭제하기 버튼이 뜨게하기 위해 showCancelButton값을 true로 만들었다.. 만약 그 버튼이 눌리면 다시 false값이 들어가고 searchBar의 텍스트가 지워진 후 모든 정보들이 tableView에 뜨도록 만들었다.

```swift
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "showDetail" {
        if let detailController = segue.destination as? DetailViewController {
            detailController.hospitalName =
                self.filteredData[tableView.indexPathForSelectedRow!.row].BIZPLC_NM
        }
    }
}
```

: 다음 세부 설명 페이지에 선택된 cell에 대한 데이터를 넘긴다.

● 참고 문서

① 기본적인 테이블 뷰 생성을 위해 튜토리얼 참고 (테이블 뷰 생성 및 코드와 연결, 데이터 수만큼 cell 생성)
https://youtu.be/C36sb5sc6lE

② 커스텀 cell을 만들기 위해 튜토리얼 참고 (hospitalCell을 찾아와서 직접 색상과 배치를 구성, 병원이름, 거리, 진료목록을 cell에 보여줄 수 있도록)
https://youtu.be/XPk0QeuEtFA

③ Tableview에 Search Bar 넣기 (서치바 생성 및 테이블 뷰에 삽입, 이름으로 검색, 진료과목으로 검색, contains를 통해 진료과목 string에서 '내과'라는 단어가 포함되어 있으면 검색이 되도록 개선)
https://www.youtube.com/watch?v=P5ob4TXIK90&feature=share
http://youtu.be/iH67DkBx9Jc
http://youtu.be/uBeC6PkFJog

④ 거리값 출력시 소수점 이하 2자리까지만 출력되도록 수정
https://www.it-swarm.dev/ko/swift/소수점-이하-두-자리로-반올림/822652192/

6. DataLoader.swift

● **요약:** 3개의 json 파일로부터 (hospital_data.json, pharmacy.jon, emergency_pharmacy.json) 데이터를 파싱한다. 특히 hospital_data.json으로부터 파싱한 데이터는 현재 위치를 기준으로 거리순 정렬을 한다.

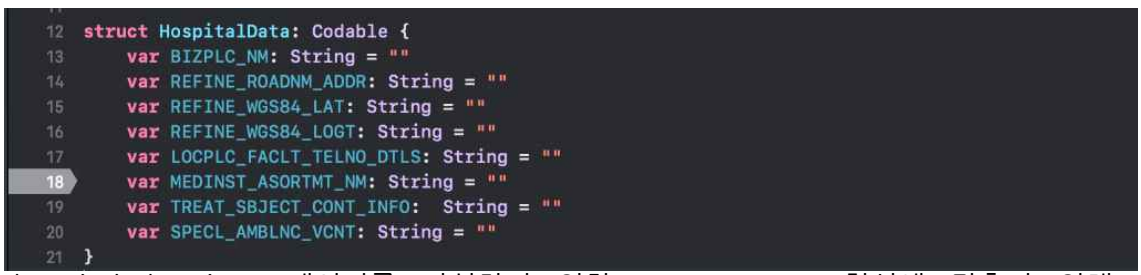
● **세부 사항:**

① load()를 통해 3개의 json 파일을 파싱한다.

② sort()를 통해 hospital 데이터 정보를 거리순으로 정렬한다.

● **주요 코드 설명:**

: hospital_data.json 데이터



: hospital_data.json 데이터를 파싱하기 위한 struct, JSON 형식에 맞추기 위해 Codable로 선언한다.

```
class Hospital: Object {
    @objc dynamic var name = ""
    @objc dynamic var address = ""
    @objc dynamic var latitude: Double = 0
    @objc dynamic var longitude: Double = 0
    @objc dynamic var telephone = ""
    @objc dynamic var medinst = ""
    @objc dynamic var subject = ""
    var reviews = List<String>()
}
```

: 데이터베이스에 담길 객체를 선언하기 위해 Hospital 클래스를 만들었습니다. 병원 이름, 주소, 경도, 위도, 전화번호, 병원유형, 진료과목, 후기 리스트를 property로 가지는 클래스입니다.

```
@Published var hospitalData = [HospitalData]()
@Published var pharmacyData = [PharmacyData]()
@Published var emergencyData = [EmergencyData]()
```

: 2번 사진과 같은 struct를 이용해 배열을 만든다. @Published property wrapper를 사용하여 시스템에 새로운 값이 할당될 때마다 관측할 수 있는 이벤트를 내보낸다.

