

모바일프로그래밍 숙제2# 최종보고서 (월요일반)

SEOUL PARK

20121332 원예생명조경 나희재

CONTENTS

- 1. 개발자 소개
- 2. 앱 개발동기 및 소개
- 3. 앱의 기능
- 4. 주요 코드
- 5. 데이터베이스
- 6. 한 학기를 마무리하며

1. 개발자 소개



- 이름 : 나희재
- 학번 : 20121332
- 전공 : 원예생명조경학과
- 이메일 주소 : nahejae533@naver.com

2. 앱 개발 동기 및 소개

■ 앱 개발 동기

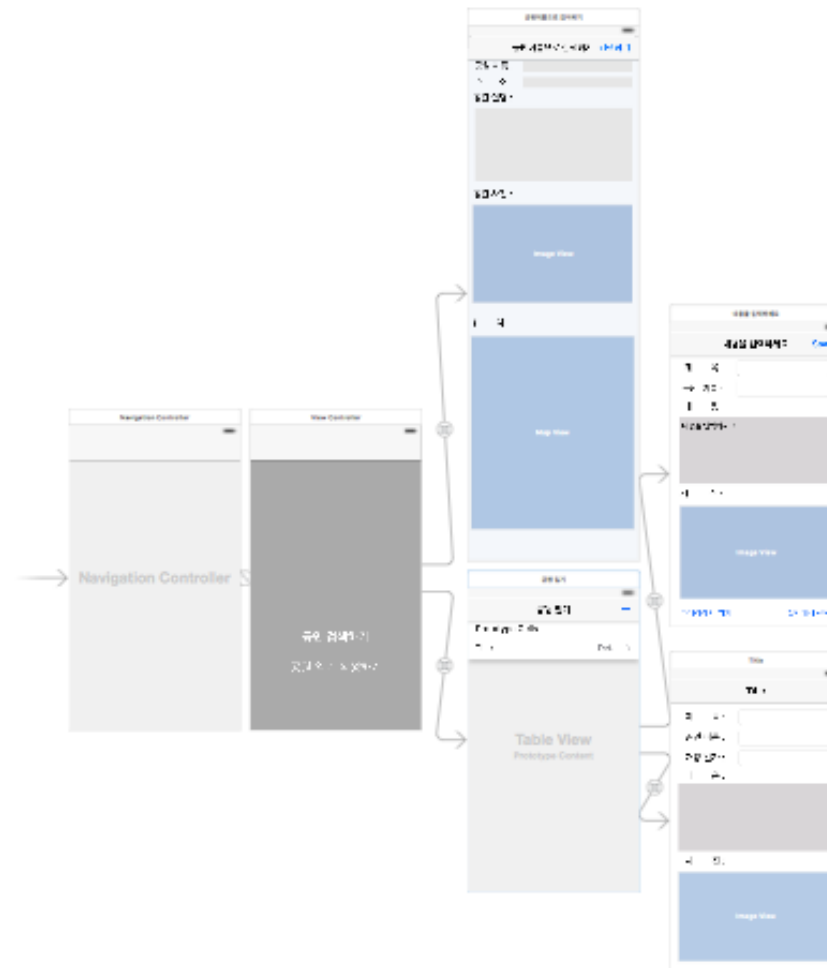
원예생명조경 전공과 관련된 앱을 개발하고 싶었습니다.
그 중에서 공원에 대한 정보를 제공해주는 앱이 있으면 좋다고 생각했습니다.
필요한 공원에 대해서 관련된 정보를 더 얻을 수 있으며,
공원에 대해서 애정을 가질 수 있도록 해주고 싶었습니다.

