

모 바 일 프 로 그 래 밍

모바일프로그래밍 숙제 #1 - 보고서

2014111514 컴퓨터학과 권민하

INDEX

01 개발자 소개

02 주제 및 개발동기

03 기본 기능 설명

04 중요 소스 코드 설명

Introduction

개발자 소개



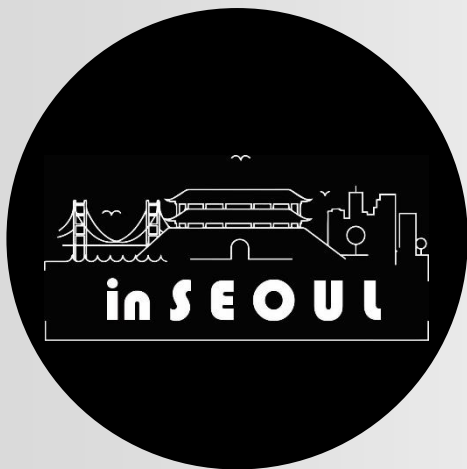
✓ 권민하

✓ 2014111514

✓ 컴퓨터학과

✓ minha61@naver.com

주제 & 앱 소개



in SEOUL

서울 안의 모든 것을 의미할 수 있는 **in SEOUL !**

서울을 여행하는, 데이트 하는, 즐기는 모두를 위해

서울에서 손에 꼽히는 명소들의 정보 및 사진들을 제공하는 어플리케이션입니다.

가고 싶은 곳에 대해 검색하거나 선택할 수 있으며 랜덤 선택 서비스를 제공합니다.

Motive for Development

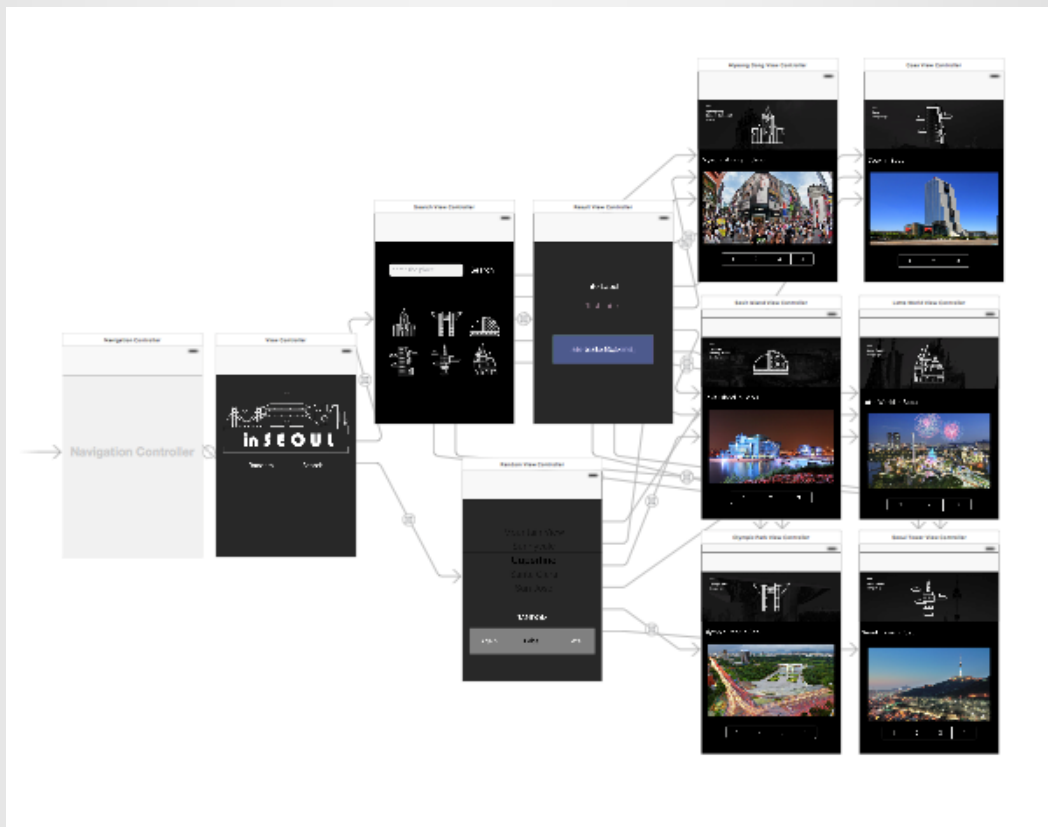
개발 동기

“

내 고향은 경상남도 창원이라 서울에 오거나 구경할 기회는 매우 희박했다.
그래서 20살, 대학생이 되어서야 한국의 수도라는 곳을 제대로 알게 되었고
지금까지 서울에서 꼭 가봐야 하는 곳들을 마치 여행하듯이 가보곤 한다.
이러한 이유로 나와 비슷하거나, 여행객들, 물론 서울에 이미 사는 사람들에게도
서울의 아름다움, 추천하는 곳들을 모아서 보여주고 싶었다.
그래서 이 앱을 개발하게 되었다.

