**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ СИСТЕМИ** **към ТЕХНИЧЕСКИ УНИВЕРСИТЕТ - СОФИЯ**

**ДИПЛОМНА РАБОТА**

Тема: Мобилно приложение, за запознаване със забележителности

Дипломант: Научен ръководител:

*Петко Дапчев Жанет Панева*

СОФИЯ

2021

**ТЕХНОЛОГИЧНО УЧИЛИЩЕ ЕЛЕКТРОННИ**

**СИСТЕМИ към ТЕХНИЧЕСКИ**  **УНИВЕРСИТЕТ - СОФИЯ**

Дата на заданието: 10.11.2020 г. Утвърждавам:.............................. Дата на предаване: 15.02.2021 г.

**ЗАДАНИЕ**

Мнение на научния ръководител

Увод:

Глава Първа

# Методи и технологии за реализиране на iOS приложения

1.1Мобилно приложение

**Мобилното приложение** е компютърна [програма](https://bg.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8E%D1%82%D1%8A%D1%80%D0%BD%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0) или софтуерна апликация, създадена, за да работи на мобилни устройства като [смартфони](https://bg.wikipedia.org/wiki/%D0%A1%D0%BC%D0%B0%D1%80%D1%82%D1%84%D0%BE%D0%BD) и [таблети](https://bg.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B5%D1%82). Обикновено приложенията са достъпни на онлайн платформи за диситрибуция като “App Store” или “Google Play Store”. Обикновено платформата е построена и се оправлява от компанията собственик на операционната система на устройството.

<https://www.techopedia.com/definition/2953/mobile-application-mobile-app>

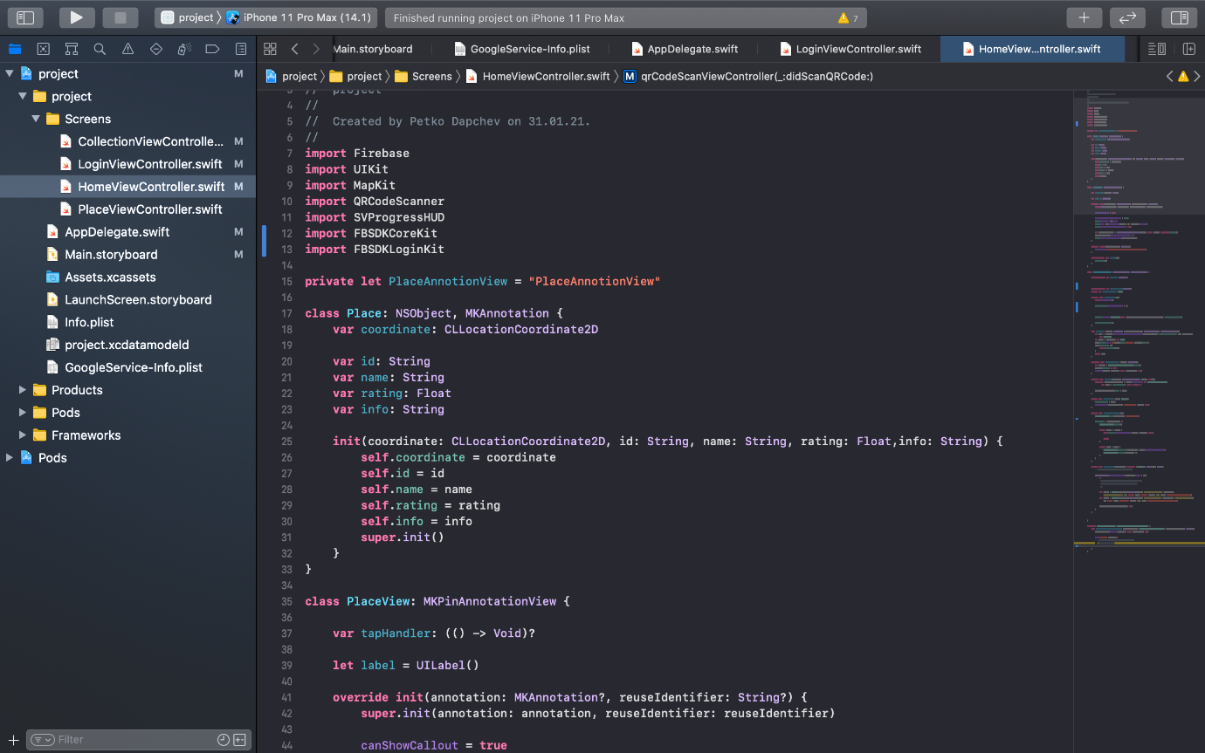
1.2 iOS мобилно приложение.

**iOS** е мобилна [операционна система](https://bg.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) на компанията [Apple Inc.](https://bg.wikipedia.org/wiki/Apple_Inc.) Разработена първоначално за [iPhone](https://bg.wikipedia.org/wiki/IPhone), тя се използва и в [мобилните устройства](https://bg.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B1%D0%B8%D0%BB%D0%BD%D0%BE_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE) [iPod Touch](https://bg.wikipedia.org/w/index.php?title=IPod_Touch&action=edit&redlink=1), [iPad](https://bg.wikipedia.org/wiki/IPad" \o "IPad). Тези приложения се използват само и единствено от продукти използващи операционната система iOS и могат да бъдат изтеглени само през платформата на Apple Inc. “App store”. Аpp Store е платформата, в която се публикуват готовите приложения за iOS устройства. Докато някои приложения са безплатни и разчитат на микротранзакции или реклами за печалба, то други предпочита да поставят фиксирана цена на продукта си. В този случай печалбата се разделя между разработчика и платформата по предварително договорени условия.

1.2.1 Създаване на мобилно приложение

Най-важното при старта на разработката на мобилно приложение е да се измисли идея. Да се изберат софтуер и средства, които ще се използват при реализацията.

1.2.2 Езици за програмиране

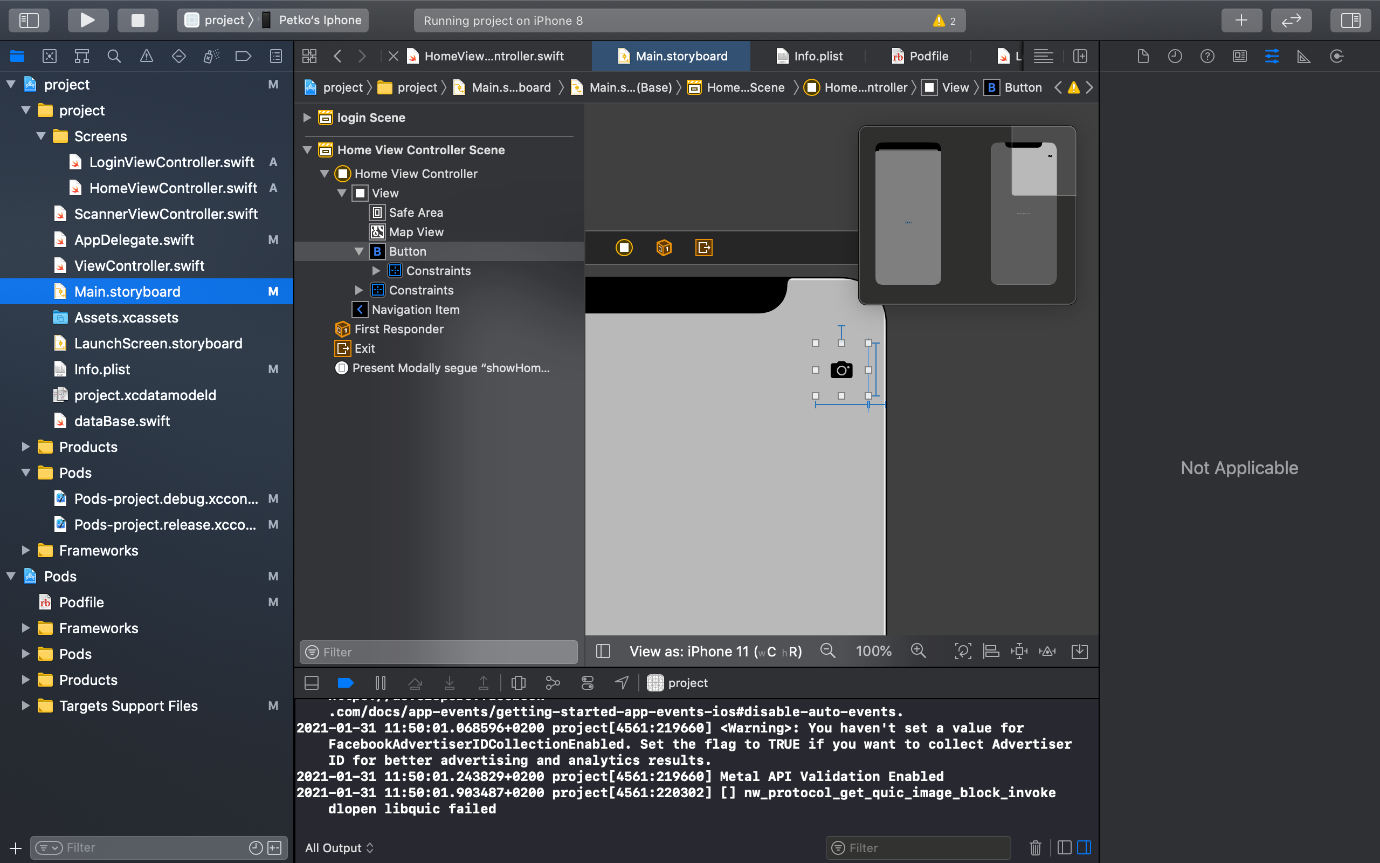
При разработката на мобилно iOS приложение освен популярните езици базирани на C (C++/C#) се използва и Swift. Swift e open source програмен език, създаден през 2014 от Apple Inc. специално за разработка на iOS приложения. Базиран е на езикът „C“ . Swift е разработен така че да има огромен спектър на функционалност, тъй като светът на мобилните приложения е много широк. Той е от т.нар. „compiled languages”, които се характеризират с това, че превръщат source code в машинен код. Основната среда за разработка на iOS приложения на езика Swift е Xcode. 