■ 앱 소개

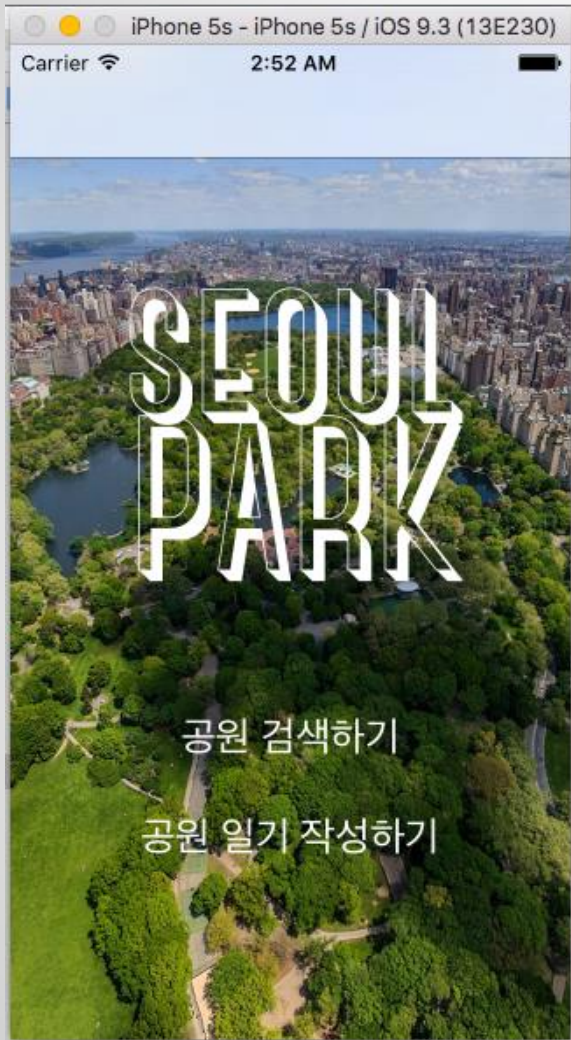
앱은 공공데이터를 통해서 공원에 대한 정보를 얻어오며,
일기장 기능을 추가하여서 자신이 경험했던 공원에 대해서 적고 저장할 수 있습니다.

3. 앱의 기능

■ 전체 Layout



3. 앱의 기능



시작화면

■ 앱의 기능

1 공원 검색하기

서울에 있는 공원을 검색할 수 있다.

2 공원 일기 작성하기

공원에 대한 일기를 작성할 수 있다.

3. 앱의 기능

1) 공원 검색하기

서울에 있는 공원을 검색할 수 있다.



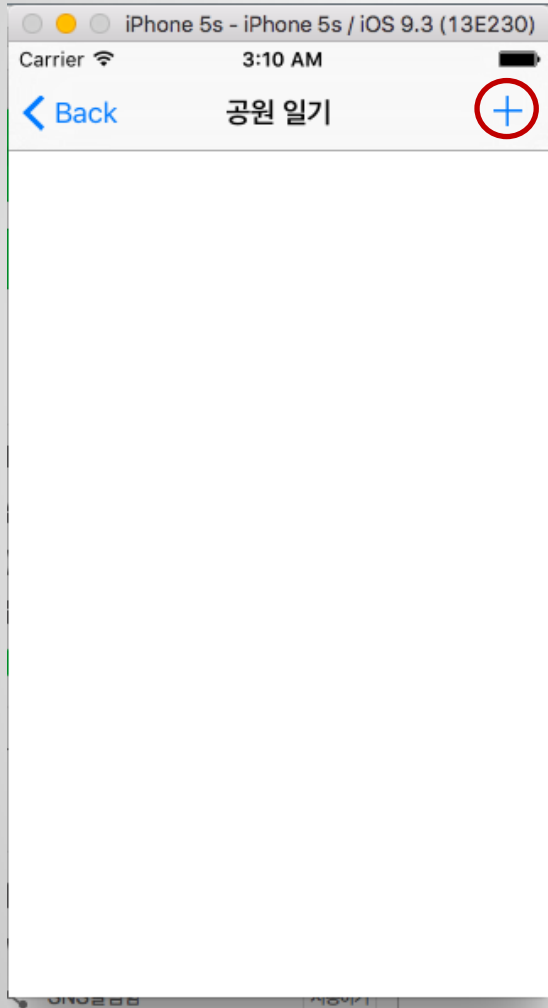
공원이름을 검색을 해서
공원이름, 주소, 공원 사진,
공원설명,
지도정보가 나오게 된다.



