**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»,  профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

**Выпускная квалификационная работа**

на тему «IOS-приложение для навигации внутри помещения»

по направлению подготовки 09.03.04 «Программная инженерия»

|  |  |
| --- | --- |
| Научный руководитель Доцент департамента Программной инженерии Национального исследовательского университета «Высшая школа экономики», к.т.н.  Х.М. Салех  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись, Дата | Выполнил  студент группы БПИ151  4 курса бакалавриата  образовательной программы «Программная инженерия»  В.А. Будруев  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись, Дата |

**Москва 2019**

**Реферат**

Целью данной работы является разработка приложения для навигации внутри помещений и в любых других местах, где недоступны общепринятые навигационные технологии для устройств на платформе Apple iOS. Задача приложения заключается в сборе данных с сенсоров устройства и сканировании информации о расположенных в помещении BLE-маяков, обработке данной информации и последующем определении положения устройства в отсканированном помещении путем сравнения текущих данных с сенсоров устройства с предварительно собранными.

Результатом данной работы является приложение, позволяющее определить местоположение устройства в закрытом помещении, таком как подземная парковка, торговый центр или выставочный центр, для устройств на платформе Apple iOS.

Приложение позволит пользователям самостоятельно, не прибегая к услугам больших компаний сделать всю необходимую подготовку помещения, расположить в нем BLE-маяки, загрузить план-схемы, проложить маршруты для навигации и создать «электронный отпечаток» помещения, а также определять местоположение устройства в предварительно подготовленном помещении.

Работа содержит 39 страниц, 3 главы, 21 рисунок, 21 источник, 4 приложения.

**Ключевые слова**: BLE-маяк, iOS, отпечаток помещения

**Abstract**

The purpose of this work is to develop an application for indoor navigation and in any other places where conventional navigation technologies such as GPS are not available for devices on the Apple iOS platform. The task of the application is to collect data from the sensors of the device and scan information about the BLE beacons located in the room, process this information and then determine the position of the device in the scanned room by comparing the current data from the sensors of the device with those previously collected.

The result of this work is an application that allows you to determine the location of the device in a room where GPS technology is not available for devices on the Apple iOS platform.

The application will allow users themselves, without resorting to the services of large companies, to do all the necessary space preparation, place BLE beacons in it, upload a plan, lay routes for navigation and create an “electronic footprint” of the room, as well as determine the location of the device in a pre-prepared room .

The work contains 39 pages, 3 chapters, 21 figures, 21 sources, 4 annexes.

**Ключевые слова**: BLE-beacon, iOS, fingerprint

**Основные определения, термины и сокращения**

**LBS** – location-based services, это сервисы, ключевой функцией которых является определение местоположения пользователя и без нее они не смогут нормально функционировать.

**iOS** – это платформа от компании Apple, на которой работают все выпущенные компанией Apple мобильные устройства.

**Swift** – это язык программирования от компании Apple, на котором можно писать приложения для устройств на платформе iOS.

**MVC** – это паттерн проектирования, который расшифровывается как Model View Controller.

**BLE-маяк** – это устройство, которое излучает сигналы Bluetooth и тратит мало энергии на свое существование, может автономно работать на протяжении до 2х лет.

**Fingerprint** **(цифровой отпечаток помещения)** – это набор данных о силе сигнала BLE-маяков в помещении с привязкам к конкретным координатам положения устройства в помещении.

**Содержание**

[Реферат 2](#_Toc9767430)

[Abstract 3](#_Toc9767431)

[Основные определения, термины и сокращения 4](#_Toc9767432)

[Введение 7](#_Toc9767433)

[Глава 1. Анализ технологий, решений и подходов для навигации внутри помещения 10](#_Toc9767434)

[1.1. Показатели эффективности систем определения положения 10](#_Toc9767435)

[1.2. Методы и техники определения местоположения 11](#_Toc9767436)

[1.3. Беспроводные технологии, используемые в локализации 14](#_Toc9767437)

[1.4. Существующие готовые решения для навигации внутри помещения 15](#_Toc9767438)

[1.5. Функциональные требования к приложению 16](#_Toc9767439)

[Выводы по главе 17](#_Toc9767440)

[Глава 2. Принцип работы и архитектурные особенности 18](#_Toc9767441)

[2.1. Принцип работы приложения 18](#_Toc9767442)

[2.1.1. Режим настройки навигационного проекта 19](#_Toc9767443)

[2.1.2. Режим локализации и навигации 20](#_Toc9767444)

[2.2. Архитектурные особенности приложения 21](#_Toc9767445)

[2.2.1. Стек используемых технологий 21](#_Toc9767446)

[2.2.2. Архитектура клиентского приложения 22](#_Toc9767447)

[2.2.3. BLE-маяки и их использование 23](#_Toc9767448)

[2.3. Проектирование пользовательского интерфейса 24](#_Toc9767449)

[Выводы по главе 24](#_Toc9767450)

[Глава 3. Описание программной реализации и интерфейса iOS-приложения 25](#_Toc9767451)

[3.1. Инструменты разработки iOS-приложения 25](#_Toc9767452)

[3.2. Хранение данных в клиентском приложении 25](#_Toc9767453)

[3.3. Реализация модуля приложения для настройки навигации 28](#_Toc9767454)

[3.3.1. Core Location для работы с BLE-маяками 30](#_Toc9767455)

[3.3.2. Структуры данных для хранения цифрового отпечатка 32](#_Toc9767456)

[3.4. Реализация модуля приложения для локализации и навигации 32](#_Toc9767457)

[3.4.1. UIScrollView и png-изображение для карты 33](#_Toc9767458)

[3.5. Использованные сторонние библиотеки 35](#_Toc9767459)

[Выводы по главе 36](#_Toc9767460)

[Заключение 37](#_Toc9767461)

[Список использованных источников 38](#_Toc9767462)

**Введение**

Сервисы, основанные на местоположении пользователя, становятся неотъемлемой частью нашей жизни. Практически каждое мобильное приложение использует их для тех или иных целей: навигация, отслеживание, таргетированная реклама, общение, знакомства и прочие. Основа любых гео-локационных сервисов – это технология определения местоположения, чьей целью является определение местоположения пользователя. После определения местоположения пользователя в приложении, эта информация передается целому ряду внутренних встроенных сервисов, таким образом, качество и эффективность их работы напрямую зависят от точности определенного положения. Есть несколько популярных способов локализации устройства на улице и в помещении и самый популярный из них – гео-позиционирование на основе данных со спутников, таких как GPS или Глонасс.

Однако, GPS, как правило, не подходит для определения местоположения в закрытых помещениях, так как работает в микроволновом диапазоне, следовательно, крыши, стены и другие предметы ослабляют сигнал.

Общепризнанная технология для определения местоположения на улице – спутниковое гео-позиционирование. Но она очень плохо работает в условиях «городских джунглей», в подземных сооружениях, в зданиях и так далее, поскольку сигналы спутников слишком слабые, чтобы проникать сквозь бетонные стены, что делает данную технологию не применимой на практике.

Помимо этого, спутниковое гео-позиционирование не позволяет определить положение по оси Z, то есть, высоту над уровнем моря, этаж здания.

Индор-навигацией можно назвать любую систему, которая позволяет получить точное местоположение внутри закрытой инфраструктуры, такой как торговый центр, подземная парковка, метро, выставочные центры и так далее. Таким образом, разработка системы позиционирования внутри здании всегда сопряжена со следующими сложностями – маленькие расстояния, NLOS (high none line of sight), влияние на сигнал таких вещей, как стены, двери и прочие.

Таким образом, возникает проблема определения точного местоположения устройства в помещении. Существует множество различных технологий, которые позволяют решить данную проблему, но, в основном, они либо не применимы в повседневной жизни, либо очень дороги в применении, либо не обеспечивают требуемую точность. В основном, суть всех подходов для навигации внутри помещения сводится к размещению маяков, излучающих сигналы, считыванию силы данных сигналов на устройстве пользователя и определении примерного расстояния до них. В последнее время набирает популярность технология Bluetooth Low Energy маяков, среди основных ее плюсов: стоимость устройства, простота использования и настройки, длительность автономной работы.

Данная работа посвящена разработке приложения, позволяющего пользователям самостоятельно настроить и использовать навигацию внутри помещений для устройств на платформе Apple iOS. Целью работы является разработка приложения, суть которого заключается в сборе данных с сенсоров устройства и сканировании информации о расположенных в помещении BLE-маяков, обработке данной информации и последующем определении положения устройства в отсканированном помещении путем сравнения текущих данных с сенсоров устройства с предварительно собранными.

Результатом данной работы является приложение для устройств на платформе Apple iOS, позволяющее определить местоположение устройства в помещении, где недоступна технология гео-позиционирования.

Приложение позволит пользователям самостоятельно, не прибегая к услугам больших компаний сделать всю необходимую подготовку помещения, расположить в нем BLE-маяки, загрузить план-схемы, проложить маршруты для навигации и создать «электронный отпечаток» помещения, а также определять местоположение устройства в предварительно подготовленном помещении.

Для достижения целей работы требуется решить следующие задачи:

1. Проанализировать существующие технологии и подходы определения местоположения в помещении;
2. Изучить основы взаимодействия устройств на платформе Apple iOS с BLE-маяками;
3. Разработать модуль приложения для сбора навигационных данных и формирования «электронного отпечатка» помещения;
4. Разработать модуль приложения для определения местоположения устройства в предварительно отсканированном помещении;
5. Разработать приложение, представляющее собой комплексное решение для навигации внутри помещения;
6. Провести тестирование приложения;
7. Разработать техническую документацию;

Документ имеет следующую структуру:

1. В первой главе рассмотрены существующие технологии, подходы и решения для навигации внутри помещения, метрики, по которым их можно сравнить, на основании этого сформулирована актуальность разрабатываемого приложения и поставлены задачи на разработку;
2. Во второй главе рассматривается архитектура приложения, выбранные технологии, алгоритмы, модели и их применение в работе;
3. В третьей главе описывается реализация приложения для устройств на платформе Apple iOS, описываются использованные подходы и инструменты разработки;
4. **Анализ технологий, решений и подходов для навигации внутри помещения**
   1. **Показатели эффективности систем определения положения**

Основными показателями эффективности систем определения местоположения являются: точность, отзывчивость, покрытие, адаптируемость, масштабируемость, стоимость и сложность реализации [4].

Начнем с точности. В данном случае – это погрешность в определении положения, а точнее разница между определенным положением и реальным положением пользователя. Иногда точность также называют областью неопределенности, то есть, чем выше точность, тем лучше система.

