

# Chapter 15

## iOS Map

# iOS Map Kit

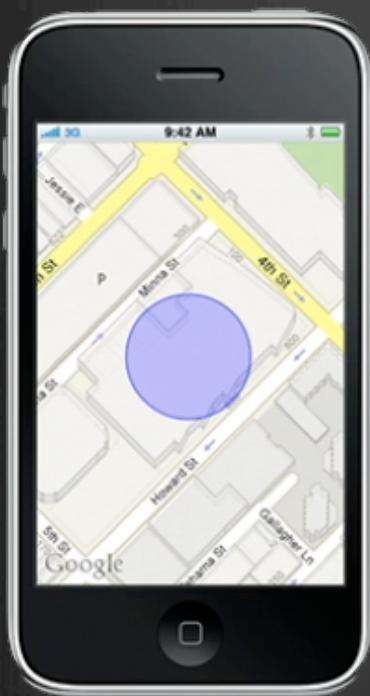


Annotation



Overlay

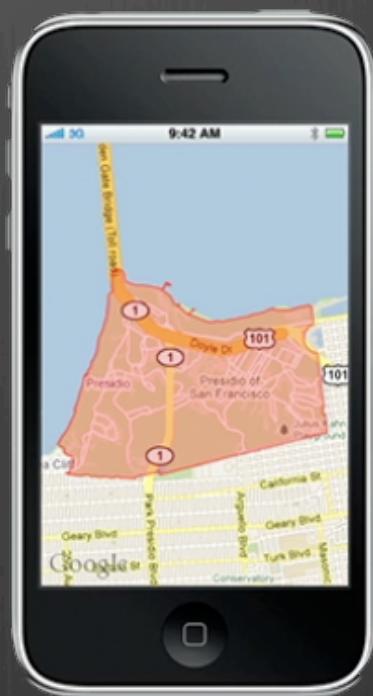
# Overlays



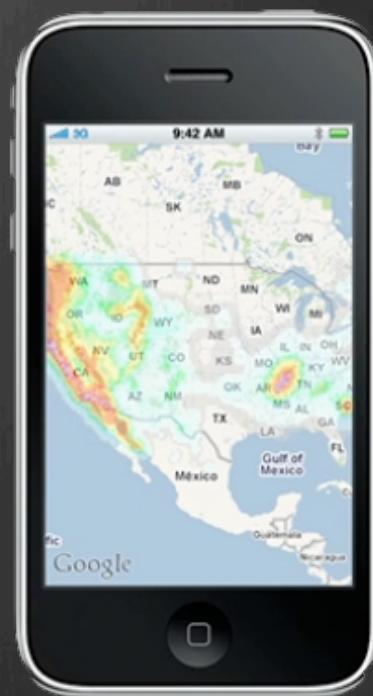
MKCircle



MKPolyline



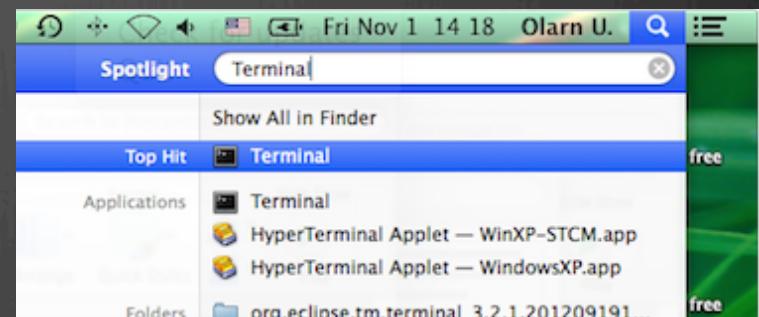
MKPolygon



Custom Overlay

# Lab 1-6 : Start Web Server (1/2)

1. เปิดโปรแกรม Terminal โดย click ที่ Spotlight และพิมพ์ Terminal ในช่อง search
2. จาก Terminal ให้ cd ไปที่ folder ..../Resources/Day4\ -\ Lab15/ RESTServer/ (หรือ drag folder Resource และ drop ลงบน Terminal)
3. Start web server ด้วยคำสั่ง ./start.sh



```
Day4 - Lab14 — Python — 75x24
=====
Simple RESTful Test Server
=====
README
=====
Info: Browse all server files with http://localhost:8000
Info: To prepare JSON GET/POST testing...
      Just create text files with ASCII code content
      and put it into the same folder
      and call... http://localhost:8000/yourfile.json
      The server will echo-back with yourfile.json contents
Note: The example JSON is name customer.json so you can try ...
      http://localhost:8000/customer.json
Note: The content-type should be - application/json -
Press ^C to terminate or just close this terminal.
=====
@rochacbruno Python http server version 0.1 (for testing purposes only)
Serving at: http://localhost:8000
```

# Lab 1-6 : Start Web Server (2/2)

4. ทดสอบว่า web server ทำงานหรือไม่ โดย
  - เปิด Safari บน Mac และ browse ไปที่ `http://localhost:8000`
  - ทดสอบ JSON โดย browse ไปที่ `http://localhost:8000/places.json`
5. เปิด iOS Emulator โดยเปิด Xcode และไปที่เมนู Xcode > Open Developer Tool > iOS Simulator
  - เปิด Safari บน iOS และ browse ไปที่ `http://localhost:8000`
  - ทดสอบ JSON โดย browse ไปที่ `http://localhost:8000/places.json`
6. ปิด web server โดยกด control + c บน keyboard

Note: ถ้า process ค้างให้เปิด Activity Monitor และล็อก quit process ชื่อ “python”

# Lab 2-6 : TrackIt (1 / 13)

## ◎ วัตถุประสงค์

- ◎ เขียน iOS App เพื่อแสดงข้อมูลแผนที่โดยใช้ iOS Map

## ◎ ขั้นตอน

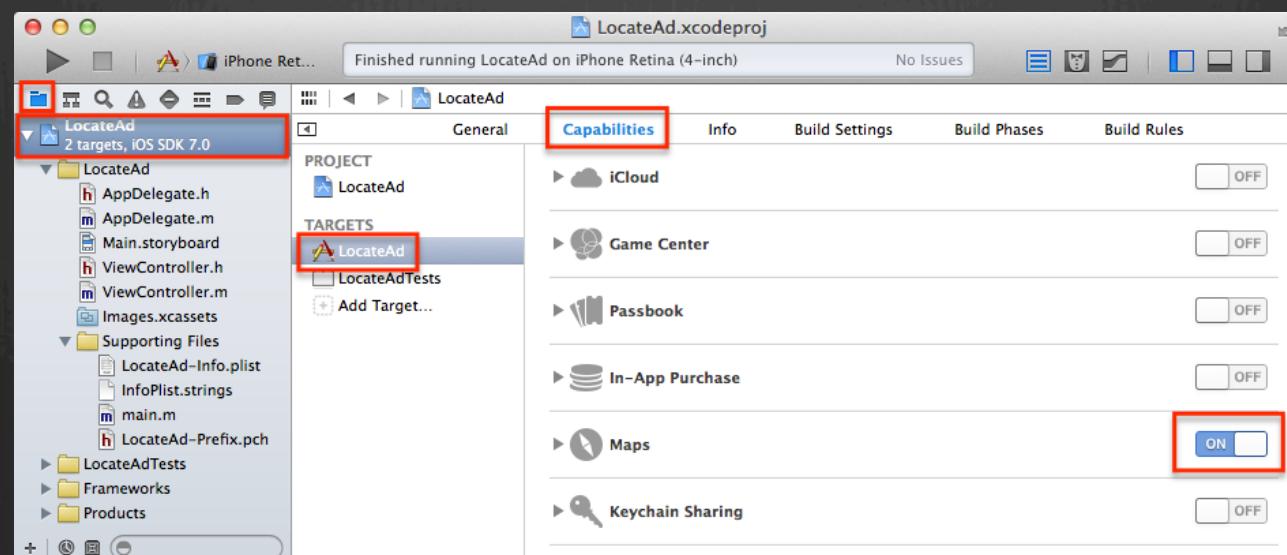
- ◎ สร้าง project
- ◎ เพิ่ม view เพื่อให้ user เปลี่ยนรูปแบบของ map ได้
- ◎ ออกแบบ UI บน Map
- ◎ เขียน code เพื่อเปลี่ยน type ของ Map
- ◎ เขียน code เพื่อ locate ตำแหน่งของ user และเปลี่ยน tracking mode