<https://developer.apple.com/swift/>

1.2.3 Xcode

Xcode е платформа, създадена от Apple Inc. , ексклузивно за разработването на приложения и програми за устройства, използващи операционните системи на компанията. Тази платформа се използва масово, защото функционалността на платформата е максимално съвместима с операционните системи на компанията, изграждайки много добра връзка между устройството и кода.

Чрез своя много детайлен симулатор, Xcode е много “user-friendly” и удобен за използване и тестване. Именно това го прави предпочитаната платформа за разработка на iOS приложения.

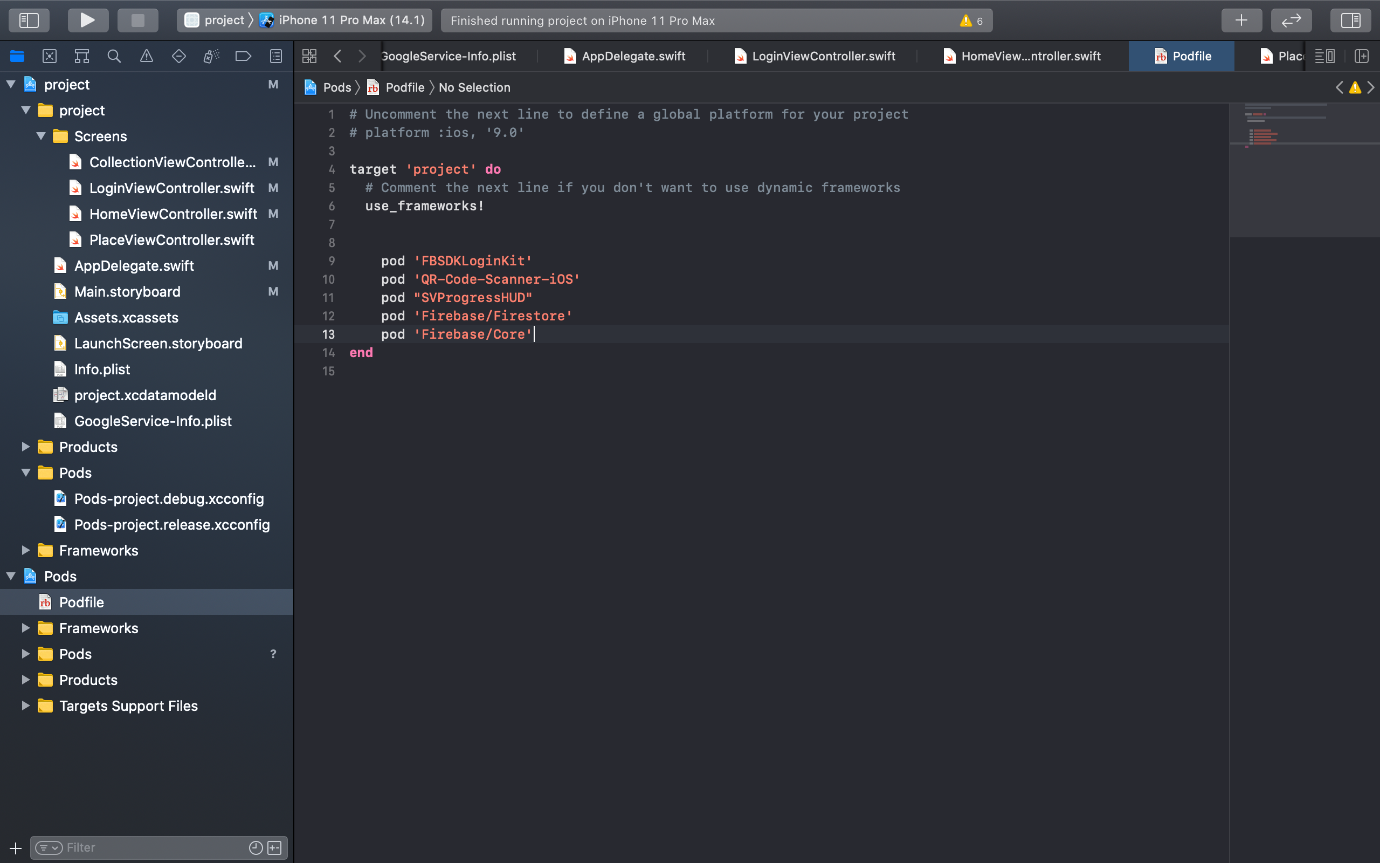


<https://developer.apple.com/xcode/>

1.2.4 CocoaPods

CocoaPods e често използвана програма за управление на външни библиотеки в Swift. Фокусира се върху разпространението независими библиотеки и интегрирането им в кода на програмата, която се разработва. Предимствата на тази технология е улесняването на разработчика при програмирането на конкретни функции, вече достъпни в споменатите библиотеки.

