



**Государственное бюджетное общеобразовательное учреждение города
Москвы «Школа № 1501»**

**Создание прототипа приложения «Weacloth» для подбора
одежды под текущие погодные условия**

**Участник:
ученик 10 «П» класса ГБОУ Школы
№1501 Никоноров Илья Анатольевич**

**Руководители:
педагог ГБОУ Школа № 1501
Мусалова Лилия Эгамовна**

Оглавление

1. Введение.....	3
2. Теоретическая часть.....	4
2.1 Изучение основ разработки Android-приложений.....	4
Среда разработки Android Studio для проекта "Weacloth".....	5
2.2 Разработка теоретического алгоритма.....	6
2.3 Планировка приложения.....	6
3. Практическая часть.....	7
3.1 Реализация базы данных и алгоритма подбора.....	7
3.2 Разработка пользовательского интерфейса.....	7
3.3 Тестирование приложения и исправление ошибок.....	7
4. Готовый продукт.....	9
5. Заключение.....	11
6. Список информационных источников.....	12

Введение

Современный ритм жизни, стремительное развитие модных тенденций и постоянная нехватка времени зачастую приводят к тому, что люди выбирают одежду, ориентируясь скорее на эстетику, чем на практичность. Это может становиться причиной дискомфорта в течение дня – от перегрева в жаркую погоду до переохлаждения в мороз. Особенно остро эта проблема стоит для людей с ограниченными возможностями здоровья (ОВЗ), которым сложно оперативно реагировать на изменения погоды или подбирать подходящие вещи из-за физических или когнитивных особенностей. Данный проект призван решить эту проблему, предлагая автоматизированный инструмент для выбора одежды на основе актуальной погоды. Особое внимание уделяется удобству использования: приложение может быть полезно не только обычным пользователям, но и людям с ограниченными возможностями, для которых выбор одежды без лишних усилий особенно важен, за счёт адаптированного интерфейса и упрощённой навигации. Основной целью работы является создание Android-приложения, способного анализировать погодные данные и предлагать оптимальный комплект одежды из гардероба пользователя. Для достижения этой цели были поставлены следующие задачи:

- Изучение основ программирования на языке Kotlin, принципов объектно-ориентированного подхода, работы с XML и Jetpack Compose для создания пользовательских интерфейсов
- Освоение инструментов навигации, включая NavHost и организацию переходов между экранами
- Разработка структуры данных для хранения информации о гардеробе пользователя и построение базы данных с использованием библиотеки Room
- Изучение принципов подбора одежды в зависимости от температуры, осадков, ветра и времени суток для создания интеллектуального алгоритма рекомендаций
- Реализация пользовательского интерфейса с акцентом на доступность и простоту использования
- Интеграция с внешними сервисами для получения прогноза погоды в реальном времени

На этапе реализации особое внимание уделялось проектированию структуры базы данных, разработке алгоритма подбора одежды с учётом сезонности и погодных условий, а также обеспечению корректной работы всех функций. После завершения разработки были проведены тестирования в реальных погодных условиях: в жару, дождь, снег и мороз. В процессе выявленные ошибки были исправлены, а алгоритмы оптимизированы для повышения качества рекомендаций. Таким образом, данный проект представляет собой полезный инструмент, помогающий экономить время, обеспечивать комфорт при выборе одежды, а также способствовать повышению качества жизни людей, в том числе инвалидов, за счёт снижения необходимости в самостоятельном принятии решений в сложных погодных ситуациях

Теоретическая часть

2.1 Изучение основ разработки Android-приложений

На первом этапе были изучены базовые технологии и инструменты, необходимые для разработки Android-приложения:

- Язык программирования **Kotlin**, как основной язык разработки Android-приложений
- Концепции **объектно-ориентированного программирования (ООП)** для построения модульной архитектуры
- **XML-разметка** для создания пользовательских интерфейсов
- **Jetpack Compose** как современная технология для декларативного создания UI
- Использование **NavController** и **Navigation Component** для построения удобной навигации между экранами
- Работа с аппаратными ресурсами устройства, такими как галерея изображений (Photo Picker API)
- Принципы работы с сетью и REST API, в частности, с использованием библиотеки **Retrofit** для получения прогноза погоды

Особое внимание уделялось построению чистой архитектуры приложения с разделением ответственности между слоями.

