

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Дальневосточный федеральный университет»**

**(ДВФУ)**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра прикладной математики, механики, управления и программного обеспечения**

НЕКРАСОВ ЭДУАРД АЛЕКСЕЕВИЧ

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ 3D ПОЗИЦИОНИРОВАНИЯ ВНУТРИ ПОМЕЩЕНИЯ

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

по направлению подготовки 02.03.03 – Математическое обеспечение и администрирование информационных систем

профиль «Технология программирования»

г. Владивосток

2020 г.

|  |  |
| --- | --- |
|  | Автор работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись)  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  | Руководитель ВКР  Профессор, д.т.н, доцент \_\_\_\_\_  (должность, степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гриняк В.М.  (подпись) (Ф.И.О.)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |
|  |  |
| Защищена в ГЭК с оценкой \_\_\_\_\_\_\_\_\_  Секретарь ГЭК  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Крестникова О.А.  (подпись) (Ф.И.О.)  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | «Допустить к защите»  Заведующая кафедрой, д.т.н., профессор  (степень, ученое звание)  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Артемьева И.Л.  (подпись) (Ф.И.О.)  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

Оглавление

[Введение 5](#_Toc44074139)

[1 Обзор существующих систем позиционирования 8](#_Toc44074140)

[1.1 Технология Beacon 8](#_Toc44074141)

[1.1.1 Схема работы 8](#_Toc44074142)

[1.1.2 Физическая реализация 8](#_Toc44074143)

[1.1.3 Достоинства и недостатки 9](#_Toc44074144)

[1.2 Примеры готовых решений 10](#_Toc44074145)

[1.2.1 Система DECK WAYZ 10](#_Toc44074146)

[1.2.2 Система Navigine 11](#_Toc44074147)

[1.2.3 Navizon indoor location solutions 12](#_Toc44074148)

[1.2.4 Система SkyLab Beacon VG02 Detecter 3000 Ultra HD - система, разработанная студентами ДВФУ 13](#_Toc44074149)

[1.3 Словарь терминов предметной области 14](#_Toc44074150)

[1.4 Задачи indoor-навигации в разных предметных областях 15](#_Toc44074151)

[1.4.1 Навигация в медицинских учреждениях 15](#_Toc44074152)

[1.4.2 Внутренняя навигация в супермаркетах 15](#_Toc44074153)

[1.4.3 Инвентаризация товара в складских помещениях 16](#_Toc44074154)

[1.4.4 Навигация во время пожаротушения 16](#_Toc44074155)

[1.5 Методы indoor-навигации 18](#_Toc44074156)

[1.5.1 Wi-Fi 18](#_Toc44074157)

[1.5.2 Bluetooth 18](#_Toc44074158)

[1.5.3 Системы спутниковой навигации 19](#_Toc44074159)

[1.5.4 GSM 20](#_Toc44074160)

[1.5.5 Инерциальные навигационные системы 21](#_Toc44074161)

[2 Анализ предметной области «Indoor-навигация» 23](#_Toc44074162)

[2.1 Анализ профессиональной деятельности 23](#_Toc44074163)

[2.2 Анализ решаемых задач обработки, передачи и хранения информации 23](#_Toc44074164)

[2.2.1 Задачи передачи информации 24](#_Toc44074165)

[2.2.2 Задачи обработки информации 24](#_Toc44074166)

[2.2.3 Задачи хранения информации 24](#_Toc44074167)

[2.3 Анализ задач, требующих автоматизации 24](#_Toc44074168)

[2.3.1 Калибровка Bluetooth датчиков 24](#_Toc44074169)

[2.3.2 Проверка работоспособности Bluetooth датчиков 25](#_Toc44074170)

[2.4 Объекты профессиональной деятельности 25](#_Toc44074171)

[2.5 Модель предметной области 28](#_Toc44074172)

[2.5.1 Неподвижные объекты 28](#_Toc44074173)

[2.5.2 Bluetooth маячки 29](#_Toc44074174)

[2.5.3 Подвижные объекты 29](#_Toc44074175)

[2.6 Онтологические соглашения 29](#_Toc44074176)

[2.7 Анализ знаний 30](#_Toc44074177)

[2.7.1 Определение местоположения пользователя 30](#_Toc44074178)

[Концепция решения 31](#_Toc44074179)

[2.7.2 Определение калибровки Bluetooth-маяков 34](#_Toc44074180)

[3 Проектирование системы Indoor-навигация 35](#_Toc44074181)

[3.1 Архитектура системы Indoor-навигация 35](#_Toc44074182)

[**3.1.1 Use-cases. Диаграмма вариантов использования 35**](#_Toc44074183)

[**3.1.2 Диаграмма перехода из состояния в состояние 36**](#_Toc44074184)

[**3.1.3 Диаграмма потоков данных 39**](#_Toc44074185)

[3.2 Библиотека для работы с Bluetooth маячками 41](#_Toc44074186)

[3.3 Требования 42](#_Toc44074187)

[**3.3.1 Требования к навигационной системе Indoor-навигация 42**](#_Toc44074188)

[**Функциональные требования 42**](#_Toc44074189)

[**3.3.2 Требования к подсистемам мобильного приложения «NaviBlue» 42**](#_Toc44074190)

[3.4 Рекомендации для кодирования 45](#_Toc44074191)

[4 Реализация и тестирование мобильного приложения «NaviBlue» для 3D позиционирования внутри помещения с использованием Bluetooth-маячков. 47](#_Toc44074192)

[4.1 Спецификация 47](#_Toc44074193)

[4.2 Среда разработки 48](#_Toc44074194)

[4.3 Описание методов и классов 51](#_Toc44074195)

[4.4 Интерфейс пользователя 57](#_Toc44074196)

[**4.4.1 Стартовый экран 57**](#_Toc44074197)

[**4.4.2 Экран списка маяков 58**](#_Toc44074198)

[**4.4.3 Экран информации о маяке 59**](#_Toc44074199)

[**4.4.4 Экран добавления маяка 59**](#_Toc44074200)

[**4.4.5 Экран определения местоположения 61**](#_Toc44074201)

[4.5 Тестирование 62](#_Toc44074202)

[**4.5.1 Тестирование подсистем 62**](#_Toc44074203)

[**4.5.2 Покрытие тестами требований 66**](#_Toc44074204)

[**4.5.3 Тестирование системы с моделируемым сигналом 67**](#_Toc44074205)

[**4.5.4 Тестирование системы с реальным сигналом 70**](#_Toc44074206)

[4.6 Навигационная ситуация 73](#_Toc44074207)

[Заключение 76](#_Toc44074208)

[Список литературы 78](#_Toc44074209)

[Статьи 78](#_Toc44074210)

[Учебники и учебные пособия 78](#_Toc44074211)

[Электронные ресурсы 78](#_Toc44074212)

[Приложение А 80](#_Toc44074213)

# Введение

В последнее десятилетие после появления и массового распространения современных мобильных устройств, будь то Android smartphone или iPhone, стремительно возрос интерес к задачам, связанным с навигацией в пространстве, среди достаточно широкого круга обывателей. Задачи навигации – это задачи определения глобальных и относительных в зависимости от местности координат и их производных. Данный тип задач безусловно является важнейшим объектом приложения достижений в области информационных технологий.

В настоящее время традиционными источниками навигационных данных являются спутниковые навигационные системы второго поколения, на пример GPS и Глонасс. Они позволяют очень точно определять местоположение объекта при условии наблюдения достаточного числа спутников, хорошего качества их сигнала и расположения объекта вблизи поверхности Земли.

Однако данные условия не всегда могут выполняться в здании. При нахождении объекта внутри помещения число видящих его спутник может быть сильно ограниченно, прием данных о нем может быть крайне неустойчивым, а точность определения высоты объекта над уровнем поверхности Земли довольно низкой (возникает «проблема этажа»). В этом случае, чтобы определить местоположение объекта в помещении, необходимо использовать альтернативные источники навигационной информации. Среди данных источников выделяют: инерциальные навигационные средства (акселерометры, гироскопы), точки доступа в интернет Wi-Fi, Bluetooth устройства передачи данных и тому подобные. Данные, полученные с этих источников, могут использоваться как по отдельности, так и совместно, в зависимости от решаемой задачи.

Благодаря все растущему интересу к задачам навигации и позиционирования в 2010 году была впервые созвана и проведена Международная Конференция Позиционирования и Навигации Внутри Помещений (International Conference on Indoor Positioning and Indoor Navigation). Данная конференция была призвана собрать вместе экспертов в области электроники, инженерии и IT-технологий со всех стран мира и позволить им представить свои разработки и исследования при решении задач позиционирования и навигации. Конференция 2010 года с более, чем несколько сотен участников, доказала, что в мире существует достаточно большое сообщество, работающее над развитием данной области. Поэтому данная конференция проводится каждый год и по сей день.

Однако несмотря на все приложенные усилия как со стороны простых обывателей, так и со стороны исследователей и специалистов к настоящему времени пока что не существует стандартных способов решения задач позиционирования и навигации внутри помещений. Данное обстоятельство позволяет рассматривать и пробовать все новые и новые методы, алгоритмы и подходы к решению данных задач.

Использование Bluetooth устройств имеет целый ряд преимуществ. Во-первых, это построение инфраструктуры при помощи недорогого и достаточно распространенного оборудования, которое не требует от использующих его людей каких-либо специальных профессиональных навыков. Во-вторых, возможность практически произвольного расположения датчиков на стенах помещения в силу их малого размера и отсутствия необходимости подводить к ним электропитание, что дает возможность оптимизировать условия наблюдения. В-третьих, дальность видимости Bluetooth устройств невелика, а их сигнал практически не проходит сквозь стены, что полностью решает «проблему этажа».

В данной выпускной работе будет рассмотрена реализация прототипа системы, которая смогла бы решить проблему определения местоположения объекта в 3-х мерном пространстве. В данном прототипе, будет реализован функционал, позволяющий провести анализ получаемых данных, определить дистанцию до объекта, а также сами координаты нашего объекта.

Целью выпускной квалификационной работы является создание прототипа системы, предназначенного для определения местоположения объекта в здании с использованием Bluetooth-маяков, разработка алгоритма и оценка точности получаемых данных.

Для достижения поставленной цели необходимо решить следующие задачи:

1) Обзор существующих систем в области «Indoor-навигация».

2) Анализ ПО «Indoor-навигация», построение модели.

3) Проектирование системы «Indoor-навигация».

4) Реализация и тестирование Android приложения для добавления данных о маяках, вычисления дистанции от маяка до объекта, а также нахождения координат местоположения объекта. Исследование точности вычислений местоположения устройства при решении задачи трилатерации с помощью Bluetooth-маяков.

Выпускная квалификационная работа имеет следующую структуру:

В первой главе проведен анализ рассматриваемой профессиональной деятельности, а также обзор существующих систем, которые позволяют определить местоположение объекта.

Во второй главе выполнен анализ предметной области, где были определены объекты и задачи, построена ее модель, рассмотрены методы решения задач.

В третьей главе приведены результаты разработки проекта информационной системы для определения местоположения объекта в трехмерном пространстве.

В четвертой главе представлено описание реализованной программной системы, а также приведены результаты тестирования и испытаний.

В заключении приведены итоговые результаты выпускной квалификационной работы.

# 1 Обзор существующих систем позиционирования

Прежде чем приступить к разработке собственной системы позиционирования, необходимо сначала рассмотреть уже существующие системы.

## 1.1 Технология Beacon

Технология Beacon – технология навигации внутри помещения, которая основана на использовании Bluetooth-маячков Beacon и дает достаточно большую точность при приемлемом уровне финансовых затрат [1].

### 

## 1.1.1 Схема работы

По всему периметру устанавливаются Bluetooth-маячки, координаты расположения которых известны заранее. Эти маячки с заданной периодичностью производят широковещательную рассылку, содержащую идентифицирующую их информацию. Пользовательское приложение циклично получает эти данные, по базе данных определяет координаты маячков, и на основе силы сигнала (позволяющей определить удаленность от каждого из них) определяет свое местоположение.