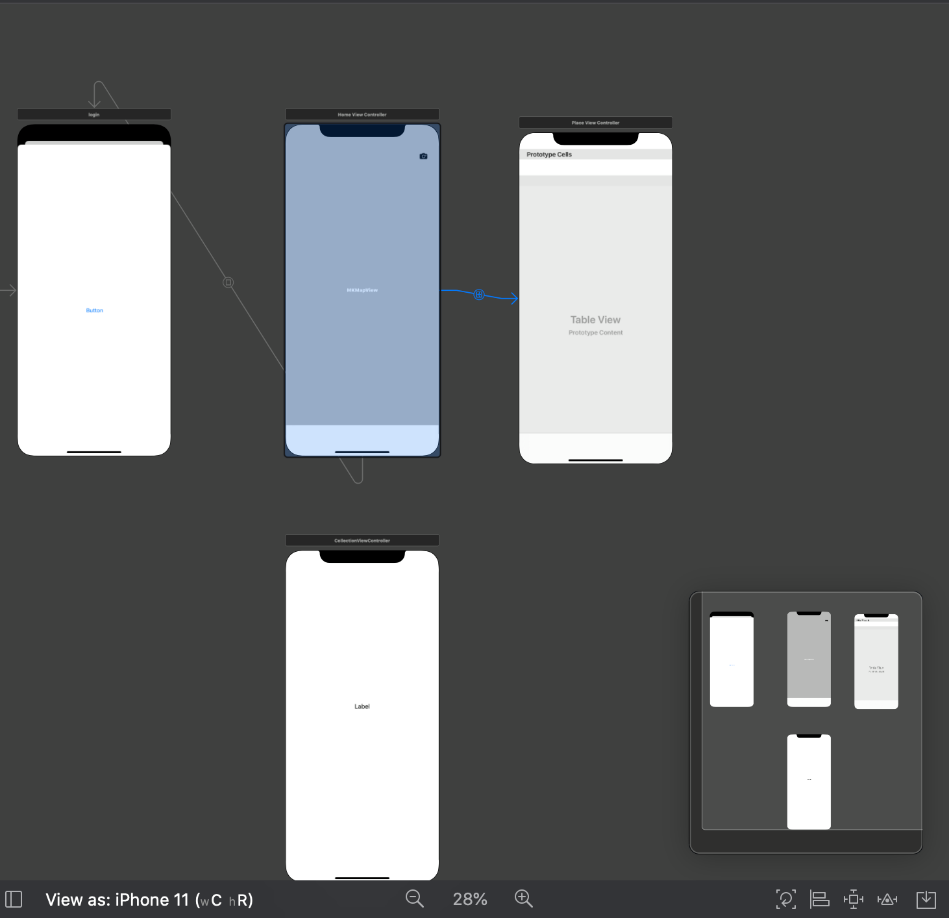
https://cocoapods.org/



1.2.5 Storyboards

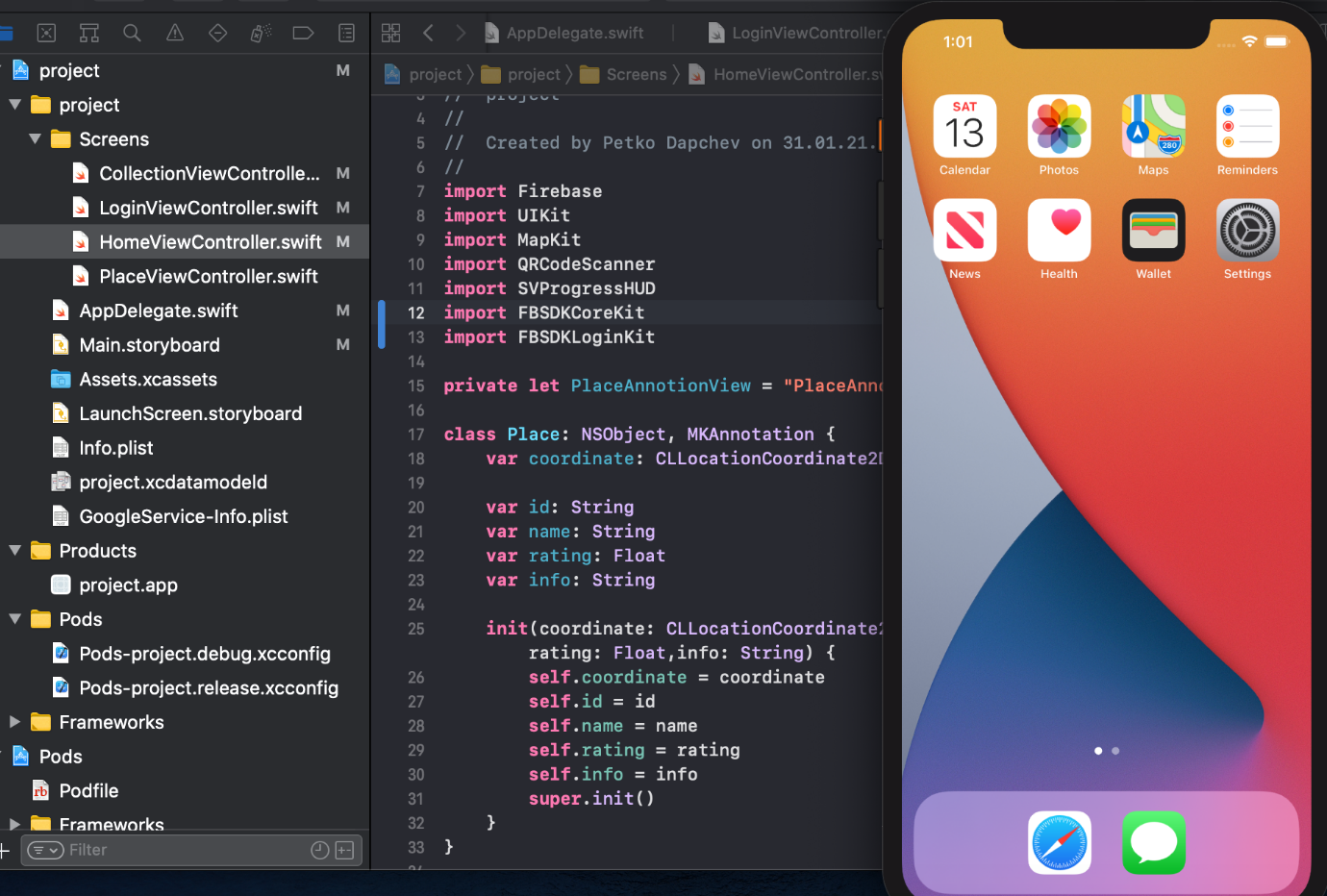
Едно от другите големи предимства на Xcode е именно опцията Storyboard. Това е много удобна технология, която позволява графически да се оформя приложението в самия софтуер. Така нагледно може да се види на симулатор как изглежда дизайна, което много улеснява разработчика.

Освен това много лесно се свързват части от Storyboard към самият код. Може ясно да се направят segues (препратки) между две страници на едно приложение, което подпомага навигацията в него на самия потребител.



1.2.6 Симулатор

Много важно за едно мобилно приложение е то да бъде тествано. За тази цел Xcode има един от най-добрите симулатори. Който е много детайлен, разполагащ с наличен симулатор за всяко едно устройство на Apple Inc. Това дава възможност на разработчика да вижда нагледно как би работило приложението върху реално устройство.



1.3 База данни

База данни е организирана информация, която обикновено се пази в електронно-достъпна компютърна система. DBMS( [database management system](https://en.wikipedia.org/wiki/Database" \l "Database_management_system)) e софтуерът, с който са свързани крайните потребители, програмта и самата база данни, където се взима цялата информация и се организира. Именно цялостната DBMS се асоциира с названието „база данни“ и обикновено като се използва израза „база данни“ се има предвид именно това.

1.3.1 MySql database

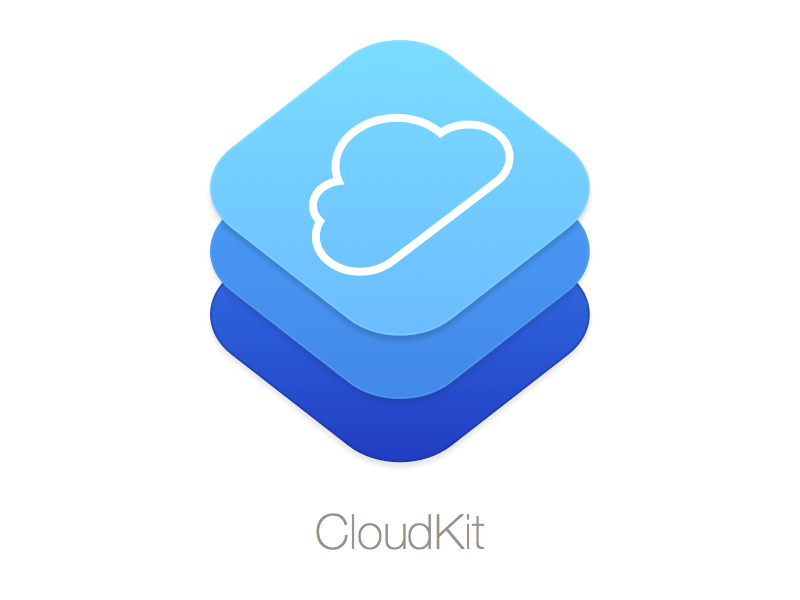
MySql е open-source система за релационни бази данни. Релационните бази данни се организират в таблици, които може да се свързват помежду си. Това е безплатен софтуер, който може да се използва самостоятелно, но в повечето случаи се имплементира в други програми и апликации. Но не е от най-подходящите при създаването на мобилно приложение.