The screenshot shows the Android Studio interface with the following details:

- Project Files:** Shows the project structure under `C:\Users\Kuki\AndroidStudioProjects\Weacloth`. It includes `build`, `libs`, `src`, `androidTest`, and `main` directories. `main` contains `java`, `res`, and `layout` sub-directories with various files like `AddClosing.kt`, `MainScreen.kt`, etc.
- Code Editor:** Displays the `AddClosing.kt` file. The code defines a `Fragment` named `AddClosing` with fields for spinners, checkboxes, and buttons, along with logic for handling image selection via a launcher.
- Device Manager:** Shows a device configuration for "Medium Phone API 35".
- Terminal:** Shows a Windows PowerShell window with the command `PS C:\Users\Kuki\AndroidStudioProjects\Weacloth>`.
- Bottom Status Bar:** Includes icons for Weacloth, app, src, main, java, com, kuki, AddClosing, pickImageLauncher, 45:55, LF, UTF-8, 4 spaces, and a refresh icon.

Среда разработки Android Studio для проекта "Weacloth"

2.2 Разработка теоретического алгоритма

Для реализации функционала подбора одежды был разработан алгоритм, учитывающий следующие факторы:

- Температуру окружающей среды
- Вероятность и наличие осадков
- Скорость ветра
- Сезонность (весна, лето, осень, зима)
- Материал и назначение одежды

Для каждого типа одежды (базовый слой, средний слой, внешний слой, обувь, аксессуары) были разработаны отдельные критерии подбора в зависимости от погодных условий. Например, для базового слоя учитывалась только температура, в то время как для внешнего слоя также анализировались осадки и сила ветра.

