

Geekbrains

**Создание Android приложения для GPS навигации в полевых условиях на основе библиотеки ArcGIS Runtime SDK for Android**

Программа: Разработчик - Программист

Специализация: Android-разработка

ФИО: Эпов Семён Алексеевич

Иркутск

2024 г.

**Дипломный проект**

**" Создание Android приложения для GPS навигации в полевых условиях на основе библиотеки ArcGIS Runtime SDK for Android".**

## Содержание

Введение.....	4
Глава 1. Обзор мобильных приложений для GPS навигации с использованием ArcGIS Runtime SDK for Android.....	9
1.1 Особенности мобильного приложения для GPS навигации, его особенности.....	9
1.2 Анализ существующих приложений для GPS навигации .....	10
1.3 Определение требований к новому приложению .....	15
Глава 2. Разработка мобильного приложения “GPS Tracker” .....	18
2.1 Создание нового проекта в Android Studio.....	18
2.2 Разработка основного функционала приложения .....	19
2.2.1 Разработка пользовательского интерфейса.....	19
2.2.2 Программирование основных операций с GPS и файлами .....	23
2.2.3 Библиотеки и технологии.....	33
2.3. Тестирование GPS Tracker на эмуляторах и реальном устройстве	35
Глава 3. Анализ результатов и предложения по улучшению .....	40
3.1. Обзор достигнутых результатов .....	40
3.2. Предложения по улучшению .....	40
3.3. Взгляд в будущее .....	41
Заключение .....	42
Список использованной литературы.....	43
Приложение 1 .....	44
Приложение 2. ....	45
Приложение 3 .....	46

## **Введение**

В современном мире геопространственные данные играют ключевую роль в самых разных сферах деятельности, от научных исследований до управления ресурсами. Особую значимость они приобретают в области геофизических исследований, где точность и эффективность работы с геоданными напрямую влияют на качество получаемых результатов.

Одним из важных методов геофизической разведки является метод зондирования становлением поля (ЗСБ), применяемый для изучения геологического строения земной коры. ЗСБ основан на измерении изменений электромагнитного поля во времени, вызванных искусственным возбуждением импульсным током. Полученные данные позволяют определить распределение удельного электрического сопротивления в земной коре, что, в свою очередь, дает информацию о наличии и расположении полезных ископаемых, водоносных горизонтов, геологических структур и других объектов.

Современные геофизические исследования, включая метод ЗСБ, неразрывно связаны с использованием геоинформационных систем (ГИС). ГИС предоставляют мощный инструментарий для работы с пространственными данными, что существенно расширяет возможности электроразведки и повышает ее эффективность.

ГИС играют важную роль на всех этапах электроразведки методом ЗСБ:

- Планирование работ: ГИС позволяют создавать и анализировать цифровые модели местности, проектировать сети наблюдений с учетом рельефа, инфраструктуры, геологических особенностей и других факторов. Это обеспечивает оптимальное расположение пунктов измерений и повышает эффективность исследований (приложение 2).
- Проведение работ: ГИС используются для навигации и позиционирования в полевых условиях, что облегчает процесс сбора данных и повышает его точность. Также ГИС позволяют оперативно контролировать ход работ и вносить корректировки в план исследований.

- Обработка и анализ данных: ГИС предоставляют инструменты для обработки и интерпретации данных электроразведки, построения карт распределения электропроводности, выявления аномалий и геологических структур.
- Представление результатов: ГИС позволяют создавать наглядные карты, графики и отчеты, которые облегчают понимание результатов исследований и принятие решений.

Примеры успешного применения ГИС в электроразведке методом ЗСБ:

- Поиск месторождений полезных ископаемых: ГИС используются для анализа геологических данных и выявления перспективных участков для проведения электроразведки.
- Изучение геологического строения: ГИС помогают строить трехмерные модели геологической среды на основе данных электроразведки.
- Мониторинг геологических процессов: ГИС используются для отслеживания изменений электропроводности земной коры, связанных с геологическими процессами (например, сдвигами земной коры, изменением уровня грунтовых вод).

Таким образом, интеграция ГИС в процесс электроразведки методом ЗСБ является необходимым условием для получения качественных и достоверных результатов, а также для повышения эффективности и безопасности исследований.

Сбор и анализ геоданных в полевых условиях при проведении ЗСБ — трудоемкий и ответственный процесс. Мобильные устройства, оснащенные GPS-модулями и мощными процессорами, открывают новые возможности для создания удобных и функциональных приложений, способных значительно упростить и ускорить работу с геопространственной информацией.

В рамках данной дипломной работы разработано мобильное приложение для платформы Android под названием "GPS Tracker", которое предоставит

специалистам, работающим методом ЗСБ, удобный и функциональный инструмент для работы с геоданными и GPS-трекингом, способствующий повышению эффективности и точности полевых исследований. Для достижения поставленной цели необходимо решить следующие задачи:

1. Анализ требований к функционалу приложения: детальное изучение потребностей пользователей ЗСБ в области работы с геоданными в полевых условиях, определение ключевых функций и возможностей приложения.

2. Проектирование интерфейса пользователя: создание интуитивно понятного и удобного интерфейса, обеспечивающего быстрый доступ к основным функциям приложения и комфортную работу с геоданными.

3. Реализация функционала приложения: разработка программного кода, реализующего запланированный функционал, включая:

- Запись треков: фиксация маршрута движения с привязкой к координатам и времени.
- Отображение местоположения на карте: визуализация текущего местоположения на интерактивной карте.
- Разработка функционала для работы с геопространственными данными:
  - Загрузка и отображение данных из файлов SHP, TIF и KML на карте.
  - Визуализация точек, линий и полигонов с использованием соответствующей символики.
  - Возможность анализа данных (например, определение расстояний, площадей, построение профилей).
  - Интеграция данных GPS с геопространственными данными.

4. Тестирование приложения на эмуляторах и реальном устройстве: проведение комплексного тестирования приложения на различных устройствах и в различных условиях, выявление и исправление ошибок, оптимизация производительности.

5. Отладка и оптимизация приложения: устранение выявленных ошибок и недочетов, оптимизация работы приложения для обеспечения максимальной производительности и стабильности.

Особое внимание при разработке уделено вопросам безопасности и конфиденциальности данных, а также удобству использования приложения в полевых условиях. Приложение построено на основе архитектурного паттерна Model-View-ViewModel (MVVM), что обеспечивает высокую степень модульности и расширяемости, открывая возможности для дальнейшего развития и адаптации под конкретные задачи пользователей.

Для выполнения текущего проекта планируется использование библиотеки ArcGIS Runtime SDK for Android и Android Studio как основной среды разработки.

В дальнейшем планируется расширение функционала приложения, включая:

- Добавление функций поиска по содержимому карт: реализация возможности поиска объектов и информации на карте по ключевым словам.
- Форматирование данных: предоставление пользователям инструментов для форматирования и структурирования геоданных.
- Закрепление карт в списке: возможность сохранения часто используемых карт в отдельном списке для быстрого доступа.
- Скрытие данных в раздел “скрытые”: реализация функции скрытия определенных данных от отображения на карте для удобства работы с большим объемом информации.
- Работа с .mdb: интеграция возможности работы с файлами Microsoft Access (.mdb), что позволит пользователям взаимодействовать с данными, хранящимися в этом формате.

В целом, Данный проект представляет собой важный шаг в области разработки мобильных приложений для работы с геоданными в контексте электроразведочного метода ЗСБ. Он направлен на удовлетворение конкретных

потребностей пользователей, повышение их продуктивности и эффективности работы с геопространственной информацией. Разработка приложения "GPS Tracker" позволит не только создать полезный инструмент для специалистов ЗСБ, но и получить ценный опыт в разработке мобильных приложений, который может быть использован в будущих проектах.



## **Глава 1. Обзор мобильных приложений для GPS навигации с использованием ArcGIS Runtime SDK for Android**

### **1.1 Особенности мобильного приложения для GPS навигации, его особенности**

Мобильные приложения для GPS навигации — это категория программного обеспечения, предназначенная для обеспечения навигации и работы с геопривязанными данными в цифровом формате. Они обеспечивают пользователям инструменты для отслеживания местоположения, планирования маршрутов и работы с геопривязанными данными, такими как карты и SHP-установки. Эти приложения часто используются в полевых исследованиях, где требуется точное позиционирование и работа с геоданными.

Основные характеристики мобильных приложений для GPS навигации включают:

1. Интуитивно понятный интерфейс: Приложения разработаны таким образом, чтобы минимизировать кривую обучения и позволить пользователям быстро начать работу.
2. Работа в оффлайн режиме: Большинство приложений предлагают возможность загрузки карт и геоданных для работы без доступа к интернету.
3. Интеграция с внешними устройствами: Многие приложения предоставляют возможность интеграции с внешними GPS-устройствами и другими датчиками.
4. Безопасность: Защита конфиденциальности и безопасность данных являются ключевыми аспектами, с функциями шифрования и парольной защиты.
5. Персонализация: Пользователи могут настраивать внешний вид и ощущение своих карт, выбирая темы, шрифты и цветовые схемы.
6. Расширенные функции: Некоторые приложения предлагают дополнительные инструменты, такие как запись треков, создание маршрутов и работа с SHP-установками.

