

# Swift Term Project

## (여행 기록 애플리케이션)



과 목 명	Swift 프로그래밍
담 당 교 수	김계숙 교수님
프로젝트 팀명	여행 기록 애플리케이션
계 열	컴퓨터정보
학 번	201540537
	201550241
	201620463
이 름	박선재, 유성민, 김병철

# 목차

□ 프로젝트 소개	-----	1
주제선정 선정이유		1
주간별 작업		2
개발 환경		3
□ 프로그램 UI 구성	-----	2
스토리보드 구성		1
Google Maps		2
SearchBar		3
Navigation		4
메모장		5
□ 주요 코드 소개	-----	3
UISearchController		1
MKLocalSearch		2
GoogleMaps Coordinate		3
Alert		4
UITextView		5

## □ 프로젝트 소개.

여행자들을 위한 애플리케이션으로, 현재 위치에서 목적지를 검색 하여 경로를 찾고, 해당 지역에 대해 메모하는 기능.





## □주제 선정 이유.

개인적인 취미생활로 배낭 여행다니는 것을 좋아하는데, 네비게이션을 자주 사용했던 기억이 있습니다. 경로를 보면서 걷다가 문득 메모할 일이 생기는데 그때마다 메모장을 열어 메모를 하고, 다시 네비게이션을 오픈해야하는 불편했던 경험이 있어 이를 해소하고자 이 주제를 선정을 했습니다.

## □ 주간별 작업.

구 분	내 용	비 고
1주차	주제선정 및 개발환경 설정	
2주차	UI 설계	
3주차	시스템 기능 구현	
4주차	테스트 및 보고서 작성	
5주차	최종 보고서 작성	