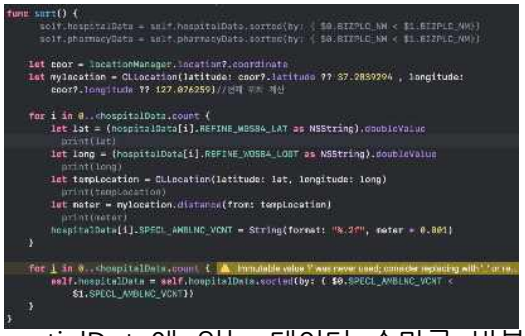
```
init() {
    load()
    sort()
}
```

: 데이터 가공을 위해 가장 먼저 호출되는 함수들이다.

```
func load() {

    if let filehosLocation = Bundle.main.url(forResource: "hospital_data", withExtension:
        "json") {

        // do catch in case of an error
        do {
            let data = try Data(contentsOf: filehosLocation)
            let jsonDecoder = JSONDecoder()
            let dataFromJson = try jsonDecoder.decode([HospitalData].self, from: data)

            // don't add if address is nil
            for i in 0..<dataFromJson.count {
                if(dataFromJson[i].REFINE_ROADNM_ADDR != "") {
                    self.hospitalData.append(dataFromJson[i])
                }
            }

        } catch {
            print(error)
        }
    }
```

: 파일의 이름과 자료형을 Bundle.main.url에 입력하여 파일을 열고 그 후 JSONDecoder를 통해 json 파일을 decode 한다. 주소에 값이 없는 경우는 제외하고 append하여서 후에 mapview에서 오류가 생기지 않도록 한다.

: 상단에서 선언한 CLLocationManager의 객체인 locationManager와 coordinate를 사용하여 사용자의 위치를 가져온다. 시뮬레이터에서 gps가 동작하지 않는 경우를 대비하여 ??를 통해 coor값이 nil인 경우 정해진 위도, 경도를 넣는다. 그 후



hosptialData에 있는 데이터 수만큼 반복하며 현재 위치와 데이터의 위치간의 거리를 구해 struct에 넣는다. 이를 가지고 오름차순으로 가까운 거리부터 정렬한다.

● 참고 문서

① local에 있는 json 파일을 swift에서 파싱하는 방법 (특히 Codable 과 JSONDecoder의 사용법에 대해 조사)
https://youtu.be/Oj-2s0ALACE
https://youtu.be/cyWrj61vpCY
https://youtu.be/9tkoD0fSxmU

② CLLocationDistance 반환값 확인
https://developer.apple.com/documentation/corelocation/cllocationdistance

③ sort()에서 CLLocation을 계산하는 방법에 대해 아이디어를 얻은 사이트
https://ko.coder.work/so/swift/698254

④ 경기도 병원, 약국 공공 데이터
https://data.gg.go.kr/portal/adjust/selectThemeServicePage.do?infId=5E9F96P9YXCJ8ZTFHFF721022502&cateId=T101&infSeq=1&layout=1
https://data.gg.go.kr/portal/adjust/selectThemeServicePage.do?page=1&rows=10&sortColumn=&sortDirection=&infId=374OQ18937P0828Q981618796&infSeq=1&temaId=T101&sigunNm=&orgNm=&index=&tab=&layout=1&parSigunNm=&mParSigunNm=&mSigunNm=&cateId=T101

7. AppDelegate.swift

● **요약: 파싱한 json 데이터를 데이터베이스에 넣는다.**

● **세부 사항:**

①json 파일을 파싱한 데이터를 데이터베이스에 저장한다.