<https://www.oracle.com/mysql/>

1.3.2 CloudKit

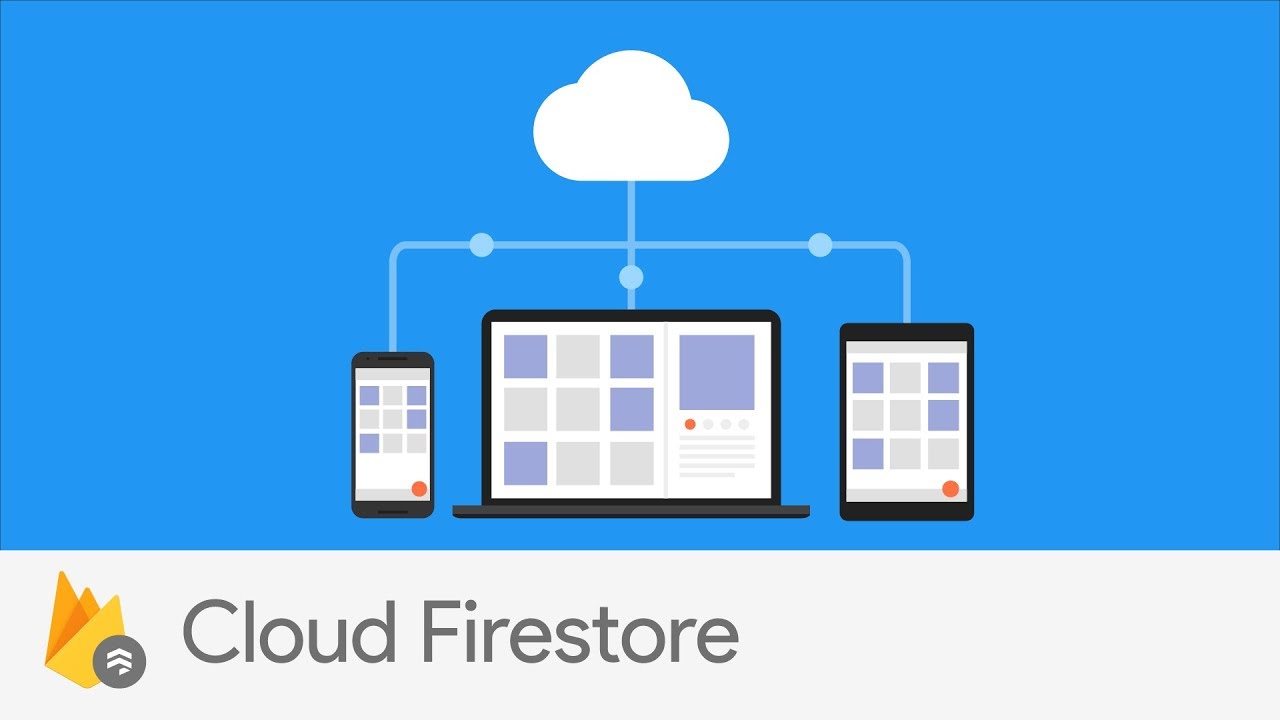
CloudKit е интерфейс вграден в операционните системи на Apple Inc. Основната му функционалност е именно пазенето на данни на web или мобилни апликации в отделено пространство в iCloud. iCloud e специалният облак на Apple Inc., който се използва от всички iOS или macOS устройства. Много удобен при разработката на мобилни и web приложения за устройства на Apple Inc., но няма безплатна версия.



<https://developer.apple.com/icloud/cloudkit/>

1.3.3 Cloud Firestore

Cloud Firestore e част от платформата Firebase, създадена от Google, с цел създаване на мобилни и web апликации. Cloud Firestore е нерелационна база данни, в която се съхранява информация. Тя може да се синхронизира с приложенията на потребителите, което я прави много удобна за мобилни приложения. Също така предоставя и offline функции, които намират много употреба с цел приложението да може да се използва винаги и по всяко време независимо от връзката с интернет.



https://firebase.google.com/docs/firestore

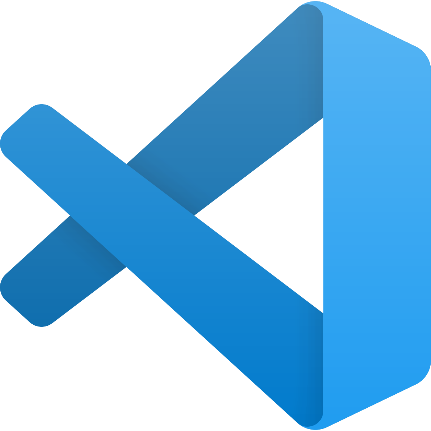
1.4 Текстови редактори

Текстови редактор е компютърна програма, предназначена за писане редактиране на текстови файлове. Текстовите редактори се делят на два типа: едни, ориентирани за писане на текст(word processors) и други, насочени към писането на технически текст(код). При създаването на приложения се използва вторият тип.

<https://www.techopedia.com/definition/5478/text-editor>

1.4.1 Visual Studio Code

Visual Studio Code е един от най-популярните текстови редактори сред програмистите и поддържа голямо разнообразие от програмни езици под формата на extensions (удължения). То работи само на операционните системи: Windows и Linux, затова е решение само за разработчици, които използват тези операционни системи и пишат iOS приложения.



<https://code.visualstudio.com/learn>

1.4.2 Xcode

Платформата на Apple Inc. съдържа собствен текстови редактор, специално предназначен за езика Swift и разработването на web и мобилни апликации. Xcode има и опциите за създаване, компилиране и тестване на приложенията.

Втора глава

# Изисквания към приложението.

2.1.1. Регистрация и аутентикация със социална мрежа

2.1.2 Удостоверяване на посещение на забележителност с QR код

2.1.3 Заснемане и споделяне на снимките от забележителностите в социалната мрежа

2.1.4. Рейтинг и ревюта на забележителностите

2.1.5. Система за прогрес и виртуални награди

2.1.6. Интернационализация на информацията за обектите

2.1.7. Карта за визуализиране на обектите

2.1.8. Нотификация при преминаване покрай непосетен обект

2.1.9. Статистика за посещаемост на обектите по часове и дни.

2.2 Аргументация на избора на средите за разработка

2.2.1 Xcode

Xcode e специално изграден и формулиран в своята цялост да се възползва от предмиствата на устройствата на Apple Inc. Освен това е напълно безплатен за употреба. Единствената законно одобрена програма за създаването на iOS приложения специално за конкретните операционни системи. Много удобен за използване, тъй като отчита непрекъснато синтаксиса и го проверява за грешки и предложения за debug. Също така си заслужава усилията за научаване при разработката на приложения, тъй като превъзхожда абсолютно всички конкурентни платформи.

2.2.2 Swift

Моят избор за програмен език е Swift. Езикът Swift е създаден от Apple Inc. специално за разработката на web и мобилни приложения на iOS и macOS. Той е фаворит сред езиците, използвани при програмирането на iOS приложения.

2.2.3 CocoaPods

CocoaPods e програма за вкарването на външни библиотеки към приложението. Те значително улесняват работата на разработчика, като му спестяват нуждата и времето да прави неща, които вече съществуват. И са доста често използван метод при разработката на iOS приложение.

2.2.3.1 FACEBOOK SDK за iOS

Този SDK се добавя към проекта чрез CocoaPods. Това позволява да интегрираме Facebook в нашето приложение. Както и да използваме функции на Facebook като Login(влизане) или Sharing(споделяне).

<https://cocoapods.org/pods/Facebook-iOS-SDK>

2.2.3.2 Firebase SDK за iOS

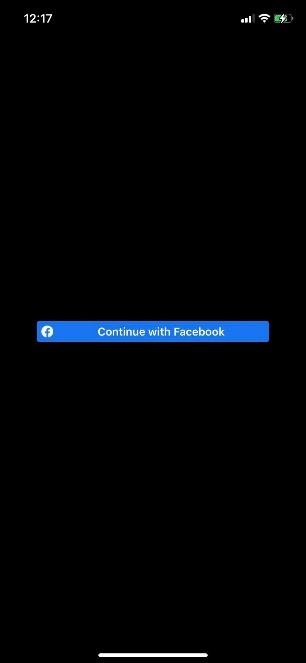
Този SDK позволява на приложението да се свърже с базата данни, която се намира в облака на Firebase – Firestore. Така то ще може да обменя информация със съответната база данни.

2.3 Навигация в приложението

Навигацията в моето мобилно iOS приложение се извършва чрез Storyboard. При определени събития в приложението се изпълняват segues, които отговарят за прехвърлянето от един екран (ViewController Scene) в друг. Като така навигацията се изпълнявa бързо и сигурно.

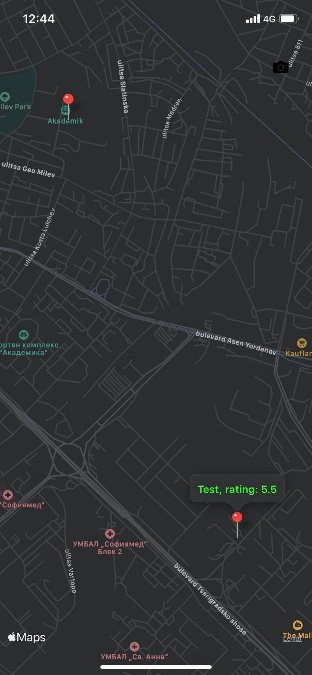
2.3.1 LoginView

Приложението ще се стартира с екран за вход в социална мрежа. Когато потребителят успешно влезе в своя профил, ще бъде прехвърлен чрез segue към началната страница.