# Task : Create Project (2/13)

1. จาก Xcode สร้าง project ใหม่ โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “LocateAd” และเลือก Devices เป็น iPhone
3. Click “Next” เลือก folder ที่จะ save project แล้ว click ปุ่ม “Create”
4. ใช้ Finder เปิดไปที่ path ../Resources/Day4 – Lab15/Images/ และ<sup>1</sup> ลากไฟล์ทั้งหมดเข้าไปไว้ใน Images.xcassets ของ project

# Task : Enable Map in iOS (3/13)

- Click ที่ root ของ project ใน Project Navigator เลือก “LocateAd” ใต้ “TARGETS” เปิดหน้า “Capabilities” และเปลี่ยน switch เป็น “On” ในหัวข้อ Maps

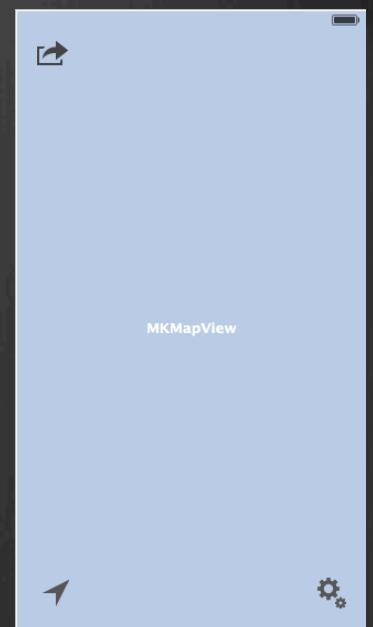


# Task : UI Design – Map (4/13)

6. เปิด Main.storyboard เพิ่ม control ลงบน view ดังนี้

- Map View :-  
    วางตำแหน่งและขนาดของ map ให้เต็มพื้นที่ของ view
- Button :- Image = “action”  
    วางตำแหน่งที่มุ่งช้ายบน
- Button :- Image = “settings”  
    วางตำแหน่งที่มุ่งขวาล่าง
- Button :- Image = “location”  
    วางตำแหน่งที่มุ่งช้ายล่าง

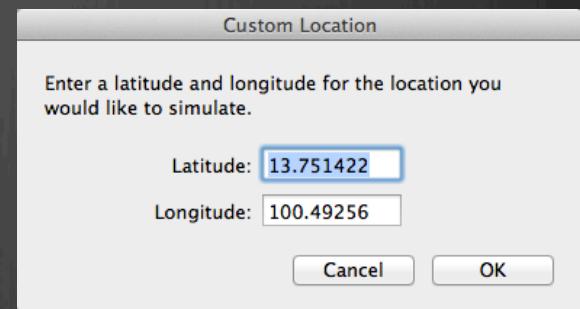
7. Click ที่ Map และเปิด Attributes inspector กำหนด property  
“Show User Location” เป็น YES



# Task : Simulate Location (5/13)

8. Run โปรแกรมเพื่อดูผลลัพธ์

  - เมื่อ โปรแกรมเริ่มทำงาน iPhone จะขอ permission เกี่ยวกับ Location-Based (เฉพาะเมื่อเราเลือก “Show User Location” เท่านั้น)
  - Default location ของ user บน emulator จะอยู่ที่สังหนาดใหญ่ของ Apple (Cupertino, CA)
  - มา เราสามารถกำหนดตำแหน่งเพื่อใช้จำลองตำแหน่งปัจจุบันของ user ได้โดย run Simulator และเปิดเมนู Debug → Location → Custom... และสามารถจำลองการขับรถ ปั่นจักรยาน และการวิ่งได้ด้วย (กำหนดเส้นทางเองไม่ได้)
  - เราสามารถจำลองการ zoom โดยใช้ 2 นิ้วได้โดยกด alt บน keyboard ค้างไว้บนหน้าจอ emulator จะมี cursor สีเทา 2 จุดแสดงขึ้นมาให้ drag mouse เพื่อให้จุดทั้งสองออกห่าง หรือเข้าหากันเพื่อ zoom เข้าออก



# Task : UI Design – Settings View (6/13)

9. เพิ่ม View ใหม่ลงบน Storyboard
10. เพิ่ม control “Segmented Control” ลงบนบริเวณล่างสุดของ View ใหม่
11. Click ที่ Segment Control จากนั้นแก้ property “Segments” ใน Attributes inspector เป็น “3”
12. ทำการสร้าง Segue จากปุ่มที่อยู่ในตำแหน่งขวาล่างของ view แรก (รูป settings) โดยกดปุ่ม control บน keyboard ค้างไว้ แล้ว drag จากปุ่มไปยัง view ใหม่ เลือกเมนู “modal” จากเมนูที่แสดงขึ้นมา
13. Click เลือก Segue ที่สร้างขึ้นมาใหม่ แล้วกำหนด property
  - Identifier = segueMapType
  - Transition Type = Partial Curve
14. Run เพื่อดูผลลัพธ์



# Task : Coding – Change Map Style (7/13)

15. สร้าง class ใหม่เพื่อใช้เป็น view controller ของ view ที่เพิ่มเข้ามา โดย click ขวาที่ project เลือก New File... > iOS > Cocoa Touch > Objective-C class กำหนด “Subclass of” เป็น “UIViewController” และตั้งชื่อ class ว่า “SettingsViewController”
16. เปลี่ยน class ของ view ใหม่ โดย click ที่ view ใหม่แล้วเปิด Identity inspector บน Inspector Pane และเลือก class เป็น “SettingsViewController”
17. Click ที่ view ใหม่ และเปลี่ยน editor mode เป็น Assistant Editor โดยให้ code ด้านขวาเป็นไฟล์ “SettingsViewController.h” จากนั้น
  - ทำการผูก Action ให้กับ Segment control ตั้งชื่อว่า “segMapTypeChanged”
  - ทำการผูก Outlet ให้กับ Segment control ตั้งชื่อว่า “segmentMapType”
18. Click ที่ view เดิม ผูก Action ของปุ่มทั้ง 3 ปุ่มเข้ากับไฟล์ “ViewController.h” โดย
  - ตั้งชื่อ Action ว่า “btnActionTapped” สำหรับปุ่มซ้ายบน
  - ตั้งชื่อ Action ว่า “btnSelfLocationTapped” สำหรับปุ่มซ้ายล่าง
  - ผูก Outlet ให้กับ MapView และตั้งชื่อ property ว่า “Map”

