

Radionica 2

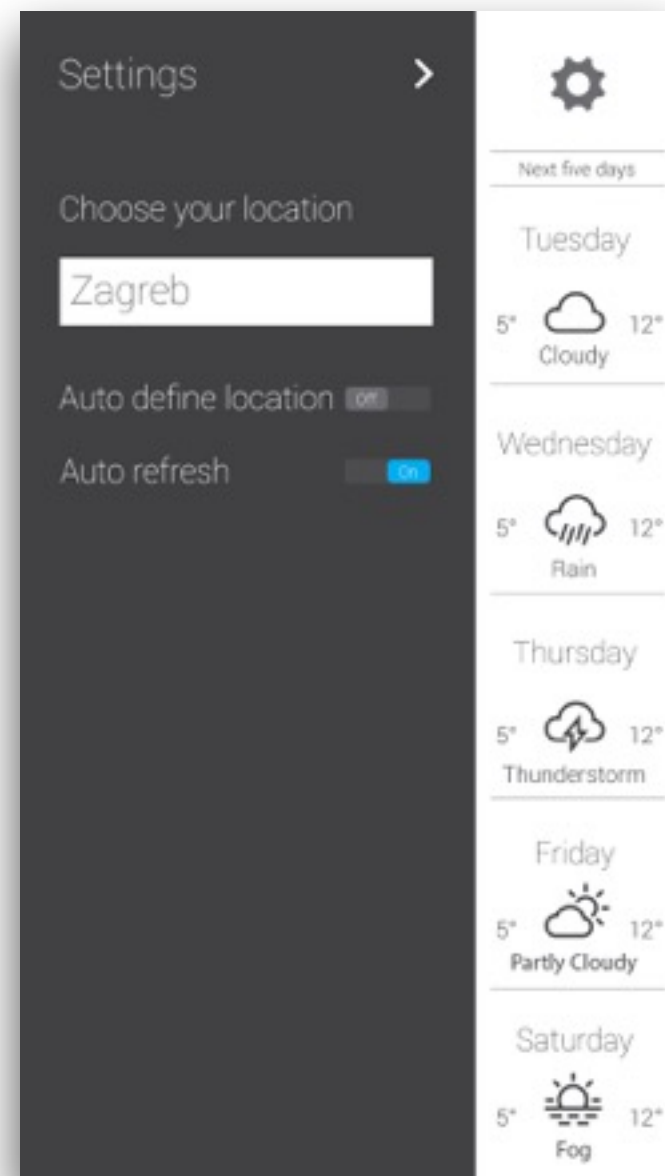
Ponavljjanje

- Xcode
 - znamo što su controller, storyboard, resursi (slike)
- Napravili smo aplikaciju s dva screena, akcijama na buttone i alertom
- Upoznali smo najbitnije klase iz Foundation frameworka
- Koristili smo neke klase iz UIKit frameworka
 - UIButton, UIAlertView, UILabel

Što dalje?

- Cocoa framework je izuzetno velik
- upoznavanje po potrebi - pronalazak rješenja ili odgovora na sličan problem na Internetu
- učenje na primjeru

WeatherApp



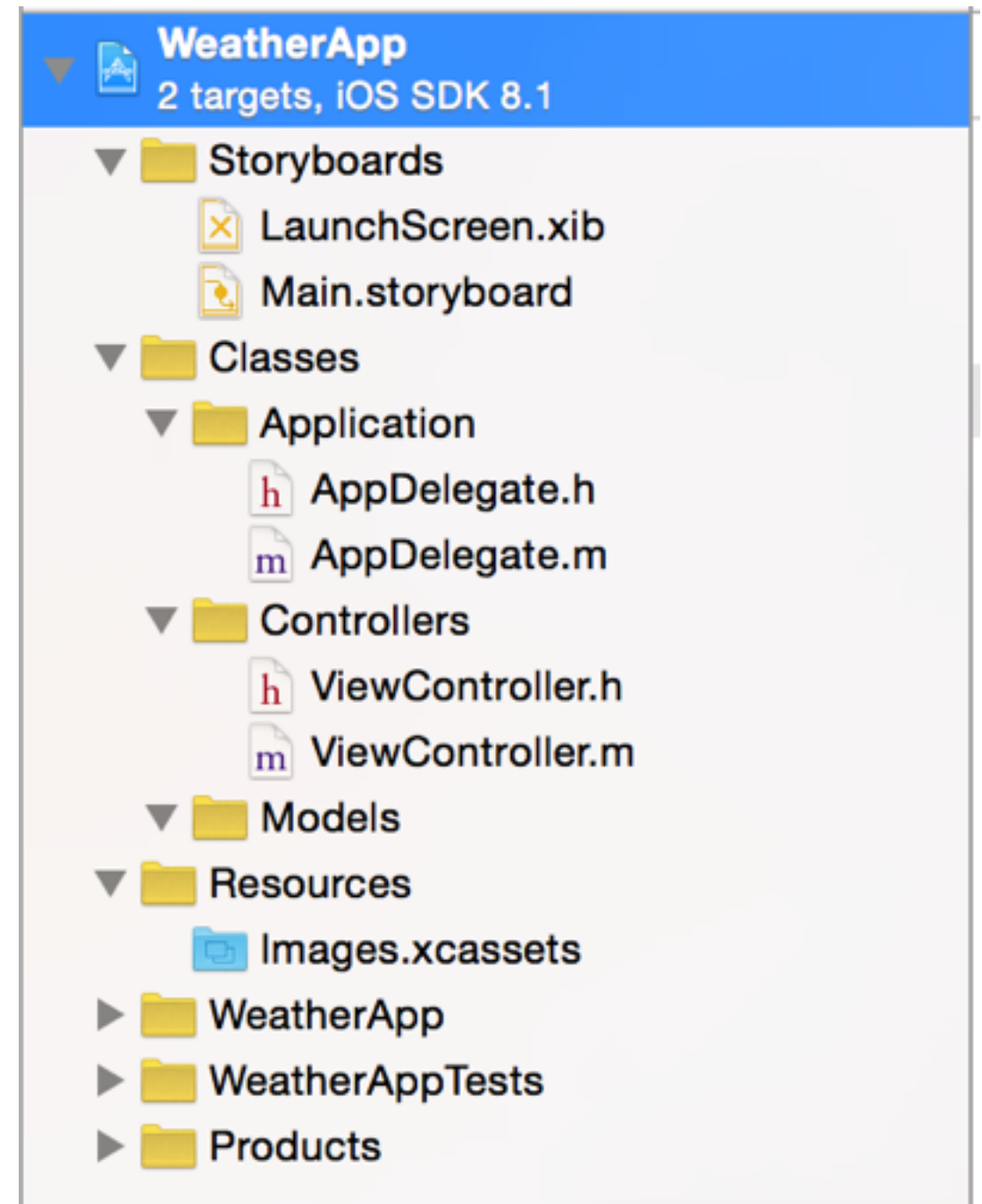
- Zbog specifičnosti iOS platforme i nedostatka vremena dizajn ćemo malo promijeniti