”

Storyboard



기본 기능 설명

메인 화면



☒ Random 버튼 : 가고 싶은 장소(명소)를 선택하지 못하였을 경우
기존에 등록된 장소들 중 하나를 랜덤하게 선택
하기 위한 Scene으로 이동하는 버튼

☒ Search 버튼 : 가고 싶은 장소를 검색하거나 선택하기 위한
Scene으로 이동하는 버튼

메인 화면 : ViewController.m

```
#import "ViewController.h"
#import "RandomViewController.h"
#import "SearchViewController.h"

@interface ViewController ()
@end

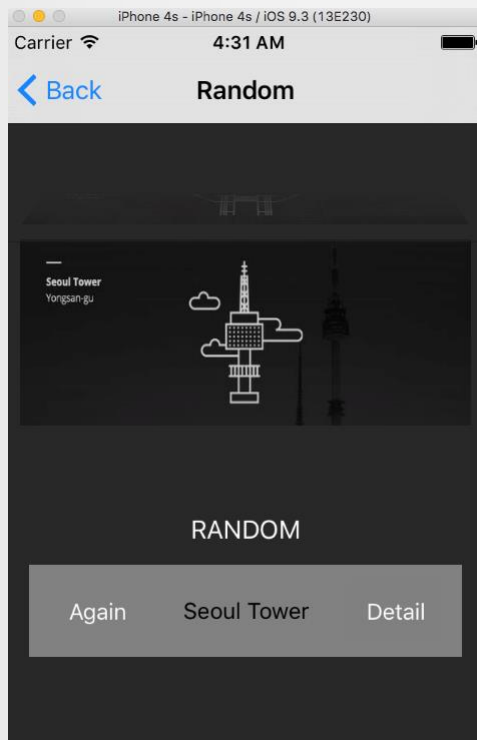
@implementation ViewController
@synthesize mainImageView, randomButton, searchButton;

- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    // Random View로 갈 때
    if([segue.identifier isEqualToString:@"toRandomView"]) {
        RandomViewController *rc = [segue destinationViewController];
        rc.title = randomButton.titleLabel.text; // 네비게이터의 타이틀 값 지정
    }
    // Search View로 갈 때
    if([segue.identifier isEqualToString:@"toSearchView"]) {
        SearchViewController *sc = [segue destinationViewController];
        sc.title = searchButton.titleLabel.text;
    }
}

- (void) viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    SeoulAtoZ = [UIImage imageNamed:@"inSeoul.jpg"]; // 메인 이미지 지정
    [mainImageView setImage:SeoulAtoZ];
}
```


기본기능설명

Random View



- Navigator Controller를 통해 메인 화면에서 Random View로 이동
- RANDOM 버튼을 누르면 UIPickerView를 통해 그 안의 장소 이미지 데이터들 중 하나를 랜덤하게 선택
- 데이터가 선택되면 Sub View가 나타나며 데이터의 명소 이름을 Label에 표시
- Again 버튼으로 다시 선택 가능
- Detail버튼으로 선택된 데이터의 상세 정보를 볼 수 있는 View로 이동

RandomViewController.h

```
@interface RandomViewController : UIViewController <UIPickerViewDataSource, UIPickerViewDelegate> {  
    BOOL started; }
```

```
@property (strong, nonatomic) IBOutlet UIPickerView *picker;  
@property (strong, nonatomic) IBOutlet UILabel *info;  
@property (strong, nonatomic) IBOutlet UIView *viewHidden;
```

// 랜덤 선택된 장소에 따라 각각의 상세정보 View와 연결하기 위함

```
@property (strong, nonatomic) IBOutlet UIView *coaxView; // 랜덤선택 완료 후 Again, Label, Detail을 다룰 Sub View  
@property (strong, nonatomic) IBOutlet UIView *coaxButton; // 연결 Button
```

```
@property (strong, nonatomic) IBOutlet UIView *sevitView;  
@property (strong, nonatomic) IBOutlet UIButton *sevitButton;
```

```
@property (strong, nonatomic) IBOutlet UIButton *lotteButton;  
@property (strong, nonatomic) IBOutlet UIView *lotteView;
```

```
@property (strong, nonatomic) IBOutlet UIButton *myeongButton;  
@property (strong, nonatomic) IBOutlet UIView *myeongView;
```

```
@property (strong, nonatomic) IBOutlet UIButton *olyButton;  
@property (strong, nonatomic) IBOutlet UIView *olyView;
```

```
@property (strong, nonatomic) IBOutlet UIButton *seoulButton;  
@property (strong, nonatomic) IBOutlet UIView *seoulView;
```

```
- (IBAction)selectRandom:(id)sender;  
- (IBAction)selectAgain:(id)sender;
```