3. 앱의 기능

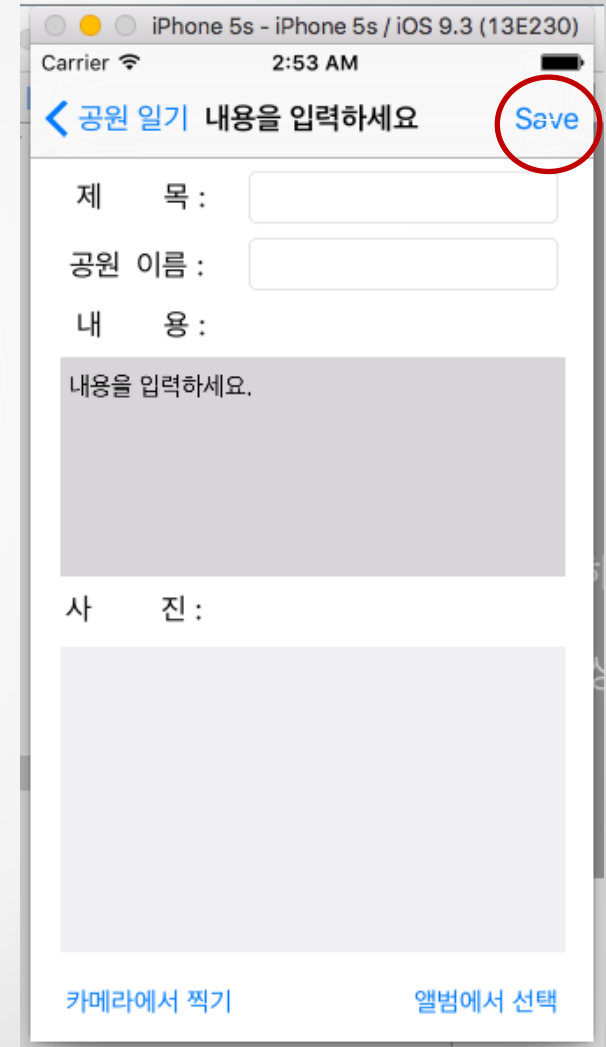
2) 공원 일기 작성하기

공원에 대한 일기를 작성할 수 있다.



시작화면에서 '공원 일기 작성하기' 버튼을 누르면 tableview로 저장된 일기의 리스트가 나온다.

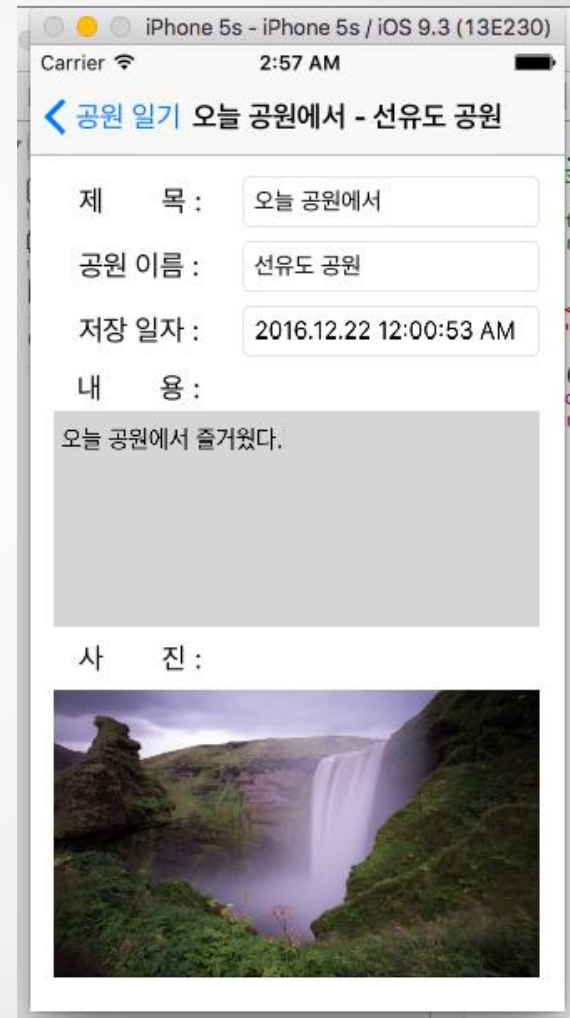
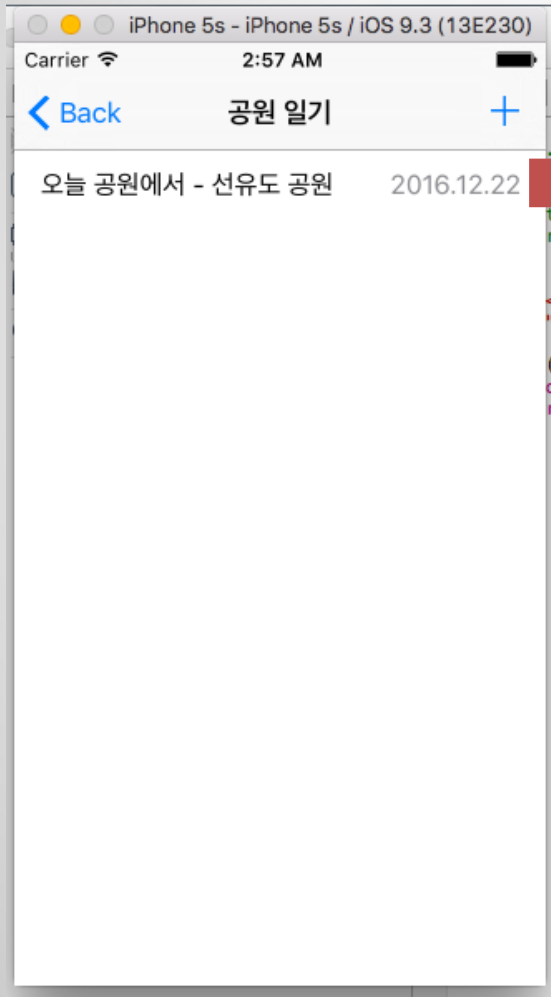
오른쪽 위의 '+' 버튼을 눌러서 제목, 공원이름, 내용, 사진 정보들을 추가할 수 있다.