Отзывчивость – это показатель, который говорит нам о том, на сколько быстро система определит измененное местоположение после того, как пользователь совершит передвижение.

Покрытие очень тесно связано с точностью и, на самом деле, может быть классифицировано как локальное покрытие, масштабируемое покрытие и глобальное покрытие. Проблема определения покрытия сети для обозначенной области важна при оценке эффективности системы определения местоположения. Местное покрытие – это небольшая, четко определенная ограниченная область, которая не может быть расширена (например, отдельная комната или здание). В этом случае указывается размер покрытия (например, (м), (м2) или (м3)). Масштабируемое покрытие означает системы с возможностью увеличения площади за счет добавления аппаратного обеспечения, а глобальное покрытие означает производительность системы во всем мире или в пределах желаемой или указанной области.

Следующая метрика – адаптируемость. Изменения влияния окружающей среды могут повлиять на производительность системы локализации. Способность системы локализации справляться с этими изменениями называется ее адаптивностью. Система, которая способна адаптироваться к изменениям окружающей среды, может обеспечить лучшую точность локализации, чем системы, которые не могут адаптироваться. Адаптивная система также может предотвратить необходимость повторной калибровки.

Масштабируемость – это очень важная метрика, которая может сообщить нам о том, насколько хорошо будет работать система с увеличением количества пользователей, с ростом покрытия, на котором работает система и с учащением количества запросов на определение положения. Масштабируемость является желательным свойством практически любой системы, которая показывает, насколько хорошо система работает, когда она работает с большим количеством запросов местоположения и большим охватом. Плохая масштабируемость может привести к снижению производительности системы, что требует реинжиниринга или дублирования систем. Масштабируемая система позиционирования должна обрабатывать большое количество меток без лишних усилий.

Стоимость реализации системы складывается из нескольких показателей – это стоимость дополнительной инфраструктуры, которая потребуется для создания рабочей системы, дополнительные затраты для ее масштабирования, непосредственная стоимость написания самой системы, долговечность, тяжесть, энергетические затраты и влияние на природу самой системы. Если система позиционирования может повторно использовать существующую коммуникационную инфраструктуру, некоторая часть инфраструктуры, оборудования и полосы пропускания может быть сохранена. Сложность же определяется алгоритмами, которые использует система и созданием сигналов с физических датчиков, которые будут ей использоваться. Сложность обычно варьируется и прямо пропорциональна точности определения местоположения, чем точнее нужна система – тем сложнее будут алгоритмы, которые будут в ней использоваться.

* 1. **Методы и техники определения местоположения**

Используется несколько разных подходов, технологий и алгоритмов определения местоположения, основанных на технологии беспроводных сетей. Методы определения местоположения можно разделить на три основные категории: близость к источнику сигналов, триангуляция и создание электронного отпечатка помещения [5].

Определение близости к источнику сигналов.

Это один из самых простых способов позиционирования. Однако, точность данной технологии оставляет желать лучшего. Принцип работы заключается в следующем. В помещении размещаются датчики, количество которых примерно равно количеству возможных для определения местоположений пользователя. Устройство пользователя обнаруживает сигналы датчиков. Если обнаружен всего один датчик – то его местоположение принимается за местоположение пользователя. Если же несколько – то местоположение датчика с самым сильным сигналом принимается за местоположение пользователя. Точность данной техники определяется лишь количеством датчиков, которые будут размещены в помещении. Данная техника может быть реализована с помощью целого ряда беспроводных технологий определения местоположения: технологии на основе IR (инфракрасного излучения), RFID, GSM, Bluetooth и настраиваемые радиоустройства.

Триангуляция основана на геометрических свойствах треугольников и использует их для определения местоположения пользователя. Ее можно разделить на две основных категории алгоритмов: латерация и ангуляция Методы, которые основаны на измерении времени перемещения, такие как TOA, RTOF, TDOA и основанные на RSS и фазах принятого сигнала называются техниками латерации. Метод оценки угла прихода называется техникой ангуляции.

Угловой метод.

Данная техника определяет угол прихода сигнала от мобильного устройства, который поступает из известного местоположения, в котором он принимается на нескольких базовых станциях. Для определения местоположения на плоскости, данная техника требует наличия лишь двух маяков. Чтобы улучшить точность или находить местоположение пользователя в трехмерном пространстве, потребуется три и более маяка. При использовании направленных антенн или антенных решеток, можно находить направление движения пользователя. Но, у техник, основанных на угле прихода есть ряд своих серьезных минусов – достаточно высокая стоимость внедрения технологии, которая получается за счет дорогих антенн и практическая неприменимость алгоритмов на практике в навигации внутри помещений из-за того, что большое количество стен, дверей и других препятствий очень сильно изменяют угол путем большого количества отражений и преломлений. Это значительно снижает точность определения.

Методы, основанные на времени.

Латерация, трилатерация, мультилатерация. Все три термина обозначают технологии нахождения местоположения пользователя на основе данных о расстоянии и измерении этого растояния. Латерация и трилатерация определяет положения пользователя путем нахождения расстояний от него до опорных точек, положение которых для нас известно. На основании данных о расстоянии до опорных точек вычисляется точное местоположение пользователя.

Время прибытия.

Системы времени прибытия основаны на точной синхронизации времени получения сигнала, передаваемого с мобильного устройства на несколько приемных маяков. В данном методе мобильное устройство передает сигнал с меткой времени на приемные маяки. Когда он получен, расстояние между мобильным узлом и принимающими маяками рассчитывается из задержки времени передачи и соответствующей скорости сигнала.

Навигационное счисление.

Данная технология работает следующим образом: берется последняя, определенная алгоритмом позиция пользователя и на основании данных о его перемещении, скорости и направлении движения, полученных с помощью акселерометра, гироскопа и педометра мобильного устройства – постоянно изменяется. Данная технология относится к, так называемым, инерциальным системам определения местоположения пользователя и очень широко применяется. Основной недостаток данной техники в том – что ошибка определения местоположения является накопленной. Поэтому с каждой новой погрешностью ошибка увеличивается. Причиной является то, что новая позиция пользователя определяется исходя из предыдущей.

Создание отпечатка помещения.

Практически все подходы к реализации навигации внутри помещения используют отпечаток (fingerprint) помещения как основную технологию. Основная суть данного подхода заключается в том, чтобы собрать полный слепок данных о сигналах в помещении, в котором требуется осуществлять навигацию и сохранить ее в базу данных. Затем, во время определения местоположения, данные о сигналах с устройства пользователя сравниваются с данными в базе и берется ближайшее положение из базы данных. Этот метод не требует специального оборудования ни на мобильном устройстве, ни на принимающей стороне, а также не требуется синхронизация времени между станциями. Он может быть полностью реализован в программном обеспечении, которое может значительно снизить сложность и стоимость по сравнению с системами ангуляции или задержки, основанными исключительно на времени [6].

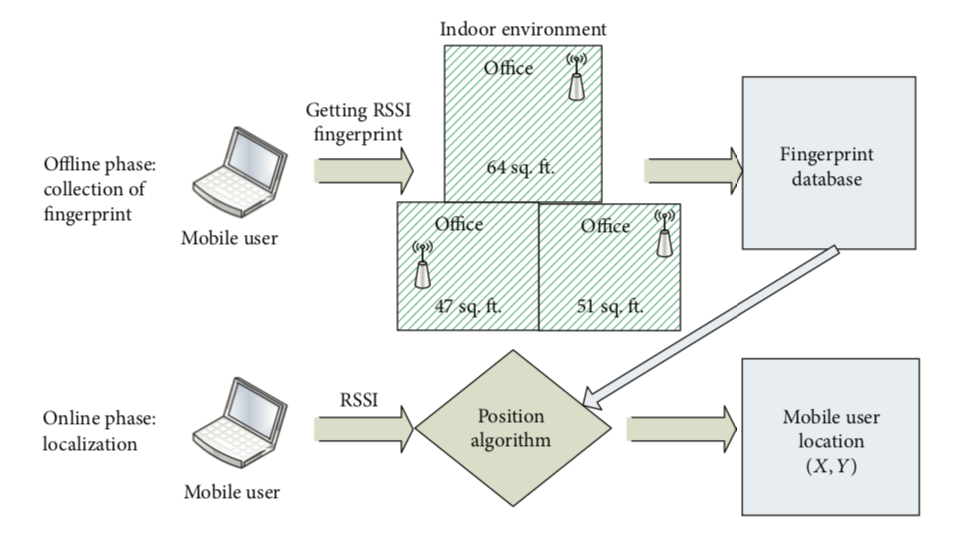


Рис. 1 – Технология отпечатка помещения

* 1. **Беспроводные технологии, используемые в локализации**

В этом разделе представлен обзор наиболее известных современных систем беспроводного позиционирования. Основное внимание уделяется системам на основе радиосвязи, особенно в позиционировании беспроводной локальной сети WiFi [7] и Bluetooth [8].

Глобальная система позиционирования (GPS) – это самая популярная и общепризнанная радионавигационная система для определения местоположения и положения объектов в пространстве. Тем не менее, она не очень хорошо работает в помещениях, потому что наличие препятствий на линии прямой видимости между спутником и приемником ослабляет силу сигнала. Поскольку люди проводят большую часть своего времени в помещениях, GPS не очень подходит для отслеживания местоположения.

Системы позиционирования с инфракрасным излучением (IR) являются одними из наиболее распространенных систем позиционирования, использующих беспроводные технологии. Спектральная область инфракрасного излучения используется различными способами для обнаружения или отслеживания объектов или людей и доступна в различных проводных и беспроводных устройствах, таких как мобильный телефон, компьютер и телевизор. В большинстве беспроводных устройств на основе ИК-излучения требуется режим прямой видимости (LOS) между передатчиком и приемником без помех от сильных источников света. Основными преимуществами использования системных устройств на основе ИК является маленький размер, легкость и простота в использовании. ИК-системы точно определяют положение в помещении. Помимо этого, системы внутреннего позиционирования, основанные на инфракрасном излучении, имеют ряд недостатков, таких как проблемы безопасности и конфиденциальности.

Радиочастотная идентификация (RFID) была признана будущей многообещающей технологией для обслуживания систем определения местоположения объектов или людей. RFID обеспечивает одностороннюю беспроводную связь с использованием бесконтактной и усовершенствованной технологии автоматической идентификации, которая использует радиосигналы. Отслеживание перемещений объектов в RFID осуществляется через сеть сканирующих радиоустройств на расстоянии нескольких метров. Технология RFID используется в широком спектре областей, включая автомобильную сборку, управление складом, сеть поставок и другие.