RandomViewController.m

```
// Picker View 컴포넌트의 수
- (NSInteger) numberOfComponentsInPickerView:(UIPickerView *)pickerView {
    return 1;
}
// 컴포넌트 안의 열의 수
- (NSInteger) pickerView:(UIPickerView *)pickerView numberOfRowsInComponent:(NSInteger)component {
    return MAX_NUM;
}
// Picker View 열 높이
- (CGFloat) pickerView:(UIPickerView *)pickerView heightForComponent:(NSInteger)component {
    return 120;
}
- (UIView *) pickerView:(UIPickerView *)pickerView viewForRow:(NSInteger)row forComponent:(NSInteger)component reusingView:(UIView *)view {
    // 인덱스에 따른 이미지 지정
    NSString *imagePath = [NSString stringWithFormat:@"pimage%d.png", row]; // Image 준비
    UIImage *image = [UIImage imageNamed:imagePath];
    UIImageView *imageView; // 이미지가 보여질 준비
    if (nil == view) { // View가 없다면
        imageView = [[UIImageView alloc] initWithImage:image]; // Image View 만들기
        imageView.frame = CGRectMake(0, 0, 300, 125); // Image View 위치
    }
    else {
        imageView = (UIImageView *)view;
        imageView.image = image;
    }
    return imageView;
}
```

RandomViewController.m

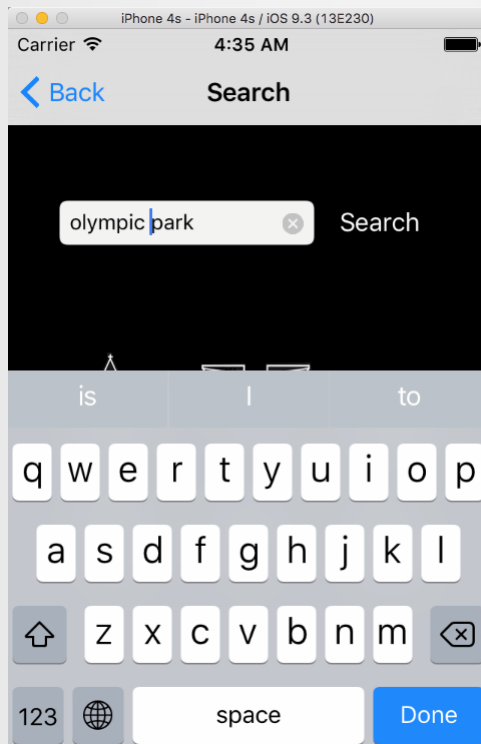
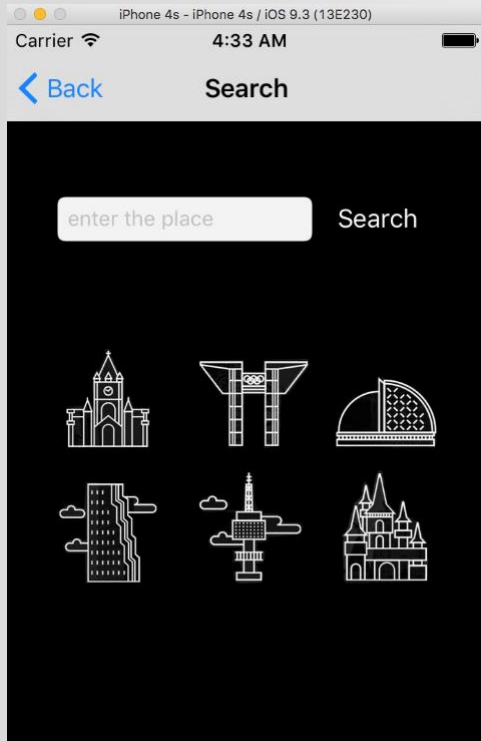
```
- (IBAction)selectRandom:(id)sender {
    // 랜덤 숫자 만들기 0~6
    if(started ==NO){
        int r = arc4random() % MAX_NUM;
        [self.picker selectRow:r inComponent:0 animated:YES];
        [self.viewHidden setHidden:NO];
        started = YES;

        // 각각의 Index값에 따라 장소에 맞는 Label반환
        NSInteger index = [self.picker selectedRowInComponent:0];
        if(index == 0) {
            [self.coaxView setHidden:NO]; // Sub View로 Again, Label, Detail 관리
            info.text = @"Coax";
        }
        else if (index == 1) {
            [self.sevitView setHidden:NO];
            info.text = @"Sevit Island"; } // Index == 5 까지 반복
        // .
        // .
        // .
    }
    // 랜덤선택 초기화 후 다시 랜덤을 돌리기 위한 버튼
- (IBAction)selectAgain:(id)sender {

    [self.viewHidden setHidden:YES];
    [self.picker selectRow:0 inComponent:0 animated:YES];
    started = NO;
}
```