## 1.2 Анализ существующих приложений для GPS навигации

В области электроразведки методом зондирования становлением поля (ЗСБ) эффективная работа с геопространственными данными играет ключевую роль. Мобильные приложения для GPS-навигации открывают новые возможности для специалистов ЗСБ, предоставляя инструменты для точного позиционирования, планирования работ и анализа геоданных непосредственно в полевых условиях.

Для анализа популярных приложений для GPS навигации, мы рассмотрим следующие аспекты:

1. Функциональность: Какие основные и дополнительные функции предлагает каждое приложение?
2. Пользовательский интерфейс: насколько удобен и понятен интерфейс приложения?
3. Производительность: как быстро работает приложение, и насколько оно стабильно?
4. Отзывы пользователей: какие плюсы и минусы отмечают пользователи в своих отзывах?

Этот анализ поможет нам лучше понять потребности пользователей и определить, какие функции должны быть включены в наше приложение.

Рассмотрим такие приложения как: “Maps.me”, “HERE WeGo” и “Mapbox”.

### **Maps.me [6] (Рис 1.1)**

- Разработчик: My.com B.V. Функциональность: Maps.me - это простое и удобное приложение для просмотра карт и навигации, особенно полезное при путешествиях и активном отдыхе. Оно предлагает следующие функции:

- Просмотр карт: Поддерживает оффлайн карты, основанные на данных OpenStreetMap, которые можно загрузить для использования без подключения к интернету. Карты содержат информацию о дорогах, зданиях, достопримечательностях и других объектах.

- Поиск мест: Позволяет искать места по названию или категории (рестораны, отели, магазины и др.).
- Планирование маршрутов: Позволяет создавать маршруты для пешеходов, велосипедистов и автомобилистов с учетом дорожных условий и пробок (при наличии интернет-соединения) .
- Запись треков: Фиксирует маршрут движения с привязкой к координатам и времени, позволяя сохранять и делиться своими путешествиями.
- Закладки: Позволяет сохранять любимые места и маршруты для быстрого доступа. Пользовательский интерфейс: Maps.me имеет простой и интуитивно понятный интерфейс, который легко освоить даже новичкам. Карта занимает большую часть экрана, а основные элементы управления расположены удобно и доступно.
- Производительность: Приложение работает стабильно и быстро, как в онлайн, так и в оффлайн режиме. Загрузка карт и поиск мест происходят без задержек.
- Отзывы пользователей: Maps.me получает положительные отзывы за свою простоту, удобство использования и надежность, особенно в оффлайн режиме.

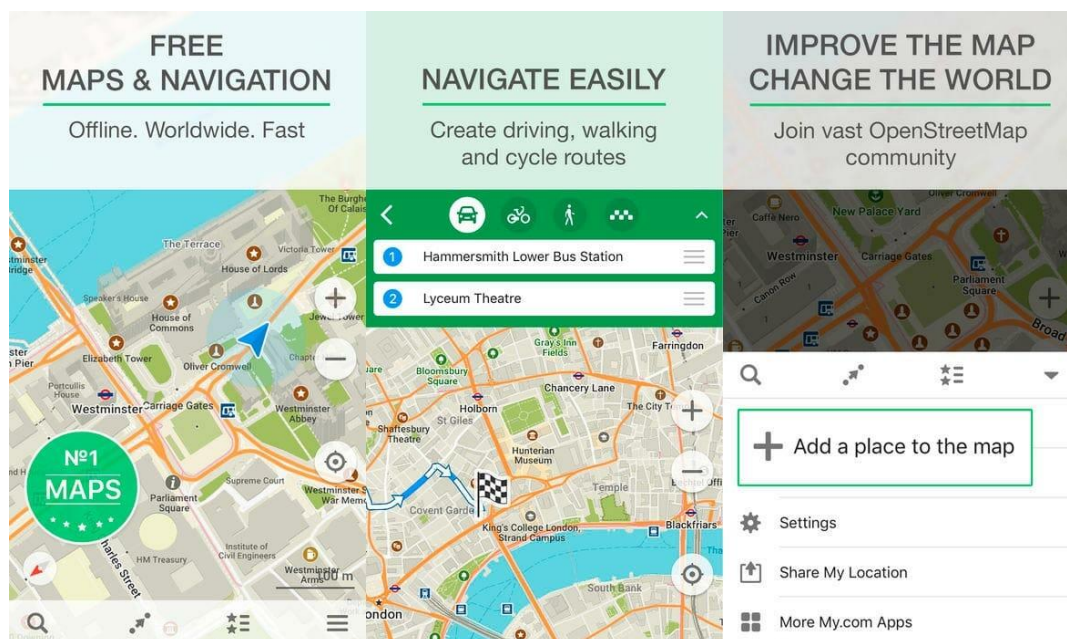


Рисунок 1.1 Интерфейс приложения Maps.Me

## **OsmAnd [7] (Рис 1.2)**

- Разработчик: OsmAnd Development Team Функциональность:

OsmAnd - это приложение с открытым исходным кодом, предлагающее широкий спектр функций для навигации и работы с картами:

- Просмотр карт: Поддерживает оффлайн карты OpenStreetMap и других провайдеров. Карты содержат подробную информацию о дорогах, зданиях, достопримечательностях, туристических маршрутах и других объектах.

- Навигация: Обеспечивает голосовую и визуальную навигацию для пешеходов, велосипедистов и автомобилистов.

- Поиск мест: Позволяет искать места по названию или категории.

- Запись треков: Фиксирует маршрут движения с привязкой к координатам и времени.

- Планирование маршрутов: Позволяет создавать маршруты с учетом различных параметров (тип транспорта, дорожные условия и др.).

- Информация о пробках: В некоторых регионах предоставляет информацию о пробках в режиме реального времени (при наличии интернет-соединения) .

- Пользовательский интерфейс: Интерфейс OsmAnd достаточно функциональный, но может показаться сложным для новичков из-за большого количества настроек и опций.

- Производительность: Приложение работает стабильно и быстро, как в онлайн, так и в оффлайн режиме.

- Отзывы пользователей: OsmAnd ценится за свой открытый исходный код, широкий функционал и возможность работы в оффлайн режиме.

