Санкт-Петербургское государственное бюджетное

профессиональное образовательное учреждение

«Ижорский колледж»

КУРСОВАЯ РАБОТА

ТЕМА: «Разработка мобильного приложения «Экологические

советы»»

по ПМ.01 МДК.01.03 Разработка мобильных приложений

Выполнил

обучающийся группы 331с

Кондратьева Мария Алексеевна

Проверил

преподаватель спец. дисциплин

специальности 09.02.07

Информационные системы и программирование

Венедиктов Д.В.

Санкт-Петербург, 2025

ЗАДАНИЕ ДЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Цель данной курсовой работы заключается в разработке мобильного приложения, отвечающего современным требованиям пользователей и особенностям платформы. На первом этапе необходимо провести исследование целевой аудитории и определить её потребности, что послужит основой для дальнейшей разработки.

Важным аспектом является создание прототипа приложения, в котором будут учтены все основные функции и пользовательский интерфейс. Рекомендуется использовать методологии Agile или Scrum для гибкой разработки, что позволит вовремя вносить изменения по мере необходимости.

При реализации приложения следует учитывать принципы юзабилити и дизайна, чтобы обеспечить максимальный комфорт и удовлетворение пользователей. Неотъемлемой частью работы станет тестирование: как функциональное, так и пользовательское, что позволит выявить и исправить возможные ошибки.

Завершением курсовой работы станет представление итогового проекта, включающего в себя документацию и описание процесса разработки. Ожидается, что разработанное приложение будет иметь потенциал для дальнейшего улучшения и внедрения в реальную среду.

ПЛАН-ГРАФИК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

|  |  |  |
| --- | --- | --- |
| Дата | Содержание работ | Отметка о выполнении |
| 09.09 | Выбор темы курсовой работы |  |
| 16.09 | Знакомство с методическими рекомендациями по выполнению курсовой работы |  |
| 23.09 | Составление развёрнутого плана выполнения курсовой работы |  |
| 07.10 | Работа с нормативно-правовыми документами, учебной литературой |  |
| 21.10 | Анализ предметной области. Постановка задачи |  |
| 28.10 | Составление введения к курсовой работе в соответствии с методическими рекомендациями |  |
| 04.11 | Описание предметной области и функции решаемых задач |  |
| 11.11 | Выбор средств для выполнения курсовой работы. Выбор среды разработки |  |
| 25.11 | Разработка интерфейса (frontend-разработка) |  |
| 13.01 | Разработка логики работы приложения (backend-разработка) |  |
| 17.02 | Разработка подключаемой базы данных (backend -разработка) |  |
| 24.03 | Составление заключения к курсовой работе |  |
| 31.03 | Составление списка источников информации, используемых при выполнении курсовой работы |  |
| 07.04 | Подготовка курсовой работы к защите |  |

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 2](#_Toc180878817)

[ГЛАВА 1 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ РАЗРАБАТЫВАЕМОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ 2](#_Toc180878818)

[1.1 Анализ предметной области 2](#_Toc180878819)

[1.1.1 Исследование рынка существующих мобильных приложений похожей тематики 2](#_Toc180878820)

[1.1.2 Выявление требований к разрабатываемому мобильному приложению 2](#_Toc180878821)

[1.1.3 Анализ технологий для разработки мобильного приложения 2](#_Toc180878822)

[1.2 Постановка задачи курсовой работы 2](#_Toc180878823)

[1.2.1 Цели курсовой работы 2](#_Toc180878824)

[1.2.2 Задачи курсовой работы 2](#_Toc180878825)

[1.2.3 Оценка рисков разработки мобильного приложения 2](#_Toc180878826)

[1.2.4 Техническое задание на разработку мобильного приложения 2](#_Toc180878827)

[1.3 Описание предметной области и функции решаемых задач 2](#_Toc180878828)

[1.4 Выбор программных средств для разработки мобильного приложения 2](#_Toc180878829)

[1.5 Выбор среды для разработки мобильного приложения 2](#_Toc180878830)

[ГЛАВА 2 РЕАЛИЗАЦИЯ РАЗРАБАТЫВАЕМОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ 2](#_Toc180878831)

[2.1 Разработка интерфейса мобильного приложения (frontend-разработка) 2](#_Toc180878832)

[2.2 Разработка логики работы мобильного приложения (backend-разработка) 2](#_Toc180878833)

[2.3 Разработка подключаемой базы данных (backend-разработка) 2](#_Toc180878834)

[ЗАКЛЮЧЕНИЕ 2](#_Toc180878835)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 2](#_Toc180878836)

# ВВЕДЕНИЕ

В последние десятилетия мобильные технологии стали неотъемлемой частью нашей повседневной жизни. Смартфоны и планшеты являются основными инструментами для получения информации, общения и развлечения. В связи с этим, разработка мобильных приложений приобретает всё большее значение, как для бизнеса, так и для пользователей.