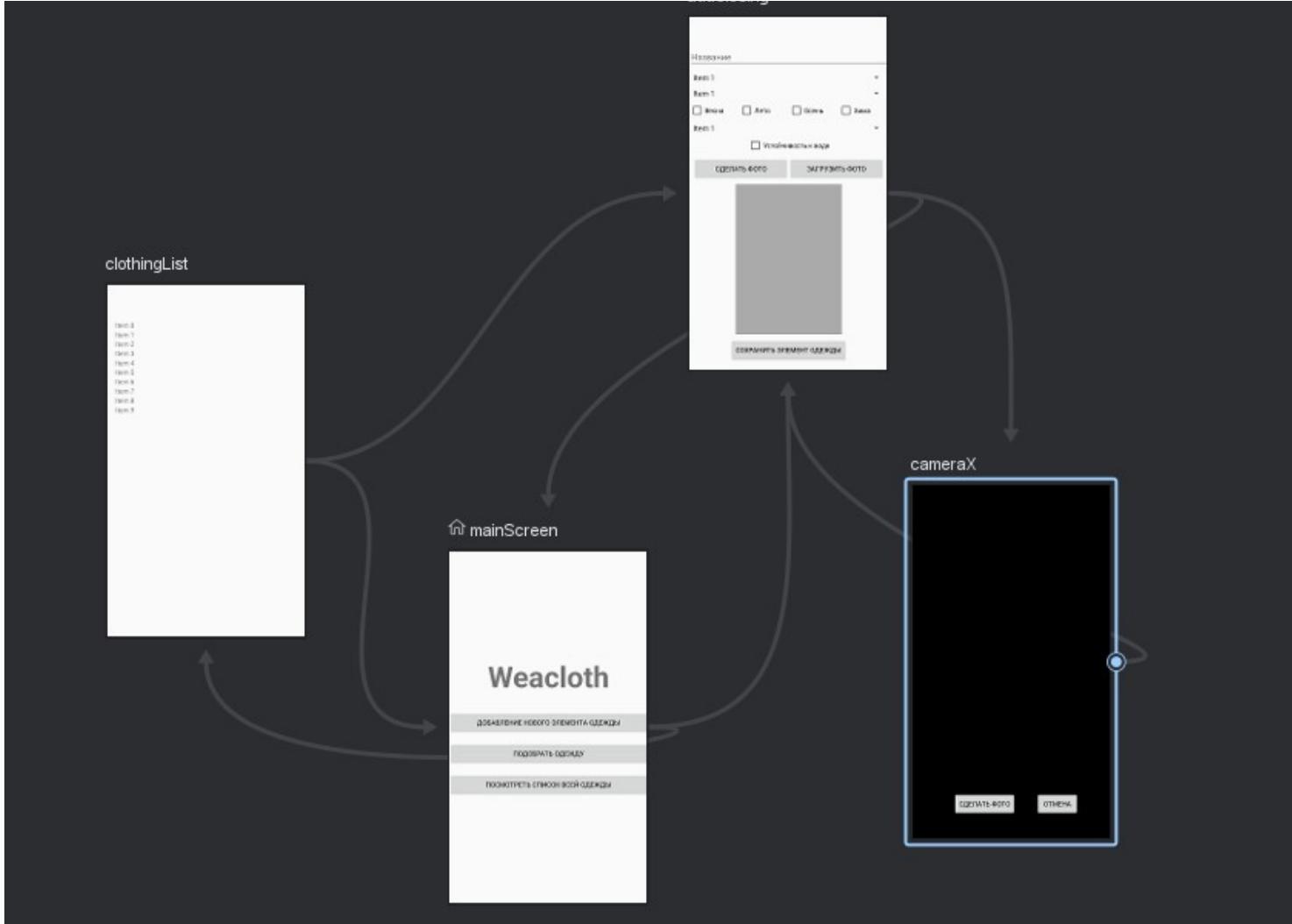
2.3 Планировка приложения

На этапе планировки были сформулированы основные требования к приложению:

- Возможность добавления элементов гардероба с указанием их характеристик (категория, подкатегория, материал, сезоны)
- Сохранение изображений одежды
- Получение актуальной информации о погоде
- Генерация рекомендуемого комплекта одежды
- Просмотр списка всей одежды
- Удобная навигация между экранами

Была определена структура базы данных и основные экраны пользовательского интерфейса:

- Экран добавления одежды
- Экран списка всей одежды
- Экран подбора одежды
- Главное меню приложения



Навигационная структура приложения

Практическая часть

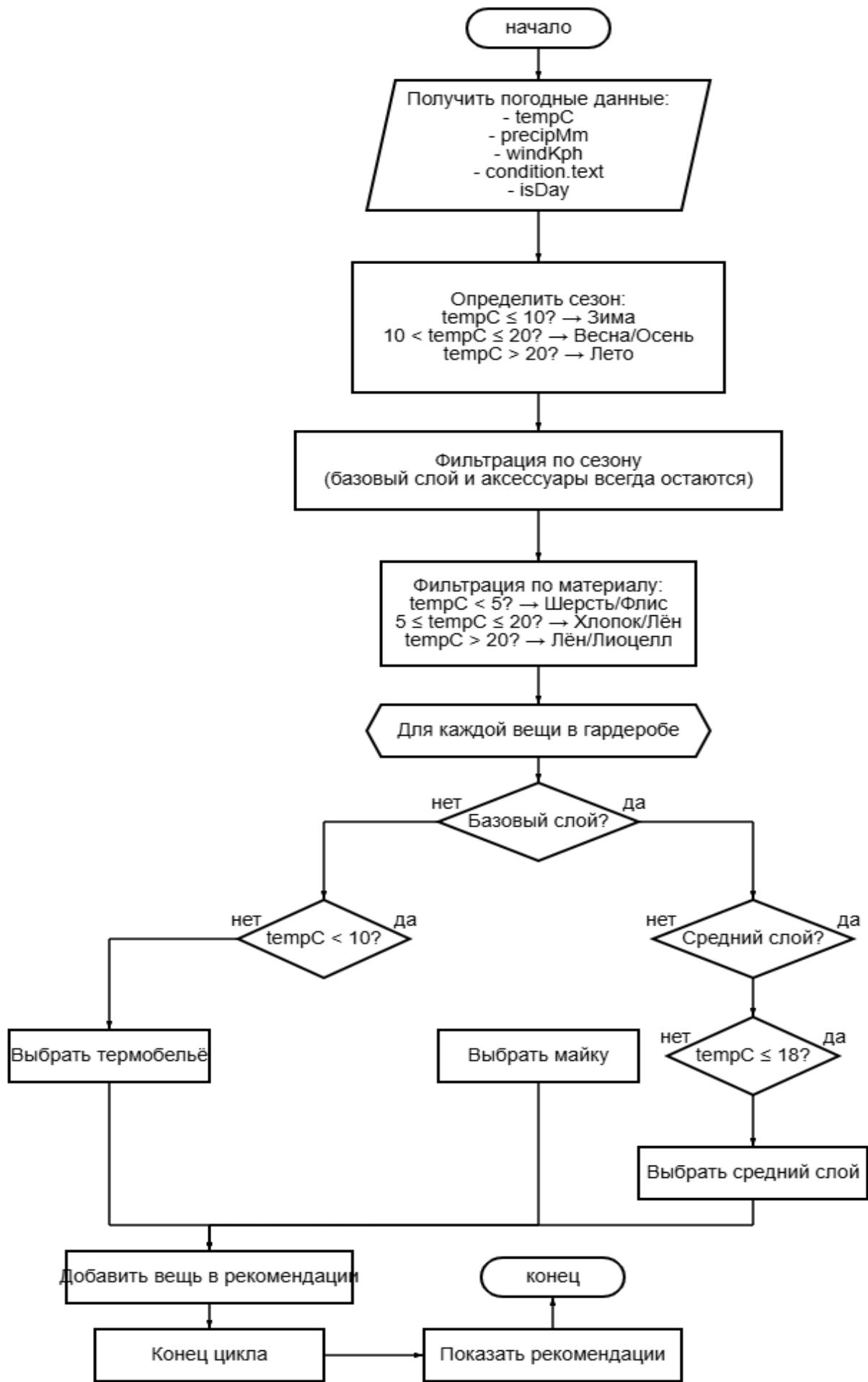
3.1 Реализация базы данных и алгоритма подбора