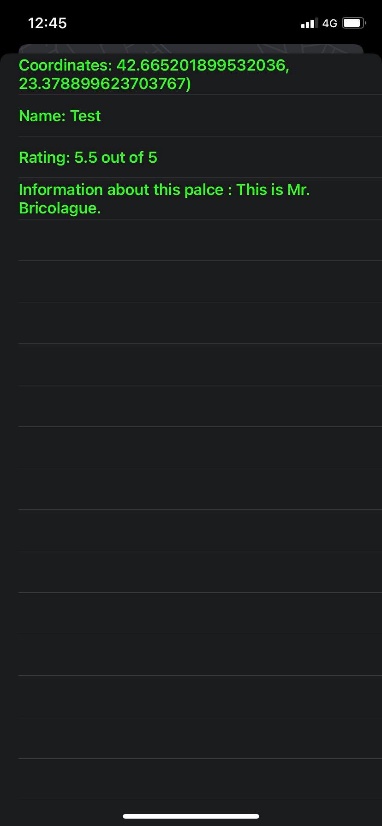
2.3.2 HomeView

Това е началният екран, който ще се състои от карта, върху която ще се нанесени местоположенията на забележителностите, зададени в базата данни. Когато потребителят натисне някой от маркираните забележителности по картата, чрез друг segue ще бъде прехвърлен в отделна сцена с информация за мястото. Друга функция ще е при натискането на специален бутон, да се отваря нов екран, в който да се намира QR четеца, който ще разпознава името на конкретното място.



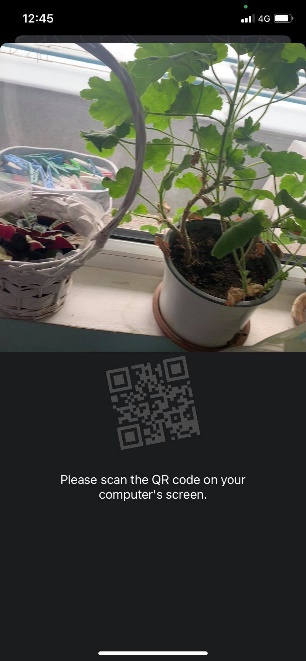
2.3.3 PlaceView

Тук ще се изкарва информация за конкретния обект, избран от потребителя.



2.3.4 QRView

В този прозорец потребителя ще бъде питан за разрешение за достъп до камерата и ще разчита QR код, ако е представен пред камерата.

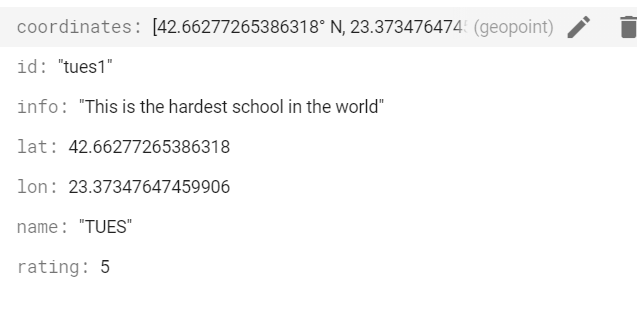


2.4 База данни

Моята база данни, която се намира в Cloud Firestore ще съдържа два основни документа(таблици). Едната ще има атрибутите на едно място (име, координати, информация, рейтинг и променлива, която следи дали обекта е бил посетен от конкретния потребител). Вторият документ ще е за потребител (там ще се съдържа уникалния идентификационен номер, facebook email и колекция от обектите, които е посетил).

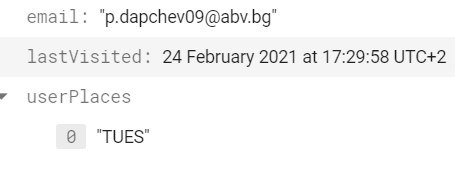
2.4.1 Places (Таблицата за обектите)

Тук се нанасят информация за обектите, която после се тегли в приложението и се използва за конструирането им върху картата.



2.4.2 User (Таблицата за потребителите)

В тази таблица ще се записва информацията за всички потребители, които се регистрират в приложението.



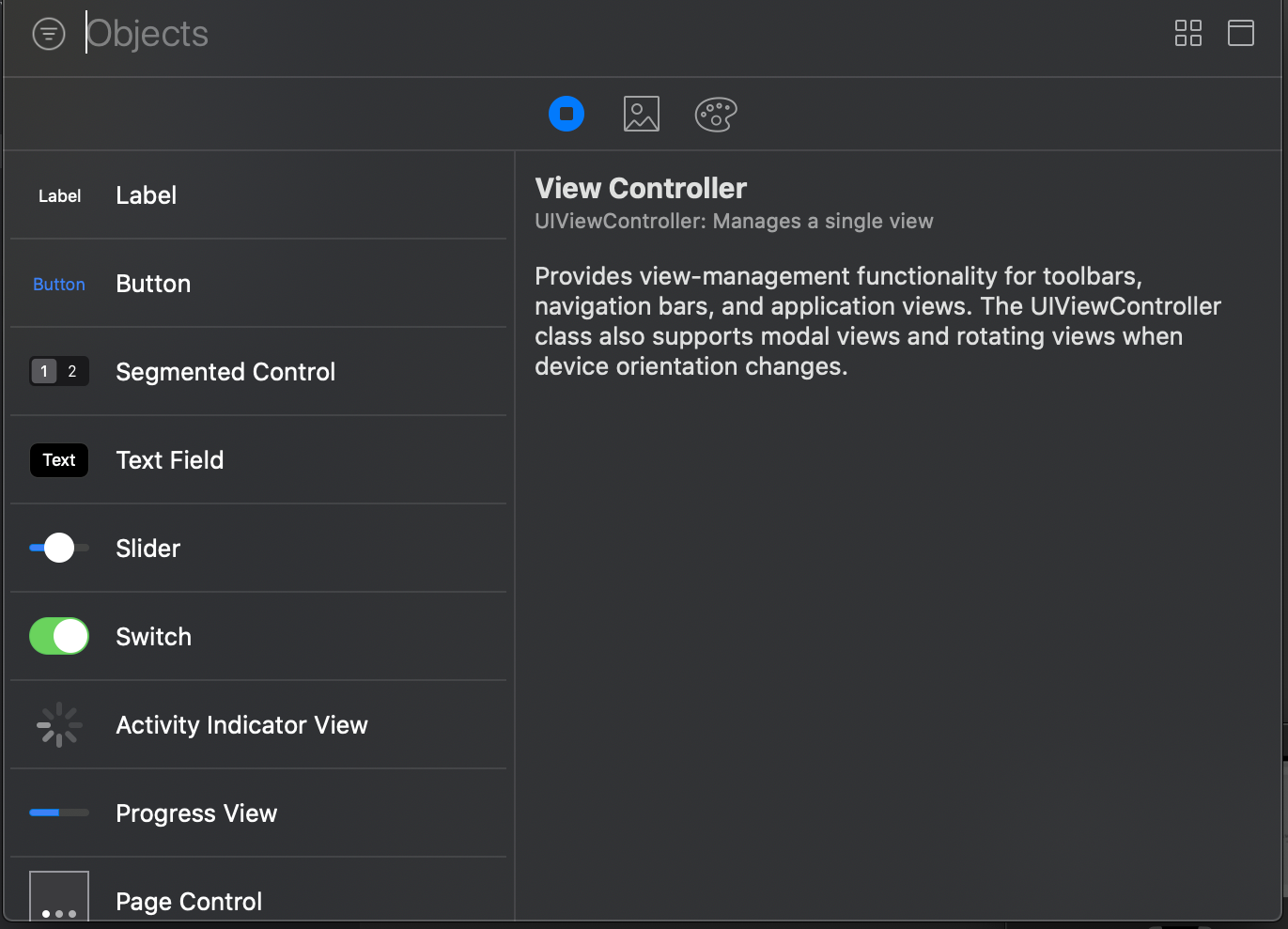
2.5 Тестване на програмата.

За да тествам моето приложение преди да го сваля на реално iOS устройство, го пускам на различни устройства в вградения симулатор на Xcode, за да съм сигурен, че ще работи във всеки един случай.



2.6 Обекти

В Xcode, за да се покажат съответните обекти на екрана, както ги искаме трябва да използваме вградените в програмата обекти (предефинирани от Apple Inc. класове), които помагат за изграждането на интерфейса на приложението.