# Task : Coding – Change Map Style (8/13)

19. เปิดไฟล์ “SettingsViewController.h” เพิ่ม delegate code สำหรับ call back กู้ลับไปยัง view ที่เรียก และ default ที่ segment ถูกเลือกตาม map type ที่เปลี่ยนไป ดังนี้

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

@protocol SettingsViewControllerDelegate <NSObject>
- (void)didSelectMapType:(MKMapType)mapType;
@end

@interface SettingsViewController : UIViewController

@property (nonatomic, assign) id<SettingsViewControllerDelegate> delegate;
@property (nonatomic) int mapTypeIndex;

- (IBAction)segMapTypeChanged:(id)sender;

@end
```

# Task : Coding – Change Map Style (9/13)

20. เปิดไฟล์ “SettingsViewController.m” เพิ่ม code ใน method “segMapTypeChanged:” และเพิ่ม method “viewWillAppear” ดังนี้

```
- (void)viewWillAppear:(BOOL)animated
{
    self.segmentMapType.selectedIndex = self.mapTypeIndex;
}

- (IBAction)segMapTypeChanged:(id)sender
{
    if (self.delegate) {
        switch ([sender selectedSegmentIndex]) {
            case 0:
                [self.delegate didSelectMapType:MKMapTypeStandard];
                break;
            case 1:
                [self.delegate didSelectMapType:MKMapTypeHybrid];
                break;
            case 2:
                [self.delegate didSelectMapType:MKMapTypeSatellite];
                break;

            default:
                break;
        }
    }
}
```

# Task : Coding – Change Map Style (10/13)

21. เปิดไฟล์ ViewController.h เพิ่ม code เพื่อ implement delegate ของ class SettingsViewController ดังนี้

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

@interface ViewController : UIViewController <SettingsViewControllerDelegate>

@property (weak, nonatomic) IBOutlet MKMapView *map;

- (IBAction)btnActionTapped:(id)sender;
- (IBAction)btnSelfLocationTapped:(id)sender;

@end
```

# Task : Coding – Change Map Style (11/13)

22. เปิดไฟล์ “ViewController.m” เพิ่ม method “prepareForSegue:sender:” และ “didSelectMapType:” เพื่อเรียก view Setting ขึ้นมาแสดงและเขียน code เพื่อเปลี่ยน map type ที่ user เลือกจาก setting

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"segueMapType"]) {
        SettingsViewController * settingViewController =
            [segue destinationViewController];
        int selectedIndex = -1;
        switch (self.map.mapType) {
            case MKMapTypeStandard:
                selectedIndex = 0; break;
            case MKMapTypeHybrid:
                selectedIndex = 1; break;
            case MKMapTypeSatellite:
                selectedIndex = 2; break;
            default:
                break;
        }
        settingViewController.mapTypeIndex = selectedIndex;
        settingViewController.delegate = self;
    }
}

- (void)didSelectMapType:(MKMapType)mapType
{
    self.map.mapType = mapType;
}
```

# Task : Coding – Change User Tracking Mode (12/13)

23. เพิ่ม code ใน method “btnSelfLocationTapped:” เพื่อให้ Map เลื่อนไป focus ที่ตำแหน่งปัจจุบันของ user (ตามที่เรากำหนด ใน custom location บน Simulator)

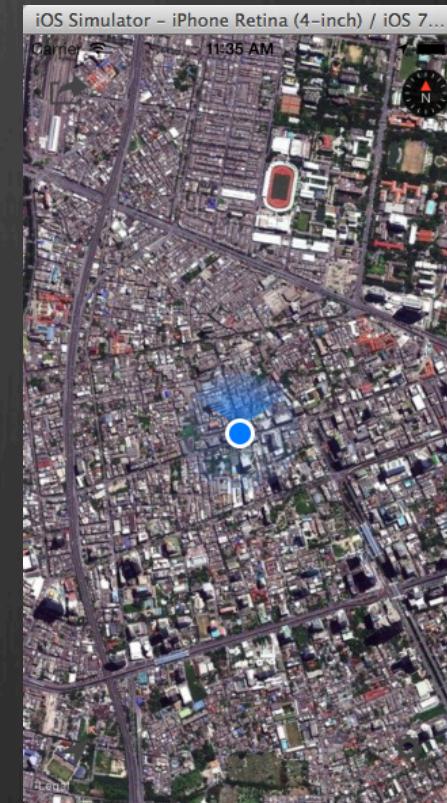
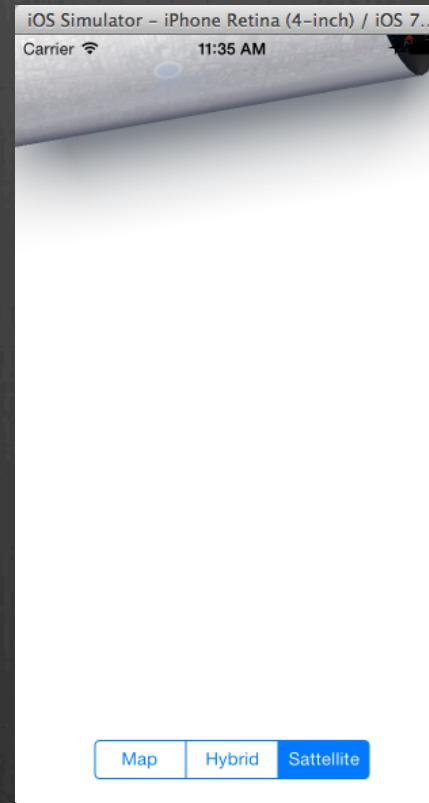
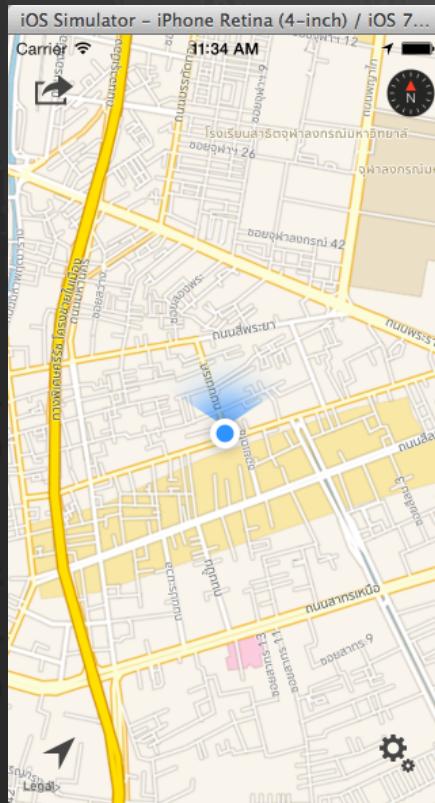
```
- (IBAction)btnSelfLocationTapped:(id)sender
{
    if (self.map.userTrackingMode == MKUserTrackingModeNone)
        self.map.userTrackingMode = MKUserTrackingModeFollow;
    else