## □개발환경.

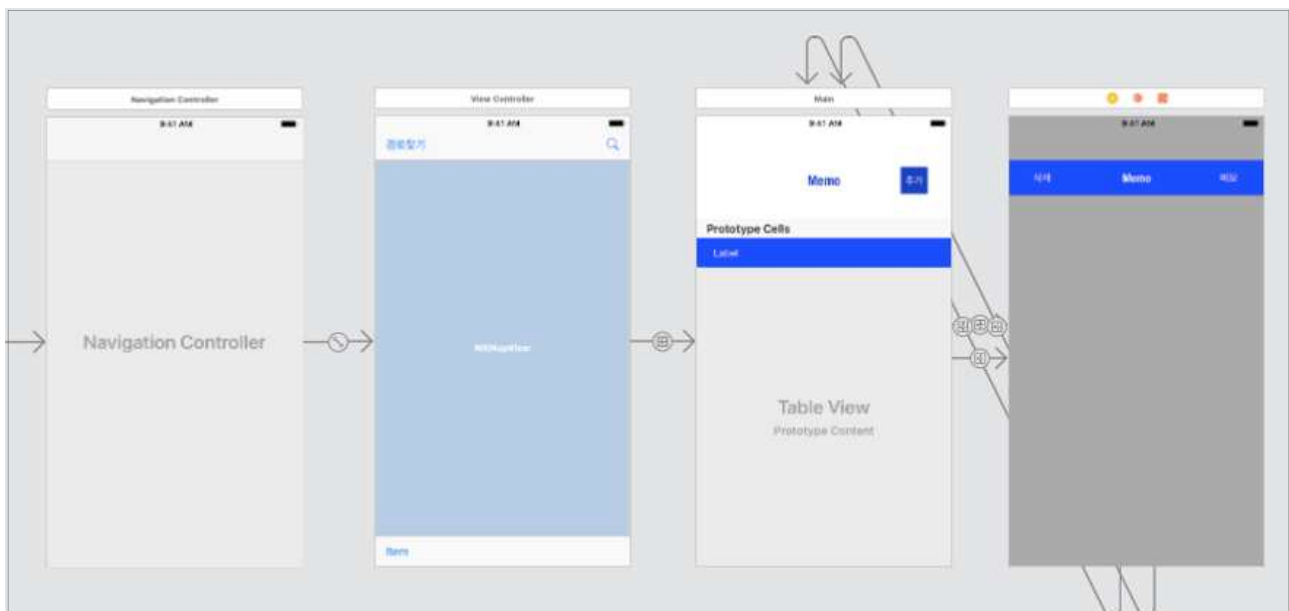
Apple System	Tool Platform
	
Language	DataBase
	

## □ 수업 계획서

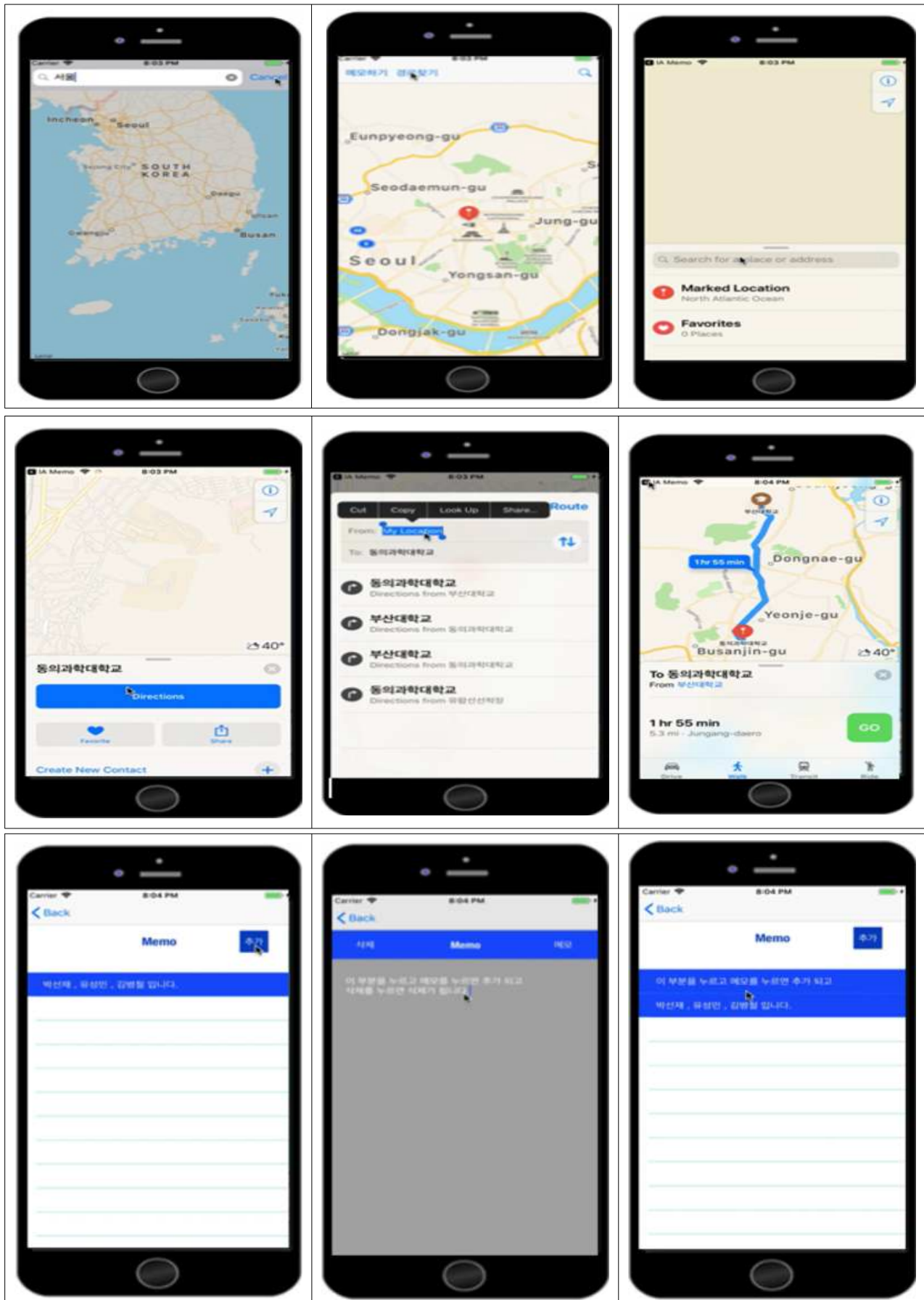
과제번호				
Objecd-C 프로젝트 수업 계획서				
1. 프로젝트 명	(한글) 여행 메모 어플리케이션 (영문) DirectionTravelMemo Application ( Direction + Travel + Memo Application )			
2. 지도교수	소속	컴퓨터 정보 계열 모바일 인터넷	성명	김계숙 (인)
3. 조 편성	조장 및 조원	성 명	학 번	담당 역할
		박선재	201540537	조장, 프로젝트 설계 · 계획 및 전체 보완
		유성민	201550241	조원, 프로젝트 설계 · 계획 및 전체 보완
		김병철	201620463	조원, 프로젝트 설계 · 계획 및 전체 보완
4. 학습 목표	프로젝트를 진행하므로써 역할 분담과 팀원 사이의 의사 소통을 배우고, 각자 책임과 의무를 가지고 맡은 역할을 수행하므로써 팀원들의 실력 향상과 책임감을 양성하고자 합니다. 프로젝트라는 공동체 안에서 구성원의 실수나 오류는 공동체 내에서 책임을 맡아야 할 것이며, 구성원들은 프로젝트의 일원으로서 각자 맡은 책임과 의무에 최선을 다해야 할 것입니다.			
5. 프로젝트 과제의 개요	여행을 가기전 여행을 계획하면서 여행지에 가면 할 체험이나 맛집정보,여행지에서 실시간으로 느낀 점,여행을 하고 난 후 후기 등 생각을 정리할수 있도록 메모 기능을 만들어 일상 생활에서 메모하는 습관을 만들고자 생각하였음.			
6. 프로젝트 수행 방법 (개선방안,연구방법 등)	질문의 제작 및 제공 방법, 여행지에 맞게 질문을 던지는 방법 등을 프로젝트에 실질적으로 필요한 기능을 파악하고 그것을 구현하는 방법을 회의를 통해 분석 합니다. 각 세부적인 항목을 프로그래밍 했을 때 그 흐름을 연결 할 수 있는 인터페이스를 협의한 후에 프로그래밍을 시작할 것 입니다. 팀 단위의 프로젝트는 팀원들이 제작한 프로그램 사이의 흐름 연결이 중요하므로, 프로젝트 진행 중 수시로 의사소통을 통한 인터페이스의 교환이 이루어지게 할 것입니다. 프로그램의 기본이 제작된 이후에는 데이터 입 · 출력을 통한 테스트를 가질 것이고 버그 · 오류의 정정과 기능의 보완 · 추가에 대한 업데이트를 가질 것 입니다.			