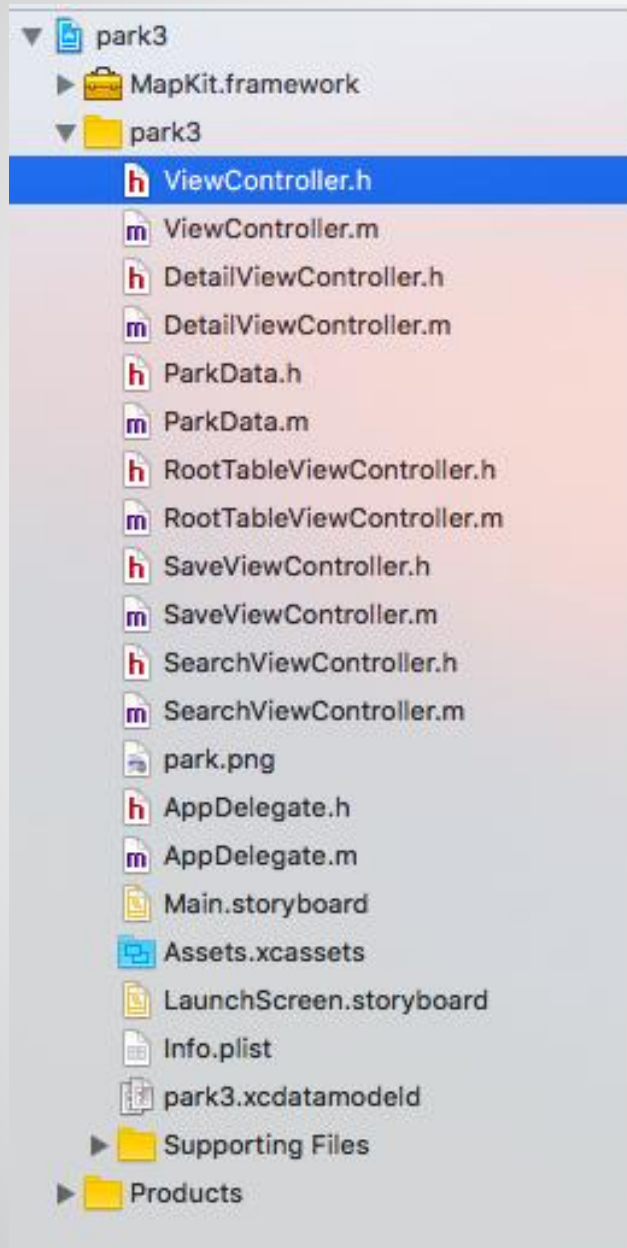
3. 앱의 기능

2) 공원 일기 작성하기

공원에 대한 일기를 작성할 수 있다.



4. 주요 코드



1. 초기화면

ViewController.h
ViewController.m

2. 공원 검색

SearchViewController.h
SearchViewController.m
ParkData.h
ParkData.m

3. 공원 일기

SaveViewController.h
SaveViewController.m
RootTableViewController.h
RootTableViewController.m
DetailViewController.h
DetailViewController.m

4. 주요 코드

1) 초기화면

ViewController.m

```
//  
// ViewController.m  
// park2  
//  
// Created by SWUComputer on 2016. 12. 20..  
// Copyright © 2016년 SWUComputer. All rights reserved.  
//  
  
#import "ViewController.h"  
  
@interface ViewController ()  
  
@end  
  
@implementation ViewController  
  
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
    UIImage *img=[UIImage imageNamed:@"park.png"];  
    UIImage *scaleImage=[UIImage imageWithCGImage:[img CGImage]  
                                     scale:(img.scale *2.0) orientation:(img.imageOrientation)];  
  
    self.view.backgroundColor = [UIColor colorWithPatternImage:scaleImage];  
  
}  
  
- (void)didReceiveMemoryWarning {  
    [super didReceiveMemoryWarning];  
    // Dispose of any resources that can be recreated.  
}  
  
@end
```