### 

## 1.1.2 Физическая реализация

В плане физической реализации Beacon-маячки – это обычные Bluetooth 4.0 LE (Low Energy) устройства, таким образом, их роль может выполнять любое устройство, оснащённое BLE-чипом – например, cмартфоны на базе Android, а также iPhone, iPad, обычные ноутбуки и т.д., на которое установлено специальное приложение, реализующее функции Beacon-маячка. Типичный Beacon-маячок, имеет довольно компактные размеры, и способен проработать всего лишь от одной батарейки до двух лет. Дальность действия маячка – в среднем 10 метров (варьируется от 15-20см до 25-40м в зависимости от модели и настроек). Периодичность выдачи данных – 200мс, но можно настроить и на более частую периодичность, и на более редкую. Срок службы от одной батарейки – в зависимости от модели от чуть менее одного года до трёх лет (в среднем – 2 года). Цена одного маячка – порядка 15-20 долларов. Маячок является простым устройством, который только выдаёт всем подряд в эфир свои данные, используя Bluetooth профиль GATT (при этом к нему даже не нужно выполнять подключение), тем не менее, производители, как правило, закладывают возможность подключения к маячку с целью его удалённого конфигурирования.

## 1.1.3 Достоинства и недостатки

При работе с технологией Beacon можно выделить следующие достоинства и недостатки.

Достоинства:

* относительно небольшой размер Bluetooth-маячков;
* большой срок работы на одном заряде батареи;
* относительно небольшая стоимость Bluetooth-маячков.

Недостатки:

* слабый уровень защиты Bluetooth-маячков.

# 1.2 Примеры готовых решений

Ниже будут рассмотрены системы DECK WAYZ, Navigine, Navizon indoor location solutions, работа которых основана на технологии Beacon, а также система, над которой велась работа студентами ДВФУ.

## 

## 1.2.1 Система DECK WAYZ

DECK WAYZ - это система навигации внутри помещений и трекинга. Реализована с использованием двухмерного пространства.

WAYZ – это комплекс, состоящий из клиентского приложения, сервера приложений и различных источников данных и подключаемых внешних систем.

В качестве источника данных о местоположении WAYZ может использовать:

* информацию о географических координатах систем ГЛОНАСС/GPS;
* информацию о находящихся рядом датчиках Bluetooth LE (iBeacon);
* информацию о текущем состоянии сотовой сети на устройстве;
* информацию о текущем подключении WI-FI;
* информацию с внешней системы WI-FI геолокации (Cisco CMX/MSE).

Рабочий цикл WAYZ состоит из трех основных блоков:

1. получение на устройство информации из все доступных источников и передача ее на сервер;
2. получение сервером информации из внешних источников координат и принятие решений на основе информации клиента и внешней информации;
3. обработка «действий», т.е. инициация неких сценариев в зависимости от координат пользователя.

Возможности системы:

* определение местоположения пользователя;
* проложение маршрута до нужного объекта;
* инициирование различных бизнес сценариев в зависимости от местонахождения пользователя;
* получение аналитики по передвижению пользователя.
* Отросли применения системы:
* ритейл;
* музеи, выставки, конференции;
* логистика.

## 1.2.2 Система Navigine

Navigine – система навигации внутри помещения и аналитики. Данная система реализована на двухмерном пространстве.

Платформа представляет собой набор инструментов для разработчиков, которые создают веб-сервисы для отслеживания перемещения пользователей. Веб-интерфейс адаптируется под потребности заказчика и включает в себя следующие модули: история перемещения, онлайн мониторинг, оповещение.

Мобильное приложение, оснащенное Navigine SDK, помогает пользователю перемещаться внутри здания и использует внешнюю радиочастотную инфраструктуру, внутренние датчики мобильных устройств.

Возможности системы:

* отслеживание перемещения пользователя в режиме реального времени;
* настройка оповещения сотрудников;
* сбор геоданных для анализа;
* оптимизация логистических процессов.
* Отросли применения системы:
* ритейл;
* цифровая реклама;
* музеи;
* транспорт;
* недвижимость и офисы;
* логистика и складирование.

## 

## 1.2.3 Navizon indoor location solutions

Navizon Indoor Location Solution, предлагает 2 программных решения для поиска активных устройств Wi-Fi: смартфоны, планшеты, ноутбуки, теги Wi-Fi и нестандартные устройства, независимо от их операционной системы. Эти решения предназначены для разных целей [12].

Первое решение, Navizon I.T.S. представляет собой аппаратное решение, предназначенное для поиска активных устройств Wi-Fi, включая смартфоны, планшеты, ноутбуки, теги Wi-Fi и нестандартные устройства, независимо от их операционной системы. Navizon I.T.S. полагается на аппаратное обеспечение, развернутое по всему объекту для обнаружения устройств, определяя их местоположение путем триангуляции радиосигналов, которые они передают. Это позволяет приложениям, таким как оценка количества посетителей сайта, отслеживать перемещение устройств или действия пользователя.

Второе решение, Navizon Indoors — это программное решение, которое позволяет мобильным приложениям выполнять действия в зависимости от их местоположения. Navizon Indoors работает, заставляя устройство сканировать точки доступа Wi-Fi вокруг него, чтобы определить его местоположение. Типичные области применения включают навигацию в помещении и поиск маршрута.

Недостатками данной системы являются:

1) высокая цена;

2) для использования этой системы нужно составлять радио-отпечаток помещения;

3) Работает на двухмерном пространстве.

## 1.2.4 Система SkyLab Beacon VG02 Detecter 3000 Ultra HD - система, разработанная студентами ДВФУ

Данное программное средство работает с использованием Bluetooth маячков. Что позволяет точно и дешево определить местоположение объекта на местности.

На основе двухмерной модели пользователь может отследить своё местоположение, а также построить кратчайший путь из точки А в точку В. Это позволяет решать задачи навигации в торговых центрах, музеях, в университетах, а также задачи складского характера.

Преимущества:

* относительно небольшой размер Bluetooth-маячков;
* большой срок работы на одном заряде батареи;
* относительно небольшая стоимость Bluetooth-маячков;
* простота работы.

Недостатки:

* слабый уровень защиты Bluetooth-маячков;
* невозможность определения высоты объекта;
* ограниченная область применения.

# 1.3 Словарь терминов предметной области

Bluetooth-маячки – это тип миниатюрных батарейных устройств для передачи информации малого объема [7].

Bluetooth Low Energy – это интеллектуальная и экономичная версия беспроводной технологии Bluetooth. Она уже играет значительную роль в превращении разумных гаджетов в еще более разумные, делая их компактными, доступными и менее сложными [8].

Bluetooth – открытый стандарт беспроводной связи с низким энергопотреблением, обеспечивающий передачу данных и звука между совместимыми устройствами [9].

Мобильное приложение – программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы (iOS, Android, Windows Phone и т. д.) [10]. Многие мобильные приложения предустановлены на самом устройстве или могут быть загружены на него из онлайновых магазинов приложений, таких как App Store, Google Play, и других, бесплатно или за плату.

Android устройство – устройство на операционной системе Android.

Indoor-навигация – навигация внутри помещения.

iBeacon - это протокол передачи Bluetooth-сигнала между локационным маячком и любым устройством (смартфоном, планшетом), которое совместимо с Bluetooth Low Energy [11]. Принцип работы iBeacon прост: через равные промежутки времени маячки отправляют Bluetooth-сигнал на устройства, находящиеся в радиусе их действия.

# 1.4 Задачи indoor-навигации в разных предметных областях

Задачи indoor-навигации встречаются в самых различных предметных областях, таких как:

Предметные области, в которых встречаются задачи indoor-навигации:

1. Медицинские учреждения
2. Супермаркеты
3. Складские помещения
4. Пожаротушение

## 1.4.1 Навигация в медицинских учреждениях

В медицинских учреждениях единовременно находятся сотни больных. Многие из больных не сразу могут ориентироваться в здании, находить нужные кабинеты и т.д. Из-за этого случаются опоздания на процедуры, долгие поиски кабинета.

Пунктами навигации, в данной предметной области, могут быть кабинеты, столовые, места для отдыха, палаты. Должна быть возможность врачам отслеживать местоположение больного, что эффективнее скажется на их работу.

По прибытию в больницу, больному уже не нужно будет тратить свое время на поиски входной двери, поиски регистратуры, кабинета.

Bluetooth-маяки установлены на стенах больницы. Для удобного взаимодействия с пользователем, разработано мобильное приложение, которое принимает сигналы от Bluetooth-маяков и используя эти данные вычисляет местоположение пользователя.

## 1.4.2 Внутренняя навигация в супермаркетах

Современные торговые центры и супермаркеты отличаются большими размерами, они увеличиваются на столько, что могут представлять собой целые торговые города, которые занимают несколько уровней. Увеличивается количество магазинов и торговых точек, это дает возможность покупателям выбирать то, что больше нравится. Это приводит к тому, что клиенты запоминают планировку тех магазинов, в которых они бывают чаще всего. Некоторые ТЦ не посещают регулярно, поэтому покупателям каждый раз приходится заново узнавать, где что находится.

Люди, которые оказываются в незнакомой среде, должны знать, в каком месте здания они находятся, иметь представление о планировке здания и определять направления, в которых им надо двигаться, чтобы прийти в необходимый магазин. Архитектура здания и графические средства навигации должны решать данную проблему.

Indoor-навигация помогает покупателям быстро находить продукты, которые им необходимо приобрести, а также найти как добраться до нужных торговых точек в торговом центре. Для владельцев торговых центров Indoor-навигация предоставляет возможность мониторинга посещений клиентов, что позволяет сделать большой анализ для выставления продукции в более проходные места.

## 1.4.3 Инвентаризация товара в складских помещениях

В многих крупных компаниях присутствуют огромные складские помещения, в которых находятся тысячи единиц товаров. Для того чтобы провести инвентаризацию каждого товара, компании необходимо содержать десяток специалистов и тратить большое количество ресурсов и времени на выполнение данной задачи.

Автоматизация данного процесса необходима и может быть достигнута с использованием Indoor-навигации и специально запрограммированного квадрокоптера или робота, который бы выполнял данные задачи в автоматическом режиме.

## 1.4.4 Навигация во время пожаротушения

Такая задача, как спасение людей при пожаре, является достаточно важной для увеличения эффективности пожарных. Это позволит уменьшить количество несчастных случаев. Во время пожара здания, торговые центры, квартиры настолько задымлены, что быстрая и эффективная навигация в данных помещениях просто не возможна.

Смысл пожарной разведки заключается не только в том, чтобы вывести людей в безопасное место, но и в том, чтобы обследовать здание до того, как начнется тушение. Однако прежде всего пожарная разведка – это навигация внутри здания. Если специалист не в состоянии осуществлять навигацию, вам не удастся найти пострадавших, которых можно было бы спасти, выбраться наружу также будет проблематично.

Умение упаковывать и транспортировать пострадавших жизненно важно, но, прежде чем вы сможете этими делами заняться, вам сначала нужно добраться до пострадавших, а также иметь чёткое представление, по какому пути вы будете выходить в безопасное место.

Indoor-навигация в данном случае поможет пожарникам быстрее добираться до очага возгорания, а также максимально безопасно выводить пострадавших.

# 1.5 Методы indoor-навигации

Навигация внутри помещения осуществляется разными методами, такими как:

1. Wi-Fi
2. Bluetooth
3. Системы спутниковой навигации
4. GSM
5. Инерциальные навигационные системы

## 1.5.1 Wi-Fi

Для определения местоположения пользователя с использованием метода позиционирования по Wi-Fi используются данные, полученные от точек доступа Wi-Fi. При вычислении координат пользователя используется метод триангуляции относительно точек доступа (AccessPoints) с известными данными MAC, SSID (рисунок) и координатами. Координаты точек доступа не всегда определены.