Bluetooth – это стандарт для беспроводных персональных сетей (WPAN). Почти каждое мобильное устройство с поддержкой WiFi, такое как мобильный телефон или компьютер, также имеет встроенный модуль Bluetooth. Bluetooth работает в диапазоне 2,4 ГГц ISM. Преимущество использования Bluetooth для обмена информацией между устройствами состоит в том, что эта технология отличается высокой безопасностью, низкой стоимостью, низким энергопотреблением и небольшим размером. Каждый Bluetooth-маяк имеет уникальный идентификатор, который можно использовать для определения местоположения. Одним из недостатков использования технологии bluetooth в локализации является то, что при каждом обнаружении местоположения выполняется процедура обнаружения устройства, из-за этого значительно увеличивается задержка локализации (на 10 – 30 секунд) и энергопотребление.

Разрабатываемое решение должно иметь доступную для пользователя стоимость и простоту настройки и реализации, эти ограничения накладывают на нас дополнительные ограничения в выборе технических средств и методов реализации приложения. Единственная технология, подходящая под наши требования – Bluetooth-маяки, которые достаточно только поставить в помещении и они не требуют практически никакой настройки. Каждое BLE-устройство отдает свой уникальный идентификатор и силу сигнала, по которой можно построить отпечаток помещения.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Система | Точность | Методы | Покрытие | Сложность | Стоимость |
| GPS | 6-10 м | Время прибытия | Вне помещений | Высокая | Высокая |
| ИК | 1-2 м | Дальность, время прибытия | В помещении | Низкая | Средняя |
| RFID | 1-2 м | Дальность, Время прибытия, Сила сигнала | В помещении | Низкая | Низкая |
| Bluetooth | 2-5 м | Сила сигнала, отпечаток помещения | В помещении | Низкая | Средняя |

Таблица 1. – сравнение технологий позиционирования

* 1. **Существующие готовые решения для навигации внутри помещения**

Проблема определения местоположения в закрытом пространстве появилась довольно давно и уже не один год множество компаний занимается разработкой промышленных решений для организации сложных систем позиционирования. Основными потребителями таких систем являются большие торговые центры, научные центры, университеты, медиа-пространства и экспозиционные центры, в которых часто проводятся конференции, выставки с множеством количеством однотипных павильонов, участников. Все эти решения объединяет большая цена, долгий срок внедрения, сложность коммуникации, отсутствие гибкости решения.

Основные компании, которые занимаются разработкой таких систем: Indoors Navigation, Visioglobe, Locus Labs.

Все решения основаны на Bluetooth маяках, WiFi-роутерах и на технологии предварительного сканирования «электронного отпечатка» помещения с дальнейшим сравнением полученных данных с сенсоров с первоначально собранной базой [9].

Большинство компаний не раскрывает внутреннюю информацию о алгоритмах, которые они применяют для повышения точности определяемого местоположения, поэтому сравнить решения не представляется возможным.

Прямых конкурентов и аналогов у разрабатываемого продукта нет, поскольку он нацелен на предоставление конечному пользователю возможности самостоятельно настроить навигацию в своем помещении с нуля до определения местоположения.

* 1. **Функциональные требования к приложению**

Разрабатываемое приложение должно позволять конечным пользователям с нуля самостоятельно без посторонней помощи настраивать навигацию в собственном помещении и в конечном итоге позволять определять местоположение устройства в пространстве, где недоступна технология GPS. Исходя из этого можно сформировать следующие функциональные требования к приложению:

1. Приложение должно предоставлять возможность создавать новый объект (помещение) для настройки навигации в нем;
2. Приложение должно предоставлять возможность загрузки план-схемы помещения в формате png;
3. Приложение должно предоставлять возможность расстановки основных путей внутри помещения, где будет проходить навигация и определение местоположения;
4. Приложение должно предоставлять возможность прохождения пользователя по путям и собирать электронный отпечаток помещения;
5. Приложение должно предоставлять возможность определения местоположения устройства в помещении, по которому был собран полный электронный отпечаток.

**Выводы по главе**

В первой главе были рассмотрены метрики эффективности локализационных систем, существующие методы и подходы к определению местоположения устройства в пространстве, описаны основные беспроводные технологии, которые чаще всего используются в локализационных сервисах, проведен анализ конкурентов и описаны преимущества разрабатываемого приложения. Описаны преимущества использования технологии Bluetooth, метода создания электронного отпечатка помещения и был обоснован выбор технологий. Также были описаны высокоуровневые требования к разрабатываемому приложению.

В следующей главе будет рассмотрена архитектурная организация приложения, описана работа Bluetooth-маяков и встроенных сенсоров устройств на платформе Apple iOS и предоставляемые ими возможности. Будут описаны детали реализации приложения.

1. **Принцип работы и архитектурные особенности**

В данной главе подробно описано устройство работы приложения, два основных больших процесса, на которые можно разделить все функции приложения, подробное описание всех конечных действий, которые может произвести пользователь, а так же архитектурные особенности подготовленного решения.

* 1. **Принцип работы приложения**

Весь функционал мобильного приложения можно разделить по принадлежности к двум основным процессам, в которые будет вовлечен пользователь при пользовании программой. Первый – это создание навигационного проекта под конкретное помещение и подготовка данного помещения к дальнейшему осуществлению навигации и локализации в нем, что включает в себя загрузку карт, создание этажей, расстановку путей навигации и сбор цифрового отпечатка помещения (рис. 2). Второй – собственно, сама навигация и локализация в заранее подготовленном помещении (рис. 3). При входе в приложение пользователю сразу будет предложено выбрать, какой режим он в данный момент хочет использовать – настройка или локализация в помещении.



Рис. 2 – Use Case Model режима настройки навигационного проекта



Рис. 3 – Use Case Model режима локализации

* + 1. **Режим настройки навигационного проекта**

Самое первое, с чего начинается пользование мобильным приложением – это создание нового навигационного проекта. Допустим, у пользователя есть некое помещение, будь то магазин, собственная квартира или любая другая площадь, будь то даже открытая площадка на улице. Пользователь придумывает условное название для данного помещения, вводит его в специальное поле и можно считать проект созданным. У каждого навигационного проекта помимо названия должно быть этажи и план-схемы. Если помещение расположено на одном уровне – это не является проблемой и пользователь просто создает один уровень. У каждого этажа навигационного проекта есть два обязательных параметра – это название (названия могут быть в виде «1 этаж», «2 этаж» и так далее, либо это могут быть условные названия раздельных помещений в рамках одного проекта «Оранжевая зона», «Фиолетовая зона» и так далее) и неделимая план-схема в формате png. Таким образом, после создания навигационного проекта пользователь должен наполнить его информацией о существующих помещениях и привязать к ним план-схемы.

Как только создание проекта и наполнение его информацией закончено, пользователь может перейти к следующему пункту для настройки навигации в помещении – это расстановка всех возможных путей, по которым он хочет осуществлять навигацию и в рамках которых будет осуществляться локализация устройства в помещении. Это может быть как всего лишь один путь (если помещение – длинный коридор) или большое количество переплетающихся путей, если это помещение, например, продуктовый магазин. Чем больше пользователь создаст навигационных путей, тем точнее затем будет локализовано устройство в рамках данного проекта. Расстановка путей в мобильном приложении осуществляется путем последовательных касаний по предварительно загруженным план-схемам.

У каждого BLE-маяка есть три настройки, которые нас интересуют, по которым мы сможем точно идентифицировать и отличать маяки друг от друга – это UUID, major, minor. Если пользователь имеет возможность настраивать устройства, то он может собственноручно изменить данные параметры, в противном же случае он может оставить заводские настройки. Главное – внести информацию о маяках в приложение. В платформе iOS программно ограничено количество маяков, которые приложение может одновременно сканировать – 20 устройств.

После того, как все пути, по которым будет осуществляться навигация в помещениях, расставлены, можно переходить к завершающему этапу настройки навигационного приложения – сбору цифрового отпечатка помещения. На этом этапе нужно расставить BLE-маяки по помещению и не менять в дальнейшем их положение (в случае малейших изменении в положении маяков в помещении, требуется сбор нового цифрового отпечатка, поскольку точность определяемого местоположения устройства резко упадет и будет велика вероятность ошибок). Чем больше будет размещено устройств в сканируемом помещении, тем лучше. Программа работает таким образом, что ей не нужна информация о том, где находятся сами маяки, ей важны лишь полученные данные RSSI, поэтому вносить информацию о положении маяка не нужно. После того, как все BLE-маяки расставлены, начинается сбор цифрового отпечатка. Это последовательный проход по всем расставленным пользователем путям с включенным режимом сканирования. В процессе режима сканирования, в каждой точке созданного пользователем маршрута приложение получает информацию о находящихся поблизости маяках и сохраняет информацию об их RSSI. Данный этап завершается после того как пользователь соберет цифровые отпечатки всех созданных путей на всех уровнях навигационного проекта.

* + 1. **Режим локализации и навигации**

После того, как навигационный проект создан, настроен и собран цифровой отпечаток карты, можно переходить к основной функции, ради которой и создавалось данное мобильное приложение – локализация устройства в пространстве. В данном режиме приложение начинает сканировать находящиеся поблизости BLE-маяки и получать их RSSI, далее полученные данные сравниваются с данными цифрового отпечатка, берется точка в помещении, отпечаток которой наиболее приближен к полученным данным. Эту точку и будем принимать за местоположение устройства. Таким образом, чем лучше и подробнее собран отпечаток помещения, тем точнее будет последующая локализация.

Если местоположение, полученное с помощью алгоритма, находится на другом этаже (уровне), нежели отображаемом на экране устройства, то приложение автоматически переключит карту помещения на требуемый уровень и заменит план-схему.

Помимо определения местоположения устройства в помещении, в приложении реализована упрощенная форма навигации в нем. Пользователь может нажать на любую точку на план-схеме и построить маршрут к ней. Приложение возьмет все размеченные заранее пользователем пути навигации на план-схемах, найдет ближайшую к требуемой точке точку, в которой собран цифровой отпечаток и построит маршрут к ней, проложенный исключительно по путям, размеченным пользователем на этапе настройки навигационного проекта.

* 1. **Архитектурные особенности приложения**

В данной подглаве будут рассмотрены архитектурные особенности и выбранные решения для реализации требуемых функций мобильного приложения, аргументирован их выбор и описано использование в проекте.