База данных была реализована с использованием библиотеки **Room**, обеспечивающей удобную работу с локальными данными. Каждая единица одежды сохранялась как отдельный элемент с полями: название, категория(пример: «Базовый слой»), подкатегория(пример: «Майка»), материал, сезоны и ссылка на изображение.

Алгоритм подбора одежды был реализован на основе анализа прогноза погоды, получаемого через API WeatherAPI. В процессе работы алгоритм фильтрует элементы гардероба, оставляя только те, которые соответствуют текущим условиям.

Основные этапы алгоритма:

1. Фильтрация одежды по сезону
2. Фильтрация по материалу
3. Выбор одежды по категориям
4. Учет специфических условий (дождь, ветер, солнце)



Алгоритм подбора одежды в виде блок-схемы

3.2 Разработка пользовательского интерфейса

Пользовательский интерфейс был создан с использованием стандартной XML-разметки для Android в сочетании с элементами Jetpack Compose. Основные принципы разработки UI:

- Простота и минимализм
- Четкая навигация между экранами
- Чтение и доступность текста
- Адаптивность под различные размеры экранов
- Использование встроенных тем для поддержки контрастного режима

Для организации навигации между экранами использовался `NavController` и граф переходов `NavHostFragment`. Действие кнопки "Назад" было синхронизировано с поведением стрелки в тулбаре для удобства пользователя. Всего приложение состоит из 3 окон:

- **Главное меню** — представляет собой окно с тремя кнопками: «Добавление нового элемента одежды», «Подобрать одежду» и «Посмотреть список всей одежды»
- **Окно добавления нового элемента одежды** — окно для добавления нового элемента одежды, вводится имя(пример: «Моя любимая куртка»), затем из выпадающих списков выбирается категория и подкатегория одежды(выпадающие списки использованы в целях упрощения пользования приложением), после чего отмечаются сезоны, для которых предназначен элемент одежды, следующим шагом является выбор материала, из которого сделана одежда(если их несколько, выбирается самый преобладающий в процентном соотношении) и, наконец, прикрепляется фотография одежды(эта функция по большей части рассчитана на людей с ОВЗ(поэтому для облегчения задачи были сделаны две отдельные кнопки), однако также может быть полезна и для обычных пользователей)
- **Список одежды** — окно, в котором отображается вся(или только отфильтрованная — в зависимости от выбранной кнопки в главном меню), в этом окне можно увидеть фотографию и дополнительную информацию о каждом отдельно взятом элементе одежды, помимо этого можно также удалить элемент одежды

