



DipMatBeacon

Diego Russo



Who I am

- Diego Russo
- Studente Specialistica, 231423
- Living and working in Cambridge
- Staff Software Engineer @ ARM Ltd
- <http://www.diegor.uk> - me@diegor.it
- @diegor



The project

- Developing an iOS app using Swift (2.2) language. The app is used to check the booking state of department rooms
- Using iBeacon technology within the app <https://developer.apple.com/ibeacon/>
- Other features within the app

Outline

- Xcode
- Framework iOS
- Swift Language
- App

Xcode

- Editor
- Compiler
- Interface Builder
- iOS simulator
- Debugger
- Profiler
- Playground
- Version Control



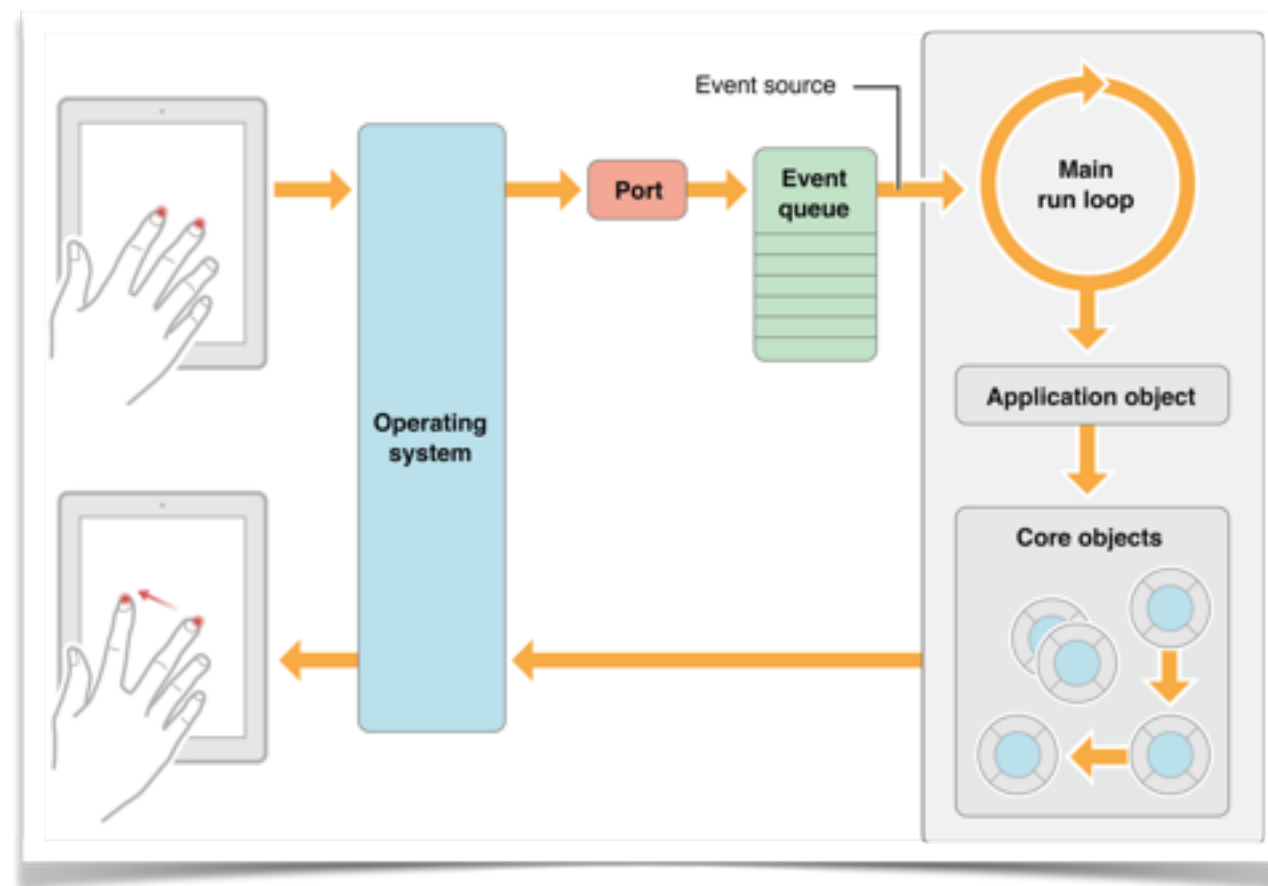
Framework iOS

- Frameworks: WebKit, Cocoa Touch Layer, Media Layer, Core Services Layer, Core OS Layer...
- 3000+ Libraries
- Threads: GCD
- Resources: <https://developer.apple.com/library/ios/navigation/>
- MVC: Model-View-Controller
- Delegation: the most common pattern in iOS



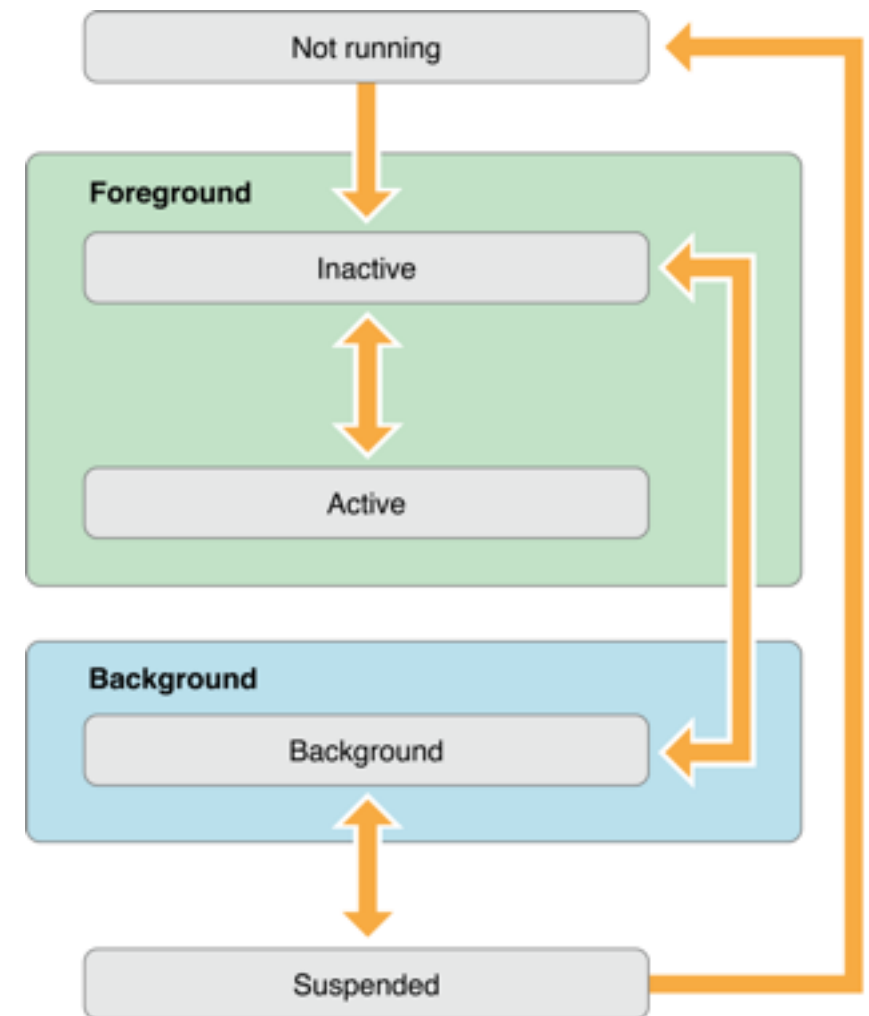
Main run loop

- Main run loop processes all user-related events and it runs on main thread
- Events: touch, remote control, location, accelerometer, etc...