* + 1. **Стек используемых технологий**

Для разработки приложения практически без раздумываний была выбрана среда разработки XCode [16], разработанная и поддерживаемая компанией Apple. Первоначально рассматривалась IDE App Code [14] от компании Jet Brains, которая несмотря на свою молодость показывает неплохие результаты, однако следующий ряд неопровержимых плюсов нативной среды разработки от Apple не оставил шансов другим вариантам:

1. Разрабатывается и поддерживается компанией Apple, создателем платформы iOS, под которую разрабатывается приложение;
2. Присутствует на рынке уже много лет и постоянно обновляется, уменьшая количество багов и ошибок;
3. Отличные возможности для отлаживания написанного кода и исследования его быстродействия, нагрузки на мобильное устройство;
4. Встроенные графический редактор интерфейсов и инструменты autolayout для более удобной работы с интерфейсами;

Самая актуальная на момент разработки приложения версия XCode – 10.2.1

Что касается языка программирования – изначально рассматривались несколько вариантов – это Swift [13], Objective-C, React Native и Flutter. Первые два – это классика от компании Apple, React Native – это фреймворк от компании Facebook, с помощью которого можно создавать кроссплатформенные мобильные приложения, Flutter – сравнительно новый инструмент от компании Google, позволяющий переписывая минимальное количество кода писать программы для разных устройств, в том числе устройств на платформе iOS и Android. React Native и Flutter – это технологии, которые удобно использовать в проектах, в которых требуется два мобильных приложения под разные платформы и у них большая общая кодовая база. В таком случае данные технологии уменьшат дублирование кода, ускорят разработку приложений, но усложнят работу с интерфейсами приложений. Разрабатываемое же приложение для навигации внутри помещения не подходит под эти параметры – работа с технологиями для навигации устроена абсолютно по разному в платформах iOS и Android, переиспользовать код не получится, требуется доступ к встроенным в саму платформу фреймворкам, а так же требуется нативный и удобный интерфейс для работы с картой помещения, который будет очень сложно, если вообще возможно реализовать с помощью сторонних библиотек. Таким образом, выбор сузился до двух языков программирования от компании Apple – Swift и Objective-C. Выбор практически сразу пал на новый, активно развивающийся Swift, поскольку он обладает следующими неоспоримыми плюсами над изрядно устаревшим конкурентом:

1. Проекты, написанные на Swift намного легче поддерживать – намного меньше шаблонных строк кода, которые постоянно нужно писать, отсутствие двух файлов, которое тянется в Objective-C из C;
2. Файлы, написанные на Swift – намного более читаемые;
3. Количество разработчиков, которые пишут на Objective-C с каждым годом все меньше, а количество разработчиков Swift – увеличивается;
4. Все большие компании медленно переходят на Swift;

И огромное количество других неоспоримых плюсов.

* + 1. **Архитектура клиентского приложения**

Наиболее распространенная архитектура для приложений на платформе Apple iOS – это MVC, или, если расшифровать – Model-View-Controller. Это паттерн проектирования (design pattern), предлагаемый компанией Apple для разработки мобильных приложений. Но, следует сделать замечание, что MVC, описываемый компанией Apple имеет несколько различий с общепризнанным паттерном, его скорее можно назвать Model-View-Adapter. Суть различий заключается в том, что в iOS слой Model не имеет возможности общаться со слоем представления (View) и общается с ним лишь через Controller, чего нельзя сказать про обычный MVC.

Есть еще несколько довольно популярных архитектурных паттернов, которые используются при разработке приложений для устройств на платформе Apple iOS – такие как MVVM (Model-View-View Model), MVP (Model-View-Presenter). Эти нестандартные паттерны очень хороши, когда применяешь их для решения конкретной задачи или возникшей проблемы, в обычных же случаях использования они лишь усложняют читабельность кода и увеличивают количество шаблонного кода. Поэтому было принято решение использовать классический, предлагаемый компанией Apple MVC.

В архитектурном паттерне MVC можно выделить три слоя – Model, View и Controller, которые мы дальше рассмотрим более подробно.

Слой Model. Данный слой отвечает, в основном за представление всех используемых в приложении структур данных, описывает всю основную логику их хранения и обработки, никак не должен быть связан с интерфейсом пользователя. Обычно сюда относятся все структуры данных, модели, объекты, объекты для хранения данных, а так же для сетевого взаимодействия.

Слой View. Данный слой отвечает за отображение всех возможных данных, за пользовательский интерфейс, проще говоря. В случае iOS все нативные элементы уже существуют и элементарное приложение можно создать без написания собственного кода в этом слое, лишь используя Interface Builder и встроенные там настройки элементов. Однако, очень часто UI-элементы в iOS переиспользуются и наследуются, в таком случае уже появляется код в данном слое.

Слой Controller. Этот слой отвечает за все возможные взаимодействия внутри приложения: обрабатывает полученные действия пользователя, вносит изменения в модель, получает уведомления от модели и изменяет пользовательский интерфейс. В случае платформы iOS за этот слой отвечают файлы ViewController, которые отвечают за управление интерфейсом, обработку пользовательских действий, отображение моделей данных и многое другое.

* + 1. **BLE-маяки и их использование**

В связи с тем, что изначально мы были ограничены выбранной платформой – это Apple iOS, сильно сузились все возможные технологии, которые мы могли бы использовать для локализации устройства в пространстве. Apple не дает разработчику доступ к получению информации о силе сигнала WiFi и другим радио-сигналам за исключением BLE-маяков, поэтому сразу было решено двигаться в этом направлении. У каждого Bluetooth маяка есть 4 требуемых нам параметра – UUID, major, minor и RSSI. Первые три – константы до тех пор, пока не потребуется полная перенастройка устройства. Последнее же – изменяемая полученная сила сигнала устройства. На основании ее можно примерно оценить расстояние до устройства. Но это расстояние будет очень неточным, поскольку при разных условиях эта величина может сильно разниться. Тогда было решено использовать сравнение изначально собранного отпечатка помещения с полученными данными в реальном времени путем выбора точки с наиболее приближенными данными о силе сигнала всех маяков.

* 1. **Проектирование пользовательского интерфейса**

Apple очень строго относится к пользовательскому интерфейсу в приложениях и на сайте Apple Developer этой теме посвящен целый раздел с большим количеством рекомендаций и советов по использованию и доработке нативных элементов управления интерфейсом [10].

При проектировании интерфейса приложения для навигации внутри помещения использовался инструмент Sketch и Adobe XD. На этапе проектирования было решено максимально придерживаться требований компании Apple, не отходить от нативных элементов. Поскольку цель приложения – навигация, то самое большое внимание было уделено именно этому элементу, а точнее - карте, работа с ним должна быть привычной пользователю, не вызывать негативных эмоций. Этот элемент используется в приложении несколько раз и пользователь большую часть времени находится именно на этом экране – будь то расстановка путей для навигации, сбор цифрового отпечатка помещения или же непосредственно локализация и навигация в помещении.

**Выводы по главе**

Во второй главе были подробно рассмотрены и разделены на два основных процесса все поддерживаемые функции приложения с подробным описанием их использования пользователем, описана проработанная схема взаимодействия пользователя с приложением. Так же были рассмотрены архитектурные особенности разработки мобильного приложения, которое бы удовлетворяло изначально поставленным задачам и соответствовало всем функциональным требованиям.

В следующей главе будет рассмотрена программная реализация приложения для мобильных устройств на платформе Apple iOS и рассмотрены интерфейсные особенности приложения.