## □ 프로그램 UI 구성

### 1. 스토리보드 구성도.



## 2. SWIFT UI 설계 및 구현.



### 3. 주요 코드 소개.

- UISearchController



```
@IBAction func searchButton(_ sender: Any)
{
    let searchController = UISearchController(searchResultsController: nil)
    searchController.searchBar.delegate = self
    present(searchController, animated: true, completion: nil)
}
```

1. UISearchController : 동일한(해당) 뷰에 결과를 표출합니다.
2. searchBar.delegate : 동일한(해당) 뷰에 이벤트를 발생시킵니다.

```
func searchBarSearchButtonClicked(_ searchBar: UISearchBar) {
    UIApplication.shared.beginIgnoringInteractionEvents()

    //Activity Indicator
    let activityIndicator = UIActivityIndicatorView()
    activityIndicator.style = UIActivityIndicatorView.Style.gray
    activityIndicator.center = self.view.center
    activityIndicator.hidesWhenStopped = true
    activityIndicator.startAnimating()

    self.view.addSubview(activityIndicator)

    //Hide search bar
    searchBar.resignFirstResponder()
    dismiss(animated: true, completion: nil)

    //Create the search request
    let searchRequest = MKLocalSearch.Request()
    searchRequest.naturalLanguageQuery = searchBar.text

    let activeSearch = MKLocalSearch(request: searchRequest)
```