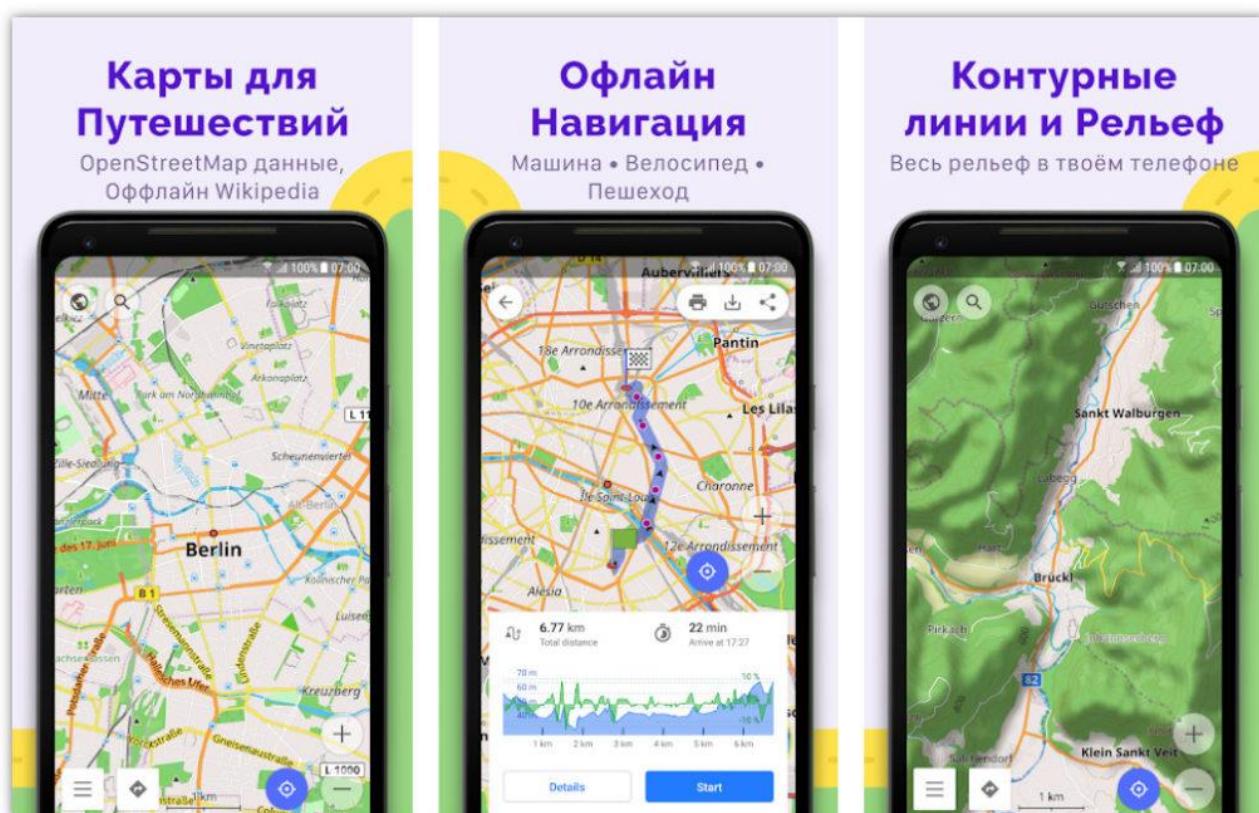


Рисунок 1.2 Интерфейс приложения OSMAnd

### GPS Navigation & Maps Sygic [8] (Рис 1.3)

- Разработчик: Sygic a.s. Функциональность: Sygic - это приложение для навигации, предлагающее хороший баланс функциональности и простоты использования:
- Просмотр карт: Поддерживает оффлайн карты TomTom и других провайдеров. Карты содержат подробную информацию о дорогах, зданиях, достопримечательностях и других объектах.
- Навигация: Обеспечивает голосовую и визуальную навигацию для пешеходов, велосипедистов и автомобилистов.
- Поиск мест: Позволяет искать места по названию или категории.
- Информация о пробках: Предоставляет информацию о пробках в режиме реального времени (при наличии интернет-соединения).
- Предупреждения о камерах контроля скорости: Информировает о приближении к камерам контроля скорости.



- **Head-up Display (HUD):** Проецирует навигационную информацию на лобовое стекло автомобиля (при наличии специального устройства).  
**Пользовательский интерфейс:** Sygic имеет современный и удобный интерфейс, который легко освоить.  
**Производительность:** Приложение работает стабильно и быстро, как в онлайн, так и в оффлайн режиме.  
**Отзывы пользователей:** Sygic получает положительные отзывы за свой удобный интерфейс, точную навигацию и полезные функции, такие как предупреждения о камерах контроля скорости.

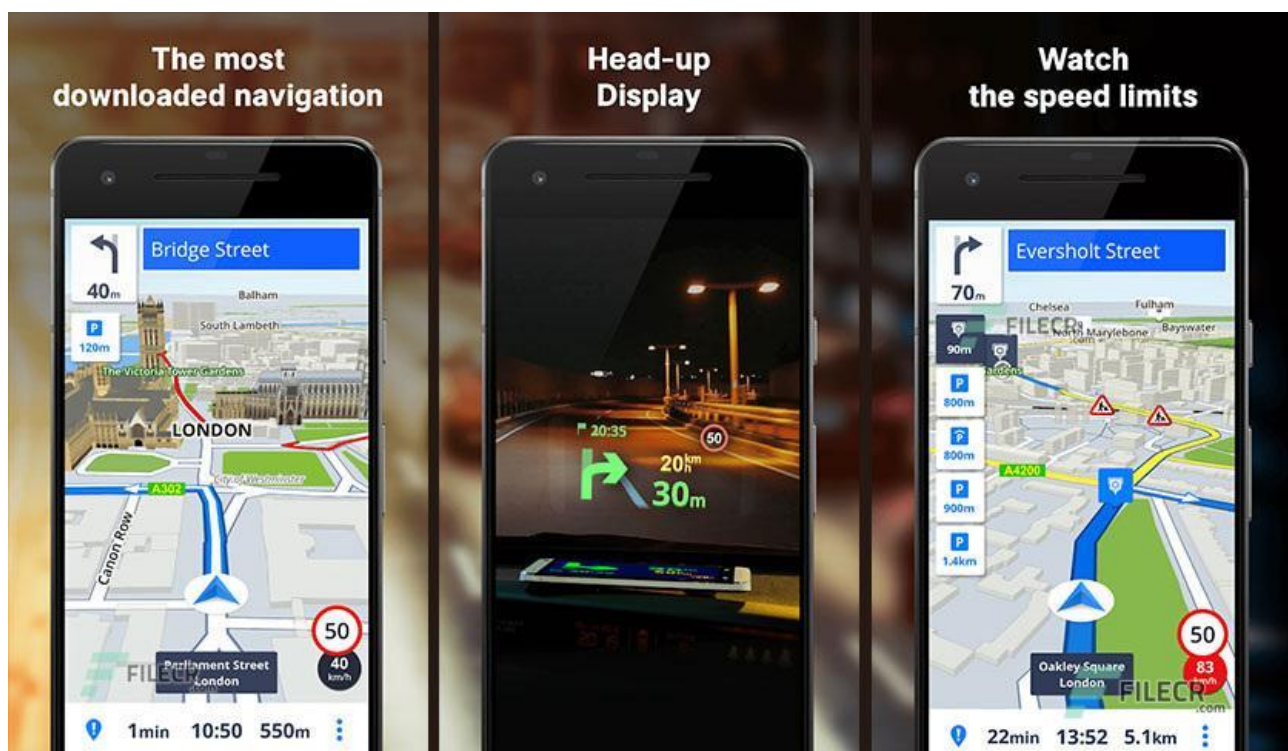


Рисунок 1.3 Интерфейс приложения GPS Navigation & Maps Sygic

**Таблица 1.1 Сравнение характеристик приложений**

	Maps.me	OsmAnd	Sygis
Разработчик	My.com B.V.	OsmAnd Development Team	Sygis a.s.
Просмотр карт	Оффлайн карты (OpenStreetMap)	Оффлайн карты (OpenStreetMap и др.)	Оффлайн карты (TomTom и др.)
Навигация	Пешеходная, велосипедная, автомобильная	Пешеходная, велосипедная, автомобильная	Пешеходная, велосипедная, автомобильная
Поиск мест	Да	Да	Да
Запись треков	Да	Да	Нет
Планирование маршрутов	Да	Да	Нет
Информация о пробках	При наличии интернет-соединения	В некоторых регионах	Да
Предупреждения о камерах	Нет	Нет	Да
Head-up Display (HUD)	Нет	Нет	Да
Работа с шейп-файлами (.shp)	Нет	С помощью плагинов	Нет
Работа с KML-файлами (.kml)	Нет	Да	Нет
Работа с растровыми данными (.tif)	Нет	С помощью плагинов	Нет
Расширенные инструменты анализа данных	Нет	Ограниченные	Нет

### **1.3 Определение требований к новому приложению**

На основе детального анализа существующих приложений для GPS навигации (Таблица 1.1) и отзывов пользователей, мы можем сделать вывод, что идеальное приложение для GPS навигации должно сочетать в себе простоту использования с глубокими функциональными возможностями. Пользователи ценят интуитивно понятный интерфейс, но также ожидают от приложения гибкости и мощности для работы с геоданными и картами. Следовательно, новое приложение должно учитывать как положительные стороны существующих решений, так и их недостатки, предлагая улучшенный пользовательский опыт.

Требования к новому приложению:

1. Удобство работы с геоданными: Приложение должно предоставлять легкие в использовании инструменты для загрузки и отображения карт, работы с SHP-установками и записи треков.
2. Работа в оффлайн режиме: Приложение должно предлагать возможность загрузки карт и геоданных для работы без доступа к интернету.
3. Безопасность: Защита конфиденциальности и безопасность данных являются ключевыми аспектами, с функциями шифрования и парольной защиты.
4. Персонализация: Пользователи должны иметь возможность настраивать внешний вид и ощущение своих карт, выбирая темы, шрифты и цветовые схемы.
5. Расширенные функции: Приложение должно предлагать дополнительные инструменты, такие:
  - Расчет расстояний и отклонений от заданных точек и линий.
  - Возможность создания и загрузки проектов, объединяющих загруженные файлы и настройки отображения.
  - Редактирование геометрии и атрибутов SHP-файлов.
  - Поддержка различных систем координат.



Эти требования отражают желание пользователей иметь универсальное приложение, которое будет одинаково удобно как для быстрой навигации, так и для сложной работы с геоданными. Новое приложение должно предложить баланс между простотой и функциональностью, учитывая современные тенденции и ожидания пользователей. Основываясь на этих требованиях, мы можем приступить к следующему этапу проекта — проектированию и разработке приложения.

## Глава 2. Разработка мобильного приложения “GPS Tracker”

### 2.1 Создание нового проекта в Android Studio

Первый этап создания нового проекта в Android Studio является ключевым, поскольку он определяет основу для всего процесса разработки. Этот этап начинается с запуска Android Studio и выбора опции “Создать новый проект Android Studio” (Рис. 2.1).

Здесь важно выбрать название проекта, которое будет отражать его функциональность и будет легко восприниматься пользователями. Например, для приложения “GPS” подходящим названием может быть “GPS Tracker”.