②sort()를 통해 hospital 데이터 정보를 거리순으로 정렬한다.

● **주요 코드 설명:**

```
let allHospital = DataLoader().hospitalData
let realm = try! Realm(configuration: Realm.Configuration(deleteRealmIfMigrationNeeded: true))
```

: hospital 데이터 정보를 파싱해서 allHospital property에 저장한다. import 해온 RealmSwift framework에 정의되어 있는 클래스 Realm의 객체를 선언한다.

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
    [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    // Override point for customization after application launch.
    locationManager = CLLocationManager()
    locationManager?.requestWhenInUseAuthorization()

    for i in 0..<allHospital.count {
        let insertData = Hospital()
        insertData.name = allHospital[i].BIZPLC_NM
        insertData.address = allHospital[i].REFINE_ROADNM_ADDR
        insertData.latitude = (allHospital[i].REFINE_WGS84_LAT as NSString).doubleValue
        insertData.longitude = (allHospital[i].REFINE_WGS84_LOGT as NSString).doubleValue
        insertData.telephone = allHospital[i].LOCPLC_FACLT_TELNO_DTLS
        insertData.medinst = allHospital[i].MEDINST_ASORTMT_NM
        insertData.subject = allHospital[i].TREAT_SBJECT_CONT_INFO

        try! realm.write {
            realm.add(insertData)
        }
    }

    return true
}
```

: application 함수 중 didFinishLaunchingWithOptions에 어플리케이션이 실행되면 가장 먼저 실행해야하는 것들을 적는다. 먼저, 데이터베이스에 담길 객체를 insertData 라고 선언했다. 파싱해서 가져온 allHospital 데이터 중 필요한 정보들만 inserDat에 담는다. 이를 allHospital 배열의 크기만큼 반복하며 데이터베이스에 담는다.

● **참고 문서**

① AppLifeCycle에 대한 문서
https://developer.apple.com/documentation/uikit/app_and_environment/managing_your_app_s_life_cycle

② realm 데이터베이스 사용법
https://realm.io/docs/swift/latest/

8. DetailViewController.swift

● **요약: TableViewController에서 선택된 cell의 데이터의 세부정보를 DetailViewController에 띄운다.**

● **세부 사항:**

① TableViewController에서 전달받은 데이터로 데이터베이스를 query해서 세부정보를 띄운다.

② 후기를 적고 추가하면 데이터베이스에 추가되고 즉시 table에 띄운다.

③ 해당 병원의 지도를 연결해주는 버튼을 만든다.

● **주요 코드 설명:**

```
class DetailViewController: UIViewController, UITableViewDataSource {


    @IBOutlet weak var nameLabel: UILabel!
    @IBOutlet weak var addressLabel: UILabel!
    @IBOutlet weak var telLabel: UILabel!
    @IBOutlet weak var medinstLabel: UILabel!
    @IBOutlet weak var subjectLabel: UILabel!
    @IBOutlet weak var reviewTextView: UITextView!
    @IBOutlet weak var reviewTable: UITableView!

    var hospitalName = ""
```

: TableViewController에서 클릭한 정보가 hospitalName에 저장되는데, 이 정보를 통해 query해와서 데이터를 띄울 label들을 선언했다. 이름(nameLabel), 주소(addressLabel), 전화번호(telLabel), 병원유형(medinstLabel), 진료과목(subjectLabel), 후기 입력창(reviewTextView), 후기 table(reviewTable)이 있다.

```
override func viewDidLoad() {
    super.viewDidLoad()

    let query = realm.objects(Hospital.self).filter("name = %@", hospitalName).first

    nameLabel.text = query?.name
    addressLabel.text = query?.address
    telLabel.text = query?.telephone
    medinstLabel.text = query?.medinst
    subjectLabel.text = query?.subject

    reviewTextView.frame.size.height = 374
    reviewTextView.frame.size.width = 133

}
```

: 가장먼저 선택한 cell의 병원 이름을 가지는 객체들을 데이터베이스에서 모두 query해서 이름, 주소, 전화번호, 병원유형, 진료과목을 띄운다.