Сначала происходит сканирование доступных точек, а затем устройство посылает данные для дальнейшей их обработки на сервере. На сервере эти данные проходят анализ с использованием известных координат точек доступа и вычисляются координаты пользователя.

Трилатерация на базе Wi-Fi передатчиков имеет погрешность 10 метров, радиокарты сигналов Wi-Fi - 5 метров.

## 1.5.2 Bluetooth

Данный метод определения местоположения пользователя применяется в помещениях для проведения форумов, супермаркетах, музеях и прочих помещениях, которые обладают достаточно сложной структурой и большой площадью.

Данная технология основывается на использовании Bluetooth-маячков, установленных по периметру помещения, базы данных с данными о них, а также мобильном приложении данного заведения.

Точность позиционирования данной системы достигает 3-5 метров. Стоимость одного маячка варьируется от 18 до 35 долларов, а их форм-фактор упрощает процесс монтажа в помещении (есть возможность распределить маячки по периметру используя простейший двусторонний скотч).

Данная система имеет возможность использования более одного раза. Для этого достаточно снять все маячки, изменить их конфигурацию и установить на новом объекте. Для использования данной системы пользователю достаточно иметь под рукой телефон с установленным приложением.

Из этого следует, что данная система имеет низкую стоимость, она проста в установке, а также имеет высокую точность позиционирования и удобство в использовании конечным пользователем. Поэтому данный метод, был выбран для решения нашей проблемы.

## 1.5.3 Системы спутниковой навигации

GPS - спутниковая система навигации, обеспечивающая измерение расстояния, времени и определяющая местоположение во всемирной системе координат.

Минус данной системы в том, что она до сих пор не может решить проблему навигации внутри помещений. Хотя была запатентована технология GPS репитеров, видимых результатов она не принесла. В достаточно больших супермаркетах или торговых центах, а также в многоэтажных домах GPS не позволяет точно определить местоположение объекта. В открытом пространстве точность GPS оставляет желать лучшего.

## 1.5.4 GSM

GSM (Global Systemfor Mobile Communications) - глобальный стандарт цифровой мобильной сотовой связи, с разделением каналов по времени и частоте [14]. Был разработан в конце 1980-х годов.

Так как связь возможно на расстоянии не более 120 км от ближайшей базовой станции, даже при использовании усилителей и направленных антенн. Поэтому для покрытия определенной площади необходимо большое количество передатчиков.

При отсутствии препятствий ослабление сигнала при распространении возрастает пропорционально квадрату расстояния, увеличиваясь, таким образом, на 6 дБ каждый раз, когда расстояние удваивается.

На изменение уровня сигнала действуют не только потери при прохождении сигналом определенного расстояния в пустом пространстве, но и всякого рода препятствия, которые могут располагаться между станцией и телефоном. Как пример, железобетонные строения способны ослаблять сигналы, проходящие через них, в 100-1000 раз (то есть на 20-30 дБ).

Список характеристик сигнала:

- способность «пробивать» стены: отличная;

- точность определения координат: от нескольких километров до 10 м;

- рабочие частоты: 850 Мгц, 900 Мгц, 1800 Мгц, 1900 Мгц;

- радиус действия: в городских условиях около 2 км, реально до 120 км;

- доступность и распространенность: более 95 % телефонов на территории Европы;

- энергопотребление: до 300 Ма.

- минимальная цена мобильного устройства, поддерживающего данную технологию: 500 рублей;

- минимальная цена дополнительного оборудования для обеспечения работы навигационной системы: точность навигации зависит только от количества базовых станций;

Отсюда можем сделать вывод, что на сегодня данную технологию нет возможности использовать для навигации внутри помещений, из-за низкой точности, отсутствие возможности определить уровень над уровнем моря (устройство не сможет понять, на каком этаже здания оно находится), высокое воздействие на сигнал внешними объектами, а также прямая зависимость точности от количества дорогостоящих базовых станций.

## 1.5.5 Инерциальные навигационные системы

Основа работы инерциальной навигационной системы заключается в измерении ускорений летательного аппарата и его угловых скоростей относительно трех осей самолета для того, чтобы исходя из этих данных определить местоположение самолета, его скорость, курс и другие параметры [13]. По результатам анализа объект стабилизируется, и может использоваться автоматическое управление.

Чтобы получить информацию о полете в состав инерциальной навигационной системы включают акселерометр и гироскоп. Точность информации зависит от характеристик данных приборов. Готовые данные поступают на компьютер, где осуществляется анализ данных, а затем с использованием определенных навигационных алгоритмов происходит корректировка движения объекта.

Инерциальные навигационной системы делятся на платформенные и бесплатформенные. Основой для платформенных инерциальных навигационных систем служит гиростабилизированная платформа. В бесплатформенных системах акселерометры и гироскопы жестко связаны с корпусом прибора. Функции платформы моделируются математически вычислительной системой. Бесплатформенные системы выгодно отличаются меньшим весом и габаритами, а также возможностью работать при значительных перегрузках.

Преимущества инерциальной навигационной системы:

1) полная независимость от внешних источников данных,

2) повышенная защита от помех,

3) высокая информативность,

4) возможности передавать информацию на большой скорости.

Недостатки инерциальной навигационной системы:

1) ошибки, которые накапливаются с течением времени в получаемой информации от приборов,

2) высокая стоимость входящего в состав оборудования.

# 2 Анализ предметной области «Indoor-навигация»

В данной главе проводится анализ предметной области «Indoor-навигация».

В рамках анализа будет сделано:

● проведен анализ профессиональной деятельности;

● рассмотрены задачи обработки, хранения, передачи информации;

● выделены объекты профессиональной деятельности и информационные объекты;

● построена модель предметной области «Indoor-навигация»;

● построена Математическая модель задачи позиционирования устройства.

## 2.1 Анализ профессиональной деятельности

Профессиональной деятельностью в данной работе является определение координат местоположения пользователя внутри помещения при помощи Bluetooth устройств.

Сейчас время новых технологий, в данное время большая часть людей не может обойтись без использования мобильного телефона. У различных предприятий начинает увеличиваться интерес к организации навигации внутри здания, так как это положительно влияет на лояльность клиентов. Из этого следует, что экспертами могут быть люди, которые работают в различных предметных областях.

## 2.2 Анализ решаемых задач обработки, передачи и хранения информации

Разрабатываемая система позиционирования должна решать задачи обработки, передачи и хранения информации следующим образом.

## 2.2.1 Задачи передачи информации

Bluetooth устройства размещаются на стенах помещений на некотором расстоянии друг от друга. Далее проводится калибровка всех датчиков и проверка их работоспособности. Только после этого программа может осуществлять сбор данных с Bluetooth устройств для определения местоположения пользователя. После обработки данных программа передает их на ПК, где они хранятся до востребования.

## 2.2.2 Задачи обработки информации

После сбора данных о местоположении объекта, программа обрабатывает полученные данные согласно некоторому алгоритму решения задач позиционирования и навигации и отображает точное его местоположение на заранее загруженном плане помещения в интерфейс пользовательского приложения.

## 2.2.3 Задачи хранения информации

Все данные о местоположении движущегося объекта, Bluetooth устройств, комнат, переходов, лестниц и лифтов, данные о здании и план здания хранятся на ПК до момента, когда они понадобятся программе для решения задачи позиционирования.

## 2.3 Анализ задач, требующих автоматизации

Для того, чтобы реализуемая система позиционирования работала наиболее эффективно необходимо проанализировать следующие задачи, требующие автоматизации.

## 2.3.1 Калибровка Bluetooth датчиков

Калибровка Bluetooth датчиков заключается в установлении единой системы счисления для всех датчиков, сопоставлении данной системы с планом помещения с последующим вычислением для каждого датчика координат его местонахождения.

## 2.3.2 Проверка работоспособности Bluetooth датчиков

В результате этой проверки при первом запуске всем Bluetooth датчикам будет присвоен некоторый диапазон координат, в пределах которого они могут видеть пользователя, для последующего определения местоположения пользователя. Если при проверке работоспособности, датчик будет давать ложные данные или вовсе не работать, то данные, собранные с этого датчика, не будут учитываться при дальнейшем сборе данных.

## 2.4 Объекты профессиональной деятельности

При анализе профессиональной деятельности можно выделить следующие объекты профессиональной деятельности с соответствующими характеристиками.

Объект: **здание (помещение)**

Описание:здание, в котором размещаются Bluetooth устройства и осуществляется навигация движущегося объекта

Характеристики:

* номер, однозначно определяющий здание;
* координаты границ расположения здания;
* количество этажей;
* список комнат;
* список переходов;
* список лестниц;
* список лифтов;
* список Bluetooth устройств.

Объект: **этаж**

Описание: часть здания, в котором осуществляется навигация пользователя и определение местоположения пользователя

Характеристика:

* номер, однозначно определяющий этаж;
* список комнат;
* список переходов;
* список лестниц;
* список лифтов;
* список Bluetooth устройств.

Объект: **комната**

Описание:комната, находящаяся внутри здания на определенном этаже, в котором осуществляется навигация пользователя

Характеристика:

* номер, однозначно определяющий комнату;
* номер здания;
* номер этажа;
* координаты границ расположения комнаты.

Объект: **переход (переход между зданиями)**

Описание:объект, помогающий перемещаться между строениями здания

Характеристика:

* номер, однозначно определяющий переход;
* номер начального здания;
* номер конечного здания;
* номер начального этажа;
* номер конечного этажа;
* координаты начального расположения перехода;
* координаты конечного расположения перехода.

Объект: **лестница**

Описание: объект, помогающий перемещаться между этажами здания

Характеристика:

* номер, однозначно определяющий лестницу;
* номер здания;
* номер начального этажа;
* номер конечного этажа;
* координаты начального расположения лестницы;
* координаты конечного расположения лестницы.

Объект: **лифт**

Описание: объект, помогающий перемещаться между этажами здания

Характеристики:

* номер, однозначно определяющий лифт;
* номер здания;
* интервал номеров этажей (перечень этажей, на которых останавливается лифт);
* координаты начального расположения лифта;
* координаты конечного расположения лифта.

Объект: **Bluetooth устройство (датчик)**

Описание: устройство, которое размещается на стене помещения для последующего определения координат местоположения движущегося объекта

Характеристики:

* номер, однозначно определяющий устройство;
* номер здания;
* номер этажа;
* координаты расположения устройства;
* диапазон координат видимости (диапазон координат, в пределах которого устройство может видеть пользователя).

Объект: **движущийся объект**

Описание: объект, движущийся внутри помещения, и местоположение которого необходимо определить с помощью Bluetooth устройств

Характеристика:

* номер здания;
* номер этажа;
* начальные координаты расположения объекта;
* конечные координаты расположения объекта.

## 2.5 Модель предметной области

## 2.5.1 Неподвижные объекты

**сорт здания: {} N\**

Непустое конечное множество всех зданий.

**сорт координаты здания: здания -> {} I(0, ) X I(0, ) \**

Функция, сопоставляющая каждому зданию непустое конечное множество кортежей из двух натуральных чисел (x, y).

**сорт количество этажей: здания -> I(1, 1000)**

Функция, сопоставляющая каждому зданию некоторое натуральное число отличное от нуля.

**сорт этажи: {} N\**

Непустое конечное множество этажей здания.

**сорт объекты: {} N\**

Непустое конечное множество всех объектов.

**сорт объекты этажа: этажи -> {} объекты**

Функция, сопоставляющая каждому этажу конечное множество объектов этажа.

**объекты = (∪ (v∈этажи) объекты этажа(v))**

**сорт координаты объекта: объекты -> I(0, ) X I(0, ) X I(0, )**

Функция, сопоставляющая каждому объекту некоторый кортеж из трех натуральных чисел (x, y, z).

**сорт переходы: {} N\**

Непустое конечное множество всех переходов в здании.

**сорт лестницы: {} N\**

Непустое конечное множество всех лестниц в здании.

**сорт лифты: {} N\**

Непустое конечное множество всех лифтов в здании.

**сорт аудитории: {} N\**

Непустое конечное множество всех аудиторий в здании.