1. **Описание программной реализации и интерфейса iOS-приложения**

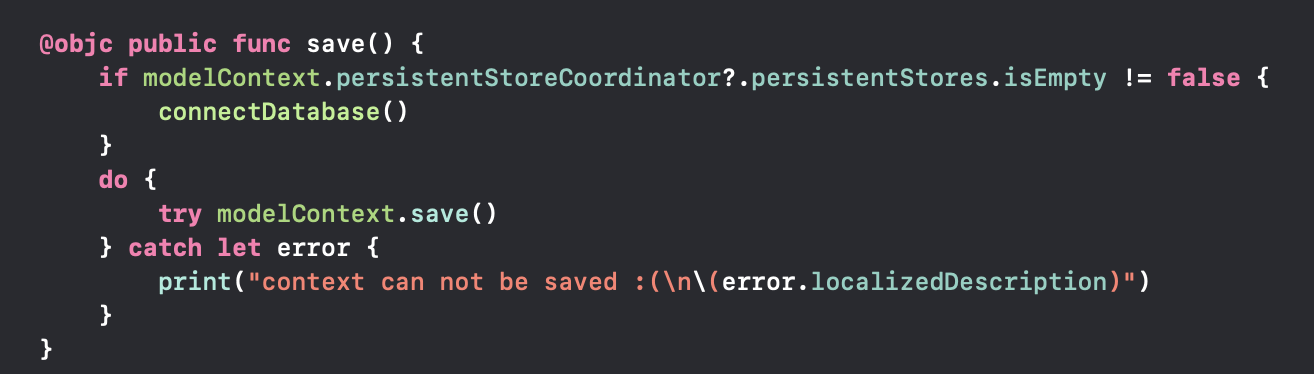
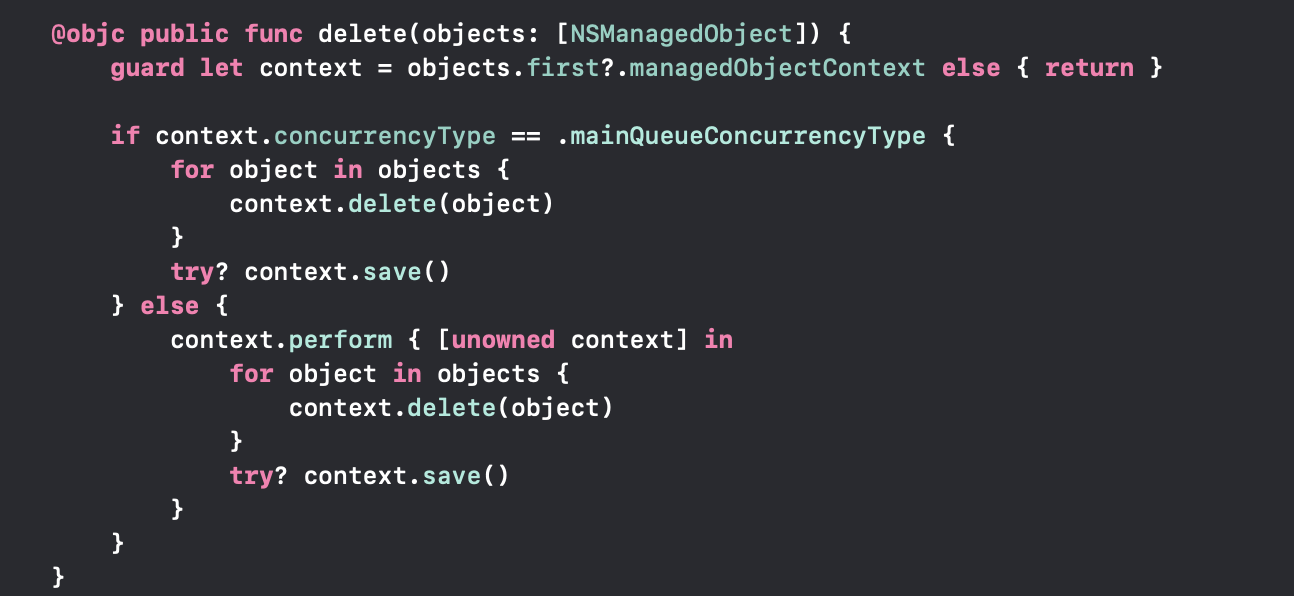
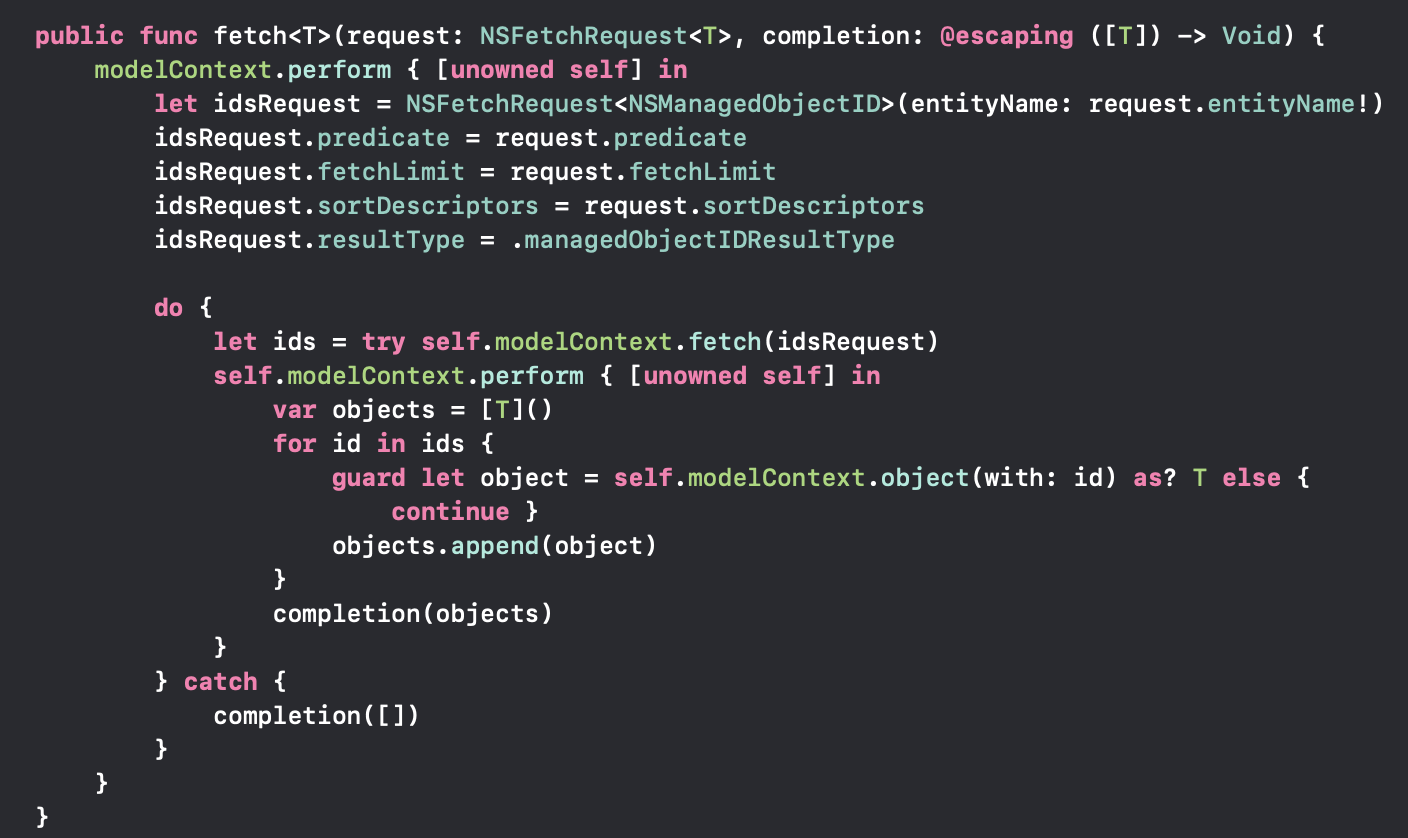
В данной главе будут рассмотрены выбранные инструменты, технологии и особенности программной реализации конечного продукта.

* 1. **Инструменты разработки iOS-приложения**

Приложение разработано на языке программирования Swift в среде разработки программного обеспечения Xcode. Для контроля версий использовалась технология git.

* 1. **Хранение данных в клиентском приложении**

Для хранения данных в приложении было принято решение использовать предлагаемый компанией Apple фреймворк Core Data [15]. Выбор стоял между Core Data, SQLite и Realm. Было решено использовать нативный фреймворк, поскольку никаких сложных структур данных в приложении не предвиделось и нужен был обычный рабочий инструмент, удовлетворяющий потребностям рядового разработчика. У него есть ряд минусов, которые постепенно решаются с обновлениями языка Swift, но все же взаимодействие и использование технологии не идеально и заставляет разработчика писать большое количество шаблонного и не наглядного, усложняющего читаемость файлов кода. По этой причине было принято решение написать вспомогательный класс для работы с Code Data – CoreDataManager, он содержит вспомогательные методы для инициализации хранилища, сохранения, удаления и запросов к объектам.

1. Сохранение  
     
    Рис. 4 – Сохранение объекта Core Data
2. Удаление  
     
    Рис. 5 – Удаление объекта Core Data
3. Запросы  
     
    Рис. 6 – Запросы к Core Data

В Core Data было принято решение хранить всего 4 сущности:

1. PlanSet – это объект, который отвечает за сам навигационный проект, по сути это самый высокоуровневый объект;
2. Plan – это объект, который отвечает за отдельные помещения или этажи с отдельной картой.
3. Image – это структура для хранения изображения, если изображение из интернета – у него есть ссылка, локальное же изображение хранится в памяти и у него есть локальный путь.
4. Track – это объект, который описывает один навигационный путь, хранит данные о точках пути, измерения, собранные в процессе формирования отпечатка помещения и прочую информацию.

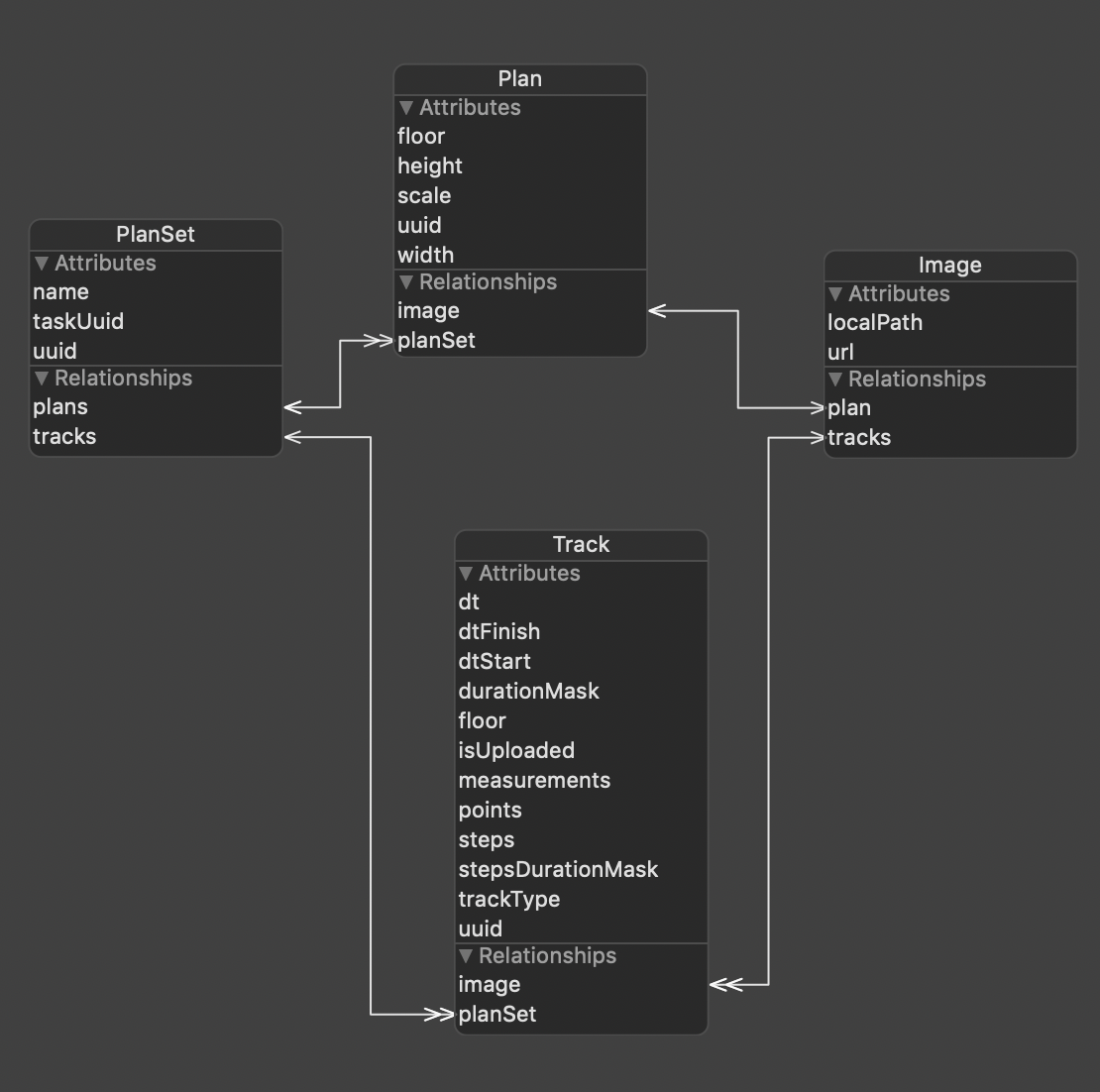


Рис. 7 – Схема данных Core Data

Каждый объект, который хранится в Core Data так же имеет вспомогательный метод, позволяющий упрощать запросы к базе, возвращающий идентификатор объекта.