초기화면에 이미지를
넣기 위한 코드

4. 주요 코드

2) 공원 검색

SearchViewController.h



```
//  
// SearchViewController.h  
// park2  
//  
// Created by SWUCOMPUTER on 2016. 12. 20..  
// Copyright © 2016년 SWUCOMPUTER. All rights reserved.  
//  
  
#import <UIKit/UIKit.h>  
#import MapKit;  
  
@interface SearchViewController : UIViewController{  
    NSDictionary *receiveDic;  
    IBOutlet UIScrollView *scroller;  
}  
  
@property (strong, nonatomic) NSString *la;  
@property (strong, nonatomic) NSString *lo;  
  
@property (strong, nonatomic) IBOutlet UILabel *info;  
@property (strong, nonatomic) IBOutlet UILabel *address;  
@property (strong, nonatomic) IBOutlet UITextView *pdetail;  
  
@property (strong, nonatomic) IBOutlet UISearchBar *searchBar;  
@property (strong, nonatomic) IBOutlet MKMapView *map;  
  
@property (strong, nonatomic) IBOutlet UIImageView *image1;  
  
- (IBAction)searchPressed:(UIBarButtonItem *)sender;  
  
@end
```

ViewController에 들어갈 내용이 많기 때문에 scrollView를 이용한다.

| 4. 주요 코드

2) 공원 검색



오픈데이터

데이터서비스

참여 · 소통

데이터셋

열린데이터를 검색하세요



서울의공원 공원명별 공원정보 조회

★★★★★ : 0(참여 - 0) |

***** ▼

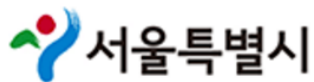
평가

♡ 찜하기

활용갤러리등록

상세정보

목록보기



제공기관 : 서울특별시

공원명으로 공원정보를 검색하는 API
로 공원번호, 공원명, 공원에 대한 선

분류체계

환경>공원녹지

원본시스템

서울의공원

저작권자명

서울특별시

제공기관

서울특별시

<http://data.seoul.go.kr/openinf/openapiview.jsp?infd=OA-211&tMenu=11>
에서 키 발급

4. 주요 코드

2) 공원 검색

SearchViewController.m

```
- (IBAction)searchPressed:(UIBarButtonItem *)sender {
```

```
    NSString *p_park=@"";  
    p_park=searchBar.text;  
    NSString* p_park1 = [p_park stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
```

검색단어를 인코딩해준다.

```
    NSString *url1=[NSString stringWithFormat:  
        @"http://openapi.seoul.go.kr:8088/714f484b6668656536324373514571/json/SearchInfoByParkNameService/1/5/%@", p_park1];  
    NSURL *url=[NSURL URLWithString:url1];
```

```
    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] init];  
    [request setURL:url];  
    //[request setHTTPBody:p_park1];  
    [request setHTTPMethod:@"GET"];
```

공공데이터 사용을 위해 가져온
openapi key를 이용해서 p_park1이라는
검색결과를 찾는다.

```
    NSError *error = [[NSError alloc] init];  
    NSHTTPURLResponse *response = nil;
```

```
    NSData *urlData= [NSURLConnection sendSynchronousRequest:request returningResponse:&response error:&error];
```

```
    //NSString *responseString=[[NSString alloc] initWithBytes:[urlData bytes] length:[urlData length] encoding:NSUTF8StringEncoding];
```

```
    receiveDic = [NSJSONSerialization JSONObjectWithData:urlData  
        options:NSJSONReadingMutableContainers  
        error:nil];
```

```
    NSDictionary *aa=[receiveDic objectForKey:@"SearchInfoByParkNameService"];  
    NSArray *result01=aa[@"row"];
```