**переходы ∪ лестницы ∪ лифты ∪ аудитории = объекты этажа ⊂ объекты**

## 2.5.2 Bluetooth маячки

**сорт Bluetooth: {} N\**

Непустое конечное множество всех Bluetooth устройств.

**сорт координаты Bluetooth: Bluetooth -> I(0, ) X I(0, ) X I(0, )**

Функция, сопоставляющая каждому Bluetooth устройству некоторый кортеж из трех натуральных чисел (x, y, z).

## 

## 2.5.3 Подвижные объекты

**сорт подвижные объекты: {} N\**

Непустое конечное множество подвижных объектов.

**сорт координаты подвижного объекта: подвижные объекты ->{}( I(0, ) X I(0, ) X I(0, ) X I(0, )) ­**

Функция, сопоставляющая каждому подвижному объекту некоторый кортеж из четырех натуральных чисел (x, y, z, t)

## 2.6 Онтологические соглашения

1. Объект не может находиться вне здания;
2. Подвижный объект может находиться вне здания;
3. Координата время (t) уникальна у подвижного объекта для каждой координаты x, y, z.

## 2.7 Анализ знаний

## 2.7.1 Определение местоположения пользователя

Рассмотрим правую декартову систему координат **oxyz** [2]. **Oxyz** – это относительные координаты, заданные системой Bluetooth-маяков. Пусть многопозиционная система наблюдения состоит из **n** радиолокационных станций (Bluetooth-маяков) с координатами **x(i), y(i), z(i) ()** и в поле её действия выделено **m** неподвижных объектов с координатами **xj, yj, zj ()**. Информационная ситуация пусть характеризуется **N** измерениями каждой станцией дальности и азимута каждого объекта. Тогда поставляемая системой Bluetooth-маяков навигационная информация может быть формализована **mnN** уравнением следующего вида:

**(1)**

где – данные об измерении i-ой станцией дальности по j-му объекту, – случайные инструментальные погрешности измерений дальности.

Целью решения задачи **(1)** является нахождение вектора **s** с компонентами **xj, yj, zj, x(i), y(i), z(i), ()** по **N** измерений , который полностью определяет конфигурацию системы наблюдения централизованного типа. Таким образом, модель **(1)** может быть принята в качестве базовой для решения задачи выставки такой системы.

Что касается систем транспондерного типа, то их элементы должны быть привязаны не относительно, а к глобальной (в данном случае - географической) системе координат. В настоящее время такая привязка выполняется, как правило, с использованием данных спутниковых навигационных систем второго поколения (GPS, ГЛОНАСС). Эти системы в настоящее время способны обеспечить точность определения местоположения до 5-10м. Указанная точность вполне достаточна для выставки системы наблюдения, построенной на базе двухкоординатных и работающей по морским целям. Однако для систем наблюдения с навигационными функциями расширенного диапазона, например, работающим также и по воздушным объектам (находящимся над акваторией в зоне действия системы), такая выставка является слишком грубой.

Нужная точность выставки, однако, может быть достигнута. Одним из путей к этому является использование специальных дифференциальных режимов работы глобальных спутниковых навигационных систем, позволяющих, наряду с определением глобальных координат, с высокой точностью определять расположение нескольких объектов по отношению друг к другу. Другой путь повышения качества решения проблемы выставки - применение специальных алгоритмов комплексирования глобальной (спутниковой) и локальной (радиолокационной) навигационной информации.

В свете вышесказанного проблема выставки транспондерной системы наблюдения может быть, таким образом, естественно разделена на две задачи. Во-первых, задачу относительной выставки элементов по типу централизованной системы **(1)**, обеспечивающей в дальнейшем работу системы в автономном режиме. Во-вторых, для поддержки транспондерного режима - задачу привязки системы **oxyz** к глобальной системе **ψλR** (**ψ** -широта, **λ** - долгота, **R** - расстояние до центра Земли).

## Концепция решения

Рассмотрим задачу (1). При решении задач такого типа (по сути - обратных задач), можно использовать два методологических подхода. Во-первых, непосредственную обработку данных всех 2mnN измерений, и, во-вторых, обработку измерений с предварительным осреднением согласно правилу:

**,**

где **(k)** - результат **k-**го измерения дальности **i-**ой станцией по **j**-му объекту. Второй подход более предпочтителен в данной задаче, так как позволяет перейти от системы **2mnN** уравнений к **2mn** уравнениям.

Что касается методов решения системы **(1)** с числом уравнений **2mn**, то в ряде простейших случаев, когда **2mn=dim(s)** (например, при **n=2, m=1, s=(x1, y1, x(2), y(2))**) она вполне может быть разрешена аналитически как система алгебраических уравнений. При больших значениях **m**, **n** и **dim(s)**, когда **2mn>dim(s)**, для решения системы **(1)** может быть успешно применён метод наименьших квадратов с минимизацией функционала:

,

где – измеренные значения дальности и угла, – оцененные значения дальности (при конкретных значениях **–** оценок искомого вектора), – значения соответствующих среднеквадратичных погрешностей измерений. При этом для анализа и решения задачи

**(2)**

целесообразно прибегнуть к методологически общему элементу решения многих нелинейных задач - линеаризации.

Линеаризация системы **(1)** около некоторого опорного решения, характеризующего априорные представления об искомых параметрах, позволяет привести её к следующей задаче «в малом»:



**(3)**

,  

, или, в общем виде,

**,**  **(4)**

где **δs** - искомый вектор (с компонентами **δxj, δyj, δzj, δx(i), δy(i), δz(i)**), **δz** - полный вектор невязок измерений (с учётом процедуры предварительного осреднения **dim(δz)=2mn**), **H** - матричный коэффициент, формируемый согласно равенствам **(3)** (**dim(H)=2mn × dim(s)**), **ξ** - вектор погрешностей измерений (**dim(ξ)=2mn**), причем введем обозначение **Q=M[ξξT] (M[\*]** - оператор математического ожидания).

Функционал **J** примет для задачи **(4)** следующий вид:

**,**

при этом решение задачи его минимизации **(2)** может быть записано следующей процедурной моделью:

**.**  **(5)**

Представление задачи **(1)** в виде **(4)** и **(5)** не только предоставляет конструктивный алгоритм ее решения, но и дает возможность априорного анализа свойств ее решения, позволяет судить о разрешимости исходной задачи. При этом разрешимость задачи понимается как совокупность трех связных понятий: общесистемная разрешимость, или наблюдаемость по Калману; разрешимость при конечной точности модельных представлений (представления чисел в ЭВМ и инструментальных погрешностей, например, измерений) отождествляемая с понятием численной устойчивости; разрешимость проблемы ускорения вычислений.

В случае задачи **(4)** её общесистемная разрешимость определяется полнотой ранга матрицы H и не вырожденностью матрицы **HTQ-1H**. Что же касается ответа на вопрос об устойчивости процедуры **(5)** в условиях погрешностей вычислений (обусловленных конечностью разрядной сетки ЭВМ), то для этого имеет смысл обратиться к следующему условию достаточного типа:

**μ <** , **(6)**

где **μ** - вычисленное значение числа обусловленности матричного оператора **HTQ-1H** по отношению к его спектральной норме, а - некоторое критическое число, в частности, при решении задачи **(5)** с помощью ортогональных преобразований Хаусхолдера определяемое как:

**,**

где **u=dim(s)**, **ε1** - относительная точность представления чисел в ЭВМ (например, при расширенной (extended, long double) точности ЭВМ IBM PC ε1≈10-20).

## 2.7.2 Определение калибровки Bluetooth-маяков

Для вычисления координат объекта необходимо произвести процесс калибровки каждого Bluetooth-маяка, т.е. подобрать такие значения коэффициентов и – калибровочные значения уровня сигнала и расстояния i-го датчика, – коэффициент влияния внешних факторов среды на точность измерений, что погрешность вычисленных значений минимальна.

Значение коэффициента , по умолчанию считаем равным 1, остальные коэффициенты заполняются путем проведения соответствующих экспериментов. Так как данные эксперименты уже были проделаны, поэтому будем считать коэффициенты и равными -68 и 2 соответственно.

# 3 Проектирование системы Indoor-навигация

## 3.1 Архитектура системы Indoor-навигация

### 3.1.1 Use-cases. Диаграмма вариантов использования

Диаграмма вариантов использования состоит из графической диаграммы, описывающей действующие лица и прецеденты, а также спецификации, представляющего собой текстовое описание конкретных последовательностей действий (потока событий), которые выполняет пользователь при работе с системой. Спецификация станет основой для тестирования и документации. Кроме того, use-case диаграмма достаточно проста, чтобы ее мог понять заказчик, следовательно, ее можно использовать для согласования ТЗ.

На диаграмме использования изображаются:

актеры – группы лиц или систем, взаимодействующих с системой;

варианты использования (прецеденты) – сервисы, которые система предоставляет актерам;

отношения между элементами диаграммы.

Для работы с прототипом приложением предусмотрен один тип пользователя: admin. Ниже на Рисунке 1 представлена диаграмма, отображающая функции пользователя admin.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – функции пользователя admin

Для работы с приложением в режиме admin необходимо запустить приложение на устройстве и нажать кнопку старт.

Под «Калибровать маячки» подразумевается ввод соответствующих калибровочных коэффициентов при добавлении маячка.

Под «Просматривать промежуточные данные по маячкам» подразумевается просмотр карточки маячка, на которой отображены координаты маячка, калибровочные коэффициенты, уровень сигнала в конкретный момент времени, а также график с отображением уровня сигнала и соответствующей ему дистанцией до маяка.

Под «Узнать свое местоположение» подразумевается автоматическое вычисление местоположения пользователя в реальном времени, относительно n – введенных маяков и k – проделанных измерений (n и k вводит пользователь).

### 3.1.2 Диаграмма перехода из состояния в состояние

Диаграмма состояний – это диаграмма состояний из теории автоматов со стандартизированными условными обозначениями, которая может определять множество систем от компьютерных программ до бизнес-процессов. Используются следующие условные обозначения:

* скругленный прямоугольник, обозначающий состояние (верхушка прямоугольника содержит название состояния, в середине может быть горизонтальная линия, под которой записываются активности, происходящие в данном состоянии);
* стрелка, обозначающая переход (название события, вызывающего переход, отмечается рядом со стрелкой);
* толстая горизонтальная линия с либо множеством входящих линий и одной выходящей, либо одной входящей линией и множеством выходящих, обозначающая объединение и разветвление.

Диаграмма состояний и переходов показывает:

* пространство состояний данного класса;
* события, которые влекут переход из одного состояния в другое;
* действия, которые происходят при изменении состояния.

На Рисунке 2 представлена диаграмма перехода из одного состояния в другое.

Для работы с приложением в режиме admin необходимо запустить приложение на устройстве и нажать кнопку старт, система проверит устройство на наличие Bluetooth. После успешной проверки, admin видит список маяков и дополнительные возможности. Admin может выполнить любую из функций:

* калибровка датчиков;
* добавление;
* удаление;
* просмотр местоположения и т.д.

Система предусматривает проверку введенных (измененных) данных на корректность и позволяет повторно выполнять имеющиеся функции.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 - Диаграмма перехода из состояния в состояние

### 3.1.3 Диаграмма потоков данных

Диаграммы потоков данных (Data Flow Diagrams) представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления — продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Основными компонентами диаграмм потоков данных являются:

Внешние сущности (англ. External Entity). Это любые объекты, которые не входят в саму систему, но являются для нее источником информации либо получателями какой-либо информации из системы после обработки данных.

Процесс (англ. Process) - функция или последовательность действий, которые нужно предпринять, чтобы данные были обработаны.

Хранилище данных (англ. Data store). Внутреннее хранилище данных для процессов в системе.

Поток данных (англ. Data flow). В нотации отображается в виде стрелок, которые показывают, какая информация входит, а какая исходит из того или иного блока на диаграмме.