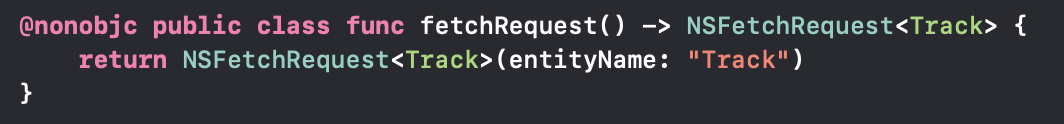


Рис. 8 – Запрос объекта из базы Core Data

Цифровой отпечаток помещения хранится в формате json, поскольку объекты такого формата очень удобны для использования на любых платформах и, с заделом на будущее, было принято решение хранить все собранные измерения в таком формате.

Пример собранных навигационных данных по одному из навигационных путей:



Рис. 9 – Пример json отпечатка помещения

* 1. **Реализация модуля приложения для настройки навигации**

Во второй главе была подробно описана последовательность действий для осуществления создания и настройки нового проекта для навигации. Ниже приведены скриншоты приложения, на которых видно, как организована настройка путей навигации и сбор цифрового отпечатка помещения.

На первом скриншоте расположен список созданных путей навигации, по нажатию на них открывается путь на план-схеме, по которому видно, где будет проходить данный путь. Пользователь может подойти к началу пути, подготовиться к сбору цифрового отпечатка, дальше по нажатию на «начать» устройство запустит таймер и будет показывать пользователю его текущее положение.

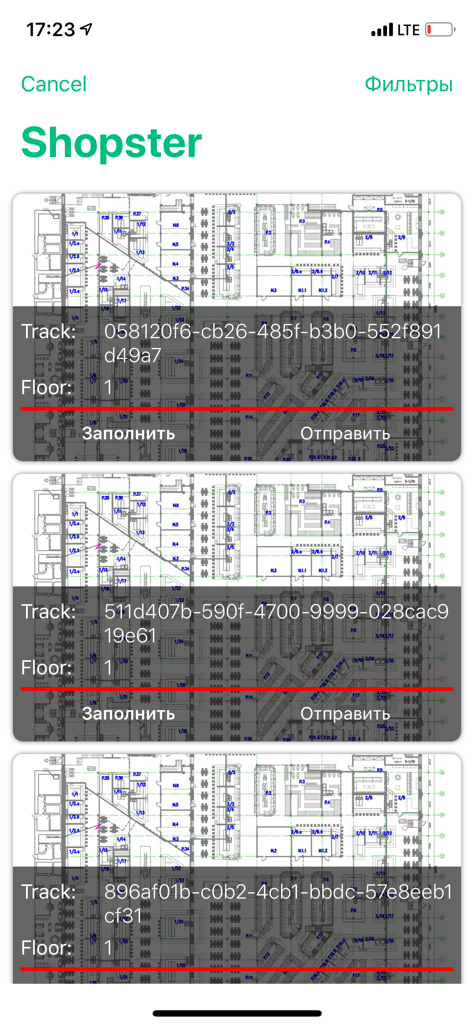
 

Рис. 10 – Экран с путями Рис. 11 – Экран с навигационным путем

После нажатия на кнопку «Начать», устройство начинает заполнять точки на карте зеленым цветом, что обозначает и указывает пользователю, где ему сейчас следует быть. В каждый момент времени прохода по навигационному пути устройство сохраняет в цифровой отпечаток BLE-устройства, находящиеся поблизости, привязывая их к данной точке на план-схеме.

По нажатию на «пауза» можно приостановить запись, если пользователю нужно отвлечься или успеть поправить свое реальное местоположение, чтобы оно соответствовало указанному на карте.

По нажатию на «стоп» можно прекратить создание цифрового отпечатка помещения, если пользователь считает, что уклонился от маршрута или же сбился и хочет начать с начала.



Рис. 12 – Экран со сбором цифрового отпечатка

* + 1. **Core Location для работы с BLE-маяками**

В предыдущей главе было решено, что за основу технологии навигации внутри помещения взять BLE-маяки и сравнение силы их сигнала. В iOS работа с Bluetooth заложена в фреймворке Core Location [12].

С помощью данного фреймворка устройство на платформе Apple iOS может сканировать находящиеся поблизости BLE-маяки. Для этого достаточно знать UUID устройства. Одновременно iOS может сканировать до 20 UUID, но никто не запрещает нам сканировать устройства с разными параметрами major и minor.

Следующий код начинает сканирование BLE маяков с определенным идентификатором:

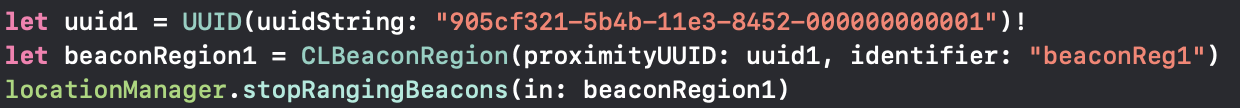


Рис. 13 – Старт сканирования beacon-ов

В нашем случае, в каждой точке нашего навигационного пути требуется сохранять текущий цифровой отпечаток помещения. Здесь мы встречаем небольшую проблему. Core Location так устроен, что нельзя запросить информацию о текущих BLE устройствах вокруг. Требуется реализовать метод didRangeBeacons делегата CLLocationManagerDelegate [11] и тогда, как только устройство на платформе Apple iOS обнаружит поблизости устройства с ранжируемым нами UUID, оно вернет нам его в данном методе.

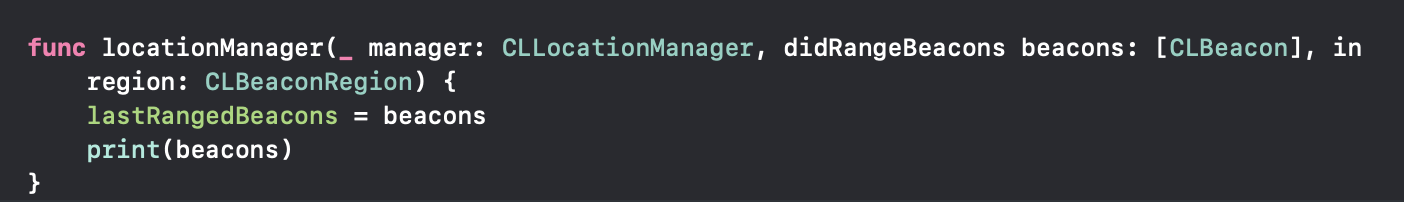


Рис. 14 – Метод для получения сканированных устройств

Таким образом, было решено создание следующей механики. Разделить создание цифрового отпечатка помещения, при каждом вызове метода делегата мы сохраняем последние отсканированные устройства Bluetooth Low Energy и при инициализации контроллера запускаем отдельный таймер, который будет брать последние данные о маяках и сохранять их в отпечаток с определенной периодичностью прохода по навигационному пути.

Таким образом, решается проблема с тем, что у нас нет доступа к текущим данным о окружающих устройство маяках.

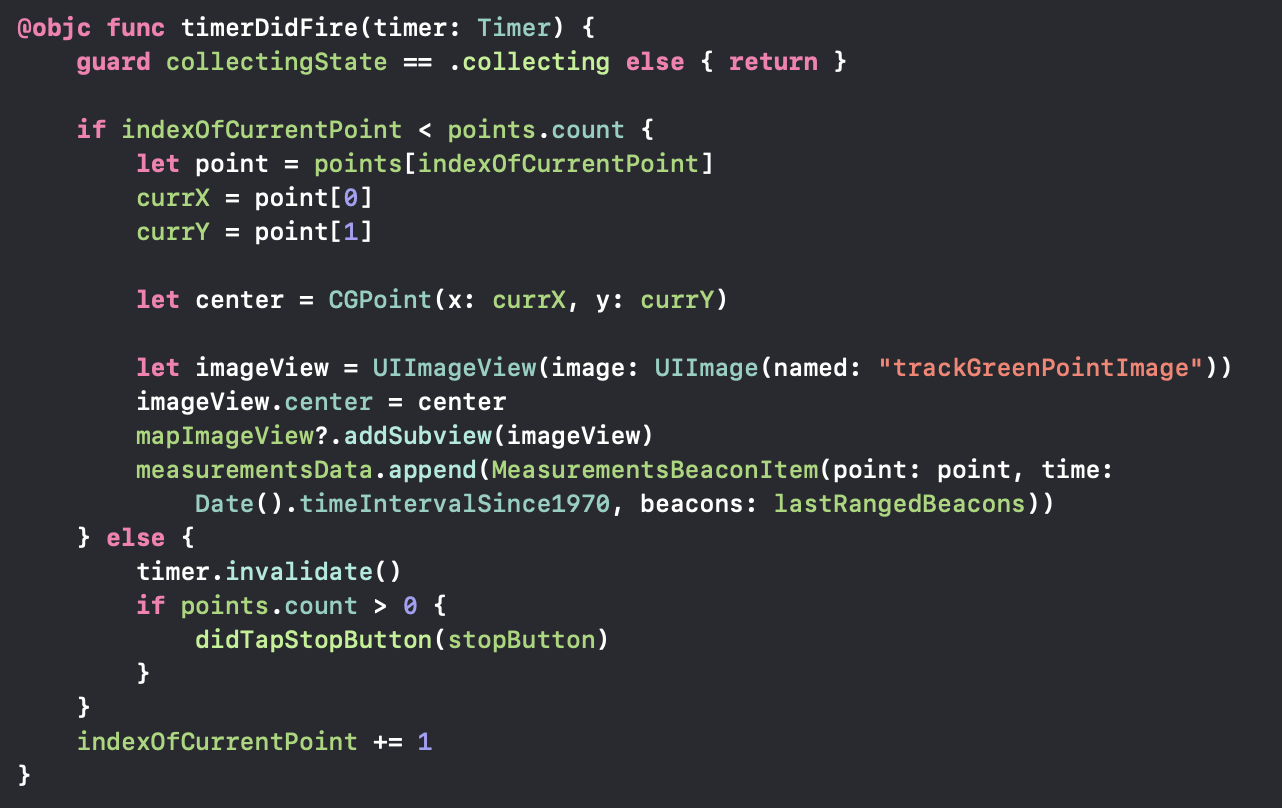


Рис. 15 – Метод таймера для сохранения отпечатка

* + 1. **Структуры данных для хранения цифрового отпечатка**

Для хранения данных о цифровом отпечатке используется следующая разработанная структура данных, в которой так же присутствуют методы для моментального формирования объекта в формате JSON.