- \* Activity Indicator : 실행 화면에 대한 지표 설정.(스타일, 표시될 위치, 숨김, 애니메이션 등)
- \* MKLocalSearch : 문자열을 사용하여 위치를 검색하고 검색이 완료시 검색한 문자열과 일치하는 지정된 영역 내의 위치로 이동합니다.



```

activeSearch.start { (response, error) in

    activityIndicator.stopAnimating()
    UIApplication.shared.endIgnoringInteractionEvents()

    if response == nil
    {
        print("ERROR")
    }
    else
    {
        //Remove annotations
        let annotations = self.myMapView.annotations
        self.myMapView.removeAnnotations(annotations)

        //Getting data
        let latitude = response?.boundingRegion.center.latitude
        let longitude = response?.boundingRegion.center.longitude

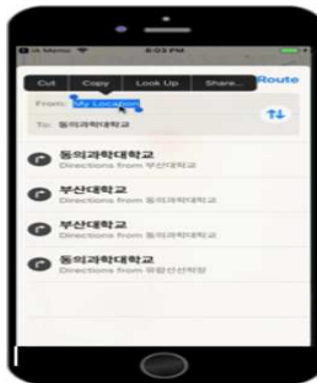
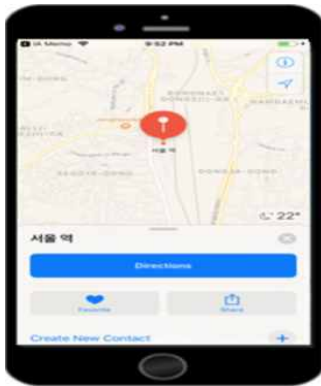
        //Create annotation
        let annotation = MKPointAnnotation()
        annotation.title = searchBar.text
        annotation.coordinate = CLLocationCoordinate2DMake(latitude!, longitude!)
        self.myMapView.addAnnotation(annotation)

        //Zooming in on annotation
        let coordinate:CLLocationCoordinate2D = CLLocationCoordinate2DMake(latitude!, longitude!)
        let span = MKCoordinateSpan(latitudeDelta: 0.1, longitudeDelta: 0.1)
        let region = MKCoordinateRegion(center: coordinate, span: span)
        self.myMapView.setRegion(region, animated: true)
    }
}

```

- \* endIgnoringInteractionEvents() : 화면 전체에 대한 터치 이벤트 풀기 설정.
- \* beginIgnoringInteractionEvents() : 화면 전체에 대한 터치 이벤트 막기 설정.
- \* annotation : 핀을 꽂아 장소를 설정하는 것.

- UISearchController



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

    // ----- 네비게이션 코드 -----
    // 목적지 찾기 서울역 위도 경도 : 37.553816, 126.969631
    let latitude:CLLocationDegrees = 37.553816 // 위도
    let longitude:CLLocationDegrees = 126.969631 //경도

    let regionDistance:CLLocationDistance = 1000;
    let coordinates = CLLocationCoordinate2DMake(latitude, longitude) // 위에 좌표(위도, 경도)를 Combine 한다.
    let regionSpan = MKCoordinateRegion(center: coordinates, latitudinalMeters: regionDistance,
                                         longitudinalMeters: regionDistance) // 지역 범위 지정 (중심 좌표, 지역 범위 지정, 거리 지정)

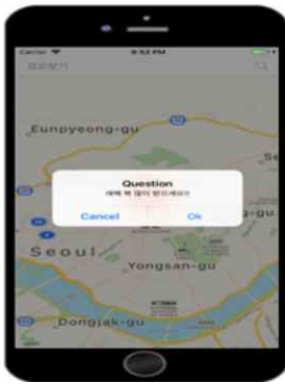
    let options = [MKLaunchOptionsMapCenterKey: NSValue(mkCoordinate: regionSpan.center),
                   MKLaunchOptionsMapSpanKey: NSValue(mkCoordinateSpan: regionSpan.span)]
    // 기능 = 좌표값 지역 범위 중앙 설정