Главная цель декомпозиции DFD-функций - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами. На схемах бизнес-процесса отображаются: функции процесса; входящая и исходящая информация, при описании документов; внешние бизнес-процессы, описанные на других диаграммах; точки разрыва при переходе процесса на другие страницы.

Правила и рекомендации построения модели DFD в основном совпадают с принятыми в IDEF0. По аналогии с IDEF0 у каждого процесса (подсистемы) на диаграмме потоков данных должен быть как минимум один входящий и один выходящий поток. Процесс должен запускаться на выполнение либо через обрабатываемый, либо через управляющий поток данных. Работа каждого процесса должна завершаться конкретным результатом.

Каждый накопитель данных также должен иметь как минимум один входящий и один выходящий поток. Наличие только входящих потоков в накопитель означает, что информация накапливается, но не используется.

На рисунке 3 представлена диаграмма потоков данных программной системы Indoor - навигация. Компоненты диаграммы:

* Внешние сущности: пользователь.
* Процессы: Система Indoor-навигации, Система расчета дистанции, Система определения местоположения пользователя
* Хранилища: Bluetooth-маяки.

Система Indoor-навигации состоит из двух подсистем: системы определения местоположения пользователя и системы расчета дистанции от маячка до пользователя.

Система определения местоположения пользователя получается из хранилища «Bluetooth-маяки» данные - уровни сигнала до видимых пользователю маяков. На основе этих данных система рассчитывает дистанцию до ближайших маячков, определяет местоположение пользователя и передает эти данные системе Indoor-навигации.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 3 – Диаграмма потоков данных

## 3.2 Библиотека для работы с Bluetooth маячками

Для решения задачи определения местоположения пользователя необходимо реализовать взаимодействие программного средства с Bluetooth маячками. Эту задачу помогает решить библиотека android.bluetooth для работы с Bluetooth устройствами и ее классы:

* android.bluetooth.BluetoothAdapter,
* android.bluetooth.BluetoothManager,
* android.bluetooth.le.BluetoothLeScanner,
* android.bluetooth.le.ScanCallback,
* android.bluetooth.le.ScanResult.

Библиотека android.bluetooth включает в себя классы, которые позволяют разработчику управлять функциями Bluetooth, такими как сканирование устройств, подключение к устройствам и передача данных между устройствами. Bluetooth API поддерживает как «Классические Bluetooth», так и BLE устройства.

API-интерфейсы Bluetooth позволяют приложениям:

1. производить сканирование других устройств Bluetooth
2. запрашивать локальный адаптер Bluetooth для сопряженных устройств
3. устанавливать RFCOMM каналы/сокеты
4. подключаться к указанными сокетам на других устройствах
5. обмениваться данными с другими устройствами
6. осуществлять связь с устройствами BLE, такими как датчики приближения, мониторы сердечного ритма, Bluetooth-маячки, фитнес устройства и т.д.
7. выступать в качестве клиента GATT или сервера GATT (BLE).

## 3.3 Требования

### 3.3.1 Требования к навигационной системе Indoor-навигация

### Функциональные требования

1.1 Система должна производить измерения уровня сигнала для всех видимых Bluetooth-маячков

1.2 Система должна выводить на экран вычисленные значения дистанции до ближайших Bluetooth-маячков

1.3 Система должна позволять пользователю указывать количество маячков для расчета и количество измерений

1.4 Система должна выводить данные в логи

1.5 Система должна запрашивать доступ к Bluetooth при первом запуске (и каждом последующем если доступ не получен)

**Требования к стабильности**

2.1 Система должна непрерывно считывать данные с Bluetooth-маячков

2.2 Время расчетов для одного объекта не должно превышать 30 секунд

**Требования к переносимости**

3.1 Система должна корректно работать при версии Android 4.4 и выше

**Аппаратные требования к устройству**

4.1 Система должна корректно работать при версии bluetooth не ниже 4.0

### 3.3.2 Требования к подсистемам мобильного приложения «NaviBlue»

Мобильное приложение «NaviBlue» представляет собой прототип системы позиционирования, предназначенной для определения координат объекта в 3х мерном пространстве.

Мобильное приложение состоит из следующих подсистем:

1. Нахождение маячка;
2. Добавление маячка;
3. Вывод списка маячков;
4. Удаление маячка;
5. Просмотр данных о маячке;
6. Вычисление координат объекта;
7. Вычисление дистанции до маячка.

**Требования к подсистеме «Нахождение маячка»**

Требование REQ\_FIND\_001

При нажатии пользователем на кнопку «Добавить маячок» система должна запустить поиск маячков, а затем вывести информацию о всех Bluetooth маяках, находящихся в пределах видимости системы.

**Требования к подсистеме «Добавление маячка»**

Требование REQ\_ADD\_001

При нажатии пользователем на кнопку «Добавить маячок» система должна отображать новое окно, позволяющее выбрать маячок из списка предложенных и ввести координаты его местоположения.

Требование REQ\_ADD\_002

После выбора маячка, ввода его координат и нажатия пользователем на кнопку «Сохранить» система должна добавить новый маячок.

Требование REQ\_ADD\_003

Если пользователь не выбрал маячок и/или не ввел его координаты, но нажал на кнопку «Сохранить» система должна вывести сообщение «Для начала заполните поля!».

**Требования к подсистеме «Вывод списка маячков»**

Требование REQ\_LIST\_001

При нажатии пользователем на кнопку «Список маячков» система должна выдавать список Bluetooth маячков, добавленных самим пользователем.

**Требования к подсистеме «Удаление маячка»**

Требование REQ\_DEL\_001

После выбора маячка и нажатия кнопки удалить, маячок должен удалиться из списка маячков.

**Требования к подсистеме «Просмотр данных о маячке»**

Требование REQ\_BEACON\_001

При нажатии пользователем на конкретный маячок из списка система должна вывести в новом окне подробную информацию о нем.

**Требования к подсистеме «Вычисление координат объекта»**

Требование REQ\_COORD\_001

Система должна высчитывать координаты объекта получая уровень сигнала и координаты Bluetooth маячков, а также отображать этот уровень сигнала пользователю.

**Требования к подсистеме «Вычисление дистанции до маячка»**

Требование REQ\_DIST\_001

Система должна высчитывать дистанцию от объекта до маячка получая уровень сигнала Bluetooth маячка, а также отображать этот уровень сигнала пользователю.

## 3.4 Рекомендации для кодирования

Ниже приведены рекомендации по тому, как писать код для языка java, включая стандартную библиотеку, входящую в состав java.

**Внешний вид кода**

Рекомендуется использовать 4 пробела на каждый уровень отступа [18]. Табуляция должна использоваться только для поддержки кода, написанного с отступами с помощью табуляции. Также рекомендуется не смешивать табуляции и пробелы в отступах.

Необходимо ограничивать длину строки в 79 символов. Предпочтительный способ переноса длинных строк является использование подразумеваемых продолжений строк.

Каждый импорт, как правило, должен быть на отдельной строке. Импорты всегда помещаются в начале файла, сразу после комментариев к модулю и строк документации, и перед объявлением констант. Шаблоны импортов (import \*) следует избегать, так как они делают неясным то, какие имена присутствуют в глобальном пространстве имён, что вводит в заблуждение как читателей, так и многие автоматизированные средства.

**Пробелы в выражениях и инструкциях**

Избегайте использования пробелов в следующих ситуациях:

* Непосредственно внутри круглых, квадратных или фигурных скобок;
* Непосредственно перед запятой, точкой с запятой или двоеточием;
* Сразу перед открывающей скобкой, после которой начинается список аргументов при вызове функции;
* Сразу перед открывающей скобкой, после которой следует индекс или срез;
* Использование более одного пробела вокруг оператора присваивания (или любого другого) для того, чтобы выровнять его с другим.

**Комментарии**

Комментарии должны являться законченными предложениями. Если комментарий – фраза или предложение, первое слово должно быть написано с большой буквы, если только это не имя переменной, которая начинается с маленькой буквы.

**Соглашения по именованию**

Модули должны иметь короткие имена, состоящие из маленьких букв. Можно использовать символы подчеркивания, если это улучшает читабельность. То же самое относится и к именам пакетов, однако в именах пакетов не рекомендуется использовать символ подчёркивания.

Имена классов должны следовать соглашению CapWords.

Имена функций должны состоять из маленьких букв, а слова разделяться символами подчеркивания – это необходимо, чтобы увеличить читабельность.

Для имен методов и переменных экземпляров классов рекомендуется использовать тот же стиль, что и для имен функций: имена должны состоять из маленьких букв, а слова разделяться символами подчеркивания, используйте один символ подчёркивания перед именем для непубличных методов и атрибутов.

Константы обычно объявляются на уровне модуля и записываются только заглавными буквами, а слова разделяются символами подчеркивания.

В приложение А показан участок кода, оформленный согласно данным рекомендациям.

# 4 Реализация и тестирование мобильного приложения «NaviBlue» для 3D позиционирования внутри помещения с использованием Bluetooth-маячков.

В рамках ВКР было разработано и протестировано мобильное приложение для позиционирования внутри помещения.

## 4.1 Спецификация

* Мобильное приложение разрабатывается для мобильных устройств на операционной системе Android начиная с версии 4.4;
* Мобильное приложение разрабатывается для мобильных устройств, которые поддерживают стандарт Bluetooth 4.0 LE и выше;
* Мобильное приложение позволяет отслеживать Bluetooth-маячки в радиусе видимости для добавления их;
* Мобильное приложение позволяет конфигурировать Bluetooth-маячки, путем добавления значений соответствующих коэффициентов, при добавлении маячков;
* Мобильное приложение позволяет отслеживать передаваемые данные добавленных Bluetooth-маячков и выводить их на экран мобильного устройства;
* Мобильное приложение в реальном времени отображает изменения местоположения пользователя на основе изменяющейся информации, получаемой от видимых маячков

## 4.2 Среда разработки

Для реализации мобильного приложения «NaviBlue», была выбрана операционная система Android.

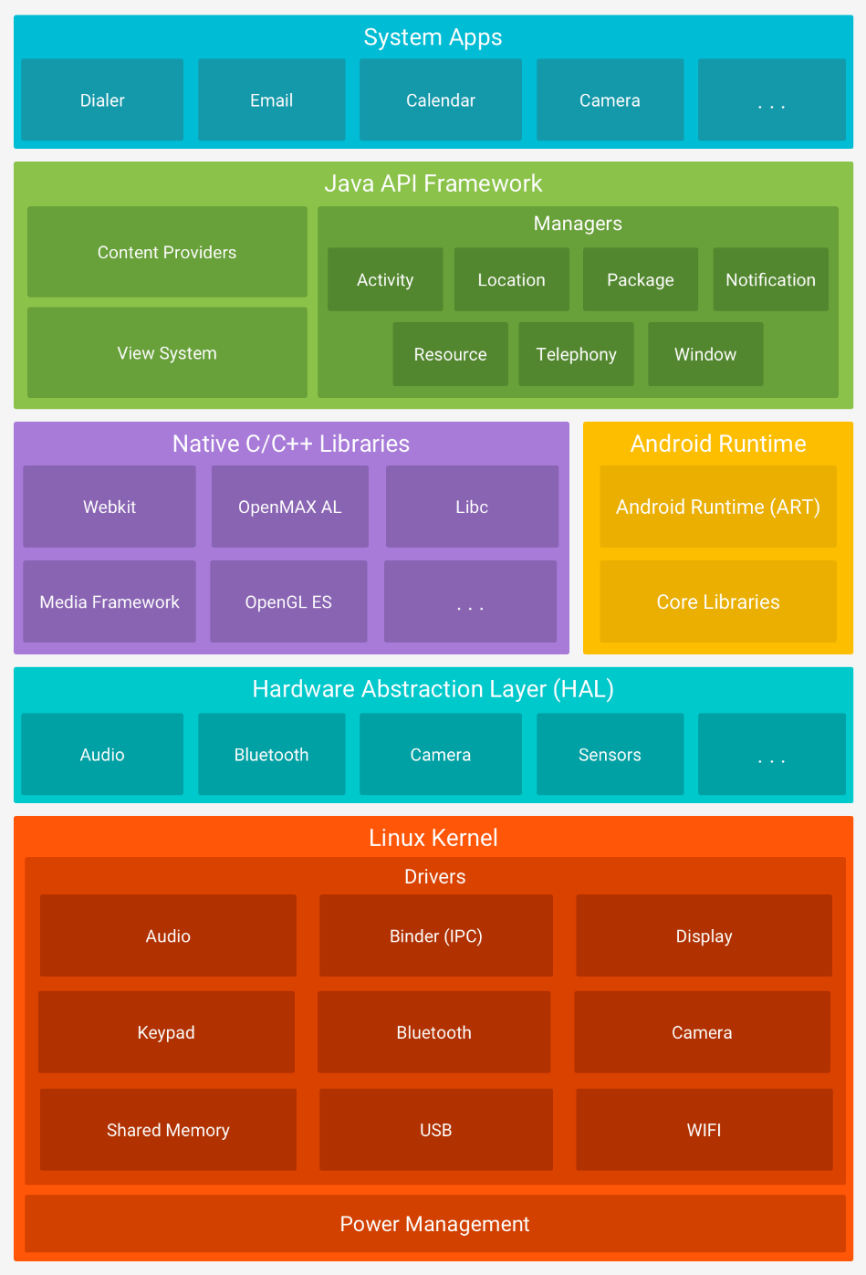


Рисунок 4 – Архитектура операционной системы Android

Android — это программный стек с открытым исходным кодом, основанный на Linux, созданный для широкого спектра устройств и форм-факторов. На рисунке 4 показаны основные компоненты платформы Android.