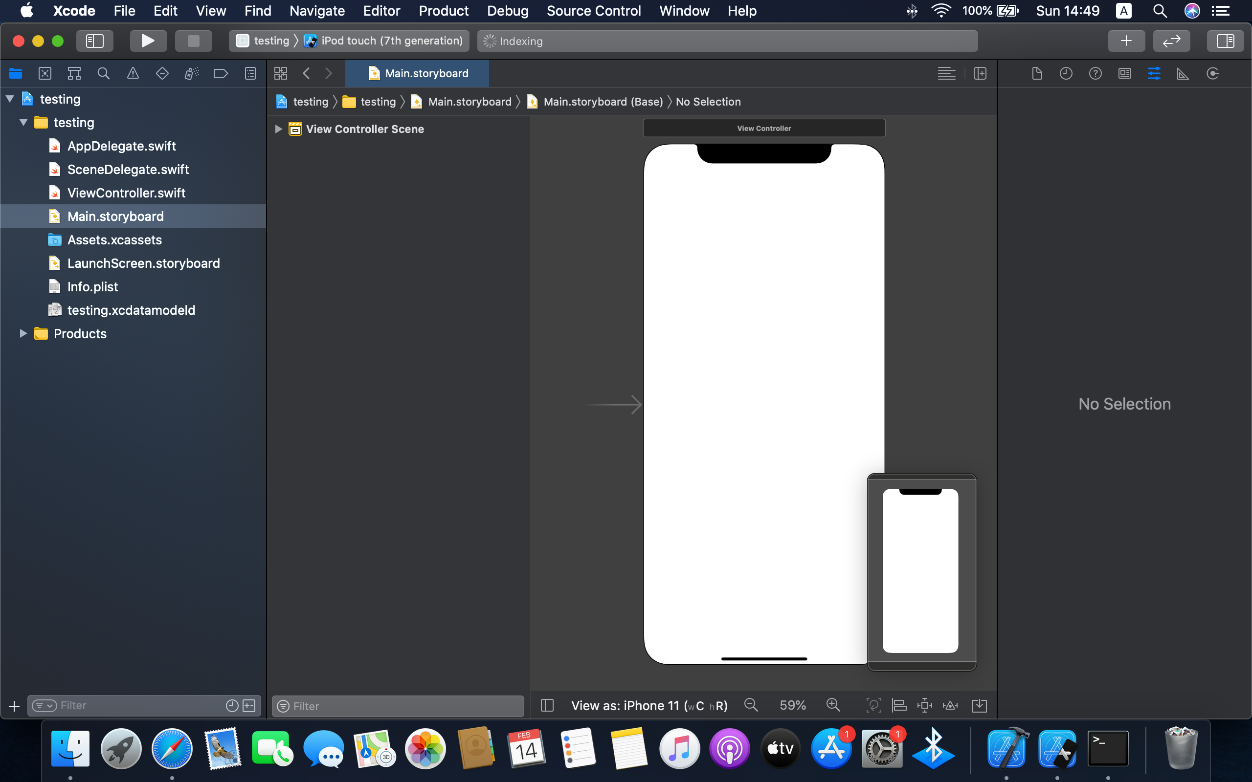


Трета глава

# Същинска част. Разработка на приложението

3.1Начално приложение

Когато се създаде нов проект в Xcode, той съдържа 4 основни файла: AppDelegate.swift; ViewController.swift; Main.storyboard; Info.plist



3.1.1 АppDelegate.swift

Този файл е изключително важен, тъй като служи като root(главен) файл за приложението. То осъществява връзките в системата. В него се описват точно тези връзки.

<https://developer.apple.com/documentation/uikit/uiapplicationdelegate>

3.1.2 ViewController.swift

Файловете от този тип са основата на инфраструктурата на всяко iOS приложение. Всеки контролер отговаря и управлява част от интерфейса на приложението и връзката между този интерфейс с back-end на това приложение. <https://developer.apple.com/library/archive/featuredarticles/ViewControllerPGforiPhoneOS/index.html>

3.1.3 Main.storyboard

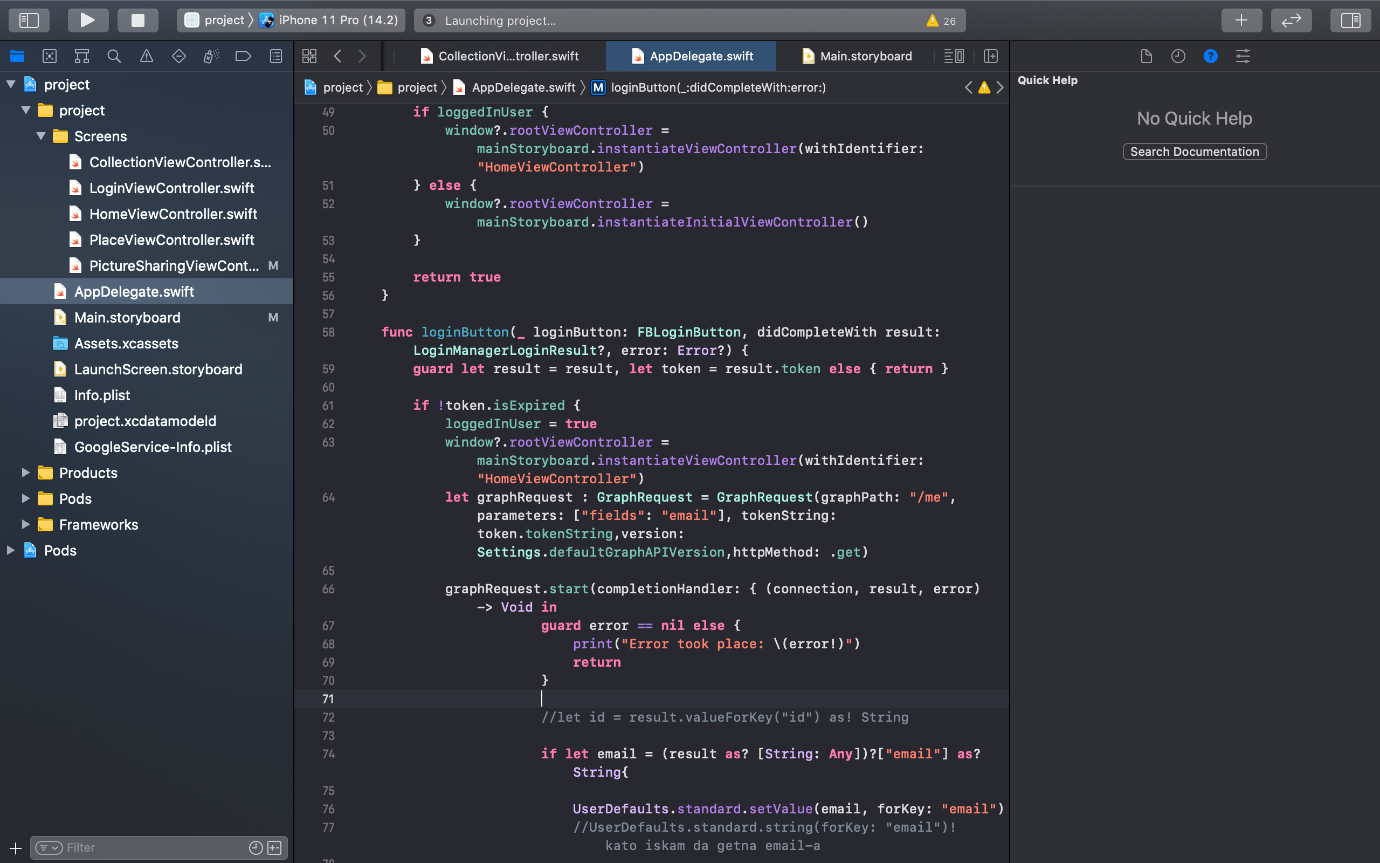
Това е модерна технология, характерна за Xcodе. Тя улеснява работата на разработчика при създаването на интерфейса на приложението. Чрез сторибордове се позволява лесно да се имплементират връзки между няколко ViewController-ите.