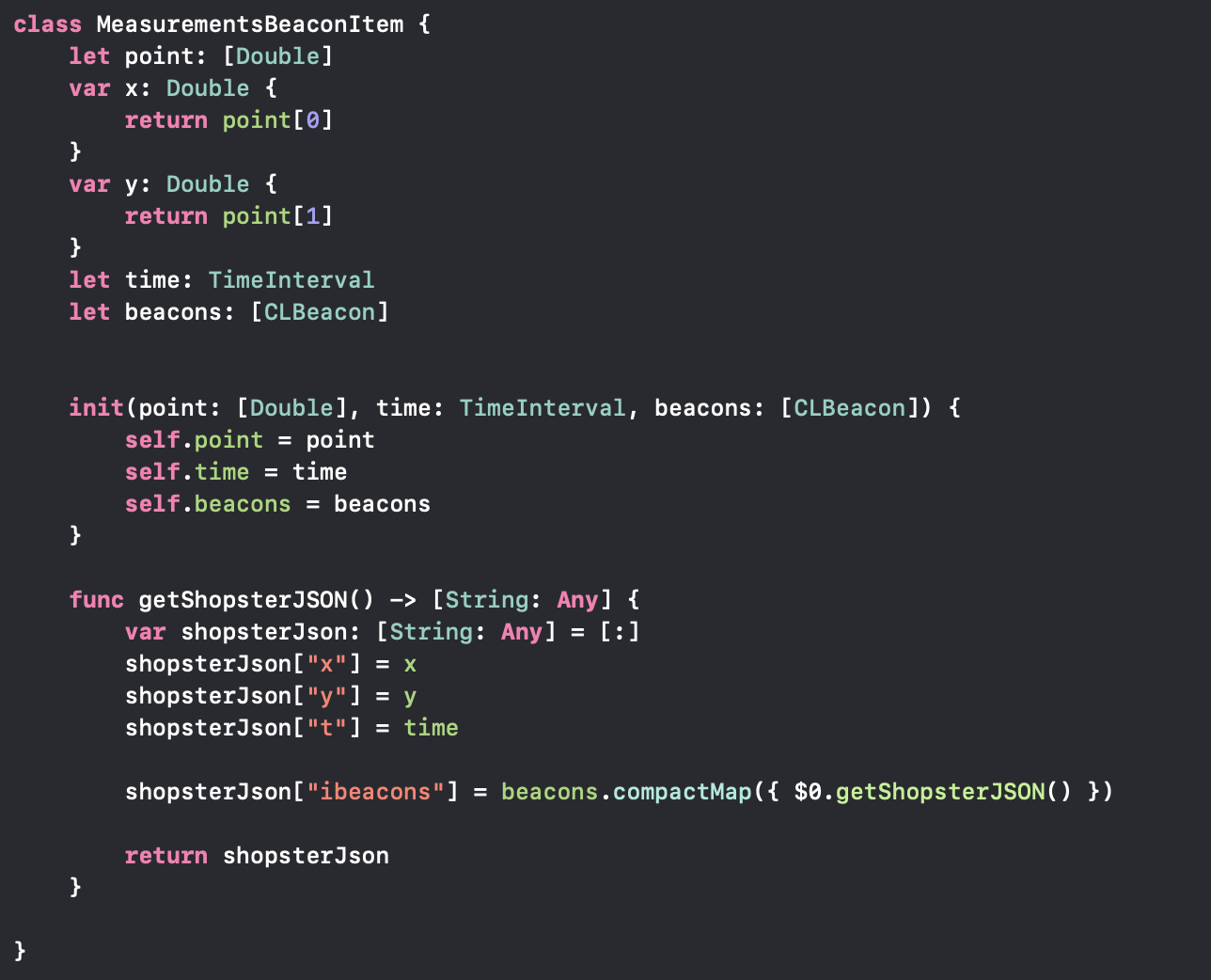


Рис. 16 – Структура данных цифрового отпечатка

* 1. **Реализация модуля приложения для локализации и навигации**

В интерфейсе данного модуля все максимально просто для пользователя, ему следует лишь начать навигацию и ждать, пока устройство обнаружит BLE-маяки поблизости и определит начальное местоположение. Карта автоматически переключится на нужный этаж и сразу будет показано местоположение пользователя.

Помимо автоматического режима, пользователь может просматривать карты, меняя последовательно этажи (уровни) помещения.

Для осуществления навигации к определенной точке пользователю следует долгим нажатием на карту определить точку, к которой он бы хотел построить маршрут, приложение найдет ближайшие точки от пользователя до выбранной им точки и по заранее проложенным навигационным путям проложит маршрут.



Рис. 17 – Экран локализаци и навигации

* + 1. **UIScrollView и png-изображение для карты**

Карта – это основной элемент как сервисного модуля для создания и настройки навигационного проекта, так и основного модуля для локализации и навигации пользователя внутри помещения. Она используется в следующих случаях:

1. Расстановка путей для навигации;
2. Сбор данных для создания цифрового отпечатка помещения;
3. Локализация и навигация пользователя;

Было принято решение использовать для этого предлагаемый компанией Apple в библиотеке UIKit элемент – UIScrollView с значительными доработками.

Во всех, привычных нам приложениях для навигации (таких как Apple Maps, Yandex Maps) используется двойной tap по карте для уменьшения масштаба. Для этого был добавлен обработчик двойного нажатия.

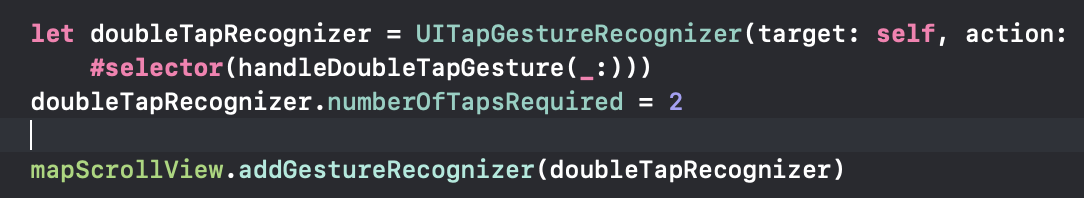


Рис. 18 – Добавление обработчика double-tap

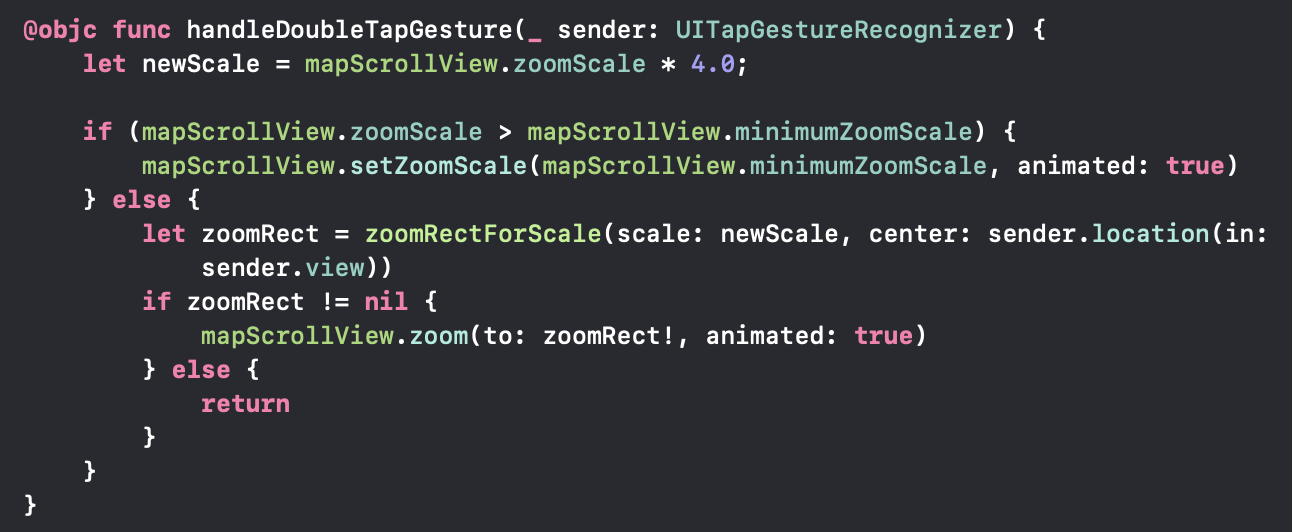


Рис. 19 – Обработчик double-tap

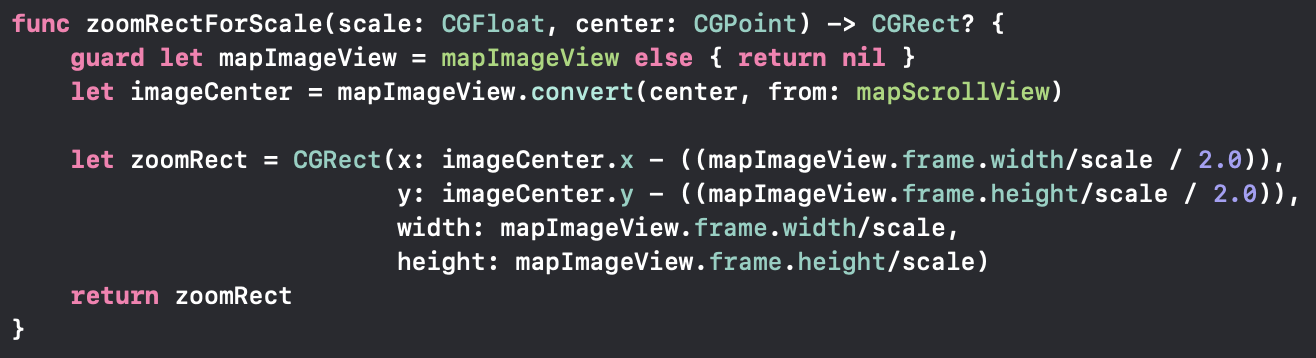


Рис. 20 – Метод для определения квадрата для приближения

Для обработки привычных движений пальцами для приближения или отдаления карты (изменения масштаба) используется делегат UIScrollView.