        if (self.map.userTrackingMode == MKUserTrackingModeFollow)
            self.map.userTrackingMode = MKUserTrackingModeFollowWithHeading;
        else

            if (self.map.userTrackingMode == MKUserTrackingModeFollowWithHeading)
                self.map.userTrackingMode = MKUserTrackingModeNone;
}
```

# Task : Coding – Change User Tracking Mode (13/13)

24. Run โปรแกรมเพื่อดูผลลัพธ์



# Lab 3-6 : Map Pin (1/7)

- ◎ วัตถุประสงค์
  - ◎ เขียน iOS App เพื่อปัก Pin ลงบน iOS Map ตาม location ที่เรากำหนด
- ◎ ขั้นตอน
  - ◎ สร้าง Class Pin Annotation
  - ◎ สร้าง Class Pin Annotation View
  - ◎ เพิ่ม code สำหรับสร้าง object Pin และปักลงบน Map
  - ◎ เพิ่ม code เพื่อ location ไปยัง Pin ที่ต้องการ และแสดง Pin ทั้งหมดบน Map

# Task : Coding – Add Annotation Class (2/7)

1. เพิ่ม class ใหม่เพื่อใช้เป็น annotation สำหรับเก็บข้อมูล location ในการแสดงหมุดบน map โดย click ขวาที่ project เลือก New File > iOS > Cocoa Touch > Objective-C Class เลือก Subclass of เป็น “NSObject” ตั้งชื่อว่า “LocationAnnotation”
2. เปิดไฟล์ “LocationAnnotation.h” เพิ่ม code ในการ import package MapKit และเพิ่ม property สำหรับแสดงข้อมูลของ Pin

```
#import <Foundation/Foundation.h>
#import <MapKit/MapKit.h>

@interface LocationAnnotation : NSObject <MKAnnotation>

@property (nonatomic, strong) NSString * placeId;
@property (nonatomic, copy) NSString * title;           // ต้องใช้คำว่า title
@property (nonatomic, copy) NSString * subtitle;        // ต้องใช้คำว่า subtitle
@property(nonatomic) CLLocationCoordinate2D coordinate;

@end
```

# Task : Coding – Add Annotation View Class (3/7)

3. Start server ตาม Lab ที่ 1
4. เปิดไฟล์ “ViewController.h” เพิ่ม code ดังนี้

```
#import <UIKit/UIKit.h>
#import "SettingsViewController.h"
#import "LocationAnnotation.h"

@interface ViewController : UIViewController
    <SettingsViewControllerDelegate, MKMapViewDelegate>

@property (nonatomic, strong) NSMutableArray * places;

@property (weak, nonatomic) IBOutlet MKMapView *map;

- (IBAction)btnActionTapped:(id)sender;
- (IBAction)btnSelfLocationTapped:(id)sender;

- (void)getPlaceFromServer;

@end
```

# Task : Coding – Download place from Server (4/7)

5. เปิดไฟล์ “ViewController.m” เพิ่ม method “getPlaceFromServer” สำหรับ download ข้อมูลสถานที่ต่างๆ มาจาก server

```
- (void)getPlaceFromServer
{
    NSURL * url = [NSURL URLWithString:@"http://localhost:8000/places.json"];
    NSURLRequest * request = [[NSURLRequest alloc] initWithURL:url];
    [NSURLConnection sendAsynchronousRequest:request
                                           queue:[NSOperationQueue mainQueue]
                                         completionHandler:^(NSURLResponse *response,
                                                             NSData *data,
                                                             NSError *connectionError)
    {
        // เพิ่ม code ในข้อ 6.
    }];
}
```

# Task : Coding – Download place from Server (5/7)

6. เพิ่ม code ที่เหลือใน method “getPlaceFromServer” จากข้อ 5. เพื่อแปลง JSON เป็น array

```
if (connectionError)
    return;

NSError * error;
NSDictionary * placeDict = [NSJSONSerialization JSONObjectWithData:data
                                                               options:NSJSONReadingMutableLeaves
                                                               error:&error];

for (NSDictionary * eachPlace in [placeDict objectForKey:@"places"]) {
    CLLocationCoordinate2D coordinate;
    coordinate.latitude = [[eachPlace objectForKey:@"lat"] floatValue];
    coordinate.longitude = [[eachPlace objectForKey:@"long"] floatValue];

    LocationAnnotation * annotation = [[LocationAnnotation alloc] init];
    annotation.placeId = [eachPlace objectForKey:@"id"];
    annotation.coordinate = coordinate;
    annotation.title = [eachPlace objectForKey:@"name"];
    annotation.subtitle = [NSString stringWithFormat:@"%@, %@", coordinate.latitude, coordinate.longitude];

    [self.map addAnnotation:annotation];
}
```

# Task : Coding – Handle Pin in Map (6/7)

7. เพิ่ม delegate method ของ map view ชื่อ “mapView:viewForAnnotation:”  
ดังนี้

```
- (MKAnnotationView *)mapView:(MKMapView *)mapView viewForAnnotation:(id<MKAnnotation>)annotation
{
    if (annotation == self.map.userLocation) {
        CLLocationCoordinate2D userCoordinate = annotation.coordinate;
        mapView.userLocation.subtitle =
            [NSString stringWithFormat:@"%@, %@", userCoordinate.latitude,
                                         userCoordinate.longitude];
        return nil;
    }
    else {
        static NSString * annotationIdentifier = @"locationPin";

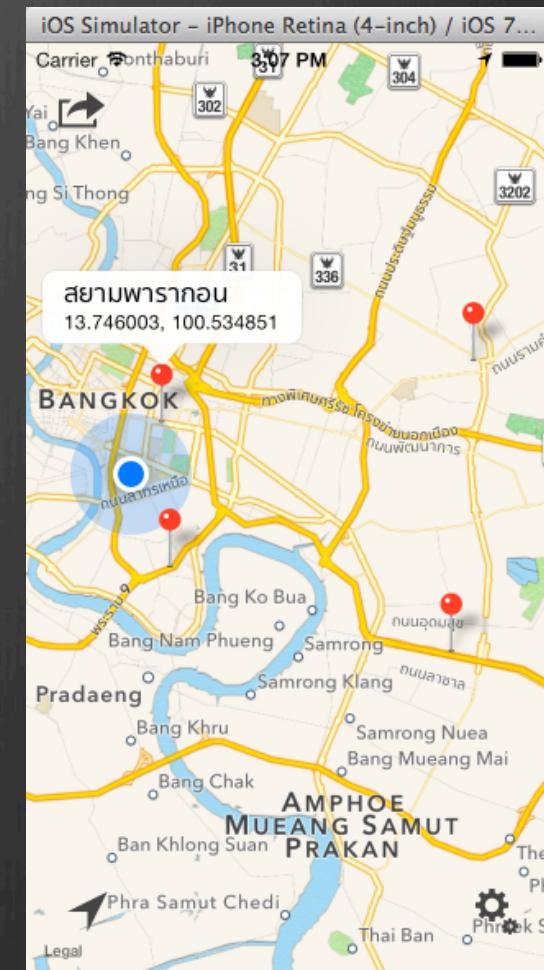
        MKPinAnnotationView * pinView = (MKPinAnnotationView *)[mapView
            dequeueReusableCellWithIdentifier:annotationIdentifier];
        if (!pinView) {
            pinView = [[MKPinAnnotationView alloc] initWithAnnotation:annotation
                reuseIdentifier:annotationIdentifier];
            pinView.canShowCallout = YES;
            pinView.animatesDrop = YES;
        }
        return pinView;
    }
}
```

# Task : Coding – Run & Test (7/7)

8. เพิ่ม code ใน method “viewDidLoad” ดังนี้