Затем необходимо определить минимальный уровень API Android, который будет поддерживать приложение. Это важно для обеспечения совместимости приложения с большим количеством устройств пользователей. Рекомендуется выбирать версию, которая охватывает как минимум 90% устройств на рынке. (Рис. 2.2)

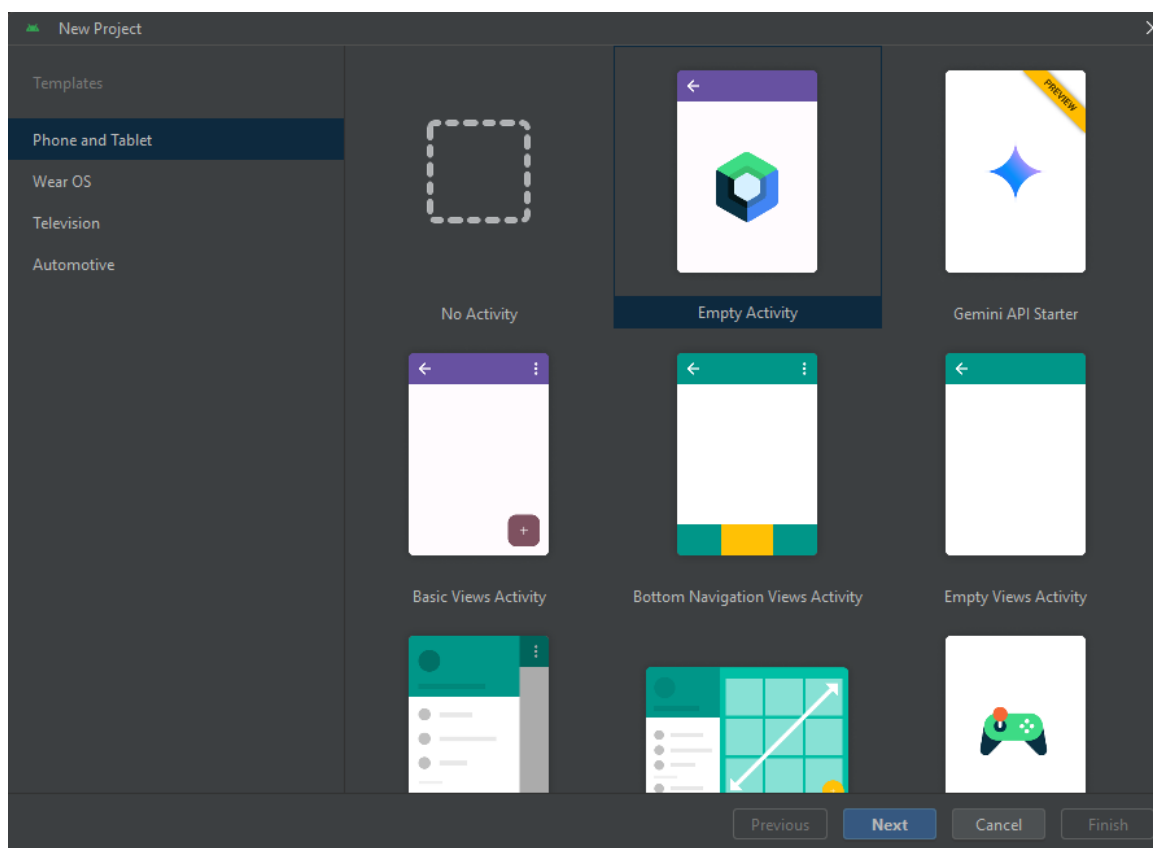


Рисунок 2.1 Окно создания нового проекта

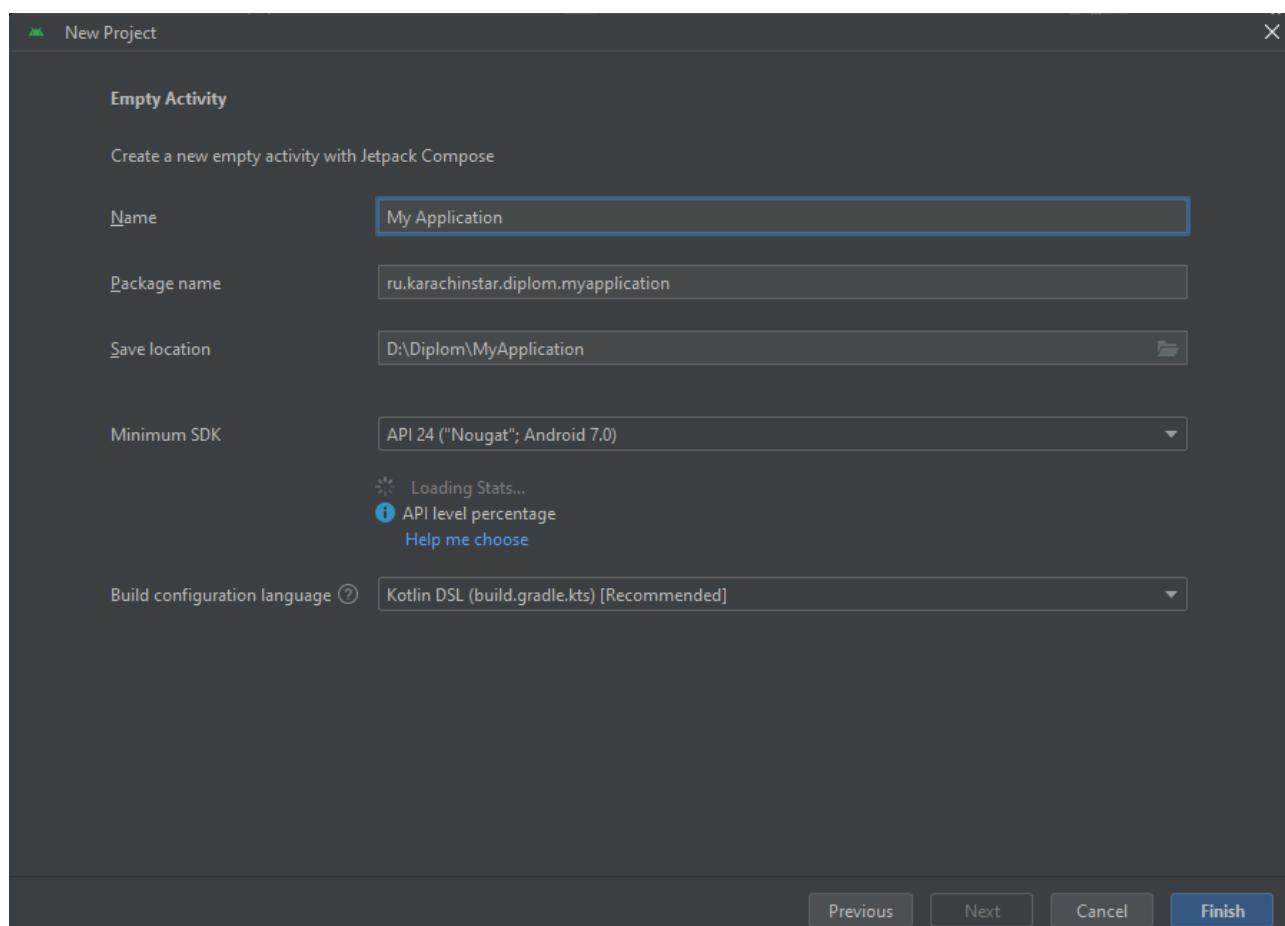


Рисунок 2.2 Окно выбора параметров нового проекта

## 2.2 Разработка основного функционала приложения

Разработка основного функционала приложения “GPS” включает в себя несколько ключевых шагов: разработка пользовательского интерфейса, программирование основных операций с GPS и файлами

### 2.2.1 Разработка пользовательского интерфейса

В начале моего пути к созданию приложения “GPS Tracker”, я сосредоточился на разработке пользовательского интерфейса, который был бы не только функциональным, но и удобным в использовании. Пользовательский интерфейс приложения "GPS Tracker" представляет собой минималистичный экран, основную часть которого занимает карта (MapView). Это позволяет

пользователю сосредоточиться на визуализации геопространственных данных и своего местоположения

Главный экран приложения.

На главном экране моего приложения я решил разместить такие элементы, как: MapView для отображения карты, TextView для отображения расстояния и отклонения, и кнопки для загрузки файлов, записи треков и перевода карты к текущему местоположению.

**MapView:** Я добавил MapView в основной экран, чтобы пользователи могли легко просматривать карту. Этот элемент был размещен так, чтобы занимать всю доступную площадь экрана, обеспечивая максимально возможное пространство для отображения карты (см. Рисунок 2.3).

**TextView:** Я добавил два элемента TextView в верхнюю часть экрана для отображения расстояния и отклонения. Эти элементы были размещены так, чтобы быть легко видимыми, но не мешать просмотру карты (см. Рисунок 2.3).

**Button:** Я добавил три кнопки в нижнюю часть экрана, чтобы пользователи могли легко загружать файлы, записывать треки и отыскивать свое местоположение. Эти кнопки были размещены так, чтобы быть легко доступными, но не мешать просмотру карты (см. Рисунок 2.3).



Рисунок 2.1 Размещение элементов пользовательского интерфейса на главном экране.