```swift
@IBAction func addReviewButton(_ sender: Any) {
    var query = realm.objects(Hospital.self).filter("address = %@", addressLabel.text!).first

    try! realm.write {
        query?.reviews.append(reviewTextView.text)
    }

    query = realm.objects(Hospital.self).filter("address = %@", addressLabel.text!).first

    print(query?.reviews)  // Expression implicitly coerced from 'List<String>?' to 'Any'

    reviewTable.reloadData()
}
```

: 후기를 작성하고 추가버튼을 누르면 실행되는 함수이다. address가 key값이기 때문에 address로 데이터베이스에서 query해서 그 객체의 review배열에 textView에 담긴 text를 추가한다. 추가된 데이터를 바로 보기 위해 reloadData함수를 사용한다.

```swift
func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    let query = realm.objects(Hospital.self).filter("address = %@", addressLabel.text!).first

    return (query?.reviews.count)!
}
```

: tableView에 띄울 cell을 개수를 반환해주어야 하는 함수이다. address가 key값이어서 address로 데이터베이스에서 query해서 그 객체의 review배열 크기를 반환해준다.

```swift
func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for: indexPath)

    let query = realm.objects(Hospital.self).filter("address = %@", addressLabel.text!).first
    let review = query?.reviews[indexPath.row]
    cell.textLabel!.text = review
    return cell
}
```

: tableView에 후기를 띄우게 하는 코드이다. cell을 dequeue해서 그 cell의 textLabel에 address로 query해온 데이터의 review들을 띄우고 cell을 반환해준다.

```swift
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    var query = realm.objects(Hospital.self).filter("address = %@", addressLabel.text!).first  // Variable 'query' was n

    if segue.identifier == "showMap" {
        if let mapController = segue.destination as? mapController {
            mapController.latitude = query?.latitude as! Double  // Forced cast from 'Double?' to 'Double' only unwraps optionals
            mapController.longitude = query?.longitude as! Double  // Forced cast from 'Double?' to 'Double' only unwraps option
            mapController.Title = query?.name as! String  // Forced cast from 'String?' to 'String' only unwraps optionals; did you m
            mapController.subtitle = query?.address as! String  // Forced cast from 'String?' to 'String' only unwraps optionals; di
        }
    }
}
```

: address로 query해서 해당 병원의 모든 정보를 받아온다. 스토리보드에서 버튼을 누르면 MapController로 이동하는 segue의 이름을 showMap으로 설정하였다. 만약 segue가 showMap이라면, mapController에 경도, 위도, 병원이름, 주소를 넘겨주었다.

● 참고 문서

① ios 개발에 관한 강좌. tableView 사용법, segue를 지정하여 다음 controller에 데이터 넘겨주는 방법 등을 참고해서 활용했다.
https://www.inflearn.com/course/swift4-스위프트-ios-개발/dashboard

9. MapController.swift

- **요약:** TableView에서 선택된 병원의 위치에 pin을 띄어주고 Segmented Control에서 길 찾기로 변경하면 현재 위치에서 병원까지의 경로를 그려준다.

- **세부 사항:**
  ① 경기도 병원 공공 데이터로부터 TableView에서 선택된 병원의 위치에 pin을 띄어준다.
  ② 하단의 Zoom을 누르면 현재 내 위치 기준 남/북 2000미터 공간을 span한다.
  ③ UISegmentedControl에서 길찾기를 선택하면 현재 위치에서 선택된 병원까지의 경로를 그려준다.
  ④ 하단의 Type를 누르면 기본지도/위성지도 변환한다.

- **주요 코드 설명:**

```
import MapKit
import CoreLocation
class mapController: UIViewController, MKMapViewDelegate {