App States

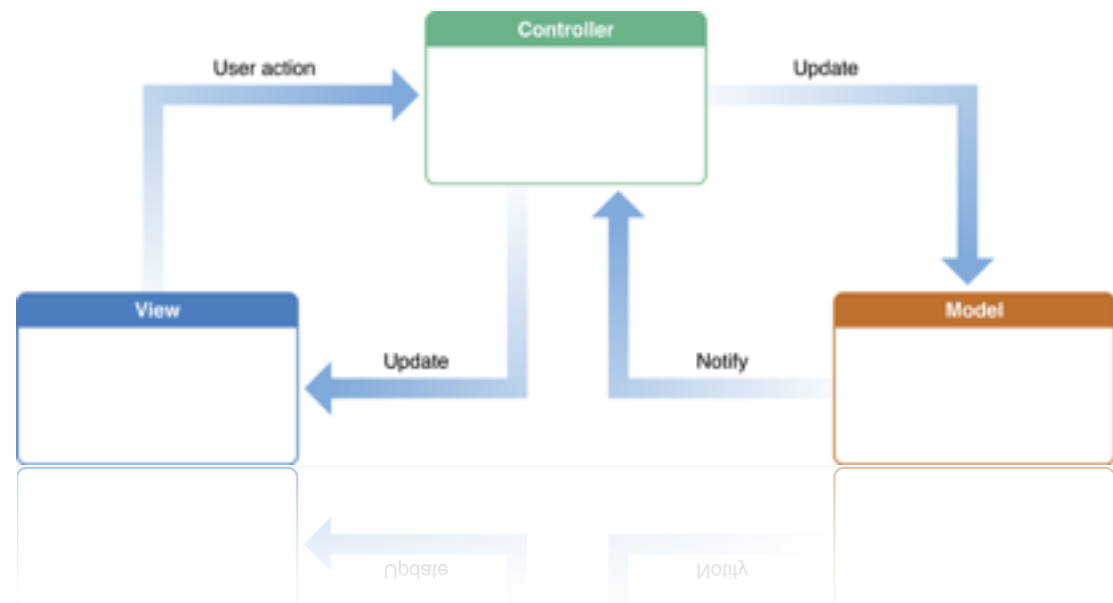
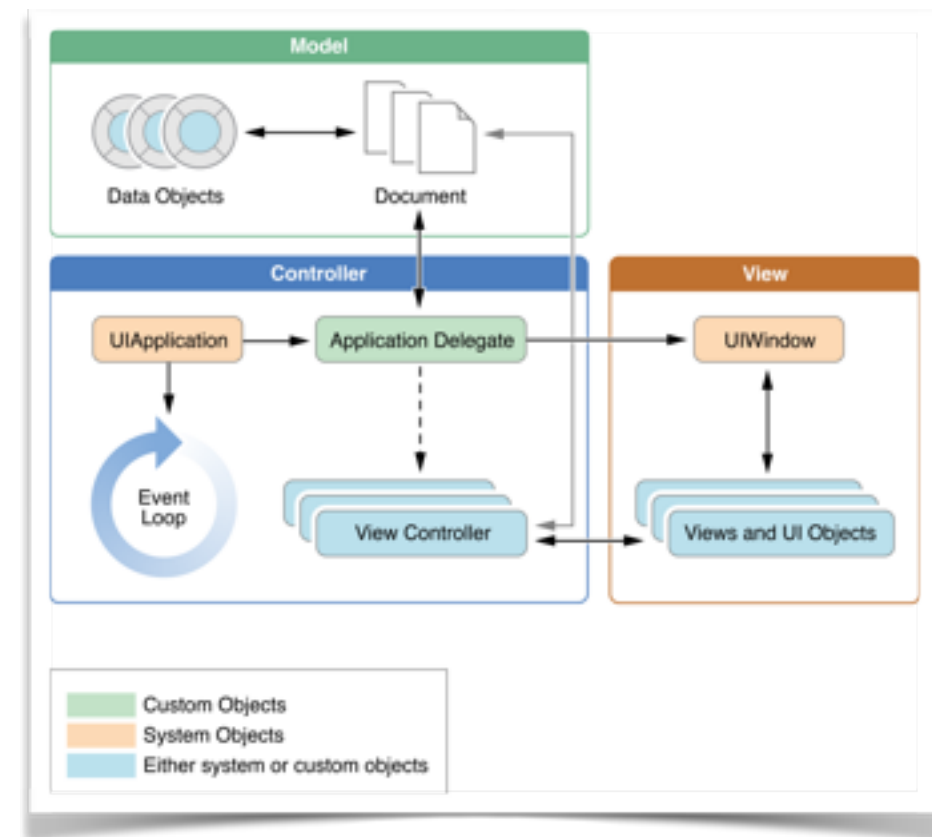
- **application:willFinishLaunchingWithOptions:**—This method is your app's first chance to execute code at launch time.
- **application:didFinishLaunchingWithOptions:**—This method allows you to perform any final initialization before your app is displayed to the user.
- **applicationDidBecomeActive:**—Lets your app know that it is about to become the foreground app. Use this method for any last minute preparation.
- **applicationWillResignActive:**—Lets you know that your app is transitioning away from being the foreground app. Use this method to put your app into a quiescent state.
- **applicationDidEnterBackground:**—Lets you know that your app is now running in the background and may be suspended at any time.
- **applicationWillEnterForeground:**—Lets you know that your app is moving out of the background and back into the foreground, but that it is not yet active.
- **applicationWillTerminate:**—Lets you know that your app is being terminated. This method is not called if your app is suspended.