```
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    self.places = [[NSMutableArray alloc] init];  
    self.map.delegate = self;  
  
    [self getPlaceFromServer];  
}
```

9. Run โปรแกรมเพื่อดูผลลัพธ์



# Lab 4-6 : Locate to Location (1/6)

## ◎ วัตถุประสงค์

- ◎ เขียน iOS App เพื่อให้ Map แสดงตำแหน่งที่เราต้องการ

## ◎ ขั้นตอน

- ◎ เพิ่ม Action Sheet และ delegate code
- ◎ เพิ่ม code เพื่อสั่งให้แผนที่เลื่อนไปยังตำแหน่งที่ใกล้กับตำแหน่งของ user มากที่สุด และแสดงตำแหน่งนั้นกับ user ในหน้าเดียวกัน

# Task : Coding – Add Delegate Handler (2/6)

- เพิ่ม Actionsheet Delegate ในไฟล์ “ViewController.h”

```
#import <UIKit/UIKit.h>
#import "SettingsViewController.h"
#import "LocationAnnotation.h"

@interface ViewController : UIViewController
<SettingsViewControllerDelegate, MKMapViewDelegate, UIActionSheetDelegate
```

# Task : Coding – Add ActionSheet & Delegate (3/6)

2. เพิ่ม code ใน “btnActionTapped:” เพื่อใช้ Action Sheet และเพิ่ม method “actionSheet:clickedButtonAtIndex:” เพื่อ Locate ตำแหน่งต่างๆ

```
- (IBAction)btnActionTapped:(id)sender
{
    UIActionSheet * actionsheet = [[UIActionSheet alloc] initWithTitle:@"Locate Place"
                                                               delegate:self
                                                               cancelButtonTitle:@"Cancel"
                                                               destructiveButtonTitle:nil
                                                               otherButtonTitles:@"Reload All Places",
                                                               @"Nearest Place", nil];
    actionsheet.delegate = self;
    [actionsheet showInView:self.map];
}

- (void)actionSheet:(UIActionSheet *)actionSheet clickedButtonAtIndex:(NSInteger)buttonIndex
{
    self.map.userTrackingMode = MKUserTrackingModeNone;
    switch (buttonIndex) {
        case 0:
            [self.map removeAnnotations:self.map.annotations];
            [self getPlaceFromServer];
            break;
        case 1: break;
        default: break;
    }
}
```

# Task : Coding – Add Delegate Handler (4/6)

- เพิ่ม code ใน case 1: ใน method “actionSheet:clickedButtonAtIndex:”

case 1:

```
{  
    CLLocationCoordinate2D userCoordinate = self.map.userLocation.coordinate;  
    CLLocation *userLocation = [[CLLocation alloc] initWithLatitude:userCoordinate.latitude  
                                         longitude:userCoordinate.longitude];  
  
    NSArray *places = self.map.annotations;  
    // initial distance and location to default, avoid runtime error  
    CLLocationDistance nearestDistance = 10000000000;  
    LocationAnnotation * nearestPlace = [places objectAtIndex:0];  
  
    for (LocationAnnotation * eachPlace in places) {  
  
        if ([eachPlace isKindOfClass:[MKUserLocation class]])  
            continue;  
  
        if (userCoordinate.latitude == eachPlace.coordinate.latitude &&  
            userCoordinate.longitude == eachPlace.coordinate.longitude)  
            break;  
  
        CLLocation *thisLocation = [[CLLocation alloc]  
                                     initWithLatitude:eachPlace.coordinate.latitude  
                                     longitude:eachPlace.coordinate.longitude];  
        CLLocationDistance distanceFromUserLocationToThisStore =  
            [userLocation distanceFromLocation:thisLocation];  
  
        // ต่อในข้อ 4.
```

# Task : Coding – Add Delegate Handler (5/6)

4. (ต่อ)

```
// ต่อจากข้อ 3.

    if (distanceFromUserLocationToThisStore < nearestDistance) {
        nearestDistance = distanceFromUserLocationToThisStore;
        nearestPlace = eachPlace;
    }
} // for loop

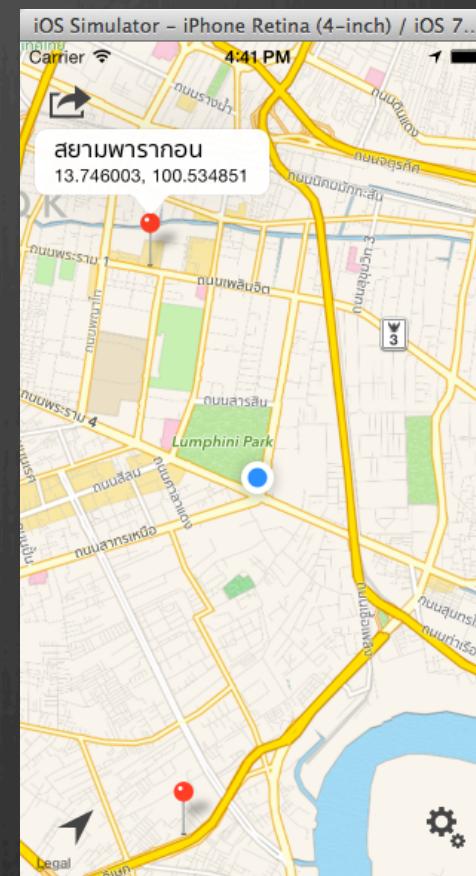
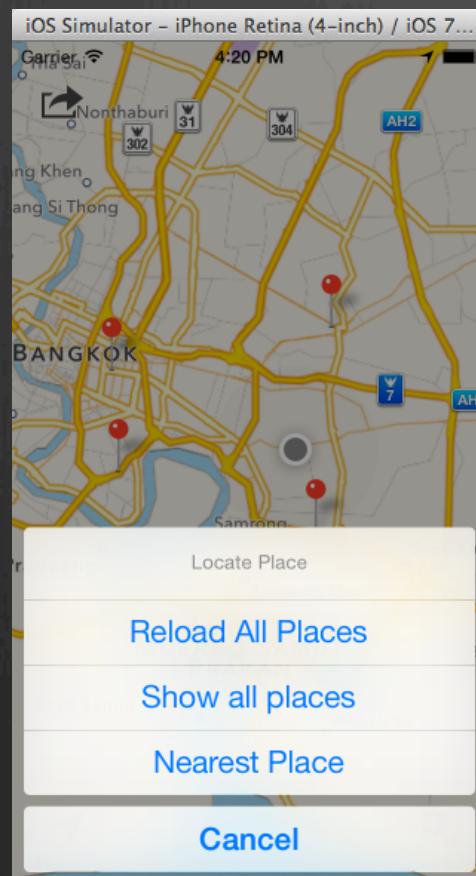
MKCoordinateRegion region = MKCoordinateRegionMakeWithDistance(userCoordinate,
    2 * nearestDistance, 2 * nearestDistance);
[self.map selectAnnotation:nearestPlace animated:YES];