По нажатию левой кнопки запускается файловый менеджер для загрузки локальных геопривязанных данных (рис. 2.2). Его вызов реализован через код.

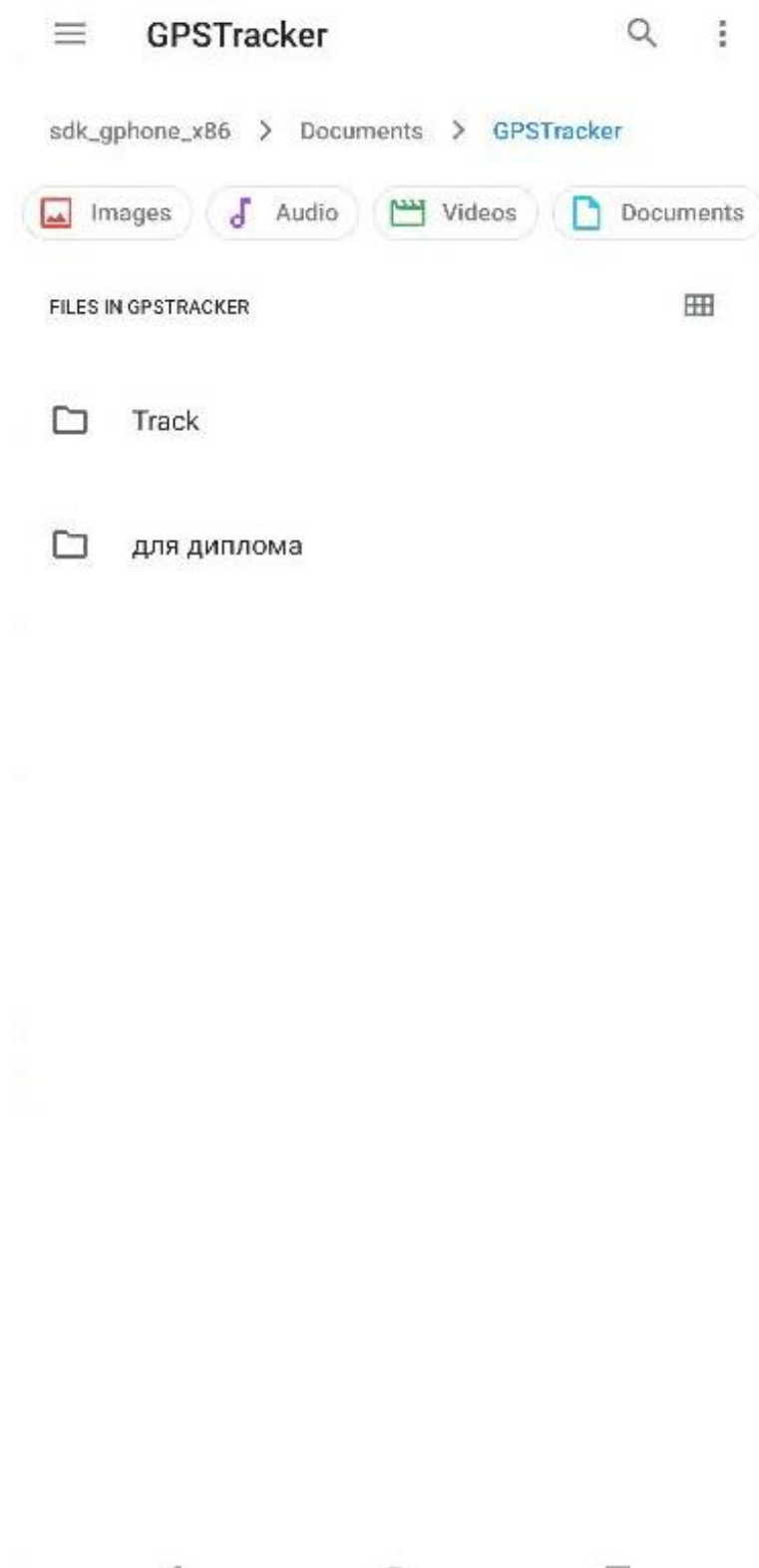


Рисунок 2.2 Вызов файлового менеджера для загрузки файлов на карту.

По нажатию на объект, загруженный на карте запускается окно с информацией об атрибутах объекта, а также там присутствует кнопка запуска следования до объекта по геодезическому пути (SHOW DISTANCE). Реализация данного окна производится через код приложения

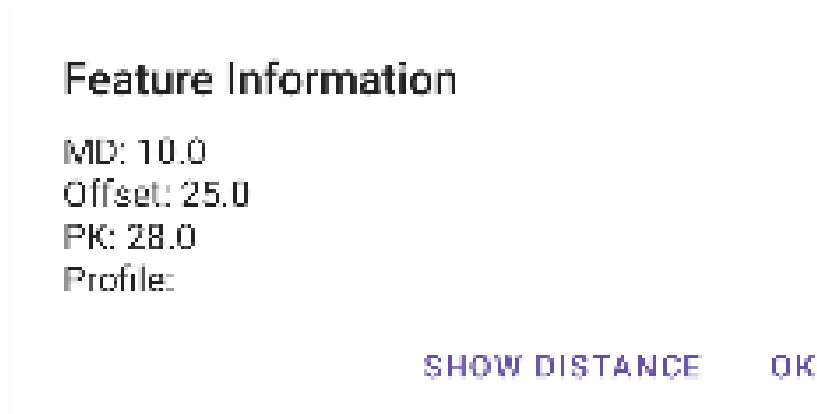


Рисунок 2.3 Вызов окна с информацией об атрибутах объекта.

### ***2.2.2 Программирование основных операций с GPS и файлами***

Этот шаг является сердцем функциональности приложения.

На начальном этапе была спроектирована архитектура приложения, включающая классы для работы с картой, загрузки данных и взаимодействия с пользователем. Оно построено на основе архитектурного паттерна Model-View-ViewModel (MVVM), что обеспечивает четкое разделение ответственности и улучшает тестируемость кода.

В основе любого приложения, работающего с геолокацией и файловой системой, лежит корректное получение разрешений.

**Класс PermissionsViewModel** (Фрагмент кода 2.4).

Он инкапсулирует логику запроса и обработки разрешений, гарантируя, что приложение получит доступ только к необходимым ресурсам.

```

class PermissionsViewModel : ViewModel() {
    // ..
    fun requestPermissions(activity: Activity,
requestPermissionLauncher:
ActivityResultLauncher<Array<String>>,
manageStorageRequestLauncher:
ActivityResultLauncher<Intent>) {
        // ... (Проверка и запрос разрешений)
    }

    fun handlePermissionsResult(permissions: Map<String,
Boolean>) {
        // ... (Обработка результатов запроса разрешений)
    } fun checkAllPermissionsGranted(activity: Activity):
Boolean {
        // ... (Проверка предоставления всех необходимых
разрешений)
    }
}

```

#### Фрагмент кода 2.4 PermissionsViewModel

В этом классе присутствуют следующие функции:

- requestPermissions(activity: Activity, requestPermissionLauncher: ActivityResultLauncher<Array<String>>, manageStorageRequestLauncher: ActivityResultLauncher<Intent>):
  - activity: Текущая активность, из которой запрашиваются разрешения.
  - requestPermissionLauncher: Объект ActivityResultLauncher для запуска запроса разрешений на доступ к местоположению и другим функциям.



- `manageStorageRequestLauncher`: Объект `ActivityResultLauncher` для запуска запроса разрешения на управление всеми файлами (для Android 11 и выше).
- Функционал: Проверяет, какие разрешения еще не предоставлены, и запрашивает их у пользователя.
- `handlePermissionsResult(permissions: Map<String, Boolean>):`
  - `permissions`: Карта, содержащая информацию о предоставленных разрешениях (название разрешения -> статус предоставления).
  - Функционал: Обрабатывает результаты запроса разрешений, обновляя состояние `_permissionsGranted`.
- `checkAllPermissionsGranted(activity: Activity):`
  - `activity`: Текущая активность.
  - Функционал: Проверяет, предоставлены ли все необходимые разрешения.

Важно отметить: Приложение запрашивает разрешения на доступ к точному и приблизительному местоположению, а также разрешения, необходимые для работы в фоновом режиме (начиная с Android 12). Это позволяет записывать треки даже когда приложение не активно, обеспечивая непрерывность и полноту данных.