기본 기능 설명

Search View



- Navigator Controller를 통해 메인 화면에서 Search View로 이동

* 검색 :

- Text Field에 찾고자 하는 장소 명 입력
 - Text Field 선택 시 키보드가 나타나며 Done 키를 누르면 키보드가 사라짐
 - Search Button을 누르면 Result View로 이동되어 존재 유/무 판단
 - Text Field 선택 시 키보드가 나타나며 Done 키를 누르면 키보드가 사라짐
 - 대소문자 구분 안함 / 붙여 써도 인식 가능
- * 선택 : Image로 표현된 버튼을 누르면 Image에 해당되는 상세정보 View로 이동

SearchViewController.h

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

```
@interface SearchViewController : UIViewController<UITextFieldDelegate>
```

```
@property (strong, nonatomic) IBOutlet UITextField *searchText;
```

```
@property (strong, nonatomic) IBOutlet UIButton *searchButton;
```

```
- (BOOL)textFieldShouldReturn:(UITextField *)textField;
```

```
@end
```

SearchViewController.m

```
#import "SearchViewController.h"
#import "ResultViewController.h"

#import "LotteWorldViewController.h" #import "CoaxViewController.h" #import "OlympicParkViewController.h"
#import "SevitIslandViewController.h" #import "MyeongDongViewController.h" #import "SeoulTowerViewController.h"

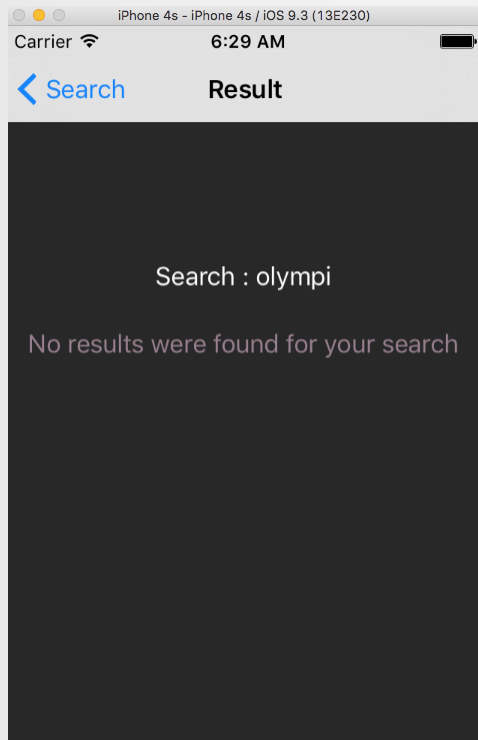
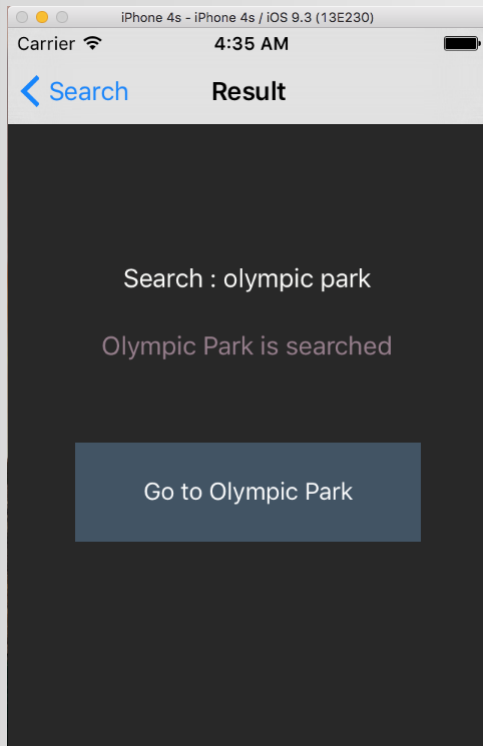
@interface SearchViewController ()
@end
@implementation SearchViewController
@synthesize searchText, searchButton;

- (void) prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    if([segue.identifier isEqualToString:@"toResultView"]) {
        ResultViewController *rc = [segue destinationViewController];
        rc.title = @"Result";
        rc.info = [NSString stringWithFormat:@"%@", searchText.text];
    }
    if([segue.identifier isEqualToString:@"toLotteWorldView"]) {
        LotteWorldViewController *rc = [segue destinationViewController];
        rc.title = @"Lotte World";
    }
    // ... 나머지 SeoulTower, SevitIsland, OlympicPark, Coax, MyeongDong ViewController도 위와 같은 작업
}

- (BOOL)textFieldShouldReturn:(UITextField *)textField {
    [textField resignFirstResponder];
    return YES;
}
```