@try {
    [self.map setRegion:region animated:YES];
}
@catch (NSEException *exception) {
}

break;
}
```

# Task : Coding – Add Delegate Handler (6/6)

5. Run โปรแกรมเพื่อดูผลลัพธ์



# Lab 5-6 : iOS 7 Direction (1/5)

## ◎ วัตถุประสงค์

- ◎ เขียน iOS App เพื่อให้ App ค้นหาเส้นทางไปยังจุดที่เราต้องการ

## ◎ ขั้นตอน

- ◎ เพิ่มปุ่มลงไปใน call out เพื่อให้ user กดได้
- ◎ เขียน code เพื่อ request เส้นทางไปที่ server ของ Apple
- ◎ Render เส้นทางลงบน Overlay ของ Map

# Task : Coding – Add Button into Callout (2/5)

1. เปิดไฟล์ “ViewController.h” เพิ่ม property “selectedAnnotation” เพื่อเก็บตำแหน่งของ callout ที่ user เลือกเอาไว้

```
#import <UIKit/UIKit.h>
#import "SettingsViewController.h"
#import "LocationAnnotation.h"

@interface ViewController : UIViewController
<SettingsViewControllerDelegate, MKMapViewDelegate, UIActionSheetDelegate>

@property (nonatomic, strong) NSMutableArray * places;
@property (weak, nonatomic) IBOutlet MKMapView *map;
@property (nonatomic, weak) LocationAnnotation * selectedAnnotation;

- (IBAction)btnActionTapped:(id)sender;
- (IBAction)btnSelfLocationTapped:(id)sender;

- (void)getPlaceFromServer;

@end
```

2. เปิดไฟล์ “ViewController.m” เพิ่ม delegate method ของ MKMapViewDelegate ชื่อ “mapView:didSelectAnnotationView:” เพื่อเก็บค่า location

```
- (void)mapView:(MKMapView *)mapView didSelectAnnotationView:(MKAnnotationView *)view
{
    self.selectedAnnotation = view.annotation;
}
```

# Task : Coding – Add Button into Callout (3/5)

3. เปิดไฟล์ “ViewController.m” เพิ่ม code  
ใน method “mapView:viewForAnnotation:”  
เพื่อสร้างปุ่มไว้ใน callout ดังนี้

```

- (MKAnnotationView *)mapView:(MKMapView *)mapView
    viewForAnnotation:(id<MKAnnotation>)annotation
{
    // ...

    if (!pinView) {
        pinView = [[MKPinAnnotationView alloc] initWithAnnotation:annotation
                                                reuseIdentifier:annotationIdentifier];
        pinView.canShowCallout = YES;
        pinView.animatesDrop = YES;

        UIButton* rightButton = [UIButton buttonWithType:UIButtonTypeDetailDisclosure];
        [rightButton addTarget:self action:@selector(btnCallOutTapped:)
                      forControlEvents:UIControlEventTouchUpInside];
        pinView.rightCalloutAccessoryView = rightButton;
    }

    return pinView;
}

```



# Task : Coding – Request Directions (4/5)

4. เพิ่ม method ชื่อ “btnCallOutTapped:” เพื่อ handle เมื่อ user แตะลงบนปุ่ม ใน callout และ request เส้นทาง (ไปยัง server ของ Apple) จากจุดที่ user อยู่ไปยังจุดที่เลือก

```
- (void)btnCallOutTapped:(id)sender
{
    [self.map removeOverlays: self.map.overlays]; // ลบ overlay เก่าออก

    MKDirectionsRequest *request = [[MKDirectionsRequest alloc] init];
    request.source = [MKMapItem mapItemForCurrentLocation];

    MKPlacemark * placemark = [[MKPlacemark alloc]
        initWithCoordinate:self.selectedAnnotation.coordinate addressDictionary:nil];
    MKMapItem * destination = [[MKMapItem alloc] initWithPlacemark:placemark];
    [destination setName:self.selectedAnnotation.title];

    request.destination = destination;
    request.requestsAlternateRoutes = NO;

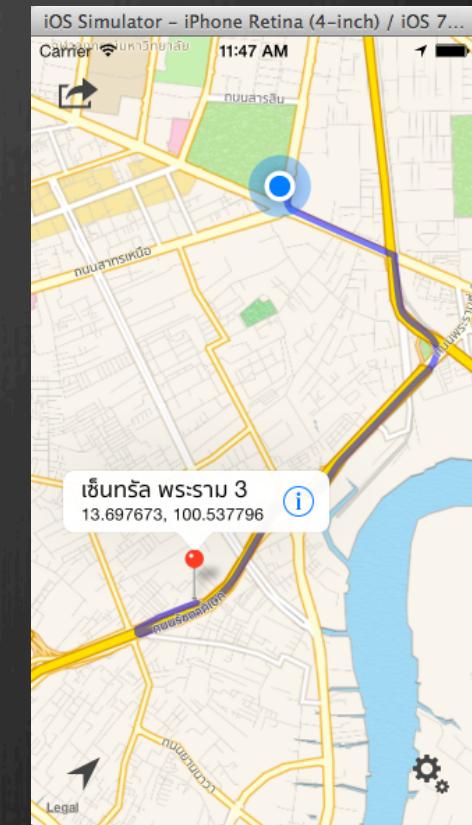
    MKDirections *directions = [[MKDirections alloc] initWithRequest:request];
    [directions calculateDirectionsWithCompletionHandler: ^(MKDirectionsResponse *response,
                                                       NSError *error) {
        for (MKRoute * route in response.routes) {
            [self.map addOverlay:route.polyline
                           level:MKOverlayLevelAboveRoads];
        }
    }];
}
```

# Task : Coding – Render Direction (5/5)

4. เพิ่ม method ชื่อ “mapView:rendererForOverlay:” เพื่อวาดเส้นทางลงบน map