Основные достоинства разработки для этой системы:

* Простота интерфейса.
* Удобство использования.
* Простая интеграция
* Простое усваивание
* Простое распространение приложений
* Лучшая совместимость устройств

Реализация программы производилась в среде разработки приложений Android Studio, на языке программирования Java.

Основные преимущества выбранного языка программирования:

* Объектно-ориентированный.
* Платформонезависимый.
* Безопасный: проверка подлинность осуществляется путем шифрования с открытым ключом.
* Портативный: не имеющий зависимости от реализации аспектов спецификаций и архитектурно-нейтральный.
* Интерпретированный: байт-код Java переводится моментально в машинные инструкции и нигде не сохраняется.
* Многопоточный: возможность написания программы, которая может выполнять множество задач одновременно.
* Динамический: язык Java предназначен для адаптации к меняющимся условиям.
* Высокопроизводительный: компилятор Just-In-Time, позволяет получить высокую производительность.

Для выбора среды разработки, был проведен сравнительный анализ нескольких сред, представлен в таблице 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Среда разработки | Язык программирования | Удобство интерфейса | Поддерживаемые платформы | Плата |
| Android Studio | Java, C/C++, Delphi, Kotlin | Да | Android | Бесплатно |
| NetBeans | Java, Python, PHP, JavaScript, C, C++, Ада | Нет | Android, IOS, Cordova | Бесплатно |
| Eclipse IDE | Java, C/C++, PHP, Ruby, Python, Cobol | Да | Android, IOS, Windows Phone, Symbian | Бесплатно |

Таблица 1. Сравнительный анализ сред разработки

После проведения анализа, максимально подходящей средой для разработки была выбрана Android Studio [15]. Среда Android Studio с 2014 года была признана компанией Google официальной средой разработки под ОС Android.

## 4.3 Описание методов и классов

Мобильное приложение «NaviBlue» состоит из следующих классов:

1. MainActivity
2. BeaconsList
3. BeaconInfo
4. DBHelper
5. MainDeviceList
6. PositionActivity

Опишем реализованные классы и их методы.

MainActivity – данный класс отвечает за отображение стартового экрана мобильного приложения и проверку устройства на наличие bluetooth (рисунки 5.1 и 5.2).

Изображение выглядит как рисунок

Автоматически созданное описание

Рисунок 5.1 – Отображение стартового экрана

Изображение выглядит как снимок экрана, монитор, телефон, экран

Автоматически созданное описание

Рисунок 5.2 – Проверка устройства на наличие bluetooth

BeaconsList – данный класс отвечает за отображения пользователю списка добавленных Bluetooth-маячков (рисунок 6).

Изображение выглядит как снимок экрана, телефон

Автоматически созданное описание

Рисунок 6 – Отображение списка маяков

BeaconInfo – данный класс отвечает за отображение информации о выбранном маяке, а также за возможность удаления маячка (рисунок 7.1). Также данный класс имеет методы непрерывное получения уровня сигнала от маячка и расчет дистанции до Bluetooth-маячка (рисунки 7.2 и 7.3).

Изображение выглядит как снимок экрана, сидит

Автоматически созданное описание

Рисунок 7.1 – Отображение информации о маяке и удаление данных о маячке

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 7.2 – Метод получения уровня сигнала маячка

Изображение выглядит как снимок экрана, черный, держит, проигрыватель

Автоматически созданное описание

Рисунок 7.3 – Расчет дистанции до маяка

DBHelper – данный класс описывает интерфейс записи и чтения из базы данных. При запуске приложения, в случае отсутствия файла базы создает новую состоящую из таблиц Beacons и Signal (рисунок 8).

Изображение выглядит как снимок экрана, экран, черный, большой

Автоматически созданное описание

Рисунок 8 – Создание и управление БД

MainDeviceList – данный класс отвечает за поиск не добавленных маячков, а также за добавление маячков в базу данных (рисунки 9.1 и 9.2).

Изображение выглядит как снимок экрана, телефон

Автоматически созданное описание

Рисунок 9.1 – Поиск Bluetooth-маячков

Изображение выглядит как снимок экрана, сидит, стол, телефон

Автоматически созданное описание

Рисунок 9.2 – Добавление маячков

PositionActivity – данный класс отвечает за вычисление местоположения пользователя. Содержит в себе методы:

* filtr – позволяет ограничить количество используемых маячков, а также изменить количество измерений для получения более точных координат (рисунок 10.1);
* showBeacons – отображает ближайшие маячки, уровень сигнала и дистанцию до каждого маячка (рисунок 10.2);
* getDistance – вычисляет расстояние до маячка (рисунок 10.3);
* getPosition – вычисляет координаты пользователя, относительно местоположения маячков (рисунок 10.4).

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 10.1 – Фильтр количества маячков и количества измерений

Изображение выглядит как снимок экрана, текст, сидит, телефон

Автоматически созданное описание

Рисунок 10.2 – Отображение ближайших маячков

Изображение выглядит как снимок экрана, черный, держит, проигрыватель

Автоматически созданное описание

Рисунок 10.3 – Расчет дистанции до маяка

Изображение выглядит как снимок экрана, текст

Автоматически созданное описаниеИзображение выглядит как снимок экрана, стол, сидит, ноутбук

Автоматически созданное описание

Рисунок 10.4 – Определение координат пользователя

## 4.4 Интерфейс пользователя

### 4.4.1 Стартовый экран

Изображение выглядит как коробка, часы

Автоматически созданное описание

Рисунок 11.1 – Иконка приложения

Изображение выглядит как рисунок

Автоматически созданное описание

Рисунок 11.2 – Стартовый экран

При нажатии на иконку приложения (рисунок 11.1) перед пользователем открывается стартовый экран (рисунок 11.2). Данный экран представляет начальное взаимодействие пользователя с приложением. При нажатии кнопки «СТАРТ», перед пользователем откроется экран «Список маяков».

### 4.4.2 Экран списка маяков

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 11.3 – Список маяков

На данном экране (рисунок 11.3) пользователю отображается весь список добавленных им маячков, при нажатии на один из маячков пользователю откроется экран «Информация». Также на данном экране присутствуют 2 кнопки «Добавить маяк» и «Местоположение», эти кнопки открывают пользователю экран добавления маяка и экран отображения местоположения пользователя соответственно.

### 4.4.3 Экран информации о маяке

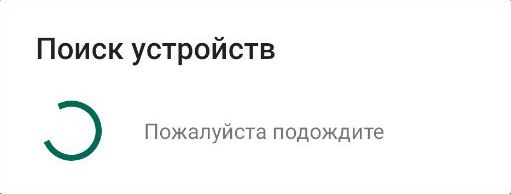
Изображение выглядит как часы, счетчик

Автоматически созданное описание

Рисунок 11.4 – Информация о маяке

На данном экране (рисунок 11.4) отображается информация о маяке, расстояние до маяка и график, на котором отображен уровень сигнала и расстояние до маяка. Еще данный экран содержит 2 кнопки, кнопка «Назад» и кнопка «Удалить». При нажатии кнопки «Назад», перед пользователем откроется окно со списком маяков, а при нажатии кнопки «Удалить» данные об этом маяке будут удалены и пользователю откроется новый список маяков.

### 4.4.4 Экран добавления маяка

Изображение выглядит как птица, дерево, цветок

Автоматически созданное описание

Рисунок 11.5 – Поиск новых Bluetooth-маяков

Изображение выглядит как текст

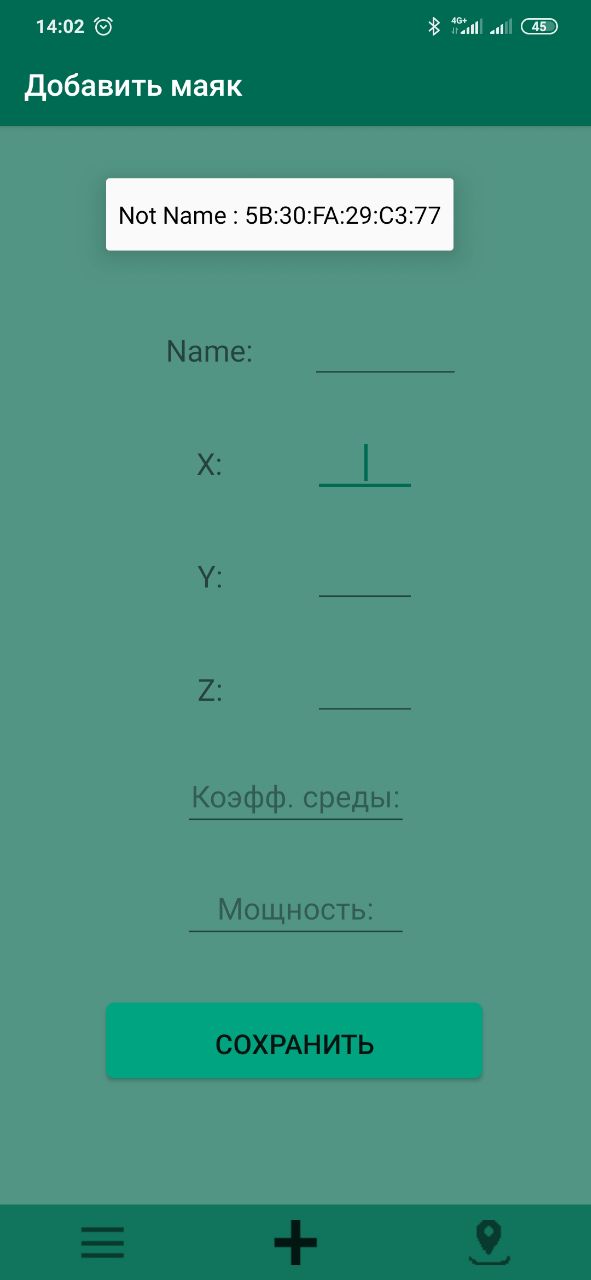
Автоматически созданное описание

Рисунок 11.6 – Добавить маяк

Изображение выглядит как нож

Автоматически созданное описание

Рисунок 11.7 – Сообщение пользователю

При открытии экрана «Добавить маяк», система осуществляет поиск новых Bluetooth-маячков (рисунок 11.5). На данном экране (рисунок 11.6) пользователь может добавить новый маяк, если он есть рядом. При нажатии кнопки сохранить, если поля X, Y, Z заполнены, перед пользователем открывается новый список маяков, если поля не заполнены, пользователю отображается сообщение (рисунок 11.7).