### Класс `MainActivity` (Фрагмент кода 2.5).

Он инициализирует `PermissionsViewModel` и запускает процесс запроса разрешений в методе `onCreate`.

```
class MainActivity : AppCompatActivity() {
    // ...

    override fun onCreate(savedInstanceState: Bundle?) {
        // ...
    }
}
```

```

        // Запрашиваем разрешения
        viewModel.requestPermissions(this,
requestPermissionLauncher, manageStorageRequestLauncher)

        // ... (наблюдение за статусом разрешений и
запуск основного функционала)
    }
}

```

Фрагмент кода 2.5 MainActivity

### Класс MyApplication (Фрагмент кода 2.5).

Расширяет класс Application и служит для инициализации глобальных объектов и настройки приложения при его запуске.

```

class MyApplication : Application() {
    lateinit var map: ArcGISMap

    override fun onCreate() {
        super.onCreate()

        val connectivityManager =
getSystemService(Context.CONNECTIVITY_SERVICE) as
ConnectivityManager

        val activeNetworkInfo =
connectivityManager.activeNetworkInfo

        val spatialReference =
SpatialReference.create(28418)
    }
}

```

```

        map = if (activeNetworkInfo != null &&
activeNetworkInfo.isConnected) {
            // Если есть интернет, устанавливаем первую
карту
            ArcGISMap(BasemapStyle.OSM_STANDARD)
        } else {
            // Если нет интернета, устанавливаем вторую
карту
            ArcGISMap(spatialReference)
        }

        AppWatcher.config =
AppWatcher.config.copy(watchActivities = true,
watchFragments = true)
    }
}

```

Фрагмент кода 2.5 MyApplication

- В методе onCreate() происходит создание объекта ArcGISMap в зависимости от наличия интернет-соединения.
  - Если интернет доступен, используется базовая карта OSM\_STANDARD.
  - Если интернет недоступен, создается карта без подложки с использованием локальной системы координат SpatialReference.create(28418).

После того, как все необходимые разрешения получены и проверено наличие интернета на устройстве, приложение запускает основной фрагмент FragmentMain, открывая перед пользователем карту и возможности GPS-трекера.

## Класс FragmentMain.

Фрагмент FragmentMain — это сердце вашего приложения, предоставляющее пользователю интуитивно понятный интерфейс для взаимодействия с картой и функционалом GPS-трекера.

*Загрузка файлов*  (Фрагмент кода 2.6):

```
binding.loadFileButton.setOnClickListener {
    openFile()
}

private fun handleFile(uri: Uri) {
    // ... (определение формата файла и загрузка)
}
```

Фрагмент кода 2.6 Загрузка файлов

Пользователь может загружать файлы различных форматов (GeoTIFF, shapefiles, KML), обогащая карту дополнительными данными и визуализируя их в удобном виде. **Важное замечание:** на текущем этапе развития приложения, во избежание ошибок программы, необходимо подгружать файлы с геопривязкой СК WGS84(EPGS:4326).

*Запись трека пути*  (Фрагмент кода 2.7):

```
binding.buttonRecordTrack.setOnClickListener {
    trackRecorderViewModel.startStopRecording()
}
```

Фрагмент кода 2.7 Запись трека пути

С помощью простого нажатия кнопки пользователь может начать или остановить запись трека, сохраняя ценные данные о своем перемещении. Данные

сохраняются в формате KML и располагаются в папке /sdcard/Documents/GPSTracker/Track смартфона.



**Отображение местоположения** (Фрагмент кода 2.8):

```
binding.buttonGPS.setOnClickListener {  
    locationDisplay.autoPanMode =  
    LocationDisplay.AutoPanMode.COMPASS_NAVIGATION  
}
```

Фрагмент кода 2.8 Отображение местоположения

Пользователь может активировать автоматическое слежение за своим местоположением на карте, всегда оставаясь в центре внимания.

**Взаимодействие с картой** (Фрагмент кода 2.8):

```
binding.mapView.onTouchListener = object :  
    DefaultMapViewOnTouchListener(requireContext(),  
binding.mapView) {  
    // ... (обработка касаний на карте)  
}
```

Фрагмент кода 2.8 Запись трека пути

Отображает данные о объекте, загруженного из SHP файла в всплывающем окне (рис 2.3)

### **Класс DataRepository.**

Класс DataRepository — это надежный хранитель данных, предоставляющий доступ к ним и выполняющий операции, связанные с загрузкой файлов, записью треков и расчетом геодезических данных.

Загрузка файлов (Фрагмент кода 2.9):

```
fun loadShapefile(uri: Uri, graphicsOverlay:
GraphicsOverlay): LiveData<FeatureLayer> {
    // ... (загрузка shapefile и отображение на карте)
}
fun loadGeoTiff(uri: Uri): LiveData<RasterLayer> {
    // ... (загрузка GeoTIFF и отображение на карте)
}
fun loadKMLfile(uri: Uri): LiveData<KmlLayer> {
    // ... (загрузка KML и отображение на карте)
}
```

Фрагмент кода 2.9 Загрузка файлов

DataRepository обеспечивает загрузку файлов различных форматов, делая карту более информативной и полезной для пользователя.

Запись треков (Фрагмент кода 2.10):

```
fun startRecording() {
    // ... (инициализация записи трека)
}

fun writeLocation(longitude: Double, latitude: Double) {
    // ... (запись данных о местоположении в файл KML)
}

fun finishRecording() {
    // ... (завершение записи трека)
}
```

Фрагмент кода 2.10 Запись треков

DataRepository заботится о записи треков, сохраняя данные о местоположении в файл KML, который в дальнейшем можно использовать для анализа и визуализации. Структуру KML файла я создавал сам. Пример кода создания в приложении (приложение 3)

Расчет геодезических данных (Фрагмент кода 2.11):

```
fun drawLineAndTrackDistance(currentPoint: Point,
targetPoint: Point): Graphic {
    // ... (отрисовка линии и расчет расстояния)
}

fun calculateDistance(currentPoint: Point, targetPoint:
Point) {
    // ... (расчет расстояния)
}

fun calculateDeviation(currentPoint: Point, polyline:
Polyline) {
    // ... (расчет отклонения)
}
```

Фрагмент кода 2.11 Расчет геодезических данных

DataRepository предоставляет функции для расчета расстояния и отклонения, позволяя пользователю получать точную информацию о своем положении относительно выбранных объектов на карте.

Мозговой центр: GeodeticPathViewModel, MapViewModel, TrackRecorderViewModel

Классы `GeodeticPathViewModel`, `MapViewModel` и `TrackRecorderViewModel` — это мозговой центр вашего приложения, отвечающий за обработку данных и взаимодействие с `DataRepository`.

`GeodeticPathViewModel` (Фрагмент кода 2.12):

```
class GeodeticPathViewModel(application: Application) :  
    AndroidViewModel(application) {  
    // ... (расчет и отображение расстояния и отклонения)  
}
```

Фрагмент кода 2.12 `GeodeticPathViewModel`

`GeodeticPathViewModel` обеспечивает расчет и отображение расстояния и отклонения от выбранного объекта на карте, предоставляя пользователю ценную информацию для анализа.

`MapViewModel` (Фрагмент кода 2.12):

```
class MapViewModel(application: Application) :  
    AndroidViewModel(application) {  
    // ... (управление загрузкой и отображением слоев на  
    карте)  
}
```

Фрагмент кода 2.12 `MapViewModel`

`MapViewModel` управляет загрузкой и отображением слоев на карте, делая ее более информативной и удобной для пользователя.



TrackRecorderViewModel (Фрагмент кода 2.13):

```
class TrackRecorderViewModel(application: Application) :  
    AndroidViewModel(application) {  
    // ... (управление записью треков)  
}
```

Фрагмент кода 2.13 TrackRecorderViewModel

TrackRecorderViewModel отвечает за управление записью треков, обеспечивая сохранение данных о местоположении пользователя.

### ***2.2.3 Библиотеки и технологии***

Разработка приложения, работающего с геолокацией, картами и файлами, требует использования специализированных инструментов. В данном проекте были использованы следующие библиотеки:

#### **1. ArcGIS Runtime SDK for Android:**

Описание: Мощная библиотека от Esri, предоставляющая широкий набор инструментов для работы с картами, геоданными и пространственным анализом. Вы используете версию 100.15.5.

Функционал:

Отображение интерактивных карт с различными базовыми картами и слоями.

Загрузка и отображение различных форматов геоданных (shapefiles, GeoTIFF, KML).

Работа с геометрическими объектами (точки, линии, полигоны).

Пространственный анализ и геообработка.

Отображение местоположения пользователя и трекинг.

Преимущества:

Богатый функционал для работы с геоданными.

Высокая производительность и оптимизация для мобильных устройств.

Поддержка различных форматов данных.

Активное сообщество и документация.

## 2. Android Jetpack:

Описание: Набор библиотек от Google, упрощающих разработку Android-приложений и улучшающих их качество.

Компоненты, используемые в проекте:

`androidx.core.ktx`: Предоставляет Kotlin-расширения для основных компонентов Android, делая код более лаконичным и выразительным.

`androidx.appcompat`: Обеспечивает обратную совместимость с более старыми версиями Android, предоставляя современные компоненты UI.

`com.google.android.material`: Библиотека Material Design, предоставляющая широкий набор готовых компонентов UI, соответствующих современным гайдлайнам.

`androidx.activity`: Упрощает работу с Activity и их жизненным циклом.

`androidx.constraintlayout`: Позволяет создавать гибкие и адаптивные макеты UI.

Преимущества:

Улучшение структуры и maintainability кода.

Упрощение работы с жизненным циклом компонентов.

Повышение производительности и стабильности приложения.