    //let data = DataLoader().hospitalData
    var latitude: Double = 0    // 디비에서 쿼리해온 위도
    var longitude: Double = 0 // 디비애서 쿼리해온 경도
    var Title : String = ""  //디비에서 쿼리해온 이름
    var subtitle = ""   //디비에서 쿼리해온 번호 or 주소

    @IBOutlet var mapView: MKMapView!
    let locationManager = CLLocationManager()
```

: AppleMap을 사용하기 위해 Mapkit, CoreLocation을 import한다.
MKMapViewDelegate를 상속받고 DB에서 query해서 받아오기 위해 변수들을 선언한다. 지도를 표시하기 위해 mapView 아울렛을 만들고 위치정보를 얻기 위해 CLLocationManager() 객체를 만든다.

```
@IBAction func zoomIn(_ sender: Any) {

let userLocation = mapView.userLocation//현재 위치 좌표, 남/북 2000미터 스팬

let region = MKCoordinateRegion(center: userLocation.location!.coordinate,
    latitudinalMeters: latitude, longitudinalMeters: longitude)
    mapView.setRegion(region, animated: true)
}
```

: Zoom 누르면 현재 위치가 지도에 포시된다. Zoom을 누르면 현재 위치기준으로 지도가 확대된다.

```
@IBAction func changeMapType(_ sender: Any) {


    if mapView.mapType == MKMapType.standard {
        mapView.mapType = MKMapType.satellite
    }
    else {
        mapView.mapType = MKMapType.standard
    }

}
```

: Type 누르면 기본지도/위성지도로 전환된다.

```
override func viewDidLoad() {
    super.viewDidLoad()
    mapView.delegate = self
    locationManager.desiredAccuracy = kCLLocationAccuracyBest//정확도 최고로 설정
    locationManager.requestWhenInUseAuthorization()//위치데이터 추적위해 사용자에게 승인요구
    locationManager.startUpdatingLocation()//위치업데이트 시작
    mapView.showsUserLocation = true//위치 보기

    setAnnotation(latitudeValue: latitude, longitudeValue: longitude, delta: 1,
        title: Title, subtitile: subtitle)
    // Do any additional setup after loading the view.
}
func setAnnotation(latitudeValue: CLLocationDegrees, longitudeValue:
    CLLocationDegrees, delta span : Double, title strTitle: String, subtitile
    strSubtitle:String)
{
    let annotation = MKPointAnnotation()
    annotation.coordinate = CLLocationCoordinate2D(latitude: latitudeValue,
        longitude: longitudeValue)
    mapView.addAnnotation(annotation)
    annotation.title = strTitle//"대구파티마병원" //디비 쿼리 병원 이름
    annotation.subtitle = strSubtitle //"053-940-7114"//디비 쿼리 병원 번호
}
```

: 처음 실행할 때 위치 정보를 업데이트 받고 선택된 병원 위치에 pin을 띄운다.
pin을 띄우는 함수로 위도, 경도, 이름, 주소 값을 parameter로 받고 addAnnotaion을 이용해서 지도에 pin을 띄운다.

```
@IBAction func sgChangeLocation(_ sender: UISegmentedControl) {
    if sender.selectedSegmentIndex == 0{
        setAnnotation(latitudeValue: latitude, longitudeValue: longitude, delta: 1,
            title: Title, subtitile: subtitle)

    }
    else if sender.selectedSegmentIndex == 1 {
        mapThis(destinationCord: CLLocationCoordinate2D(latitude: latitude,
            longitude: longitude))

    }
}
```

: Segmented Control이 0이면 현재 위치와 병원 위치의 핀을 띄운 화면을 보여주고 1이면 현재위치와 병원위치까지의 경로를 보여준다.