4. 주요 코드

2) 공원 검색

SearchViewController.m

파싱해온 데이터를 처리해준다.

```
for(int i=0;i<1;i++){  
  
    NSDictionary *jsonElement=result01[i];  
    // Create a new FavoriteData object and set its props to JsonElement properties
```

```
    ParkData *newData = [[ParkData alloc] init];  
    newData.P_IDX = jsonElement[@"P_IDX"];  
    newData.P_PARK = jsonElement[@"P_PARK"];  
    newData.P_LIST_CONTENT = jsonElement[@"P_LIST_CONTENT"];  
    newData.P_ADDR = jsonElement[@"P_ADDR"];  
    newData.P_ZONE = jsonElement[@"P_ZONE"];  
    newData.P_DIVISION = jsonElement[@"P_DIVISION"];  
    newData.P_IMG = jsonElement[@"P_IMG"];  
    newData.P_ADMINTEL = jsonElement[@"P_ADMINTEL"];  
    newData.LONGITUDE = jsonElement[@"LONGITUDE"];  
    newData.LATITUDE = jsonElement[@"LATITUDE"];  
    newData.G_LATITUDE = jsonElement[@"G_LATITUDE"];  
    newData.G_LONGITUDE=jsonElement[@"G_LONGITUDE"];
```

```
    // Add this question to the FavoriteData array  
    //[tempArray addObject:newData];  
    info.text=newData.P_PARK;  
    address.text=newData.P_ADDR;  
    pdetail.text=newData.P_LIST_CONTENT;  
    la=newData.LATITUDE;  
    lo=newData.LONGITUDE;
```

```
    NSURL *url=[[NSURL alloc] initWithString:newData.P_IMG];  
    self.image1.image=[UIImage imageWithData:[NSData dataWithContentsOfURL:url]];  
    MKCoordinateRegion region;  
    region.center.latitude=[la doubleValue];  
    region.center.longitude=[lo doubleValue];  
    region.span.latitudeDelta = 0.005;  
    region.span.longitudeDelta = 0.005;  
    [self.map setRegion:region animated:YES];  
}
```

```
//  
// ParkData.h  
// park2  
//  
// Created by SWUComputer on 2016. 12. 20..  
// Copyright © 2016년 SWUComputer. All rights reserved.  
//  
#import <Foundation/Foundation.h>  
  
@interface ParkData : NSObject  
  
@property (nonatomic, strong) NSString *P_IDX;  
@property (nonatomic, strong) NSString *P_PARK;  
@property (nonatomic, strong) NSString *P_LIST_CONTENT;  
@property (nonatomic, strong) NSString *P_ADDR;  
@property (nonatomic, strong) NSString *P_ZONE;  
@property (nonatomic, strong) NSString *P_DIVISION;  
@property (nonatomic, strong) NSString *P_IMG;  
@property (nonatomic, strong) NSString *P_ADMINTEL;  
@property (nonatomic, strong) NSString *LONGITUDE;  
@property (nonatomic, strong) NSString *LATITUDE;  
@property (nonatomic, strong) NSString *G_LONGITUDE;  
@property (nonatomic, strong) NSString *G_LATITUDE;  
  
@end
```

ParkData.h

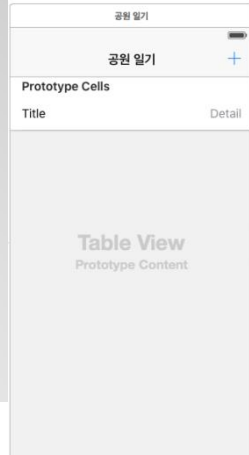
이미지 설정

지도에 좌표를 설정

4. 주요 코드

3) 공원 일기

RootTableViewController.m



```
@implementation RootTableViewController
@synthesize diarys;

//managed object context를 조회하거나 나중에 자료를 저장하기 위한 다양한 용도에 쓰임
-(NSManagedObjectContext *)managedObjectContext{
    NSManagedObjectContext *context=nil;
    id delegate=[[UIApplication sharedApplication]delegate];
    if([delegate performSelector:@selector(managedObjectContext))]{
        context=[delegate managedObjectContext];
    }
    return context;
}

//view가 보여줄때 자물르 db에서 가져오도록 한다
-(void)viewDidAppear:(BOOL)animated{
    [super viewDidAppear:animated];

    //persistent data store
    NSManagedObjectContext *moc =[self managedObjectContext];
    NSFetchRequest *fetchRequest=[[NSFetchRequest alloc]initWithEntityName:@"Diary"];

    //edit the sort key as appropriate
    NSSortDescriptor *sortDescriptor=[[NSSortDescriptor alloc]initWithKey:@"title" ascending:YES];
    NSArray *sortDescriptors=@[sortDescriptor];
    [fetchRequest setSortDescriptors:sortDescriptors];

    diarys=[[moc executeFetchRequest:fetchRequest error:nil]mutableCopy];
    [self.tableView reloadData];
}
```


4. 주요 코드

3) 공원 일기

RootTableViewController.m

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView {
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSectionSection:(NSInteger)section {
    return diarys.count;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath {
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"Diary Cell" forIndexPath:indexPath];