기본 기능 설명

Result View



- Label로 사용자가 입력한 값을 보여줌
- 입력한 장소의 정보가 APP에 있다면
성공 메시지를 Label로 보여줌
- 그리고 Sub View가 나타나며 그 안의
Go to 버튼으로 입력 장소에 해당되는
상세정보 View로 이동
- 없다면 결과가 없다고 실패 메시지를
보여줌

ResultViewController.h

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

```
@interface ResultViewController : UIViewController{  
    BOOL started;  
}
```

```
@property (strong, nonatomic) IBOutlet UILabel *infoLabel;  
@property (strong, nonatomic) NSString *info;  
@property (strong, nonatomic) IBOutlet UILabel *testLabel;
```

```
@property (strong, nonatomic) IBOutlet UIButton *lotteWorldButton;  
@property (strong, nonatomic) IBOutlet UIView *lotteWorldView;
```

```
@property (strong, nonatomic) IBOutlet UIView *seoulTowerView;  
@property (strong, nonatomic) IBOutlet UIButton *seoulTowerButton;
```

```
@property (strong, nonatomic) IBOutlet UIView *olympicParkView;  
@property (strong, nonatomic) IBOutlet UIButton *olympicParkButton;
```

```
@property (strong, nonatomic) IBOutlet UIView *sevitIslandView;  
@property (strong, nonatomic) IBOutlet UIButton *sevitIslandButton;
```

```
@property (strong, nonatomic) IBOutlet UIView *coaxView;  
@property (strong, nonatomic) IBOutlet UIButton *coaxButton;
```

```
@property (strong, nonatomic) IBOutlet UIView *myeongDongView;  
@property (strong, nonatomic) IBOutlet UIButton *myeongDongButton;
```

```
@end
```

ResultViewController.m

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

```
[self.lotteWorldView setHidden:YES];  
[self.seoulTowerView setHidden:YES];  
[self.coaxView setHidden:YES];  
[self.myeongDongView setHidden:YES];  
[self.sevitIslandView setHidden:YES];  
[self.olympicParkView setHidden:YES];
```

```
infoLabel.text = [NSString stringWithFormat:@"Search : %@", info];
```

```
// Lotte World 검사 문자열  
NSString *lotteStr1 = @"lotteworld";  
NSString *lotteStr2 = @"Lotte World";  
// Seoul Tower 검사 문자열  
// Coax 검사 문자열  
NSString *coaxStr1 = @"Coax";  
// Myeong Dong 검사 문자열  
// Sevit Island 검사 문자열  
// Olympic Park 검사 문자열 >> 위와 똑같이 반복
```

```
// Lotte World 검색 여부 판단  
if ([info compare:lotteStr1  
    options:NSCaseInsensitiveSearch] == NSOrderedSame  
    ||  
    [info compare:lotteStr2  
    options:NSCaseInsensitiveSearch] == NSOrderedSame) {  
    [self.lotteWorldView setHidden:NO];  
    testLabel.text = [NSString stringWithFormat:@"%@ is searched", lotteStr2];  
}  
// Seoul Tower 검색 여부 판단  
// Coax 검색 여부 판단  
// Myeong Dong 검색 여부 판단  
// Sevit Island 검색 여부 판단  
// Olympic Park 검색 여부 판단 >> 위와 똑같이 반복  
else {  
    testLabel.text = @"No results were found for your search";  
}  
}
```