Кнопки «Список маяков» и «Местоположение», открывают пользователю экран списка маяков и экран отображения местоположения пользователя соответственно.

### 4.4.5 Экран определения местоположения

Изображение выглядит как снимок экрана

Автоматически созданное описаниеИзображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 11.8 – Экран определения местоположения

На экране «Местоположение» (рисунок 11.8), происходит определение местоположения пользователя с изменяемыми параметрами «количество маяков» и «количество измерений», при помощи кнопок «Фильтр» и «Изменить». При отсутствии в области видимости устройства 3 и более маячков, система перестает рассчитывать координаты пользователя и выводит соответствующее сообщение.

Кнопки «Список маяков» и «Добавить», открывают пользователю экран списка маяков и экран добавления маячков соответственно.

## 4.5 Тестирование

### 4.5.1 Тестирование подсистем

**Тесты для тестирования подсистемы «Добавление маячка»**

Тест: TEST\_ADD\_001

Тестируемые требования: REQ\_ ADD\_001, REQ\_ ADD\_002

Описание теста:

Выбираем найденный маяк, вводим корректные координаты и сравниваем полученный результат с ожидаемым.

Тест: TEST\_ADD\_002

Тестируемые требования: REQ\_ ADD\_001

Описание теста:

Выбираем найденный маяк, не вводим координаты и сравниваем полученный результат с ожидаемым.

Тест: TEST\_ADD\_003

Тестируемые требования: REQ\_ ADD\_003

Описание теста:

Не выбираем маяк и сравниваем полученный результат с ожидаемым.

**Тесты для тестирования подсистемы «Удаление маячка»**

Тест: TEST\_DEL\_001

Тестируемые требования: REQ\_ DEL\_001

Описание теста:

Выбираем маяк, нажимаем кнопку удалить и сравниваем полученный результат с ожидаемым.

**Тесты для тестирования подсистемы «Вывод списка маячков»**

Тест: TEST\_LIST\_001

Тестируемые требования: REQ\_LIST\_001

Описание теста:

Нажимаем кнопку «Список маяков» и сравниваем полученный результат с ожидаемым.

**Тесты для тестирования подсистемы «Нахождение маячка»**

Тест: TEST\_FIND\_001

Тестируемые требования: REQ\_FIND\_001

Описание теста

Проверяем поиск маячков при выключенном Bluetooth и сравниваем полученный результат с ожидаемым.

Тест: TEST\_FIND\_002

Тестируемые требования: REQ\_FIND\_001

Описание теста

Проверяем поиск маячков при включенном Bluetooth и сравниваем полученный результат с ожидаемым.

**Тесты для тестирования подсистемы «Просмотр данных о маячке»**

Тест: TEST\_BEACON\_001

Тестируемые требования: REQ\_BEACON\_001

Описание теста:

Нажимаем кнопку «Список маяков», выбираем из списка маячок и сравниваем полученный результат с ожидаемым.

**Тесты для тестирования подсистемы «Вычисление координат объекта»**

Тест: TEST\_COORD\_001

Тестируемые требования: REQ\_COORD\_001

Описание теста:

Открываем местоположение пользователя, который находится в зоне действия 3-х и более Bluetooth-маячков и получаем координаты, сравниваем результат с ожидаемым.

Тест: TEST\_COORD\_002

Тестируемые требования: REQ\_COORD\_001

Описание теста:

Открываем местоположение пользователя, который находится в зоне действия менее 3-х Bluetooth-маячков и получаем координаты, сравниваем результат с ожидаемым.

Тест: TEST\_COORD\_003

Тестируемые требования: REQ\_COORD\_001

Описание теста:

Открываем местоположение пользователя, который находится вне зоны действия Bluetooth-маячков и получаем координаты, сравниваем результат с ожидаемым.

**Тесты для тестирования подсистемы «Вычисление дистанции до маячка»**

Тест: TEST\_DIST\_001

Тестируемые требования: REQ\_DIST\_001

Описание теста:

Открываем информацию о маяке, который находится в зоне действия Bluetooth и получаем дистанцию до маячка, сравниваем результат с ожидаемым.

Тест: TEST\_DIST\_002

Тестируемые требования: REQ\_DIST\_001

Описание теста:

Открываем информацию о маяке, который находится вне зоны действия Bluetooth и получаем дистанцию до маячка, сравниваем результат с ожидаемым.

### 4.5.2 Покрытие тестами требований

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TEST\_ADD\_001 | TEST\_ADD\_002 | TEST\_ADD\_003 | TEST\_DEL\_001 | TEST\_LIST\_001 | TEST\_FIND\_001 | TEST\_FIND\_002 | TEST\_BEACON\_001 | TEST\_COORD\_001 | TEST\_COORD\_002 | TEST\_COORD\_003 | TEST\_DIST\_001 | TEST\_DIST\_002 |
| REQ\_FIND\_001 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REQ\_ADD\_001 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REQ\_ADD\_002 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REQ\_ADD\_003 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REQ\_LIST\_001 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REQ\_DEL\_001 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REQ\_BEACON\_001 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REQ\_COORD\_001 |  |  |  |  |  |  |  |  |  |  |  |  |  |
| REQ\_DIST\_001 |  |  |  |  |  |  |  |  |  |  |  |  |  |

Таблица 2 – Покрытие требований

### 4.5.3 Тестирование системы с моделируемым сигналом

Для проведения теста было выбрано пустое помещение. На 3-х углах данного помещения были размещены 3 Bluetooth-маячка. Схема помещения представлена на рисунке 12.

Изображение выглядит как карта, текст

Автоматически созданное описание

Рисунок 12 – Схема помещения

Для данного тестирования была выбрана одна реперная точка в середине помещения, в которой были рассчитаны предположительные уровни сигнала, на рисунке 12 она обозначена крестиком в кружке. Предположительное расположение Bluetooth-маячков на рисунке 12 обозначено зелёными кружками.

В реперной точке уровень сигнала был рассчитан 5 раз. Это необходимо для более точного представления работы системы в реальных условиях.

Для возможности измерения расстояний, в экспериментальном помещении была задана система отчета, представленная на рисунке 13. Все точки были тщательно вымерены, поскольку в эксперименте важно расстояние между объектами.

Система координат – трёхмерная. Следовательно, расстояние между точками А1(x1, y1, z1) и A2 (x2, y2, z2) можно вычислить как:

.

Изображение выглядит как карта, текст

Автоматически созданное описание

Рисунок 13 – Заданная система отчета

Расстояния от маячков до нашей точки представлены в таблице 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Маячки | Объект | A (0;0;2.5) | B (4;0;0) | C (4;5;2.5) |
| O (2;2.5;1.25) | | 3.44 | 3.44 | 3.44 |

Таблица 3 – Расстояния от маячков до точки

Так как мы моделируем реальные условия, нам необходимо добавить коэффициент погрешности в наши измерения. Для вычисления предположительных значений уровня сигнала была написана программа, которая представлена на рисунке 14.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 14 – Вычисление прогнозных значений уровня сигнала.

Результаты выполнения данной программы, представлены в таблице 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | маячки | A (0;0;2.5) | B (4;0;0) | C (4;5;2.5) |
| 1 | | 68.42 | 63.42 | 68.42 |
| 2 | | 68.42 | 53.42 | 63.42 |
| 3 | | 48.42 | 58.42 | 63.42 |
| 4 | | 68.42 | 58.42 | 68.42 |
| 5 | | 48.42 | 48.42 | 53.42 |
| Среднее значение: | | 60.42 | 56.42 | 63.42 |

Таблица 4 – Прогнозные значения уровня сигнала

После получения уровня сигнала, произведем расчет координат нашей точки используя метод трилатерации.

В результате, координаты смоделированной точки: O (1.45; 2.64; 0.91).

Погрешность измерения: (2-1.45; 2.5-2.64; 1.25-0.91) = (0.55; -0.14; 0.34).

### 4.5.4 Тестирование системы с реальным сигналом

Проведем тест на практике с реальным использованием Bluetooth-маячков и приложения «NaviBlue».

Добавляем 3 наших Bluetooth-маячка в приложение (рисунок 15).

Изображение выглядит как снимок экрана, рисунок

Автоматически созданное описание

Рисунок 15 – Список добавленных Bluetooth-маячков

Устанавливаем маячки в помещении так же, как указано на рисунке 12.

Для проведения более точных измерений, сигнал будем принимать 5 раз и затем рассчитывать среднее.

Запустим окно «Местоположение» и заполним значения фильтра: количество маяков = 3, количество измерений = 5.

Результат измерения показаны на рисунках 16.1, 16.2, 16.3.

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 16.1 – Результат измерения местоположения

На рисунке 16.1 показаны полученные координаты объекта, а также данные дистанции и уровня сигнала с Bluetooth-маячков 2,7 и 10.

Изображение выглядит как компьютер

Автоматически созданное описание

Рисунок 16.2 – Логи измерений уровня сигнала

На рисунке 16.2 мы видим полученные значения уровня сигнала по 5 раз для каждого из маяков.

Изображение выглядит как окно, белый, женщина

Автоматически созданное описание

Рисунок 16.3 – Логи вычисления координат объекта

На рисунке 16.3 изображены промежуточные результаты вычисления координат объекта с использованием метода бинаризации, как мы видим конечный результат был получен после 3-го пересчета координат.

Сравним результаты работы программы с действительным местоположением объекта:

Полученные координаты объекта: O (0.17; 3.14; 1.72).

Погрешность измерения: (2-0.17; 2.5-3.14; 1.25-1.72) = (1.83; -0.64; -0.47).

Исходя из полученных данных, мы можем сделать вывод, что для получения реальных данных с меньшей погрешностью, необходимо произвести калибровку каждого из датчиков для уменьшения частоты появления некорректного уровня сигнала. Так как на уровень сигнала влияют внешние факторы, то для получения точного результата необходимо увеличить количество маяков и количество измерений.

## 4.6 Навигационная ситуация

Для проведения навигационной ситуации была выбрана одна комната в квартире. В углах комнаты были развешены 8 Bluetooth-маячков. Схема представлена на рисунке 17.

Изображение выглядит как карта

Автоматически созданное описание

Рисунок 17 – Схема комнаты

В таблице 5 показаны координаты маячков и пользователя.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | маяк1 | маяк2 | маяк3 | маяк4 | маяк5 | маяк7 | маяк10 | маяк12 | User |
| x | 0 | 0 | 3 | 3 | 3 | 3 | 0 | 0 | 2 |
| y | 0 | 0 | 0 | 0 | 3 | 3 | 4 | 4 | 1.5 |
| z | 0 | 2 | 0 | 2 | 0 | 2 | 1 | 2 | 0.8 |

Таблица 5 – Координаты объектов

Далее следует запустить приложение «NaviBlue» и добавить Bluetooth-маячки с соответствующими им координатами.

После этого проводится ряд экспериментов, в котором, изменяя число измерений, отслеживаем его влияние на конечный результат (рисунки 18.1, 18.2, 18.3).

Изображение выглядит как текст, карта

Автоматически созданное описание

Рисунок 18.1 – Сравнение полученных координат X

Изображение выглядит как текст, карта

Автоматически созданное описание

Рисунок 18.2 – Сравнение полученных координат Y

Изображение выглядит как текст, карта, снимок экрана

Автоматически созданное описание

Рисунок 18.3 – Сравнение полученных координат Z

Из данного эксперимента делаем вывод, что на конечную погрешность результата, достаточно сильно влияет количество проделанных измерений.

По итогу, координаты пользователя были вычислены с характерной погрешностью (рисунок 19).

Изображение выглядит как снимок экрана

Автоматически созданное описание

Рисунок 19 – Результат работы программы

Для получения более точного результата необходимо произвести калибровку каждого датчика, сделать точные измерения и увеличить количество маяков.

# Заключение

В ходе выполнения выпускной квалификационной работы были полностью решены поставленные задачи.