Demo: AppLifeCycle

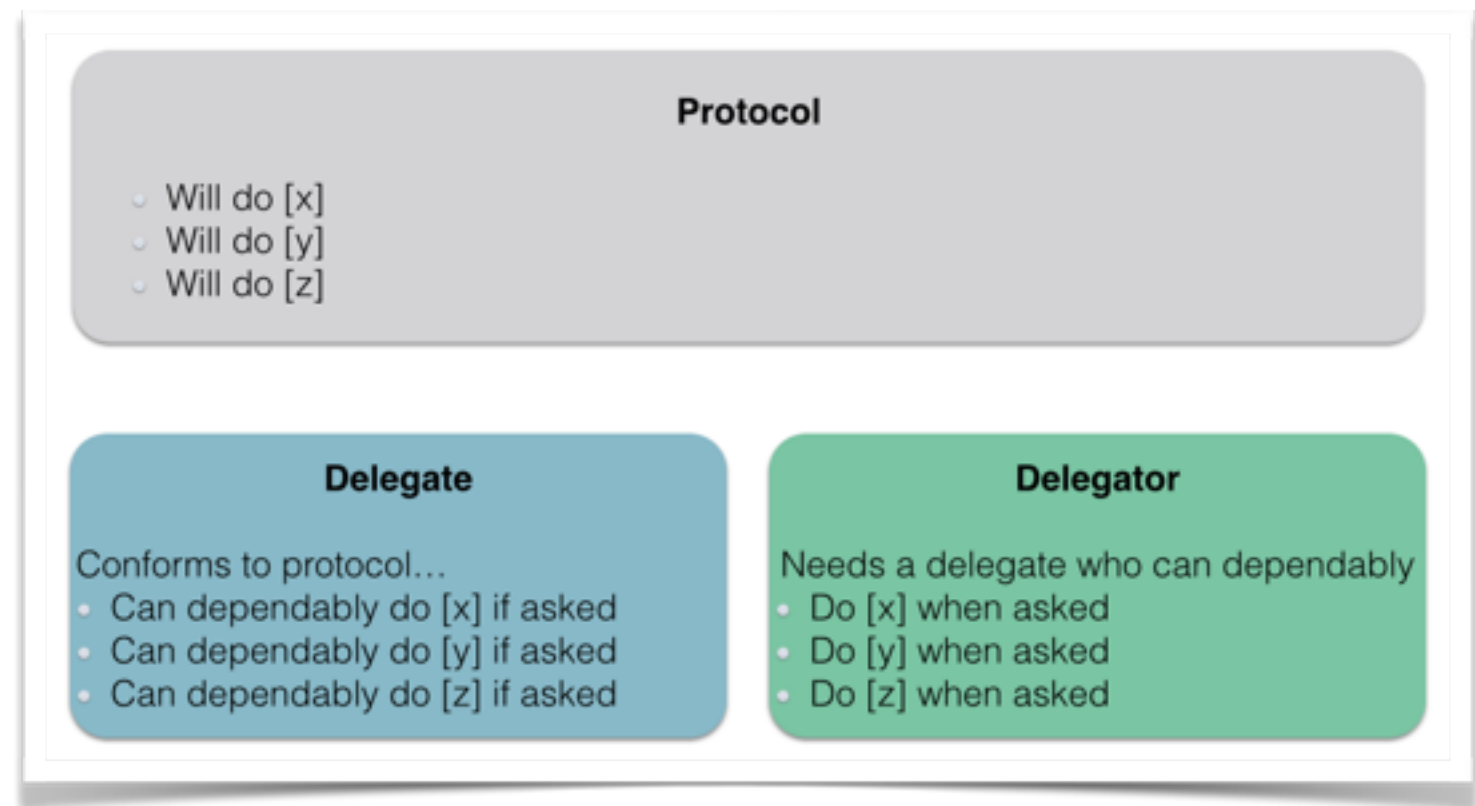
MVC: Model-View-Controller

- UIApplication object: event loop
- App delegate object: heart of custom code
- Documents and data model objects: UI independent
- View controller objects: UIViewController, brain
- UIWindow object: coordinates presentation
- View objects, control objects, and layer objects: visual representation, buttons, data representing visual content
- Communications:
 - target action/outlet
 - datasource: delegated control of data (deal with data)
 - delegation: delegated control of the user interface (deal with UI)
 - notifications: postNotification, addObserver,



Delegation

- One class (a **delegator** class) would give control or responsibility for some behavioral logic to another class (a **delegate** class)
- A **protocol** defines a blueprint of methods (no implementation), properties, and other requirements that suit a particular task or piece of functionality
- All iOS apps use delegation




Swift Language

- Language for: iOS, macOS, watchOS, tvOS, GNU/Linux (Ubuntu)
- Open Source from version 2: www.swift.org
- Version 3 in preview
- Safe programming patterns
- Memory management with ARC: Automatic Reference Counting
- Derived from and strong integration with Obj-C but much more “friendly”
- It could be used as scripting language
- **Demo with Playground**



App - Backend

- Mock up data
- Data exported from <https://servizi.dmi.unipg.it/mrbs>
- Python scripts to convert data
- Data uploaded to  **stamplay**
- Service provides JSON REST API