    //어떻게 핀을 고정 시킬지 나타내는 코드
    let placemark = MKPlacemark(coordinate: coordinates) //핀을 놓아둘 곳을 (좌표값) 설정
    let mapItem = MKMapItem(placemark: placemark) // placemark를 mapitem이라 지정하고
    mapItem.name = "서울역" // placemark(핀)을 지정한 위치의 이름을 "서울역"으로 지정
    mapItem.openInMaps(launchOptions: options) // 위에서 정의한 기능을 맵에다 지정.
}
```

1. 목적지를 검색하면 나오는 화면으로 해당 목적지에 핀 마크가 찍힙니다.
2. 초깃값 설정을 위해 서울의 위도, 경도를 기준으로 했습니다.
3. Direction 버튼을 클릭하면 해당 목적지의 경로를 탐색합니다.
4. 실제 네비와 같은 기능으로 차, 걷기, 교통 별 경로 와 시간을 탐색할 수 있습니다.



## - Alert



1. Alert를 활용하여 타이틀, 메시지, 액션 3단계로 구분하여 해당 Alert 화면 생성.
2. UIAlertController : UIAlertController 클래스는 사용자에게 표시할 얼럿 또는 액션시트의 구성에 관한 메서드와 프로퍼티를 포함하고 있다. UIAlertController 클래스를 통해 얼럿 또는 액션시트를 구성한 후 UIViewController의 present(\_:animated:completion:) 메서드를 사용하여 사용자에게 얼럿 또는 액션시트를 모달로 보여줄수 있습니다.

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

    // 1. create a gesture recognizer (tap gesture)
    // 제스처 인식기 만들기
    let tapGesture = UITapGestureRecognizer(target: self, action: #selector(handleTap(sender:)))

    // 2. add the gesture recognizer to a view
    // 뷰에 인식기 추가
    myMapView.addGestureRecognizer(tapGesture)
}

// 3. this method is called when a tap is recognized
// 터치 했을때 호출한다.
@objc func handleTap(sender: UITapGestureRecognizer) {
    print("tap")
    questionAlert()

    func questionAlert() {
        let alert = UIAlertController(title: "Question", message: "새해 복 많이 받으세요!!", preferredStyle: UIAlertController.Style.alert)

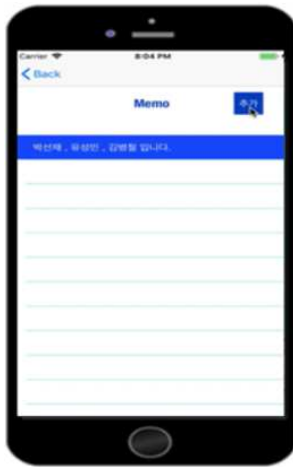
        let cancel = UIAlertAction(title: "Cancel", style: UIAlertAction.Style.cancel, handler: nil)

        alert.addAction(cancel)

        // Create OK action
        let ok = UIAlertAction(title: "Ok", style: UIAlertAction.Style.default) { (action: UIAlertAction) -> Void in
            self.performSegue(withIdentifier: "show", sender: nil)
        }
        alert.addAction(ok)

        // Present Alert Controller
        self.present(alert, animated: true, completion: nil)
    }
}
```

## - 메모장



1. 메모장 MAIN UI 화면입니다.
2. 추가 버튼 클릭시 메모를 작성할 수 있는 UI 화면으로 전환합니다.  
\* UserDefaults.standard.object(forKey: "MemoNumber") : 키 값에 해당하는 값을 기록하고, 읽거나 쓸 수 있는 기능을 하는 메소드입니다.
3. 메모장의 테이블 기준으로 메모 작성, 입력하고 첫 번째 테이블의 number가 -1 이 됨과 동시에 -1 로 사라진 테이블에 저장됩니다.
4. Delete Function : 메모 삭제 버튼 클릭 시 발생하는 이벤트.
5. func save : 코어 저장소에 저장될 이벤트

```
@IBOutlet weak var RecordTextView: UITextView!
var MemoData = [String]()
// Variable_End

override func viewDidLoad() {
    super.viewDidLoad()

    let MemoNumber = UserDefaults.standard.object(forKey: "MemoNumber") as! Int

    if MemoNumber == -1 {

        MemoData = UserDefaults.standard.object(forKey: "MemoData") as! [String]

        RecordTextView.text = "입력하세요"
    }else{

        MemoData = UserDefaults.standard.object(forKey: "MemoData") as! [String]
        RecordTextView.text = MemoData[MemoNumber]
    }

}
```

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

    let MemoNumber = UserDefaults.standard.object(forKey: "MemoNumber") as! Int

    if MemoNumber != -1 {

        MemoData.remove(at: MemoNumber)
        UserDefaults.standard.set(MemoData, forKey: "MemoData")

    }

}
// Action_End
```

```

@IBAction func save(_ sender: Any) {
    let MemoNumber = UserDefaults.standard.object(forKey: "MemoNumber") as! Int

    if MemoNumber == -1 {

        MemoData.insert(RecordTextView.text , at:0)

        UserDefaults.standard.set(MemoData, forKey: "MemoData")

    }else{

        MemoData.remove(at: MemoNumber)
        MemoData.insert(RecordTextView.text, at:MemoNumber)

        UserDefaults.standard.set(MemoData, forKey: "MemoData")