<https://www.raywenderlich.com/5055364-ios-storyboards-getting-started>

3.2 Реализация на изискванията

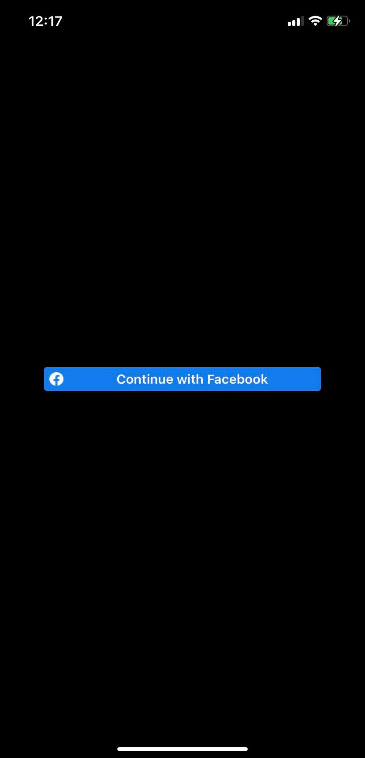
3.2.1 Регистрация и аутентикация със социална мрежа

За имплементирането на вход и аутентикация в социална мрежа се извършва със специален Pod(SDK) – FSBDK, което е специално предоставен от FacebookDevelopers с цел да се улесни връзката между приложението, което се разработва и популярната социална мрежа – Facebook. Дефинирайки бутон от тип FBLoginButton, приложението ще го използва като бутон с цел да излезе фейсбук меню, в което да въведе данните. В моето приложение предупреждавам, че имам право да виждам имейла на потребителя в GraphRequest.

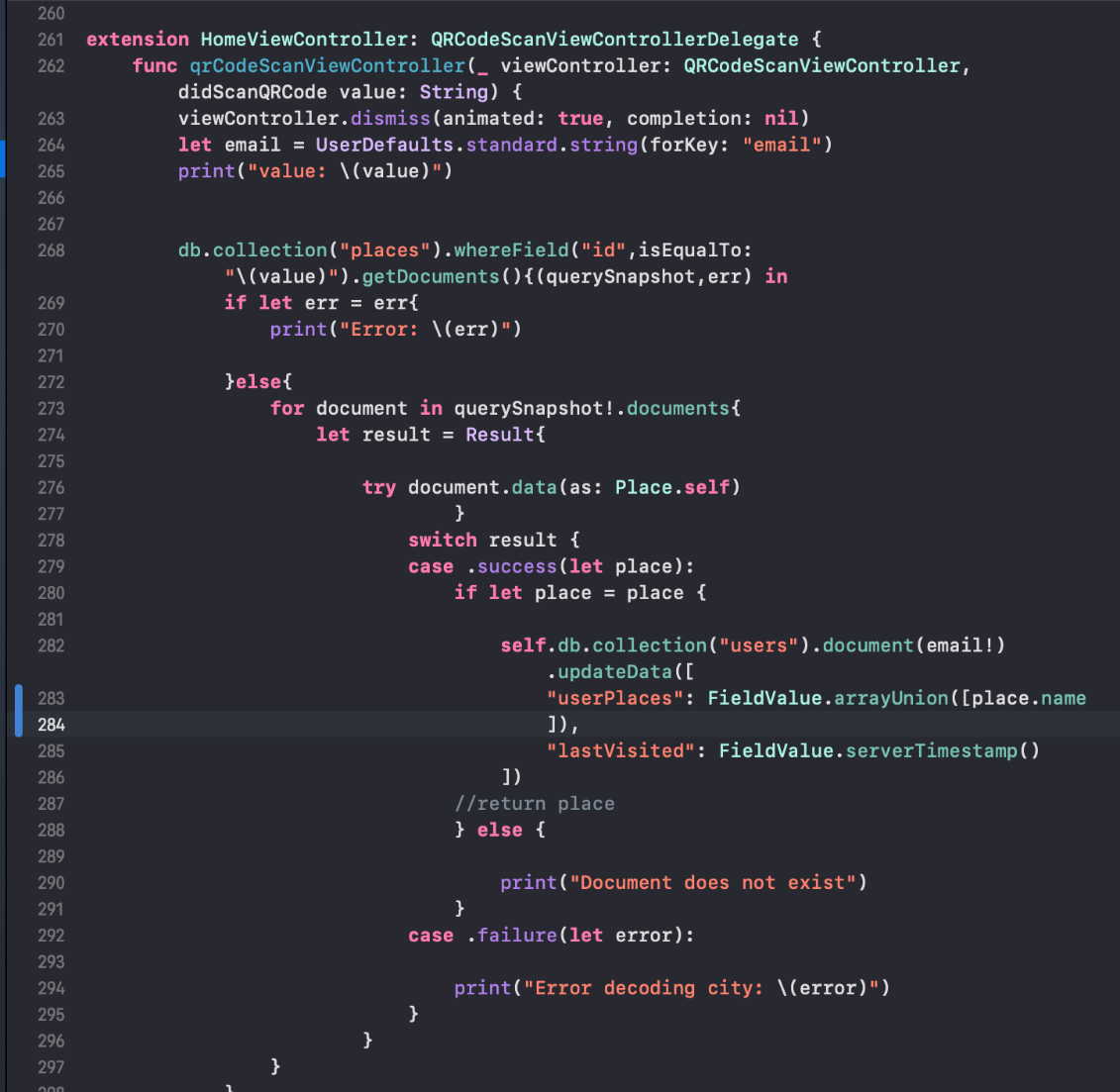


Чрез Boolean променливата - loggedInUser се проверява дали в приложението на телефона вече има потребител и ако има го препраща към HomePage, а ако няма се връща LoginViewController, където потребителя ще трябва да се аутентикира. Когато влезе имейла автоматично се запазва в базата данни.

ViewController-a изглежда така:



3.2.2 Удостоверяване на посещение на забележителност с QR код

За тази цел използвах “QR-Code-Scanner-iOS” SDK. Този SDK предоставя интерфейс и QR четец, който връща съответния резултат от кода. Но разбира се трябва да се записват единствено обекти, които присъстват в базата данни, за да не става объркване с други кодове. При сканиране на QR код обекта, на когото отговаря съответния код ще бъде добавян към посетените от потребителя обекти. Ако вече е там нищо няма да се случва. 

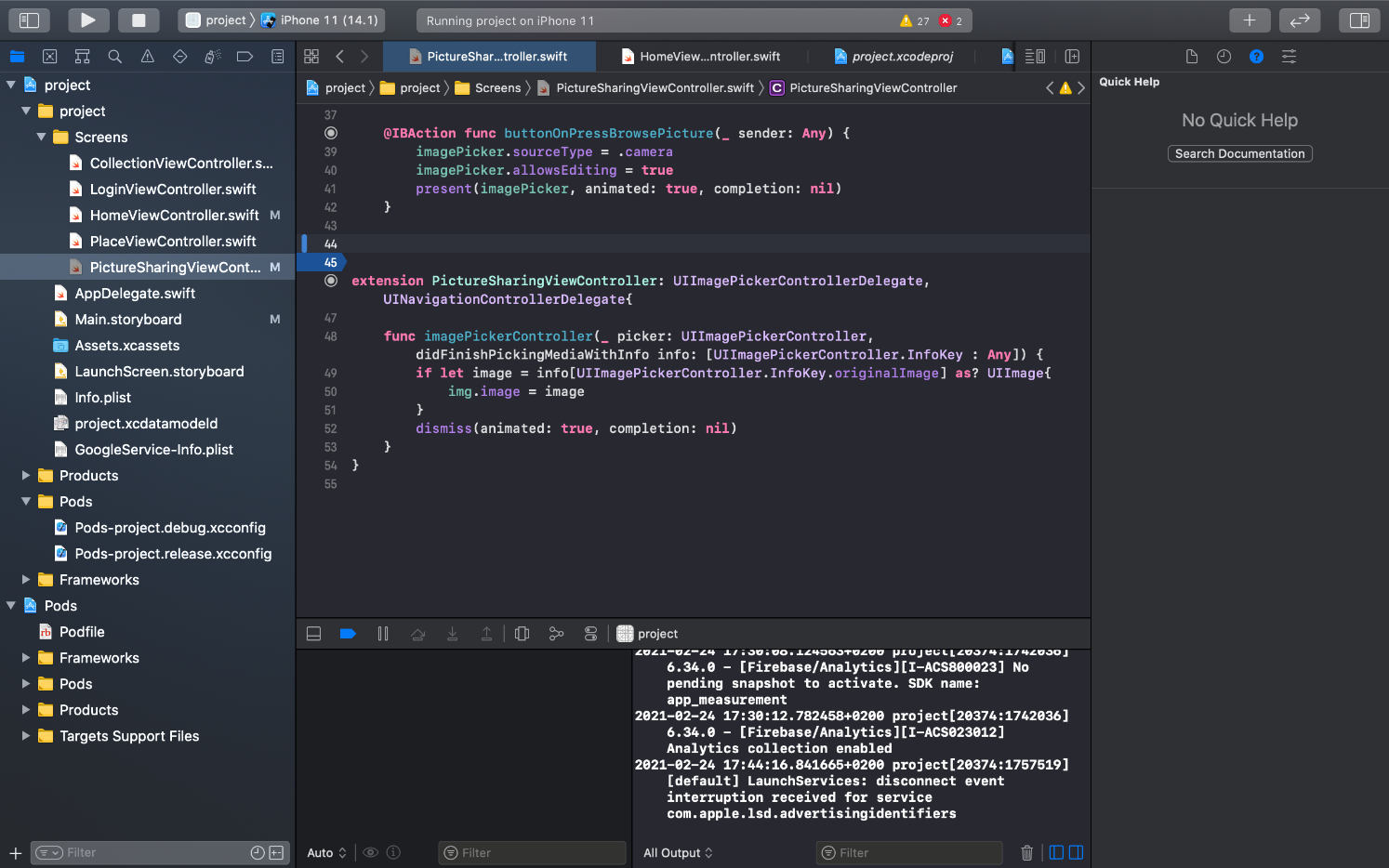
ViewController-a на скенера изглежда по този начин:



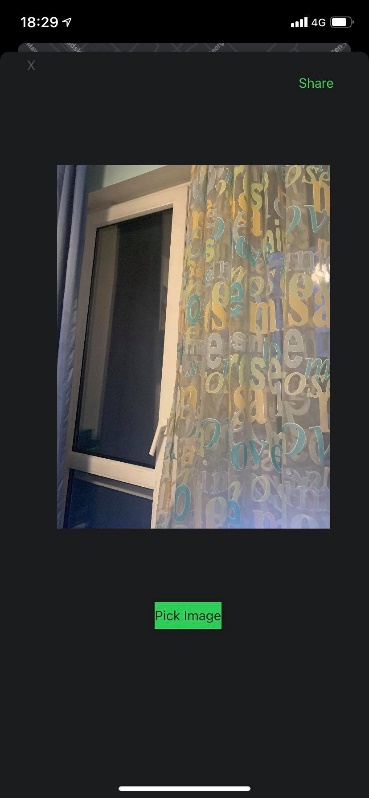
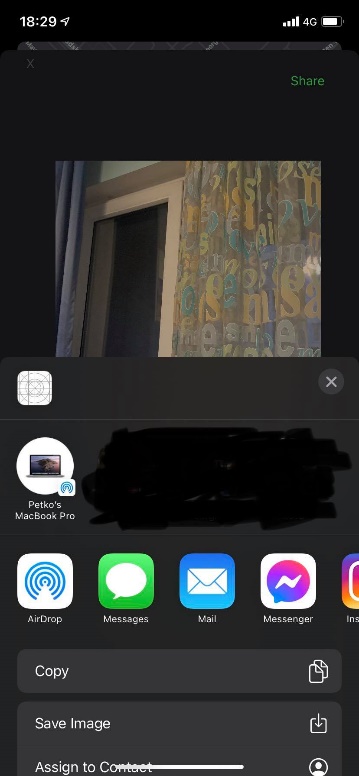
2.1.3 Заснемане и споделяне на снимките от забележителностите в социалната мрежа

Ако потребителите искат да споделят своя снимка от забележителностите, могат да се направят от PictureSharingViewController-a. Дефинирайки imagePicker(Контролер, който управлява интеракции с потребителите и показва резултатие в делигирания метод). Използвайки този imagePicker, можем да използваме камерата на устройството.

<https://developer.apple.com/documentation/uikit/uiimagepickercontroller>



За share-ing се използва UIActivityViewController. Този контролер предоставен от Apple Inc. се използва именно за допълнителния прозорец който се показва в приложението, когато потребителя се опитва да сподели дадена снимка.

2.1.4. Рейтинг и ревюта на забележителностите

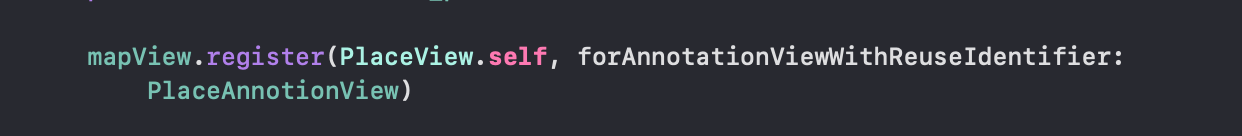
2.1.5. Система за прогрес и виртуални награди

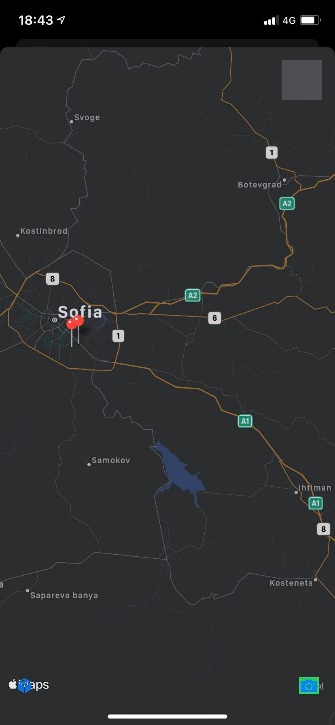
2.1.6. Интернационализация на информацията за обектите

2.1.7. Карта за визуализиране на обектите

2.1.7.1 Карта:

За картата използвах Map Kit View, което предоставя interface с картата, базиран на приложението на Apple Inc. – “Maps”. Този интерфейс е специално оптимизиран за Xcode и разработка на мобилни приложения. Като чрез класа MKMapView може да се обработва самата карта.





2.1.7.2 Обекти

Дефинирам отделен class Place(обект), който има съответните параметри: координати, идентификационен номер, информация за обекта, име рейтинг, географски дължина и ширина(последните два параметъра се използват за създаването на координатите). Като му присвояваме подклас MkAnnotation, за да се разпознава от картата като обект, който ще прави промени по MapView-то.

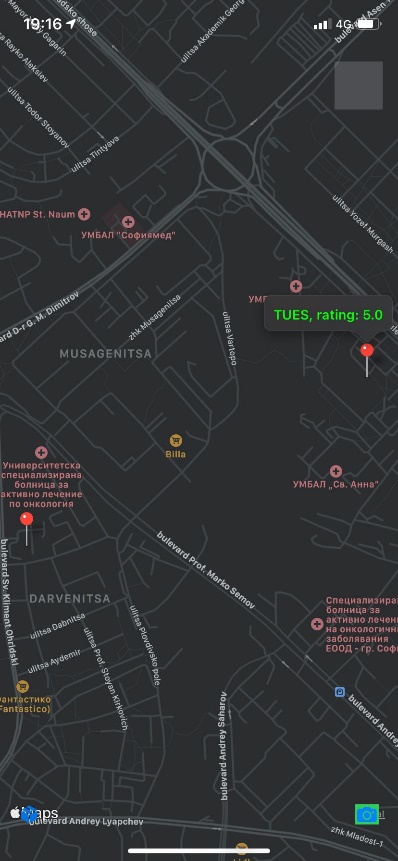
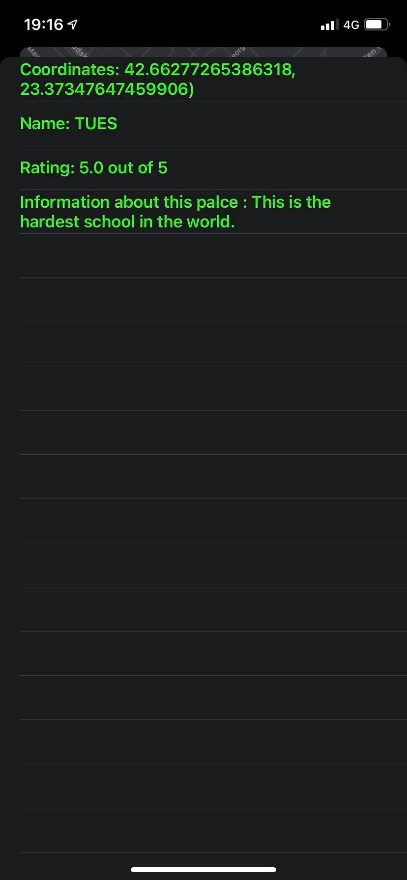


2.1.7.3 Поставянето на обекти върху картата

Това се прави в отделен клас – PlaceView, където поставяме обектите като МKPinAnnotation .Освен това при тяхното натискане като Label(), ще се показва сбита информация за конкретния обект. При докосването на съответния label ще се отваря друго меню с пълната информация за обекта.



В самия интерфейс изглежда по този начин:

2.1.9. Статистика за посещаемост на обектите по часове и дни.

# 