    // Configure the cell...
    //테이블 화면에 나올 내용
    NSMutableArray *diary=[diarys objectAtIndex:indexPath.row];
    [cell.textLabel setText:[NSString stringWithFormat:@"%d - %@",[diary valueForKey:@"title"],[diary valueForKey:@"park_name"]]];

    NSDateFormatter *formatter=[[NSDateFormatter alloc] init];
    [formatter setDateFormat:@"%yy.MM.dd"];

    [cell.detailTextLabel setText:[formatter stringFromDate:[diary valueForKey:@"date"]]];

    return cell;
}

// Override to support conditional editing of the table view.
- (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)indexPath {
    // Return NO if you do not want the specified item to be editable.
    return YES;
}

// Override to support editing the table view.
- (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyle forRowAtIndexPath:(NSIndexPath *)indexPath {
    if (editingStyle == UITableViewCellEditingStyleDelete) {
        // Delete the row from the data source
        NSMutableArray *context=[self managedObjectContext];
        [context deleteObject:[diarys objectAtIndex:indexPath.row]];

        NSError *error=nil;

        if(![context save:&error]){
            NSLog(@"Save Failed! %@",error,[error localizedDescription]);
        }
        //remove diary from table view
        [diarys removeObjectAtIndex:indexPath.row];

        [tableView deleteRowsAtIndexPaths:@[indexPath] withRowAnimation:UITableViewRowAnimationFade];
    } else if (editingStyle == UITableViewCellEditingStyleInsert) {
        // Create a new instance of the appropriate class, insert it into the array, and add a new row to the table view
    }
}
```

4. 주요 코드

3) 공원 일기

SaveViewController.m

```
#import <UIKit/UIKit.h>

@interface SaveViewController : UIViewController <UITextFieldDelegate,
UIImagePickerControllerDelegate, UINavigationControllerDelegate>

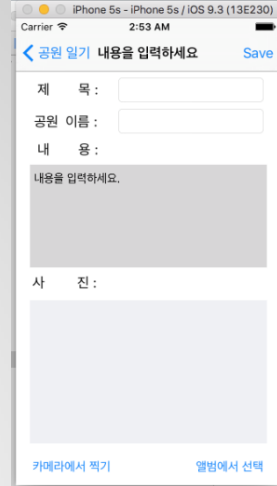
@property (strong, nonatomic) IBOutlet UITextField *textTitle;
@property (strong, nonatomic) IBOutlet UITextField *textPName;
@property (strong, nonatomic) IBOutlet UITextView *textContent;
@property (strong, nonatomic) IBOutlet UIImageView *imageView;

@property (nonatomic, retain) NSData *img;

- (IBAction)savePressed:(UIBarButtonItem *)sender;
- (BOOL)textFieldShouldReturn:(UITextField *)textField;

- (IBAction)takeButton:(UIButton *)sender;
- (IBAction)selectButton:(UIButton *)sender;

@end
```



4. 주요 코드

3) 공원 일기

SaveViewController.m

```
- (void)viewDidLoad {
    [super viewDidLoad];

    if (![UIImagePickerController
        isSourceTypeAvailable:UIImagePickerControllerSourceTypeCamera]) {

        UIAlertController *alert= [UIAlertController alertControllerWithTitle:@"Error!"
                                                                            message:@"Device has no camera"
                                                                            preferredStyle:UIAlertControllerStyleAlert];

        UIAlertAction *ok=
        [UIAlertAction
         initWithTitle:@"OK" style:UIAlertActionStyleDefault
         handler:^(UIAlertAction *action){
             [alert dismissViewControllerAnimated:YES completion:nil];
         }];
        [alert addAction:ok];
        [self presentViewController:alert animated:YES completion:nil];
    }
}
```

카메라 사용을 위해 처리를 해준다.

4. 주요 코드

3) 공원 일기

SaveViewController.m

```
- (IBAction)takeButton:(UIButton *)sender {  
    UIImagePickerController *picker = [[UIImagePickerController alloc] init]; picker.delegate = self;  
    picker.allowsEditing = YES;  
    picker.sourceType = UIImagePickerControllerSourceTypeCamera;  
    [self presentViewController:picker animated:YES completion:NULL];  
}  
  
- (IBAction)selectButton:(UIButton *)sender {  
    UIImagePickerController *picker = [[UIImagePickerController alloc] init]; picker.delegate = self;  
    picker.allowsEditing = YES;  
    picker.sourceType = UIImagePickerControllerSourceTypePhotoLibrary;  
    [self presentViewController:picker animated:YES completion:NULL];  
}  
  