```
    func mapThis(destinationCord : CLLocationCoordinate2D) {
        let souceCordinate = (mapView.userLocation.coordinate)
        let soucePlaceMark = MKPlacemark(coordinate: souceCordinate)
        let destPlaceMark = MKPlacemark(coordinate: destinationCord)

        let sourceItem = MKMapItem(placemark: soucePlaceMark)
        let destItem = MKMapItem(placemark: destPlaceMark)

        let destinationRequest = MKDirections.Request()
        destinationRequest.source = sourceItem
        destinationRequest.destination = destItem
        destinationRequest.transportType = .automobile
        destinationRequest.requestsAlternateRoutes = true

        let direction = MKDirections(request: destinationRequest)
        direction.calculate { (response, error) in
            guard let response = response else {
                if let error = error {  ⚠ Value 'error' was defined but never used; consider replac
                    print("Something is wrong")
                }
                return
            }

            let route = response.routes[0]
            self.mapView.addOverlay(route.polyline)
            self.mapView.setVisibleMapRect(route.polyline.boundingMapRect, animated:
                true)

    }
}
```

: 현재 위치와 선택된 병원까지의 경로를 보여준다.

```
func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) ->
    MKOverlayRenderer {
    let render = MKPolylineRenderer(overlay: overlay as! MKPolyline)
    render.strokeColor = .blue
    return render
}
```
: 경로를 파란색 선으로 그려준다.

- **참고소스**

① distance함수를 사용하여 현재 내 위치와 약국간의 거리 계산하는데 사용했으며, 현재 내 위치로부터 3000미터 이내의 약국만 지도에 pin이 표시되도록 하였다.

https://developer.apple.com/documentation/corelocation/cllocation/1423689-distance
https://stackoverflow.com/questions/33304340/how-to-find-out-distance-between-coordinates

② CLLLocationManager 사용해서 현재 위치의 위도, 경도 가져오기

https://sanghuiiiiii.tistory.com/entry/SWIFT-%ED%98%84%EC%9E%AC-%EC%9C%84%EC%B9%98-%EC%A3%BC%EC%86%8C-%EA%B0%80%EC%A0%B8%EC%98%A4%EA%B8%B0-%EB%AF%B8%EC%84%B8%EB%A8%BC%EC%A7%80%EC%95%B1-1-Day

③ MapView, Segmented Control 구현하는데 활용하였고, 위도/경도를 숫자로 직접 넣은 것이 아니라 함수를 사용하여 parameter로 위도/경도를 넘김.

https://books.google.co.kr/books?id=0p2CDwAAQBAJ&pg=PA234&dq=swift+%EA%B5%AC%EA%B8%80+%EB%A7%B5+%ED%98%84%EC%9E%AC+%EB%82%B4+%EC%9C%84%EC%B9%98&hl=ko&sa=X&ved=0ahUKEwim5L7Gj_3pAhUJPnAKHRk_Di0Q6AEIJzAA#v=onepage&q=swift%20%EA%B5%AC%EA%B8%80%20%EB%A7%B5%20%ED%98%84%EC%9E%AC%20%EB%82%B4%20%EC%9C%84%EC%B9%98&f=false
https://moonibot.tistory.com/27

④ Zoom, Type 활용
https://m.blog.naver.com/go4693/221223628491

10. ViewController.swift

- **요약:** FSCalendar를 이용하여 캘린더 화면을 보여준다. 날짜 선택 후 건강 기록을 추가하면 tableView에 추가되며, 기록 삭제 기능 또한 제공된다.

- **세부 사항:**
  ① 캘린더 라이브러리인 FSCalendar를 이용하여 View에 캘린더를 추가한다.
  ② 빨간색으로 표시되는 날짜는 오늘 날짜이며, 날짜를 선택하면 선택한 날짜는 파란색으로 표시된다.
  ③ 날짜를 선택하여 '내 건강정보 기록하기' 버튼을 누르면 메모를 기록할 수 있는 alert창이 뜬다.
  ④ 메모를 기록해 '추가하기'를 누르면 Realm DB에 선택한 날짜와 메모가 String 형태로 저장되며 tableView에도 업데이트 된다.
  ⑤ 다른 날짜에도 건강 상태를 기록할 수 있으며, 날짜를 누를 때 마다 해당 날짜에 저장하였던 기록이 뜬다.
  ⑥ tableView에 있던 기록들을 왼쪽으로 스와이프하면 기록 삭제도 가능하다.

- **주요 코드 설명**