기본 기능 설명

각 장소들의 상세정보 View



- 각 버튼과 연결된 장소들의 상세정보 View
(현재 Olympic Park, Myeong-Dong, Lotte World, Coax, Seoul Tower, Sevit Island가 있음)
 - View의 상단에 장소를 나타내는 Image View가 있고
그 아래의 Label이 장소 이름을 나타냄
 - 하단의 Segment를 누르면 장소들에 맞는 Image가 Image View로 나타남
(처음은 mainView로 'Click on the number below' Image가 나타나지며
세그먼트를 누르면 각 Index마다의 사진이 보여짐)
- * 현재 시점에선 상세한 text정보가 아니라 장소의 image만을 확인가능
추후 세부내용을 추가할 예정

상세 정보 View : OlympicViewController.h

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

@interface OlympicParkViewController : UIViewController {
    UIImage *olympicImage;
    UIImage *mainImage;
    UIImage *image1;
    UIImage *image2;
    UIImage *image3;
    UIImage *image4;
}

@property (strong, nonatomic) IBOutlet UIImageView *olympicImageView;
@property (strong, nonatomic) IBOutlet UIImageView *imageView;

- (IBAction)changeImage:(UISegmentedControl *)sender;

@end
```

상세 정보 View : OlympicViewController.m

```
#import "OlympicParkViewController.h"
```

```
@interface OlympicParkViewController ()
```

```
@end
```

```
@implementation OlympicParkViewController
```

```
@synthesize imageView, olympicImageView;
```

```
-(void)viewDidLoad {
```

```
    [super viewDidLoad];
```

```
    // Do any additional setup after loading the view.
```

```
    olympicImage = [UIImage imageNamed:@"pimage4.png"]; // 상단 Image View
```

```
    [olympicImageView setImage:olympicImage];
```

```
    mainImage = [UIImage imageNamed:@"black.jpg"]; // 하단 세그먼트 별 Image View
```

```
    image1 = [UIImage imageNamed:@"olym1.png"];
```

```
    image2 = [UIImage imageNamed:@"olym2.png"];
```

```
    image3 = [UIImage imageNamed:@"olym3.png"];
```

```
    image4 = [UIImage imageNamed:@"olym4.png"];
```

```
    [imageView setImage:mainImage];
```

```
}
```

```
// 세그먼트의 Index값에 따른 Image 지정
```

```
-(IBAction)changeImage:(UISegmentedControl *)sender {
```

```
    if([sender selectedSegmentIndex] == 0)
```

```
        [imageView setImage:image1];
```

```
    else if ([sender selectedSegmentIndex] == 1)
```

```
        [imageView setImage:image2];
```

```
    else if ([sender selectedSegmentIndex] == 2)
```

```
        [imageView setImage:image3];
```

```
    else if ([sender selectedSegmentIndex] == 3)
```

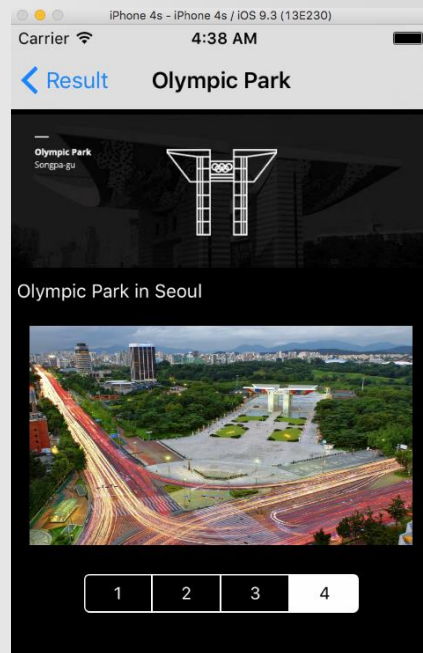
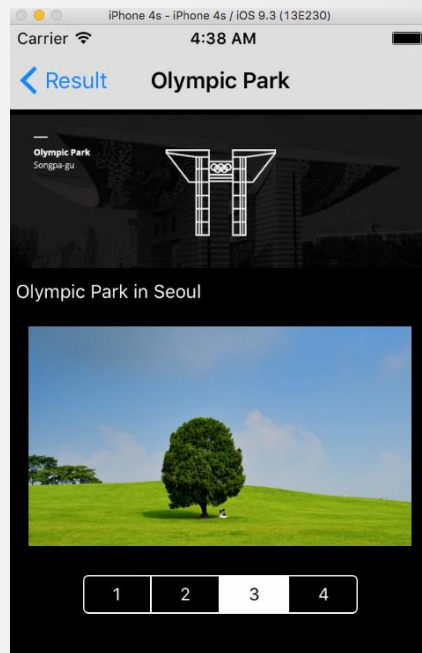
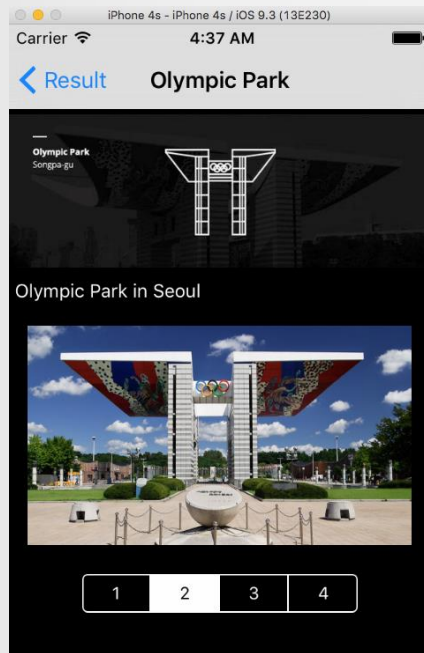
```
        [imageView setImage:image4];
```