```
- (MKOverlayRenderer *)mapView:(MKMapView *)mapView  
    rendererForOverlay:(id < MKOverlay >)overlay  
{  
    MKPolylineRenderer *renderer =  
        [[MKPolylineRenderer alloc] initWithOverlay:overlay];  
  
    renderer.strokeColor = [UIColor blueColor];  
    renderer.lineWidth = 6.0;  
    renderer.alpha = 0.5;  
  
    return renderer;  
}
```

5. Run โปรแกรมเพื่อดูผลลัพธ์



# Lab 6-6 : Background Update (1/8)

## ◎ วัตถุประสงค์

- ◎ เขียน iOS App เพื่อให้ App ค่อยอ่านค่าตำแหน่งของ user และให้แสดง promotion (Alert) เมื่อตำแหน่งของ user เข้าใกล้สถานที่ใดๆ

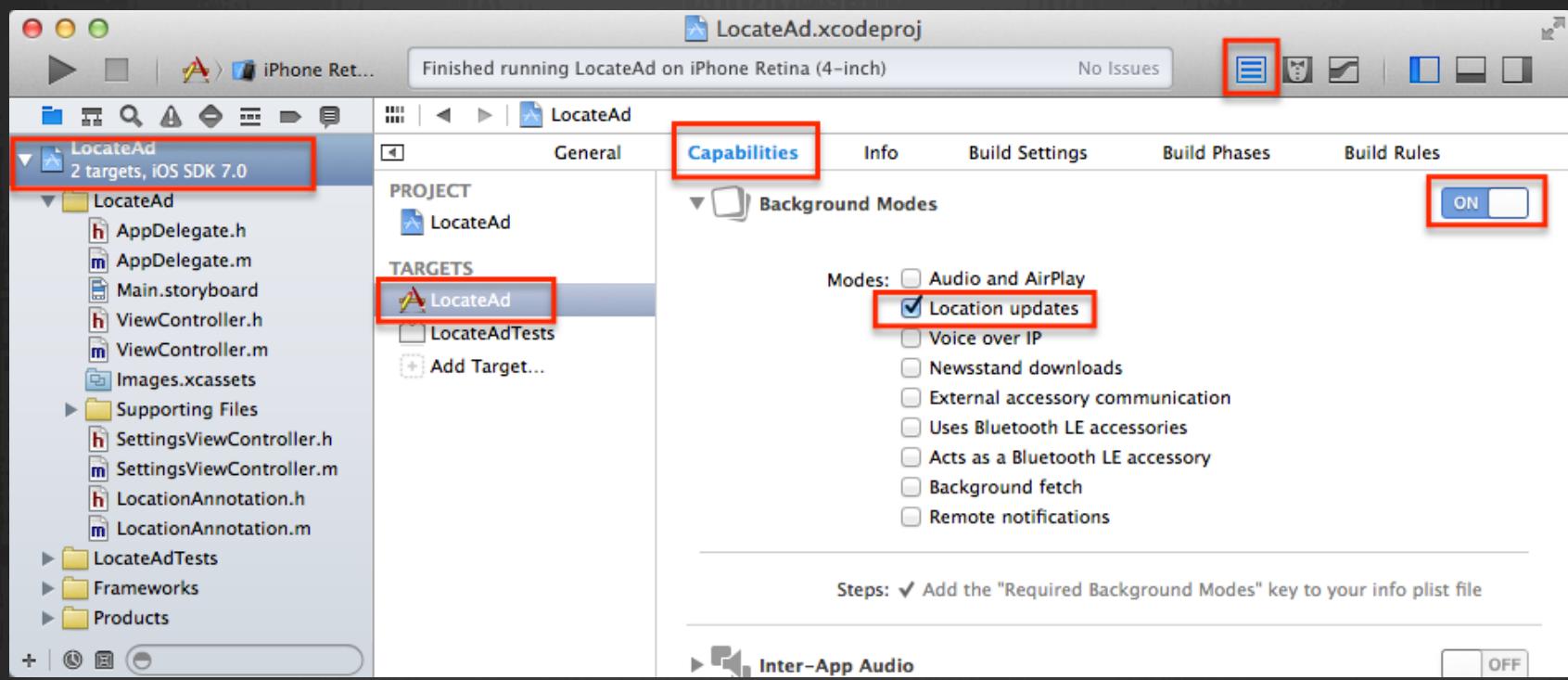
## ◎ ขั้นตอน

- ◎ เพิ่ม Object ของ class Location Manager เพื่อใช้ track ตำแหน่งของ user
- ◎ เขียน code เพื่อเช็คว่าระยะทางเข้าใกล้กว่าที่กำหนด
- ◎ Download promotion มาจาก server
- ◎ Alert promotion ที่ download มาจาก server

Note: อ่าน “App in the Real World” ท้าย lab ด้วย

# Task : Setting – Enable Background Modes (2/8)

1. Click ที่ root ของ project ใน Navigation Pane และ click ที่ “Capabilities” เลือก “Background Modes” เป็น On และเลือก “Location updates”



# Task : Coding – Add Property (3/8)

- เปิดไฟล์ “ViewController.h” เพิ่ม property “locationManager” และ import CoreLocation เพื่อ track ตำแหน่งของ user

```
#import <UIKit/UIKit.h>
#import "SettingsViewController.h"
#import "LocationAnnotation.h"
#import <CoreLocation/CoreLocation.h>

@interface ViewController : UIViewController
<SettingsViewControllerDelegate, MKMapViewDelegate, UIActionSheetDelegate,
CLLocationManagerDelegate>

@property (nonatomic, strong) NSMutableArray * places;
@property (weak, nonatomic) IBOutlet MKMapView *map;
@property (nonatomic, weak) LocationAnnotation * selectedAnnotation;
@property (nonatomic, strong) CLLocationManager * locationManager;
@property (nonatomic) int interval;

- (IBAction)btnActionTapped:(id)sender;
- (IBAction)btnSelfLocationTapped:(id)sender;
- (void)getPlaceFromServer;

@end
```

# Task : Coding – Initial Location Manager Object (4/8)

3. เปิดไฟล์ “ViewController.m” เพิ่ม code ใน method “viewDidLoad” เพื่อสร้าง object ของ Location Manager

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

    _interval = 0;

    _locationManager = [[CLLocationManager alloc] init];
    _locationManager.desiredAccuracy = kCLLocationAccuracyBest;
    _locationManager.delegate = self;

    [_locationManager startUpdatingLocation];

    self.places = [[NSMutableArray alloc] init];
    self.map.delegate = self;

    [self getPlaceFromServer];

    self.firstRouteIsRendered = NO;
}
```

# Task : Coding – Check Location Distance (5/8)

4. เพิ่ม method ชื่อ “locationManager:didUpdateToLocation:fromLocation:” เพื่ออ่านค่า location ในแต่ละวินาที และเช็คว่าเข้าใกล้ location ต่างๆ หรือไม่

```
- (void)locationManager:(CLLocationManager *)manager didUpdateToLocation:(CLLocation *)newLocation
fromLocation:(CLLocation *)oldLocation
{
    if (self.interval == 5) { // ให้เช็คทุกๆ 5 วินาที
        self.interval = 0;

        NSArray *places = self.map.annotations;
        for (LocationAnnotation * eachPlace in places) {

            if ([eachPlace isKindOfClass:[MKUserLocation class]])
                continue;

            CLLocation *thisLocation = [[CLLocation alloc] initWithLatitude:eachPlace.coordinate.latitude
                                                               longitude:eachPlace.coordinate.longitude];
            CLLocationDistance distance = [newLocation distanceFromLocation:thisLocation];

            NSLog(@"Place: %@", eachPlace.title, distance);

            if (distance < 1000)
            {
                [self getPromotionAndAlert:eachPlace.placeId];
                return;
            }
        }
    }
    else
        self.interval++;
}
```