- (void) imagePickerController:(UIImagePickerController *)picker didFinishPickingMediaWithInfo:(NSDictionary *)info {  
    //UIImage *chosenImage = [info objectForKey:UIImagePickerControllerEditedImage];  
    UIImage *chosenImage = info[UIImagePickerControllerEditedImage];  
    self.imageView.image = chosenImage;  
    [picker dismissViewControllerAnimated:YES completion:NULL];  
}  
  
- (void) imagePickerControllerDidCancel:(UIImagePickerController *)picker {  
    [picker dismissViewControllerAnimated:YES completion:NULL];  
}  
  
@end
```

카메라 사용을 위해 처리를 해준다.

4. 주요 코드

3) 공원 일기

SaveViewController.m

```
- (IBAction)savePressed:(UIBarButtonItem *)sender {
    NSManagedObjectContext *context = nil;
    id delegate = [[UIApplication sharedApplication] delegate];
    if ([delegate performSelector:@selector(managedObjectContext)]) {
        context = [delegate managedObjectContext];
    }

    //diary record 생성
    NSManagedObject *newDiary = [NSEntityDescription insertNewObjectForEntityForName:@"Diary"
                                                                    inManagedObjectContext:context];

    UIImage *image=self.imageView.image;
    //NSString *imageRoute;
    NSData *imageData = UIImageJPEGRepresentation(image, 0.5);

    [newDiary setValue:textTitle.text forKey:@"title"];
    [newDiary setValue:textPName.text forKey:@"park_name"];
    [newDiary setValue:textContent.text forKey:@"content"];
    [newDiary setValue:[NSDate date] forKey:@"date"];
    [newDiary setValue:imageData forKey:@"imageData"];

    NSError *error=nil;

    if(![context save:&error]){
        NSLog(@"Save Failed! %@ %@",error,[error localizedDescription]);
    }
    [self.navigationController popViewControllerAnimated:YES];
}
```

이미지 저장을 위해
데이터베이스에서 사진이 Binary data로
처리해준다.

4. 주요 코드

3) 공원 일기

DetailViewController.m

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view.
```

데이터베이스에서 값을 가지고 온다.

```
if(detailDiary){  
    textTitle.text=[detailDiary valueForKey:@"title"];  
    textPName.text=[detailDiary valueForKey:@"park_name"];  
    textContent.text=[detailDiary valueForKey:@"content"];  
    UIImage *image1=[UIImage imageData:[detailDiary valueForKey:@"imageData"]];  
    imagee.image=image1;  
  
    NSDateFormatter *formatter=[[NSDateFormatter alloc] init];  
    [formatter setDateFormat:@"yyyy.MM.dd h:mm:ss a"];  
  
    saveDate.text=[formatter stringFromDate:[detailDiary valueForKey:@"date"]];  
}
```

```
}  
|
```

@end



5. 데이터베이스

■ 데이터베이스 소개













- Entity : 'Diary' - 일기를 저장할 수 있는 개체이다.
- 'Diary' attribute : 'title', 'park_name', 'content', 'imageData', 'date'

일기에 저장할 속성들로 제목(title), 공원이름(park_name), 내용(content), 사진정보(imageData), 저장날짜(date)가 저장이된다.

title, park_name, content : **String type**

date : **Date type**,

imageData : **Binary Data type**

ENTITIES	▼ Attributes		
 Diary	Attribute ^	Type	
FETCH REQUESTS	 content	String	
	 date	Date	
CONFIGURATIONS	 imageData	Binary Data	
	 park_name	String	
	 title	String	
 Default	+ —		

6. 한 학기를 마무리 하며

한 학기 동안 아이폰 프로그래밍에 대해서 정말 많은 것을 배울 수 있는 유익한 시간이었습니다. 맥북도 사용해볼 수 있어서 좋았습니다.

매주 실습을 하면서 실력을 쌓을 수 있어서 실습체계를 그대로 유지하는 것이 좋을 것 같습니다.

강의가 뒤로 갈수록 내용이 어려워져서 100% 이용할 수 있을 정도로 코드를 이해하지 못했던 것 같습니다. 수업 주제 분배가 어려운 부분이 앞으로 와도 좋을 것 같습니다.

숙제로 앱을 만드는 것을 첫 번째 앱과 두 번째 앱을 나눈 것이 아니라 첫 번째 앱을 발전시키거나 시간 여유가 더 있으면 더 유익하고 좋은 앱이 탄생할 수 있을 것 같습니다.

모바일 앱을 만들 때 많이 사용하게 되는 대부분의 주제들을 배울 수 있었습니다. 지금 배우는 것 중에서 빠질 만한 것이 없는 것 같습니다.

한 학기동안 감사했습니다.