```
}
```

```
@end
```

기본 기능 설명

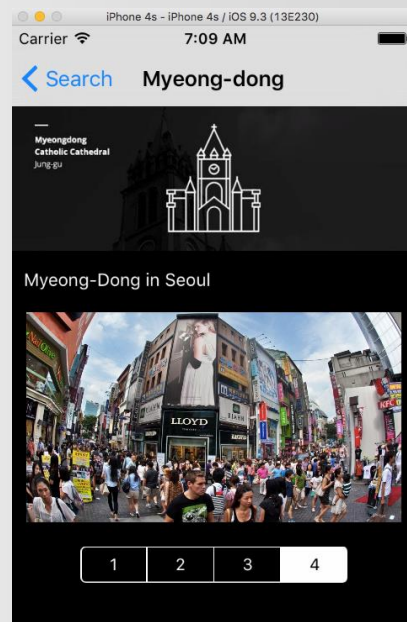
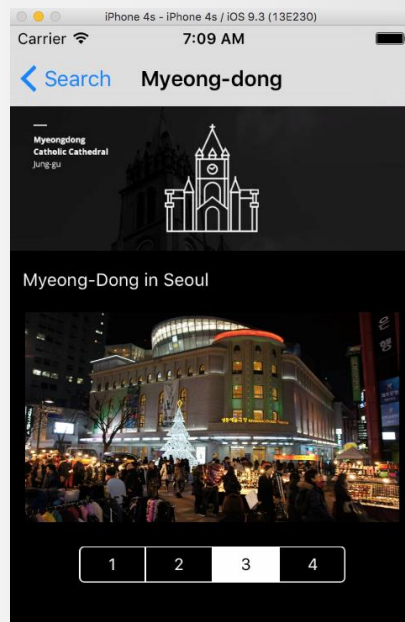
Olympic View - olympicViewController.h / olympicViewController.m



- 세그먼트의 Index별로 Image View 다른 Image가 보여짐

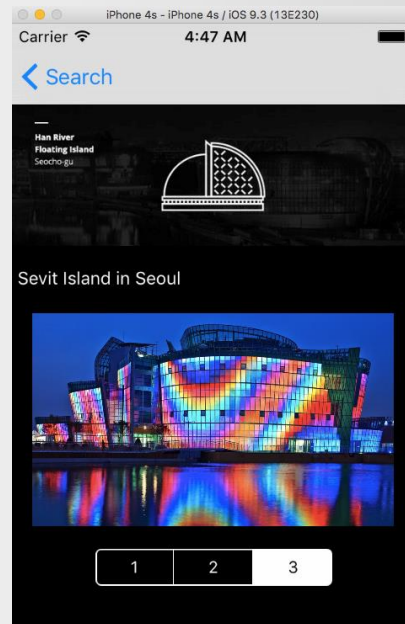
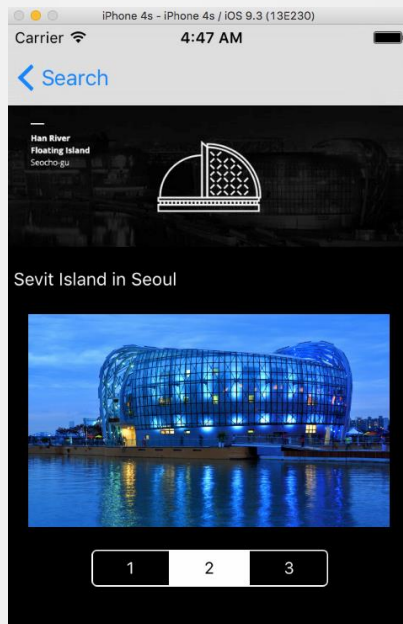