Današnji plan

- Organizacija projekta
- Settings layout
- Funkcionalnost settings ekrana
 - SettingsManager - singleton
 - Perzistentna pohrana postavki
- LocationManager
 - dohvaćanje lokacije pomoći GPS-a

Organizacija projekta

- Lakše snalaženje u projektu
- pronalazak klase ili datoteke prema njenoj funkciji



Settings layout

- UITableViewController
 - UITableView sa “static cells”
- UILabel
- UITextField
- UISwitch

Funkcionalnost settings ekrana

- Povezivanje UI elemenata s controllerom
- Povezivanje korisničkih akcija s controllerom

Singleton pattern

- Koristimo samo jednu instancu neke klase
- [SettingsManager sharedManager]
 - uvijek vraća isti objekt koji stvaramo samo kod prvog poziva metode
 - sljedeći puta uvijek vraća taj stvoreni objekt

Singleton - template

```
@implementation SettingsManager

+ (instancetype)sharedManager
{
    static SettingsManager *manager = nil;
    static dispatch_once_t onceToken;
    dispatch_once(&onceToken, ^{
        // code to be executed once
        manager = [[SettingsManager alloc] init];
    });
    return manager;
}
```

- Promjenom klase (*SettingsManager*) imamo metodu za bilo koji drugi singleton

Perzistentna pohrana podataka

- UserDefaults
 - Male količine podataka (korisničke postavke)
 - Key-value pohrana (slično kao NSDictionary)
- CoreData (izvan okvira ove radionice)
 - *an object graph and persistence framework*

NSUserDefaults

```
[[NSUserDefaults standardUserDefaults] setObject:@"value" forKey:@"KEY"];  
[[NSUserDefaults standardUserDefaults] removeObjectForKey:@"KEY"];  
[[NSUserDefaults standardUserDefaults] synchronize];
```

LocationManager

- CoreLocation framework
- praćenje korisničke lokacije, geofencing, beacon monitoring...
- moguće definiranje preciznosti
- rad u backgroundu
- dozvole za dohvaćanje korisničke lokacije - iOS 8

Dozvole za dohvaćanje lokacije - iOS 8

- Info.plist - poruke koje objašnjavaju korisniku za što će se koristiti Location Service
 - CLLocationAlwaysUsageDescription
 - CLLocationWhenInUseUsageDescription
- request(Always/WhenInUse)Authorization metode

Pokretanje LocationManagera

```
self.locationManager = [[CLLocationManager alloc] init];
self.locationManager.delegate = self;

if ([self.locationManager respondsToSelector:
    @selector(requestAlwaysAuthorization)]) {

    [self.locationManager requestAlwaysAuthorization];
}
[self.locationManager startUpdatingLocation];
```

- metodom *respondsToSelector:* pitamo imali li objekt *self.locationManager* metodu *requestAlwaysAuthorization*
 - ona ne postoji u slučaju kada se ovaj kod izvršava na uređaju s iOS 7

UX

- osigurati rad aplikacije i kada korisnik ne dopusti korištenje
- obavijestiti ga UIAlertom kada ulazi na screen ako je to bitno
- ako aplikacija bez toga uopće nema smisla prikazati poseban screen koji objašnjava kako upaliti LocationService (Settings aplikacija)
- u iOS 8 postoji opcija otvaranja Settings aplikacije iz vlastite aplikacije (Google)

Layout “home screena”

- napraviti za vježbu za sljedeću radionicu
 - **sve osim dijela s desne strane koji prikazuje vrijeme za sljedećih 5 dana, tamo staviti UITableView**
- pokušati namjestiti autolayout i testirati na iPhone 4, 5, 6 i 6 Plus simulatorima
- Oni koji nisu u mogućnosti raditi na OS X moći će preuzeti projekt s dovršenim layoutom dan prije sljedeće radionice
- adresa na kojoj se nalaze materijali (prezentacije, primjeri, WeatherApp):
 - <https://github.com/filipbec/TVZ-MC2-2015>