App - iBeacon



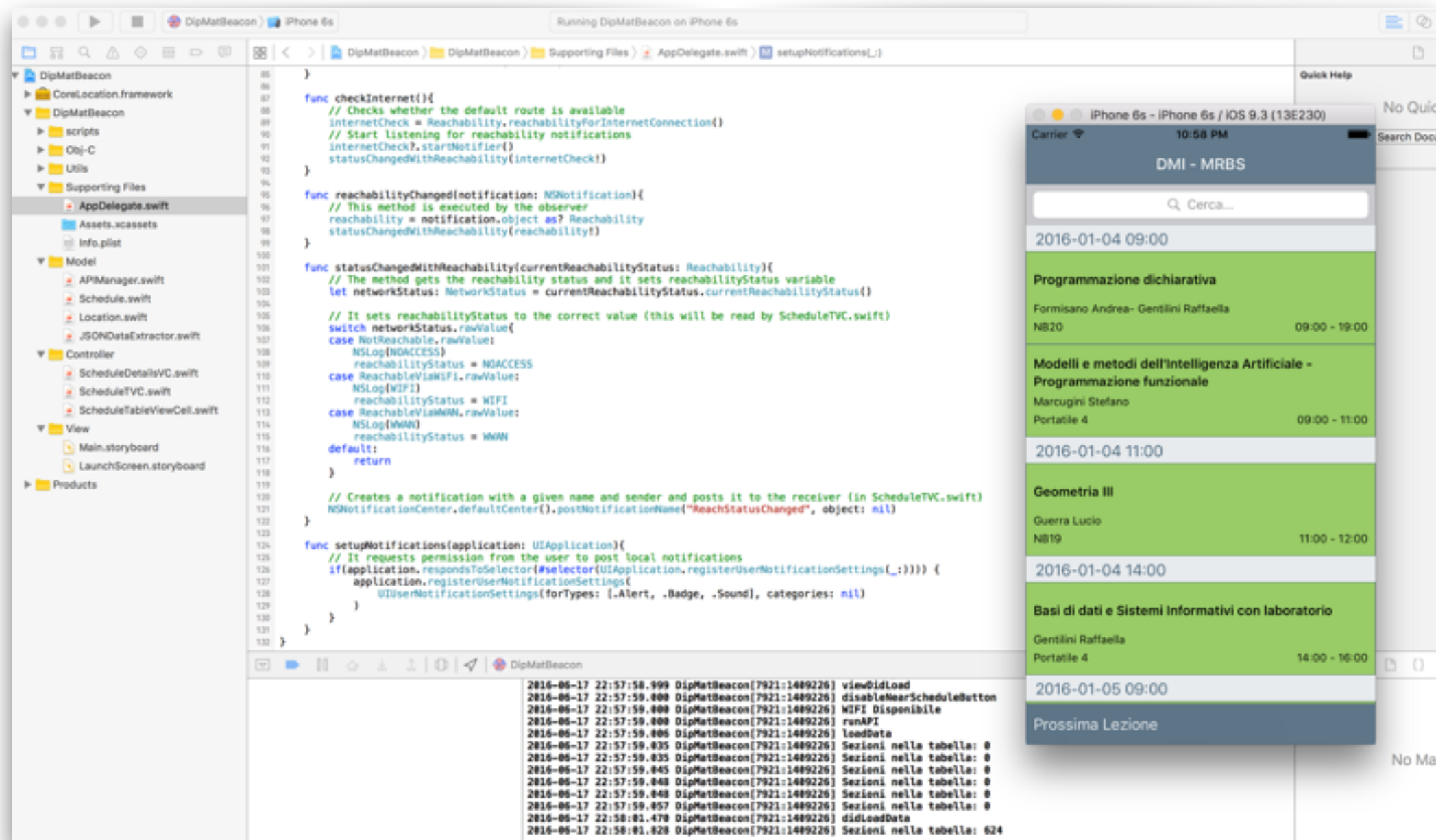
- Based on Bluetooth Low Energy
- Indoor location: used to establish a region around an object
- Proximity state: immediate, near (1-3m), far, unknown
- Calibration phase: 1m for 30 seconds to set the TXPower and to get the right RSSI

| Field | Size | Description |
|-------|----------|--|
| UUID | 16 bytes | Application developers should define a UUID specific to their app and deployment use case. |
| Major | 2 bytes | Further specifies a specific iBeacon and use case. For example, this could define a sub-region within a larger region defined by the UUID. |
| Minor | 2 bytes | Allows further subdivision of region or use case, specified by the application developer. |

App - Features

- Data split in sections
- Search through the data using any field
- Manual refresh
- Fast scroll to next schedule
- Details of the schedule
- Share schedule details through SMS/Mail/Social Media
- TouchID protection for sharing schedule details
- Landscape mode
- Reachability
- iBeacon localisation in foreground
- iBeacon localisation in background with local notifications
- Compatible with any device: from iPhone 4 to iPad Pro

App - DipMatBeacon



<https://github.com/diegorusso/DipMatBeacon>

Future improvements

- Mark the presence automatically both for students and professors
- Users management
- Modify MRBS to host minor/major information and to serve JSON REST API
- Core Data to store data locally on device
- Map every room with beacons
- Mark professors presence in their offices
- People indoor tracking