## 3. Другие библиотеки:

`org.osgeo.proj4j`: Библиотека для работы с картографическими проекциями, необходимая для корректного отображения геоданных на карте.

`javax.xml.stream.stax-api` и `org.codehaus.woodstox:stax2-api`: Библиотеки для работы с XML, вероятно, используемые для парсинга KML-файлов.

Заключение:

Использование этих библиотек позволило создать мощное и функциональное приложение для работы с GPS, картами и файлами, обеспечивая высокую производительность, удобство использования и расширяемость.

Карта становится интерактивной площадкой для пользователя. Касания позволяют получать информацию об объектах, рассчитывать расстояние и отклонение, открывая новые возможности для анализа и исследования.

### **2.3. Тестирование GPS Tracker на эмуляторах и реальном устройстве**

Чтобы убедиться, что GPS Tracker готов к любым вызовам реального мира, приложение прошло суровое испытание на прочность, включающее тестирование на эмуляторах и реальном устройстве.

#### **Виртуальные горизонты: Эмуляторы**

Приложение было запущено на эмуляторах с широким диапазоном версий Android API (от 28 до 34), имитирующих различные конфигурации устройств. Это позволило выявить и устранить ошибки, связанные с запуском и работой приложения на разных версиях Android (Рис 2.14).

В частности, были обнаружены и исправлены проблемы с отображением карты на устройствах с Android 10, а также оптимизирована работа с разрешениями на устройствах с Android 11 и выше.

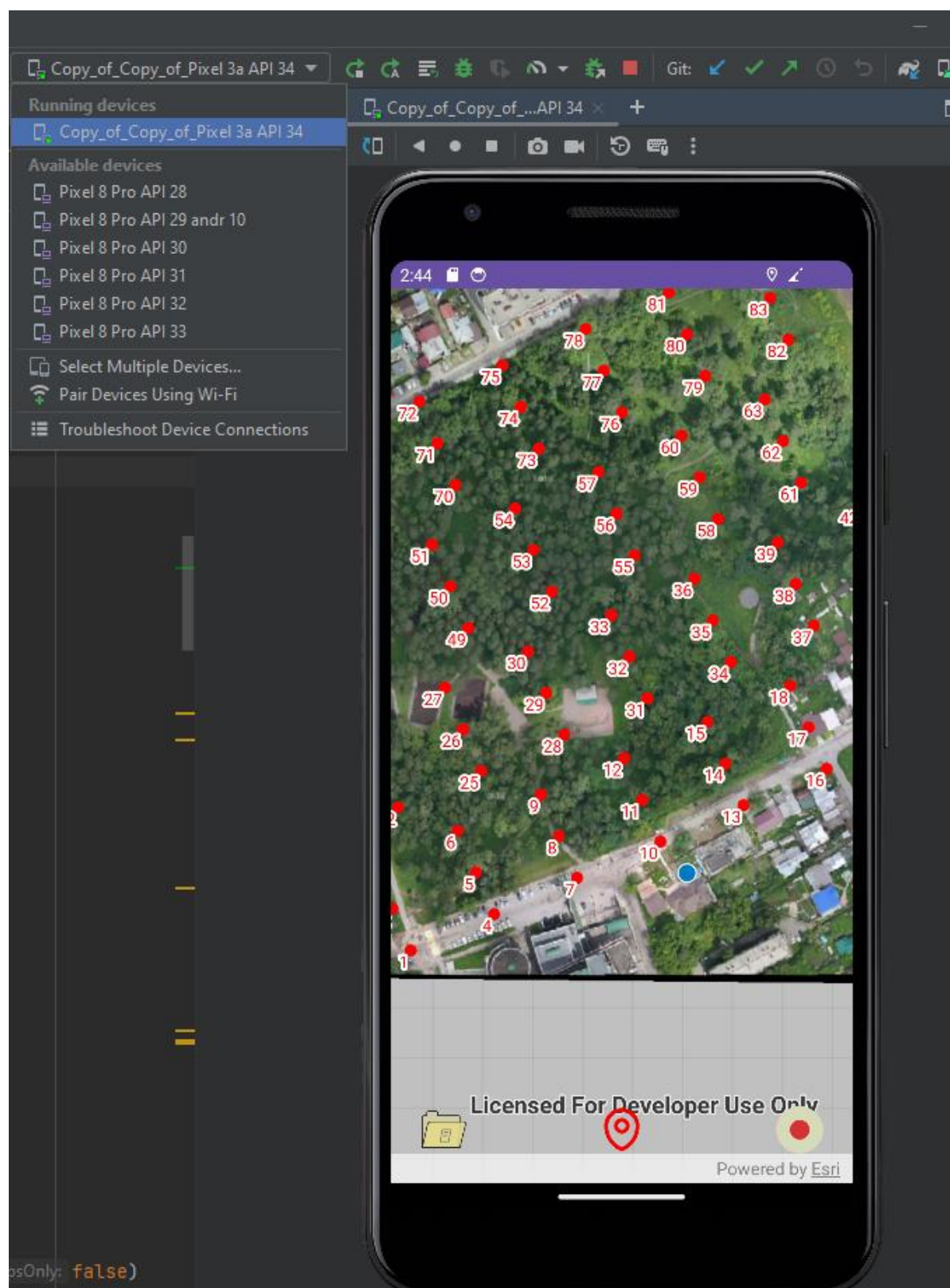


Рисунок 2.14 Тестирование на разных версиях эмулятора

## Реальный мир: Huawei P30 Pro

Для проверки производительности и отзывчивости в реальных условиях GPS Tracker был установлен на флагманский смартфон Huawei P30 Pro с Android 10. Приложение продемонстрировало стабильную работу, быстрое определение местоположения и плавное отображение карты. (рис. 2.15)



Рисунок 2.15 Тестирование на реальном устройстве



## Сравнение со специализированным устройством Garmin GPSMAP 66s

Точность записи треков — ключевой аспект любого GPS-трекера. Для оценки этого параметра треки, записанные GPS Tracker, были сравнены с треками, полученными с помощью профессионального устройства Garmin GPSMAP 66s. (рис. 2.16)



Рисунок 2.16 Сравнение приложение «GPS Tracker» с Garmin GPSmap 66s

Результаты сравнения показали высокую степень соответствия, подтверждая надежность и точность записи треков вашим приложением (Приложение 1).

## **Итоги испытаний**

Тестирование на эмуляторах и реальном устройстве позволило убедиться в стабильности, производительности и точности GPS Tracker. Приложение готово к использованию в полевых условиях и способно стать надежным помощником для пользователей, работающих с геоданными.

## **Глава 3. Анализ результатов и предложения по улучшению**

### **3.1. Обзор достигнутых результатов**

Разработка GPS Tracker успешно достигла своей основной цели — предоставления пользователям удобного инструмента для записи и визуализации треков, а также работы с геопространственными данными. Приложение обеспечивает интуитивно понятный интерфейс для загрузки и отображения различных форматов файлов (shapefiles, GeoTIFF, KML), записи треков и расчета геодезических данных. Архитектура MVVM обеспечивает четкое разделение ответственности и улучшает тестируемость кода, закладывая прочный фундамент для дальнейшего развития.

### **3.2. Предложения по улучшению**

Анализ результатов и обратная связь от пользователей (если таковая имеется) выявили ряд потенциальных улучшений, которые могут сделать GPS Tracker еще более мощным и универсальным инструментом:

#### **Поддержка баз данных .mdb:**

Интеграция возможности загрузки и отображения данных из баз данных Microsoft Access (.mdb), широко используемых в ЗСБ для хранения координат установок (точек и полигонов). Это позволит пользователям визуализировать целые спроектированные сети наблюдений одним нажатием кнопки.

#### **Создание и загрузка проектов:**

Реализация функционала для создания проектов, объединяющих загруженные файлы и настройки отображения. Пользователи смогут сохранять проекты и загружать их в любой момент, что значительно ускорит и упростит работу с повторяющимися задачами.

#### **Редактирование shp и kml файлов:**

Исследование возможности редактирования геометрии и атрибутов загруженных shapefiles и KML файлов непосредственно в приложении. Это позволит пользователям вносить корректировки и дополнения в данные без необходимости использования стороннего ПО.



### **Выпадающий список загруженных слоев:**

Добавление выпадающего списка, отображающего все загруженные слои, с возможностью изменения их порядка (перемещение на передний или задний план) и удаления.

### **Редактирование стилей слоев:**

Внедрение возможности редактирования стилей отображения для каждого слоя (цвет, размер, подписи). Для shapefiles можно добавить выбор атрибута для отображения в подписях, а также настройку стиля подписей.

### **Поддержка различных систем координат:**

Расширение функционала загрузки файлов (tif, shp, kml) для поддержки различных систем координат, помимо WGS84. Это обеспечит совместимость с данными, полученными из различных источников.

## **3.3. Взгляд в будущее**

Реализация предложенных улучшений превратит GPS Tracker в мощный и универсальный инструмент для работы с геоданными, способный удовлетворить потребности широкого круга пользователей, от специалистов ЗСБ до любителей активного отдыха.