```swift
import UIKit
import FSCalendar
import RealmSwift


class ViewController: UIViewController, FSCalendarDataSource, FSCalendarDelegate, UITableViewDelegate, UITableViewDataSource {
```

: 캘린더와 데이터베이스를 사용하기 위하여 FsCalendar와 RealmSwift library를 import한다. ViewController는 UIViewController, FSCalendarDataSource, FSCalendarDelegate, UITableViewDelegate, UITableViewDataSource를 상속받는다.

```swift
class HealthMemo: Object {

    @objc dynamic var date:String = ""
    @objc dynamic var write: String = ""
}
```

: HealthMemo 클래스를 위와 같이 선언한다. HealthMemo는 date와 write를 멤버로 가지며 date는 기록 추가 시 선택하였던 날짜, write는 그 기록을 나타낸다.

```swift
    public func calendar(_ calendar: FSCalendar, didSelect date: Date, at monthPosition: FSCalendarMonthPosition) {
          let dateFormatter = DateFormatter()
            dateFormatter.dateFormat = "YYYY-MM-dd"
        cnt = 0
        visit = []
        dateInfo = dateFormatter.string(from: date)
        dateSelected = dateFormatter.string(from: date)


        print("calendar")
        memoTable.reloadData()

       }
```

: calendar에서 날짜를 선택하면 "YYYY-MM-dd"형태로 날짜 형태를 포맷시킨다. 이때 dateInfo는 DB에 들어가는 날짜이며, dateSelected는 후에 tableView에 기록을 출

력할 때 DB에 들어가있는 정보들 중 선택된 날짜와 동일한 날짜를 가지는 기록들만 출력하도록 하기 위하여 존재한다. 사실상 dateInfo와 dateSelected는 같은 값을 가지지만 코드 가독성을 위해 두 개로 나누었다.

```
    @IBAction func insert(_ sender: Any) {
        var textField = UITextField()


        let action = UIAlertAction(title: "추가하기", style: .default) { (action) in
            let new = HealthMemo()
            new.write = textField.text!
            new.date = self.dateSelected

            self.save(healthMemo: new)
            self.memoTable.reloadData()
        }

        let alert = UIAlertController(title: "건강 상태를 기록하세요", message: "", preferredStyle: .alert)
        alert.addTextField{
            (alertTextField) in
            textField = alertTextField
        }

        alert.addAction(action)
        present(alert, animated: true, completion: nil)

    }
```

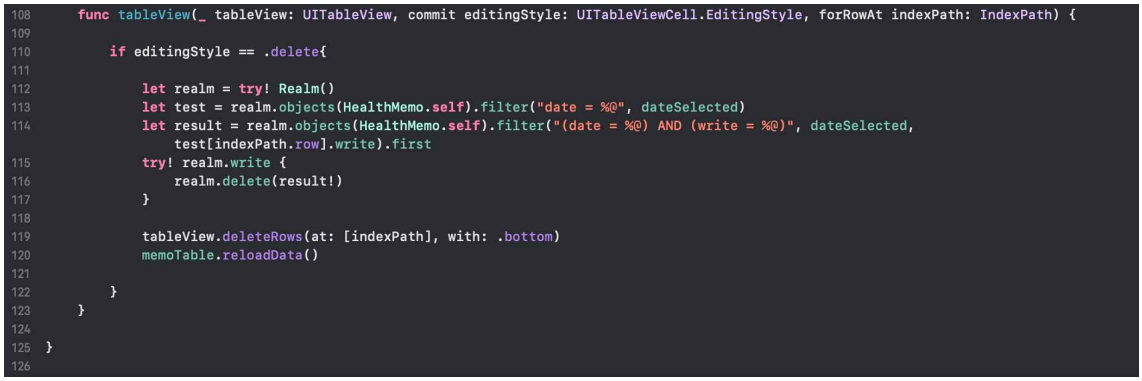
: insert 함수는 캘린더 화면에 뜨는 '내 건강상태 기록하기' 버튼의 action으로 연결된 함수이다. 버튼을 누르면 alert 창이 뜬다. alert 창 자체의 title은 "건강상태를 기록하세요" 이며, '추가하기' 를 누르면 save함수가 호출된 이후 tableView가 reload 된다. save 함수는 바로 아래에서 설명한다.