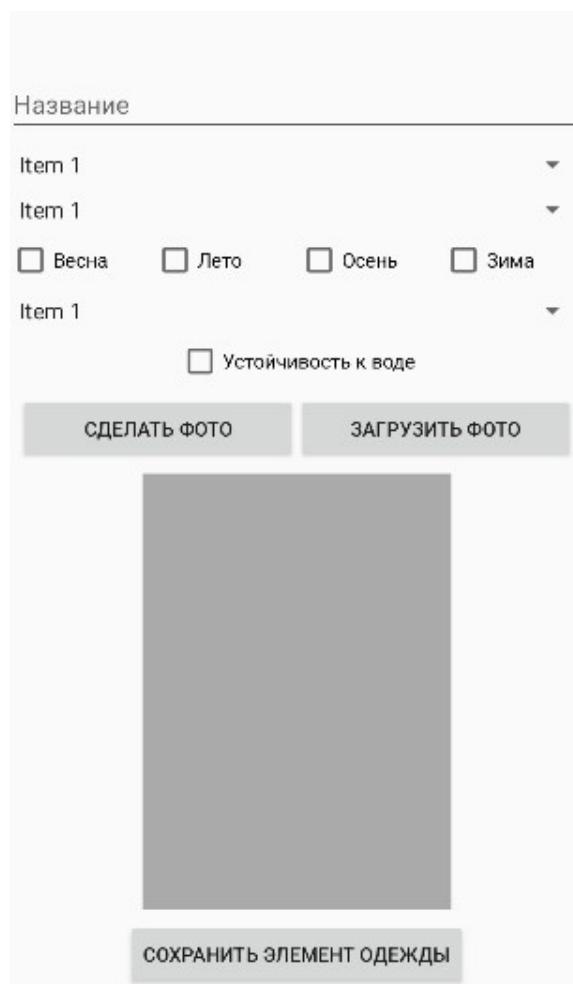
3.3 Тестирование приложения и исправление ошибок

На этапе тестирования были проверены:

- **Корректность сохранения и отображения элементов гардероба** — в начале разработки было трудно определиться со структурой данных, в следствии чего она постоянно менялась, что повлекло за собой различные

проблемы с сохранением, что создавало необходимость менять и интерфейс

- **Работа алгоритма подбора в различных погодных условиях**(дождь, жара, мороз, сильный ветер) — непосредственное тестирование приложения в реальных условиях, в первую очередь сомнению придавались значения температуры в алгоритмах и скорость работы приложения при наличии большой базы данных одеждой
- **Обработка ошибок при потере соединения с интернетом**
- **Работа с изображениями и безопасность доступа к URI файлов** — в связи с особенностями операционной системы Android любое приложение должно запрашивать у пользователя доступ к внешним модулям(камера, память и т.д.)
- **Ошибки, связанные с безопасностью доступа к изображениям из Photo Picker API и корректной обработкой URI** — в связи с допущенными в коде ошибками программа не могла корректно обрабатывать фотографии, после их исправления ближе к концу разработки выяснилось, что есть также проблемы с обработкой фотографий, которые выбирает пользователь при нажатии на кнопку «Выбрать фото»



Экран добавления элемента одежды в гардероб

Готовый продукт

На выходе был получен полностью работоспособный прототип Android-приложения, обладающий следующими возможностями:

- Добавление одежды в гардероб с фотографией;
- Автоматический подбор одежды на основе текущей погоды;
- Возможность просмотра списка всей сохранённой одежды;
- Удобная навигация и минималистичный интерфейс;
- Высокая стабильность работы приложения.

Приложение протестировано — показало высокую точность подбора в реальных погодных условиях.



Главное меню приложения

Заключение

В ходе выполнения проекта были достигнуты поставленные цели:

- Изучены основы разработки мобильных приложений для Android;
- Разработан эффективный алгоритм подбора одежды с учётом погодных условий;
- Реализован удобный и доступный интерфейс;
- Проведено всестороннее тестирование готового решения.

Созданное приложение представляет собой удобный инструмент для широкого круга пользователей. Особенno оно может быть полезно людям с инвалидностью, снижая нагрузку на принятие решений в повседневной жизни.

Дальнейшее развитие проекта может включать:

- Интеграцию системы уведомлений о смене погодных условий;
- Расширение базы данных одежды;
- Поддержку учёта личных предпочтений пользователя;
- Модуль подбора одежды для поездок в другие города.

Таким образом, проект полностью оправдал свои цели и задачи и имеет потенциал для дальнейшего развития.

Список информационных источников

1. Официальная документация Android Developers: developer.android.com
2. Официальная документация Kotlin: kotlinlang.org
3. Библиотека Jetpack Compose: developer.android.com/jetpack/compose
4. Официальная документация WeatherAPI: weatherapi.com
5. Документация библиотеки Room:
developer.android.com/topic/libraries/architecture/room
6. Статьи и примеры кода на StackOverflow
7. Документация по Retrofit: square.github.io/retrofit/
8. Курс "Основы Android-разработки" от Google Developers