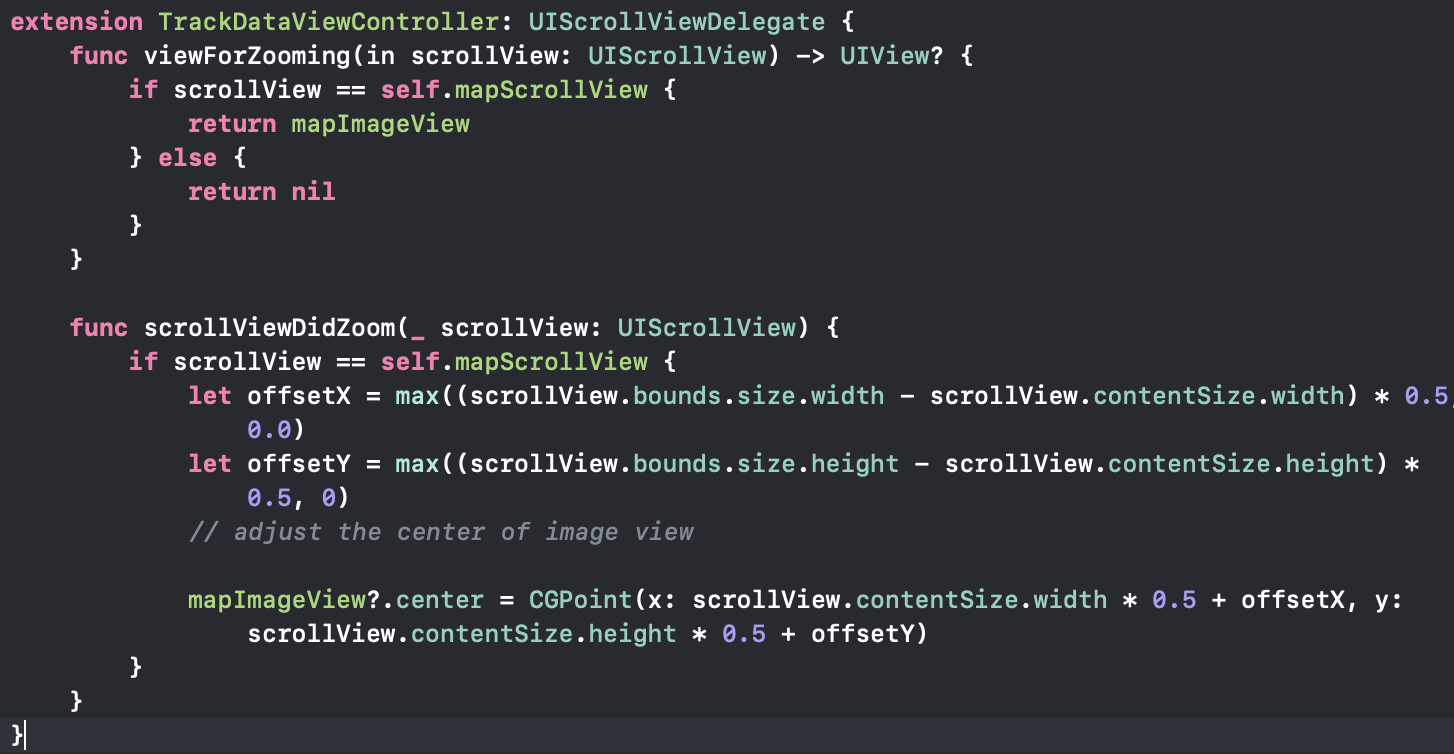


Рис. 21 – Реализация методов делегате UIScrollView

* 1. **Использованные сторонние библиотеки**

Для интеграции сторонних библиотек в приложении используется технология CocoaPods [17], позволяющая удобно оперировать библиотеками, написанными сторонними разработчиками, использовать их в своем проекте. В приложении для навигации внутри помещения используются следующие Pod-ы:

1. Alamofire [18] – это самая распространенная библиотека, упрощающая работу с сетью в проектах, написанных на Objective-C и Swift;
2. MBProgressHUD [19] – это библиотека, которая содержит различные и гибко настраиваемые всплывающие окна с подсказками, индикаторами загрузки и так далее.
3. SwiftyJSON [20] – это библиотека, которая очень упрощает работу с JSON, доступ к полям объекта по аналогии со словарем, обработку опциональных полей.
4. SwiftyUserDefaults [21] – это библиотека, которая упрощает работу с встроенной в Foundation фреймворк технологией User Defaults, которая используется для хранения простых данных, таких как пользовательские настройки приложения.

**Выводы по главе**

Во третьей главе были подробно рассмотрены все особенности реализации приложения для навигации внутри помещения для устройств на платформе Apple iOS на основании поставленных в первой главе функциональных требований и выбранных во второй главе технологий, архитектуры.

**Заключение**

В рамках выполнения выпускной квалификационной работы было разработано мобильное приложение для навигации внутри помещения для устройств на платформе Apple iOS. Для реализации данного проекта были изучены существующие технологии для локализации и навигации внутри помещения, произведено сравнение подходов, технологий, методов определения положения мобильного устройства в закрытом помещении, где недоступны сигналы общепризнанных систем, таких как GPS и Глонасс. На основании функциональных требований, ограничений выбранной платформы были проанализированы и выбраны технологии для реализации приложения.

Таким образом, в процессе выполнения выпускной квалификационной работы были выполнены следующие задачи:

1. Проанализированы существующие технологии и подходы определения местоположения в помещении;
2. Изучены основы взаимодействия устройств на платформе Apple iOS с BLE-маяками;
3. Разработан модуль приложения для сбора навигационных данных и формирования «электронного отпечатка» помещения;
4. Разработан модуль приложения для определения местоположения устройства в предварительно отсканированном помещении;
5. Разработано приложение, представляющее собой комплексное решение для навигации внутри помещения;
6. Протестирована работа приложения;
7. Разработана техническая документация;

Направления для дальнейшего развития приложения:

1. Создание серверного решения для дистанционного обновления цифрового отпечатка помещения;
2. Создание полноценной платформы с разделением сервисного приложения и клиентского приложения для локализации и навигации;
3. Создание библиотеки для того, чтобы сторонние разработчики могли использовать данное решение и интегрировать его в свои приложения;
4. Загрузка приложения в App Store;
5. Доработка навигационных алгоритмов;

**Список использованных источников**

1. Apple Developer Documentation. [Электронный ресурс]// URL: <https://developer.apple.com/documentation/> (Дата обращения: 15.02.2019, режим доступа: свободный).
2. The Swift Programming Language (Swift 4.1). [Электронный ресурс]// URL: <https://developer.apple.com/library/content/documentation/Swift/Conceptual/Swift_Programming_Language/index.html> (Дата обращения: 01.03.2019, режим доступа: свободный)
3. Y. Liu and Z. Yang, Location, Localization, and Localizability. Location- awareness Technology for Wireless Networks, Springer, 2010.
4. H.Liu,H.Darabi,P .Banerjee,andJ.Liu, “Survey of wireless indoor positioning techniques and systems,” IEEE Transactions on Systems, Man and Cybernetics Part C, vol. 37, no. 6, pp. 1067– 1080, 2007.
5. D. Zhang, F. Xia, Z. Yang, L. Yao, and W. Zhao, “Localization technologies for indoor human tracking,” in Proceedings of the 5th International Conference on Future Information Technology (FutureTech ’10), May 2010.
6. M. B. Ismail, A. F. A. Boud, and W. N. W. Ibrahim, “Implemen- tation of location determination in a wireless local area network (WLAN) environment,” in Proceedings of the 10th International Conference on Advanced Communication Technology (ICACT ’34), pp. 894–899, February 2008.
7. M. A. Youssef, A. Agrawala, and A. U. Shankar, “WLAN location determination via clustering and probability distributions,” in Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom ’03), pp. 143–150, March 2003. and Data Mining (KDD ’16), ACM, New York, NY, USA, Mar. 2016, pp. 785–794.
8. H.J.PerezIglesias,V.Barral,andC.J.Escudero,“Indoorperson localization system through RSSI Bluetooth fingerprinting,” in Proceedings of the 19th International Conference on Systems, Signals and Image Processing (IWSSIP ’12), pp. 40–43, April 2012.
9. D. Pai, M. Malpani, I. Sasi, N. Aggarwal, and P. S. Mantripragada, “A Robust pedestrian dead reckoning system on smart- phones,” in Proceedings of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom ’12), pp. 2000–2007, June 2012.
10. Human Interface Guidelines. [Электронный ресурс]// URL: https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/ (Дата обращения: 15.04.2019, режим доступа: свободный).
11. Determining the Proximity to an iBeacon. [Электронный ресурс]// URL: https://developer.apple.com/documentation/corelocation/determining\_the\_proximity\_to\_an\_ibeacon (Дата обращения: 10.03.2019, режим доступа: свободный).
12. Core Location. [Электронный ресурс]// URL: https://developer.apple.com/documentation/corelocation (Дата обращения: 10.03.2019, режим доступа: свободный).
13. Swift. [Электронный ресурс]// URL: https://developer.apple.com/swift/ (Дата обращения: 10.03.2019, режим доступа: свободный).
14. App Code. [Электронный ресурс]// URL: https://www.jetbrains.com/objc/ (Дата обращения: 10.03.2019, режим доступа: свободный).
15. Core Data. [Электронный ресурс]// URL: https://developer.apple.com/documentation/coredata (Дата обращения: 10.03.2019, режим доступа: свободный)
16. XCode. [Электронный ресурс]// URL: https://developer.apple.com/xcode/ (Дата обращения: 10.03.2019, режим доступа: свободный)
17. Cocoapods. [Электронный ресурс]// URL: https://cocoapods.org/ (Дата обращения: 10.03.2019, режим доступа: свободный)
18. Alamofire. [Электронный ресурс]// URL: https://github.com/Alamofire/Alamofire (Дата обращения: 10.03.2019, режим доступа: свободный)
19. MBProgressHUD. [Электронный ресурс]// URL: https://github.com/jdg/MBProgressHUD (Дата обращения: 10.03.2019, режим доступа: свободный)
20. SwiftyJSON. [Электронный ресурс]// URL: https://github.com/SwiftyJSON/SwiftyJSON (Дата обращения: 10.03.2019, режим доступа: свободный)
21. SwiftyUserDefaults. [Электронный ресурс]// URL: https://github.com/radex/SwiftyUserDefaults (Дата обращения: 10.03.2019, режим доступа: свободный)