# Task : Coding – Download Promotion (6/8)

5. เพิ่ม method “getPromotionAndAlert:” (ด้านบน LocationManager:) เพื่อ download promotion และ alert

```
- (void)getPromotionAndAlert:(NSString *)placeId
{
    NSURL * url = [NSURL URLWithString:
                    [NSString stringWithFormat:@"http://localhost:8000/promotions-%@.json", placeId]];
    NSURLRequest * request = [NSURLRequest requestWithURL:url];
    [NSURLConnection sendAsynchronousRequest:request
                                           queue:[NSOperationQueue mainQueue]
                                         completionHandler:^(NSURLResponse *response,
                                                             NSData *data, NSError *connectionError) {

        NSDictionary * promotion = [NSJSONSerialization JSONObjectWithData:data
                                                                     options:NSJSONReadingMutableLeaves
                                                                       error:nil];

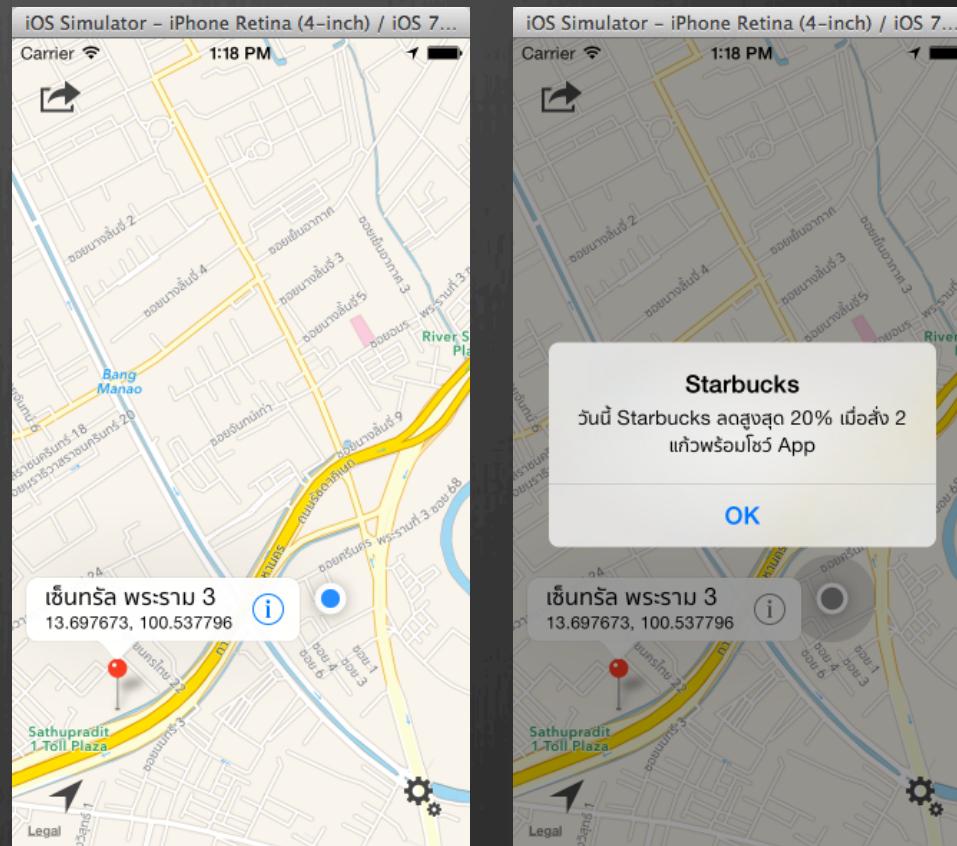
        [self.locationManager stopUpdatingLocation];

        [[[UIAlertView alloc] initWithTitle:[promotion objectForKey:@"brand"]
                                     message:[promotion objectForKey:@"description"]
                                       delegate:nil
                                      cancelButtonTitle:@"OK"
                                      otherButtonTitles:nil, nil] show];
    }];
}
```

# Task : Coding – Run & Test (7/8)

## 6. Run เพื่อติดผลลัพธ์

- ก่อน run App ให้ระบุค่า  
Lat = “13.720879”  
Long = “100.544173”
- เมื่อ run แล้วเปลี่ยนเป็น  
Lat = “**13.700879**”
- กด command + shift + h บน  
keyboard เพื่อจำลองปุ่ม home
- แม้ว่า App จะ update location ใน  
background แต่โปรแกรมจะไม่  
แสดง Alert จนกว่าจะเปิด โปรแกรม  
ขึ้นมา (สังเกตุค่าจาก log)
- ในทางปฏิบัติ ต้องส่งข้อมูลไปที่  
server และ push promotion กลับ  
มา



# Task : Coding – Fix with Local Notification (8/8)

7. ลบ UIAlertView ใน method “getPromotionAndAlert:” ออก แล้วใช้ Local Notification แทน

```
- (void)getPromotionAndAlert:(NSString *)placeId
{
    // ...
    [[UIApplication sharedApplication] cancelAllLocalNotifications];
    UILocalNotification * noti = [[UILocalNotification alloc] init];

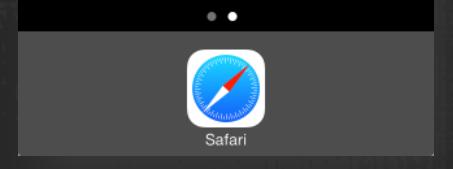
    noti.fireDate = [NSDate date];
    noti.alertBody = [promotion objectForKey:@"description"];

    noti.soundName = UILocalNotificationDefaultSoundName;
    noti.alertAction = NSLocalizedString(@"View Details", nil);

    [[UIApplication sharedApplication] scheduleLocalNotification:noti];
    // ...
}
```



8. Run โปรแกรมตามข้อ 6 เมื่อติดตั้งเสร็จแล้ว ของ user เข้าใกล้เกิน 1000 โปรแกรมจะแสดง notification ให้เห็น



# App in the Real World

- Aware Battery Consumption.
- Aware User Privacy.
- User be able to turn off Background App Refresh & Location Services. Describe it cleared why your app needs their location.
- Integration with Passbook if a good combination.

# MKMapViewDelegate Protocol Methods

## Responding to Map Position Changes

- mapView:regionWillChangeAnimated:
- mapView:regionDidChangeAnimated:

## Loading the Map Data

- mapViewWillStartLoadingMap:
- mapViewDidFinishLoadingMap:
- mapViewDidFailLoadingMap:withError:

## Tracking the User Location

- mapViewWillStartLocatingUser:
- mapViewDidStopLocatingUser:
- mapView:didUpdateUserLocation:
- mapView:didFailToLocateUserWithError:

## Managing Annotation Views

- mapView:viewForAnnotation:
- mapView:didAddAnnotationViews:
- mapView:annotationView:  
calloutAccessoryControlTapped:

## Dragging an Annotation View

- mapView:  
annotationView:  
didChangeDragState:fromOldState:

## Selecting Annotation Views

- mapView:didSelectAnnotationView:
- mapView:didDeselectAnnotationView:

## Managing Overlay Views

- mapView:viewForOverlay:
- mapView:didAddOverlayViews:

# More Details

- ⦿ iOS Map Basic Tutorial  
<http://www.raywenderlich.com/21365/>
- ⦿ iOS 7 Map Overlay  
<https://www.captechconsulting.com/blog/eric-stroh/ios-7-tutorial-series-map-kit>
- ⦿ iOS 7 Location  
[http://www.techotopia.com/index.php/An\\_Example\\_iOS\\_7\\_Location\\_Application](http://www.techotopia.com/index.php/An_Example_iOS_7_Location_Application)