```
    func save(healthMemo: HealthMemo){
        print("save")
        do{
            try realm.write{
                realm.add(healthMemo)

            }
        }catch{
            print("error")
        }
    }
```

: save함수에서는 위의 insert함수에서 인자로 받아온 healthMemo를 DB에 저장한다.

```
      public func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

        let result = realm.objects(HealthMemo.self).filter("date = %@", dateSelected)
        return (result.count)

    }

    public func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {

        let cell = tableView.dequeueReusableCell(withIdentifier: "FirstCell", for: indexPath)

        let result = realm.objects(HealthMemo.self).filter("date = %@", dateSelected)
        print(result)
        cell.textLabel!.text = result[indexPath.row].write

        return cell
    }
```

: tableView를 구현하기 위해서는 위의 두 함수를 필수적으로 구현해야 한다. tableView에서는 DB에 저장된 정보들 중 선택된 날짜에 해당하는 기록들을 보여준다. 따라서 위의 함수에서는 DB에서 데이터를 받아와 date가 선택된 날짜와 같은 데이터만 골라낸다. 그리고 result.count를 리턴하는데 이는 화면에 나타날 tableView의 셀 개수이다.



```
108     func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
109
110         if editingStyle == .delete{
111
112             let realm = try! Realm()
113             let test = realm.objects(HealthMemo.self).filter("date = %@", dateSelected)
114             let result = realm.objects(HealthMemo.self).filter("(date = %@) AND (write = %@)", dateSelected,
                    test[indexPath.row].write).first
115             try! realm.write {
116                 realm.delete(result!)
117             }
118
119             tableView.deleteRows(at: [indexPath], with: .bottom)
120             memoTable.reloadData()
121
122         }
123     }
124
125 }
126
```

: tableView에 나와있는 기록들을 삭제하기 위해 추가적으로 구현한 함수이다. delete는 목록 중 하나를 swipe 하면 삭제를 할 수 있는 데, 이때 DB에서 선택된 기록의 날짜와 기록이 같은 데이터를 삭제한다. 그리고 tableView에서도 해당 데이터는 삭제된다.

- **참고 문서**

① FSCalendar 라이브러리: https://github.com/WenchaoD/FSCalendar

② Realm DB에 데이터 저장하는 법과 FSCalendar에서 날짜를 받아오는 방법을 참고. FSCalendar에서는 날짜 데이터를 포맷하는 방식을 중심으로 참고하였으며, 날짜 선택 시 콜백메소드는 구현할 필요가 없어서 제외함.
https://hyongdoc.tistory.com/336?category=799893

③ tableView 기본과 tableView의 cell을 띄우는 방법을 참고함. 해당 글에서는 extension을 이용해 확장한 후 tableView 메소드를 구현하였지만 해당 프로젝트는 클래스 내부 메소드로 작성함.
https://shxodnr24.tistory.com/47

④ 버튼을 누르고 alert 창이 뜨며, alert 창의 textfield를 DB에 저장하는 방법을 참고함. 게시글 하단의 tableView 메소드 관련은 ③에서 확인하였기 때문에 참고하지 않음.
https://blog.naver.com/PostView.nhn?blogId=hyer1k&logNo=221323101645&parentCategoryNo=&categoryNo=14&viewDate=&isShowPopularPosts=true&from=search