    }
}

```

#### 4. 소 감

1학년 2학기에 군대 제대 후 막 복학 했을 시점에 교내 컴퓨터학과에서 열리는 Boot-Camp 아이디어 공모전에 참가하게 되었습니다. 막 복학한 시점에 무엇을 할 수 있는지 의문이 먼저 들었기에 본선에 불을 줄은 꿈에도 생각하지 못했습니다. 사실 예선전에서는 아이디어 싸움이었기 때문에 가벼운 마음으로 선정한 주제였기에 직접 개발에 들어가야 하는 본선에서는 실제 구현이 가능할지 조차 의구심이 들 정도였기 때문에 막막했습니다. 처음에는 교수님께 찾아가 중도 포기를 할까 생각도 해보았지만, 그냥 하는데 까지 해보자라는 마인드로 개발을 진행했습니다. 물론 결과물 자체는 완성도가 미흡했고, 실제 구현하고자 상상했던것과는 비교도 되지 않을 만큼 거리감이 있지만, Swift 라는 IOS 프로그래밍을 경험 할 수 있었고, BootCamp를 통해서 학생들에게 무엇을 요구하는지 깨닫게 되었고, 한층 더 성장할 수 있는 계기였습니다.

## REFERENCE

<https://zeddios.tistory.com/111> Alert 사용법

<https://m.blog.naver.com/jdub7138/220393890771> 화면전환 사용법

<https://devxoul.gitbooks.io/ios-with-swift-in-40-hours/Chapter-1/> 스위프트 기본 정보

<https://openweathermap.org> 오픈 날씨 앱 사이트

<UISearchController(searchResultsController: nil) >

<https://devmjun.github.io/archive/SearchController>

<http://blog.naver.com/PostView.nhn?blogId=scw0531&logNo=220919151737&parentCategoryNo=&categoryNo=23&viewDate=&isShowPopularPosts=false&from=postView> - <Search Bar.Delegate =self>

<https://m.blog.naver.com/jdub7138/220393890771> <Segue>

<https://developer.apple.com/documentation/uikit/uiapplication/1622938-endignoringinteractionevents> - <beginIgnoringInteractionEvents()>

<https://www.youtube.com/watch?v=INfCmCxIC0o> Navigation 기능

<https://www.youtube.com/watch?v=GYzNsVFyDru> google map

<https://www.youtube.com/watch?v=goA3kYLP3dg> 메모장 기능

<https://developer.apple.com/documentation/uikit/uiapplication/1622938-endignoringinteractionevents> - <beginIgnoringInteractionEvents()>

<http://blog.naver.com/PostView.nhn?blogId=zll11&logNo=220744646888&categoryNo=13&parentCategoryNo=0&viewDate=&currentPage=1&postListTopCurrentPage=1&from=postView>  
- < annotation.coordinate = CLLocationCoordinate2D>

<https://developer.apple.com/documentation/mapkit/mkmapview/1451889-addannotations?language=objc>  
c - <annotations>

<https://translate.google.com/translate?hl=ko&sl=en&u=https://www.mathworks.com/help/matlab/2-d-and-3-d-bounding-regions.html&prev> - <boundary>

<https://translate.google.com/translate?hl=ko&sl=en&u=https://stackoverflow.com/questions/6739505/creating-a-new-CLLocationCoordinate2D-at-custom-latitude-longitude&prev=search> - <CLLocationCoordinate2DMake>

<https://translate.google.com/translate?hl=ko&sl=en&u=https://developer.xamarin.com/api/type/MonoTouch.MapKit.MKCoordinateSpan/&prev=search> - <MKCoordinateSpan>

<https://developer.apple.com/documentation/mapkit/mkcoordinateregion> - <MKCoordinate 지역>

<https://translate.google.com/translate?hl=ko&sl=en&u=https://stackoverflow.com/questions/27751924/casting-CLLocationDistance-in-swift&prev=search> - <CLLocationDistance = Double>

<https://translate.google.com/translate?hl=ko&sl=en&u=https://stackoverflow.com/questions/30211946/change-CLLocationDegrees-into-a-double-nsnumber-to-save-in-core-data-swift&prev=search> - <CLLocationDegrees>

<https://translate.google.com/translate?hl=ko&sl=en&u=https://stackoverflow.com/questions/31445892/latitude-and-longitude-from-mkmapitem-in-swift&prev=search> - <mapKitItem>

<http://seorenn.blogspot.com/2017/01/userdefaults-nsuserdefaults.html>  
- <UserDefaults>