## **Заключение**

В ходе исследования и разработки приложения GPS Tracker была достигнута главная цель — создание функционального и удобного инструмента для работы с геоданными. Приложение успешно справляется с задачами записи и визуализации треков, загрузки и отображения различных форматов геопространственных данных, а также расчета геодезических параметров. Несмотря на некоторые ограничения, связанные с временными и ресурсными рамками, GPS Tracker уже сейчас представляет собой ценный инструмент для широкого круга пользователей. Результаты исследования имеют практическую значимость, открывая новые возможности для анализа и визуализации геоданных в различных областях, от землеустройства и строительства до туризма и активного отдыха. Однако, путешествие GPS Tracker только начинается. В будущем планируется реализация предложенных улучшений, таких как поддержка баз данных .mdb, создание проектов, редактирование файлов и расширенные возможности настройки стилей отображения. Эти нововведения сделают приложение еще более мощным и универсальным, способным удовлетворить потребности самых требовательных пользователей. Достижение поставленных целей и успешное выполнение задач подтверждают успешность проекта и открывают путь для его дальнейшего развития, обещая новые захватывающие приключения в мире геоданных.

### Список использованной литературы

1. Книга: Дон Гриффитс, Дэвид Гриффитс. Программирование для Android на Kotlin 3-е издание. Бостон: O'REILLY, 2023
2. Книга: Жемеров Дмитрий, Исакова Светлана. Kotlin в действии. Москва: ДМК издательство, 2018
3. Интернет ресурс - <https://developer.android.com/> «Основной источник информации по разработке Android-приложений, включающий руководства, справочники и примеры кода.»
4. Интернет ресурс - <https://gb.ru/> "Образовательная платформа GeekBrains"
5. Интернет ресурс - <https://developers.arcgis.com/android/> «Документация ArcGIS Runtime SDK for Android»
6. Maps.me URL:  
<https://play.google.com/store/apps/details?id=com.mapswithme.maps.pro&hl=ru&pli=1>
7. OsmAnd URL:  
<https://play.google.com/store/apps/details?id=net.osmand&hl=ru>
8. GPS Navigation & Maps Sygic. URL:  
<https://play.google.com/store/apps/details?id=com.sygic.aura&hl=ru>

## Приложение 1

Сравнение результатов записи трека м/у Garmin GPSMAP 66s и мобильным приложение «GPS Tracker»





## Приложение 2.

### Схема установок ЗСБ в ГИС системе QGIS

Скриншот интерфейса QGIS, демонстрирующий схему установок ЗСБ (Зона Сбора Близости) в ГИС системе. Интерфейс включает панель слоев (Слой), панель стилей (Стиль слоя), панель инструментов (Инструменты анализа) и панель сообщений (Отладочные сообщения).

В центре экрана отображена карта с границами территории, обозначенной красными точками (ЗСБ). Территория ограничена желтой линией. На карте отмечены населенные пункты: Жигалино, Чернорыло, Чеботаревка, Калиния, Коммуна, Плодопитомник, Ташла, Трудовое, Луговой. Также отмечены расстояния между пунктами: 4,8 км, 6,9 км, 9 км, 12,9 км, 5,4 км, 3,9 км, 3,2 км, 2,9 км, 5,5 км, 3,9 км, 4,9 км, 12,7 км, 7,2 км, 2,9 км.

Панель слоев (Слой) содержит следующие элементы:

- Финальное проектирование
  - ЗСБ
    - PKLine\_1800x1800 121 ф.н.
    - PK\_1800x1800 121 ф.н. [8367]
    - MD\_1800x1800 121 ф.н.
    - Площадь потерь ЗСБ
    - Правила по зп Вариант2
      - 1
      - 2
      - 20
      - 3
    - Профили по ГП
    - мЗСБ
      - PKLine\_100x100 шаг 100x400 горизонт
      - PK\_100x100 шаг 100x400 горизонт**
      - MD\_100x100 шаг 100x400 горизонт
      - Area
      - Площадь потерь мЗСБ
      - Полышии.мзсб

Панель стилей (Стиль слоя) показывает стиль слоя: PK\_100x100 шаг 100x400 горизонт. Включены опции: Простая символика, Маркер, Автообновление, Применить.

Панель инструментов (Инструменты анализа) включает: Поиск в QMS, Поиск геосервиса..., Фильтр по охвату, Все, Недавно использованные: Google Satellite Hybrid, TMS, Скачать геоданные для проекта.

Панель сообщений (Отладочные сообщения) отображает сообщение: 2024-06-28T21:18:00 INFO Задача выполнена : Загрузка «U:\Share\Внутренняя документация\Отдел ОП/ГИС\Висячие объекты\ЗСБ на оренбурге\Оренбург ЭСА.qgz».

Статусная панель (bottom) отображает: Искать (Ctrl+K), Координаты: 5745541, 9597706, Масштаб: 1:142954, Увеличение: 100%, Поворот: 0,0°, Отрисовка, EPSG:28409.

## Приложение 3

### Код генерации KML файла

```
fun startRecording() {
    _isRecording.value = true
    val sdf = SimpleDateFormat("dd.MM.yyyy-HH_mm_ss", Locale.getDefault())
    if (context != null) {
        val currentDate = sdf.format(Date())
        val fileName = "$currentDate Track.kml"
        val resolver = context.contentResolver
        val contentValues = ContentValues().apply {
            put(MediaStore.MediaColumns.DISPLAY_NAME, fileName)
            put(MediaStore.MediaColumns.MIME_TYPE, "application/vnd.google-earth.kml+xml")
            val documentsDir =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOCUMENTS).path
                put(MediaStore.MediaColumns.DATA,
"$documentsDir/GPSTracker/Track/$fileName")
        }
        uri = resolver.insert(MediaStore.Files.getContentUri("external"),
contentValues)
        outputStream = resolver.openOutputStream(uri!!)
        xmlSerializer = Xml.newSerializer()
        xmlSerializer?.setOutput(outputStream, "UTF-8")
        xmlSerializer?.startDocument("UTF-8", true)
        xmlSerializer?.text("\n")
        xmlSerializer?.startTag(null, "kml")
        xmlSerializer?.attribute(null, "xmlns",
"http://www.opengis.net/kml/2.2")
        xmlSerializer?.text("\n")
        xmlSerializer?.startTag(null, "Document")
        xmlSerializer?.attribute(null, "id", "root_doc")
        xmlSerializer?.text("\n\t")
        xmlSerializer?.startTag(null, "Folder")
        xmlSerializer?.startTag(null, "name")
        xmlSerializer?.text("Track")
        xmlSerializer?.endTag(null, "name")
        xmlSerializer?.text("\n\t")
        xmlSerializer?.startTag(null, "Placemark")
        xmlSerializer?.text("\n\t\t")
        xmlSerializer?.startTag(null, "Style")
        xmlSerializer?.text("\n\t\t\t")
        xmlSerializer?.startTag(null, "LineStyle")
        xmlSerializer?.text("\n\t\t\t\t")
        xmlSerializer?.startTag(null, "color")
        xmlSerializer?.text("FF000000") // Черный цвет
        xmlSerializer?.endTag(null, "color")
        xmlSerializer?.text("\n\t\t\t\t\t")
        xmlSerializer?.startTag(null, "width")
        xmlSerializer?.text("5.66928") // Толщина линии
        xmlSerializer?.endTag(null, "width")
        xmlSerializer?.text("\n\t\t\t\t\t")
        xmlSerializer?.endTag(null, "LineStyle")
        xmlSerializer?.text("\n\t\t\t\t\t")
        xmlSerializer?.startTag(null, "PolyStyle")
        xmlSerializer?.startTag(null, "fill")
        xmlSerializer?.text("0")
        xmlSerializer?.endTag(null, "fill")
        xmlSerializer?.text("\n\t\t\t\t\t")
        xmlSerializer?.endTag(null, "PolyStyle")
    }
```

```

        xmlSerializer?.text("\n\t\t")
        xmlSerializer?.endTag(null, "Style")
        xmlSerializer?.text("\n\t\t")
        xmlSerializer?.startTag(null, "LineString")
        xmlSerializer?.text("\n\t\t\t")
        xmlSerializer?.startTag(null, "coordinates")
    }
}

fun writeLocation(longitude: Double, latitude: Double) {
    xmlSerializer?.text("$longitude,$latitude,0 ")
}

fun finishRecording() {
    xmlSerializer?.endTag(null, "coordinates")
    xmlSerializer?.text("\n\t\t")
    xmlSerializer?.endTag(null, "LineString")
    xmlSerializer?.text("\n\t")
    xmlSerializer?.endTag(null, "Placemark")
    xmlSerializer?.text("\n")
    xmlSerializer?.endTag(null, "Folder")
    xmlSerializer?.text("\n")
    xmlSerializer?.endTag(null, "Document")
    xmlSerializer?.text("\n")
    xmlSerializer?.endTag(null, "kml")
    xmlSerializer?.endDocument()
    outputStream?.close()
    _isRecording.value = false
}

```