В первой главе выпускной квалификационной работы:

* был сделан обзор существующих систем в области «Indoor-навигация»
* рассмотрены прикладные задачи indoor-навигации в различных предметных областях
* рассмотрены методы indoor-навигации с помощью различных устройств

Во второй главе выпускной квалификационной работы:

* был проведен анализ предметной области «Indoor-навигация»
* были выделены задачи обработки, хранения и передачи информации
* была построена модель предметной области
* была построена математическая модель задачи позиционирования объекта

В третьей главе выпускной квалификационной работы был разработан проект системы, состоящий из:

* представления объектов в системе
* архитектурно-контекстной диаграммы
* диаграммы потоков данных
* проекта данных
* требований к навигационной системе Indoor-навигация
* рекомендаций для кодирования

В четвертой главе выпускной квалификационной работы:

* было разработано и протестировано приложение «NaviBlue» для 3D позиционирования внутри помещения
* было проведено исследование точности вычислений местоположения устройства при решении задачи трилатерации с помощью Bluetooth-маяков.

Таким образом, была достигнута цель данной ВКР: создание прототипа системы, предназначенного для определения местоположения объекта в здании с использованием Bluetooth-маяков, разработка алгоритма и оценка точности получаемых данных.

# Список литературы

## Статьи

1. А.В. Монгуш, П.М. Кикин: Обзор технологий Indoor-навигации. – 2017г. – УДК 004.
2. В.М. Гриняк: Обработка навигационных данных при решении задачи выставки многопозиционной системы наблюдения // Территория новых возможностей. Вестник Владивостокского государственного университета экономики и сервиса. 2017. Т. 9, №2. С. 126-139.

## Учебники и учебные пособия

1. Рутледж Д. Энциклопедия практической электроники. — ДМК, 2002. — С. 349-352.
2. Мурашко И.А., Храбров Д.Е. Методика локального позиционирования на основании Wi-Fi-сети университета // Веснік Гродзенскага дзяржаўнага ўніверсітэта імя Янкі Купалы. Серыя 2: Матэматыка. Фізіка. Інфарматыка, Вылічальная тэхніка і кіраванне. – 2015. - №2. – С. 119-127.
3. Казаков Е.Н. Разработка и программная реализации алгоритма оценки уровня сигнала в сети Wi-Fi // Моделирование, оптимизация и информационные технологии. – 2016. - №1. – С. 13.
4. Щёкотов М.С., Кашевник А.М. Сравнительный анализ систем позиционирования смартфонов в помещениях // Труды СПИИРАН. – 2012. - №4. – С. 459-471.

## Электронные ресурсы

1. Маячки Bluetooth Low Energy [Электронный ресурс] //Официальный сайт «Компэл» - Режим доступа: <https://www.compel.ru/lib/74346>
2. Основы Bluetooth Low Energy [Электронный ресурс] //Издательство «РАДИОЛОЦМАН» - Режим доступа: <https://www.rlocman.ru/review/article.html?di=183939>
3. Bluetooth [Электронный ресурс] //Официальный сайт Википедии - Режим доступа: <https://ru.wikipedia.org/wiki/Bluetooth>
4. Мобильное приложение [Электронный ресурс] //Официальный сайт Википедии - Режим доступа: <https://ru.wikipedia.org/wiki/Мобильное_приложение>
5. IBEACON И BLUETOOTH-МАЯЧКИ: КАК ЭТО РАБОТАЕТ В РИТЕЙЛЕ [Электронный ресурс] //Официальный сайт «Facelet» - Режим доступа: https://www.facelet.com/ru-ru/blog/ibeacon-and-bluetooth-beacons-how-to-use-them-in-retail/
6. Indoor GPS and Indoor tracking [Электронный ресурс] // официальный сайт Navizon. – Режим доступа: <https://www.navizon.com/indoors-solutions.php>
7. Инерциальная навигационная система – как это работает [Электронный ресурс] //Официальный сайт «Rostec» - Режим доступа: <https://rostec.ru/news/kak-eto-rabotaet-inertsialnaya-navigatsionnaya-sistema-/>
8. Мобильное позиционирование в GSM-сетях [Электронный ресурс] //Издательство «Нестор». – Режим доступа: <http://www.nestor.minsk.by/kg/2004/06/kg40611.html>
9. Android Studio – Википедия [Электронный ресурс] // официальный сайт Википедии. – Режим доступа: https://en.wikipedia.org/wiki/Android\_Studio
10. E. Martin, O. Vinyals, G. Friedland, and Ruzena Bajcsy Precise Indoor Localization Using Smart Phones // Proceedings of the ACM International Conference on Mul-timedia, Florence, Italy, pp. 787-790, 2010.
11. Fitchard Wifarer’s mobile app doesn’t just map the indoors: It maps the objects within // [Электронный ресурс]. <http://gigaom.com/mobile/wifarers-mobile-app-doesnt-just-map-the-indoors-it-maps-the-objects-within>
12. PEP 8 – Руководство по написанию кода на Python // [Электронный ресурс]. <https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html>

# Приложение А

Класс PositionActivity отвечающий за окно вычисления местоположения пользователя.

public class PositionActivity extends AppCompatActivity {  
  
 float last\_x = 0;  
 float last\_y = 0;  
 float last\_z = 0;  
  
 BluetoothManager btManager;  
 BluetoothAdapter btAdapter;  
 BluetoothLeScanner btScanner;  
 private final static int *REQUEST\_ENABLE\_BT* = 1;  
  
// Button btBack; //кнопка назад  
 ImageButton btSpisok; //кнопка открытия списка маяков  
 ImageButton btAdd; //кнопка открытия добавления маяка  
 Button btFiltr; //кнопка фильтра по кол-ву маяков  
 Button btCorrect; //кнопка изменить фильтр  
  
 EditText etFiltr;  
 EditText etCount;  
  
 DBBeacon dbBeacon; //База данных маячков  
  
 Integer filtr;  
 Integer count;  
  
 Integer countBeacon; //количество видимых маяков  
  
 double[][] matrixDistance; //матрица дистанций до маяков  
 int[][] matrixCoordinate; //координаты маяков  
 int[] matrixPower; // матрица мощностей у маячков  
  
 public static ArrayList <String> *filtrBeacons*; //фильтрация маячков с расстоянием для вывода  
  
 //double measuredPower = -50; //потери в свободном пространстве на расстоянии d0  
 //double n = 2; //коэффициент погрешности зависящий от типа помещения и т.п.  
  
  
  
 @RequiresApi(api = Build.VERSION\_CODES.*M*)  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_position*);  
  
 //скрыть панель навигации начало  
  
 final int flags = View.*SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN* | View.*SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION* | View.*SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY*;  
  
 getWindow().getDecorView().setSystemUiVisibility(flags);  
  
 final View decorView = getWindow().getDecorView();  
 decorView  
 .setOnSystemUiVisibilityChangeListener(new View.OnSystemUiVisibilityChangeListener()  
 {  
  
 @Override  
 public void onSystemUiVisibilityChange(int visibility)  
 {  
 if((visibility & View.*SYSTEM\_UI\_FLAG\_FULLSCREEN*) == 0)  
 {  
 decorView.setSystemUiVisibility(flags);  
 }  
 }  
 });  
  
 //скрыть панель навигации конец  
  
 ActionBar actionBar = getSupportActionBar();  
 actionBar.setTitle("Местоположение");  
 actionBar.setBackgroundDrawable(new ColorDrawable(Color.*parseColor*("#006B53")));  
  
  
 dbBeacon = new DBBeacon(this);  
 dbBeacon.open();  
 dbBeacon.clearSignalTable();  
 dbBeacon.close();  
  
 btSpisok = (ImageButton) findViewById(R.id.*bt\_map\_spisok*);  
 btSpisok.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try{  
 stopScanning();  
  
 //вывод добавленных сигналов  
 Cursor cursor;  
 dbBeacon.open();  
 cursor = dbBeacon.getAllDataSignal();  
  
 if (cursor.moveToFirst()) {  
 int idIndex = cursor.getColumnIndex(DBBeacon.*KEY\_SIGNAL\_ID*);  
 int signalIndex = cursor.getColumnIndex(DBBeacon.*KEY\_SIGNAL\_RSSI*);  
 int addressIndex = cursor.getColumnIndex(DBBeacon.*KEY\_BEACON\_ADDRESS*);  
 do {  
 String s;  
 s = "id = "+ cursor.getString(idIndex) + " : rssi = " + cursor.getString(signalIndex) +  
 " : address = " + cursor.getString(addressIndex) + " ;";  
 Log.*d*("Logm", s);  
  
 } while (cursor.moveToNext());  
 }  
 dbBeacon.close();  
  
  
 Intent intent = new Intent(PositionActivity.this, BeaconsList.class);  
 startActivity(intent);  
 overridePendingTransition(0, 0);  
 finish();  
 }catch (Exception e){  
  
 }  
 }  
 });  
  
 btAdd = (ImageButton) findViewById(R.id.*bt\_map\_add*);  
 btAdd.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try{  
 stopScanning();  
  
 //вывод добавленных сигналов  
 Cursor cursor;  
 dbBeacon.open();  
 cursor = dbBeacon.getAllDataSignal();  
  
 if (cursor.moveToFirst()) {  
 int idIndex = cursor.getColumnIndex(DBBeacon.*KEY\_SIGNAL\_ID*);  
 int signalIndex = cursor.getColumnIndex(DBBeacon.*KEY\_SIGNAL\_RSSI*);  
 int addressIndex = cursor.getColumnIndex(DBBeacon.*KEY\_BEACON\_ADDRESS*);  
 do {  
 String s;  
 s = "id = "+ cursor.getString(idIndex) + " : rssi = " + cursor.getString(signalIndex) +  
 " : address = " + cursor.getString(addressIndex) + " ;";  
 Log.*d*("Logm", s);  
  
 } while (cursor.moveToNext());  
 }  
 dbBeacon.close();  
  
  
 Intent intent = new Intent(PositionActivity.this, MainDeviceList.class);  
 startActivity(intent);  
 overridePendingTransition(0, 0);  
 finish();  
 }catch (Exception e){  
  
 }  
 }  
 });  
  
 btFiltr = (Button) findViewById(R.id.*bt\_pos\_Filtr*);  
 btCorrect = (Button) findViewById(R.id.*bt\_pos\_Correct*);  
  
 etFiltr = (EditText) findViewById(R.id.*et\_pos\_Filtr*);  
 etCount = (EditText) findViewById(R.id.*et\_pos\_Count*);  
  
 btCorrect.setEnabled(false);  
 btFiltr.setEnabled(true);  
  
 btFiltr.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try{  
 //скрыть клавиатуру по нажатию кнопки  
 InputMethodManager imm = (InputMethodManager) getSystemService(Context.*INPUT\_METHOD\_SERVICE*);  
 imm.hideSoftInputFromWindow(btFiltr.getWindowToken(),  
 InputMethodManager.*HIDE\_NOT\_ALWAYS*);  
  
 filtr();  
  
 }catch (Exception e){  
  
 }  
 }  
 });  
  
 btCorrect.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 try{  
 btCorrect.setEnabled(false);  
 btFiltr.setEnabled(true);  
 etFiltr.setEnabled(true);  
 etCount.setEnabled(true);  
 etFiltr.setText(null);  
 etCount.setText(null);  
 stopScanning();  
  
 }catch (Exception e){  
  
 }  
 }  
 });  
  
 btManager = (BluetoothManager)getSystemService(Context.*BLUETOOTH\_SERVICE*);  
 btAdapter = btManager.getAdapter();  
 btScanner = btAdapter.getBluetoothLeScanner();  
  
 if (btAdapter != null && !btAdapter.isEnabled()) {  
 Intent enableIntent = new Intent(BluetoothAdapter.*ACTION\_REQUEST\_ENABLE*);  
 startActivityForResult(enableIntent,*REQUEST\_ENABLE\_BT*);  
 }  
  
 }