기본기능 설명

Myeong Dong View

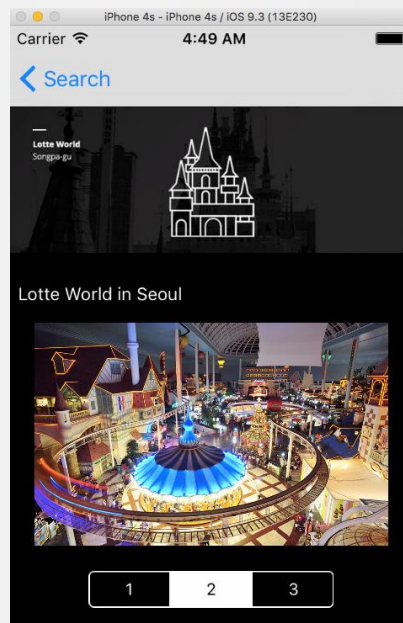
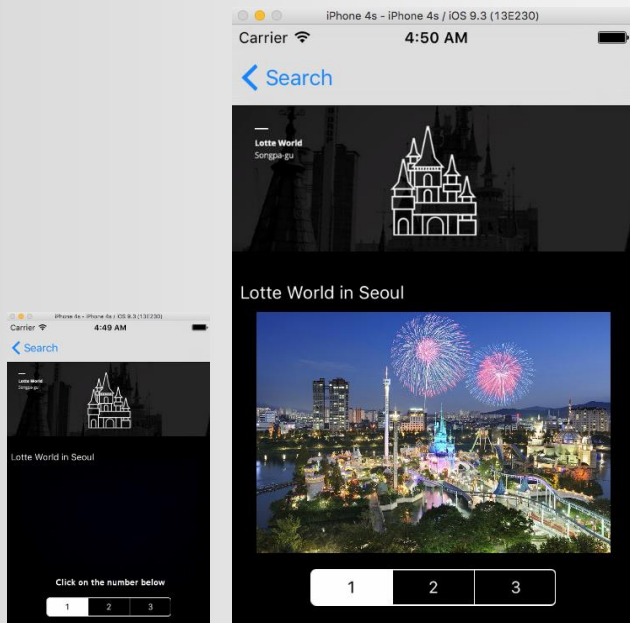


기본기능 설명

Sevit Island View

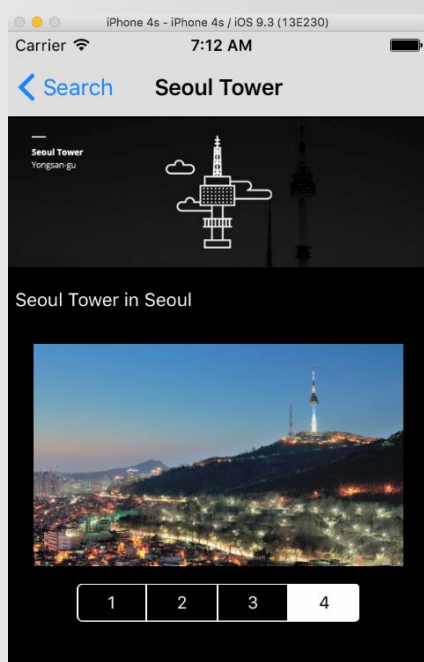
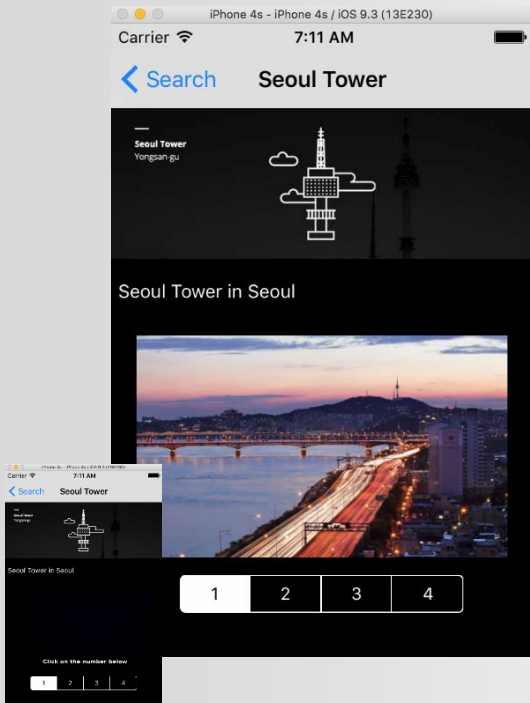


Lotte World View



기본기능 설명

Seoul Tower View



기본기능설명

Coax View