Мобильные приложения позволяют организациям улучшать взаимодействие с клиентами, оптимизировать процессы и повышать уровень сервиса. Пользователи, в свою очередь, ценят удобство, доступность и функциональность мобильных решений. Стремительное развитие технологий, таких как 5G, искусственный интеллект и облачные вычисления, открывает новые горизонты для создания инновационных приложений.

Цель данной курсовой работы – исследовать ключевые аспекты процесса разработки мобильных приложений, включая выбор платформы, языков программирования, а также принципов проектирования и тестирования. Также будет рассмотрен анализ существующих приложений и выделены основные тенденции, формирующие рынок мобильного ПО.

Актуальность выбранной темы обусловлена стремительным развитием мобильных технологий, а также растущей конкуренцией на рынке приложений. Успешная реализация мобильного приложения требует не только технических знаний, но и умения учитывать потребности целевой аудитории и трендов рынка.

# ГЛАВА 1 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ РАЗРАБАТЫВАЕМОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ

## 1.1 Анализ предметной области

Разработка мобильных приложений – это динамично развивающаяся область, охватывающая множество аспектов, включая проектирование, программирование и тестирование приложений для мобильных устройств. В этом разделе представим ключевые элементы анализа предметной области.

### 1.1.1 Исследование рынка существующих мобильных приложений похожей тематики

Изучение уже существующих мобильных приложений, которые решают схожие задачи, является важным шагом в процессе разработки. Это позволяет не только понять, как другие разработчики справляются с подобными вызовами, но и выявить сильные и слабые стороны конкурентов.

Таблица 1 – Анализ рынка существующих мобильных приложений похожей тематики

|  |  |  |  |
| --- | --- | --- | --- |
| Целевая аудитория | 30-60 лет | 20-40 лет | Для детей |
| Слабые стороны | Дизайн | Мало фото и интерактива | Дизайн |
| Сильные стороны | Много советов | Статья интересно оформлена, не перегружает информацией. | Легко и доступно объясняет детям, интерактив |
| Основные функции | Помочь, подсказать людям, как сохранить окружающую среду | Помогает сделать быт более осознанным и экологичным | Рассказывает и помогает детям понять что такое экология и для чего она нужна |
| Название приложения | Служба по контролю и надзору в сфере охраны окружающей среды, объектов живого мира и лесных отношений | Как сделать дом более экологичным: 7 принципов | Экомарафон |

### 1.1.2 Выявление требований к разрабатываемому мобильному приложению

Приложение про экологические советы нужно, чтобы помочь пользователям улучшить их экологическую осознанность и поведение.

Таблица 2 – Системные требования для разрабатываемого мобильного приложения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Операционная система | Процессор | Оперативная память | Разрешение экрана | Сетевые требования |
| Microsoft Windows |  |  | 1920\*1080 |  |

### 1.1.3 Анализ технологий для разработки мобильного приложения

Целевой платформой разрабатываемого мобильного приложения является операционная система Android не ниже версии 12.0.

Используемые языки программирования: Kotlin, Java.

Используемые фреймворки: Android SDK, Jetpack Compose, React Native, Ionic.

Используемые библиотеки: Dagger, Redux, Retrofit, Room, RxJava.

Инструменты для разработки баз данных: PostgreSQL.

## 1.2 Постановка задачи курсовой работы

Постановка задачи является важным шагом в процессе разработки мобильного приложения, так как она определяет цели, объем работы и основные требования к продукту.

### 1.2.1 Цели курсовой работы

### Приложение решает проблему недостатка информации и мотивации для ведения экологически ответственного образа жизни. Ценность приложения для пользователя заключается в доступности информации, у обращения процесса, мотивация и отслеживания прогресса, сообщество, удобства

### 1.2.2 Задачи курсовой работы

Возможность регистрации пользователя, интерактив с карточками советов, предоставление советов и рекомендаций.

### 1.2.3 Оценка рисков разработки мобильного приложения

Технические сложности, недостаток времени

Попробовать уделить больше времени техническим проблемам, попросить помощи или совета.

### 1.2.4 Техническое задание на разработку мобильного приложения

Техническое задание (ТЗ) является основанием для разработки мобильного приложения, определяет его основные требования и функции. Это приложение будет разрабатываться для платформы Android.

Создание мобильного приложения, которое предоставляет пользователю удобный доступ к экологическому миру, узнать все тонкости этой сферы и помочь себе и окружающим.

**Основные функции приложения:**

* регистрация и авторизация пользователей:
  + возможность регистрации через электронную почту, социальные сети;
  + восстановление пароля.
* личный кабинет:
  + отображение информации о пользователе: имя, контактные данные, история посещений.
* основные функции приложения
  + Регистрация
  + Личный кабинет
  + Возможность взаимодействовать с карточками
* возможность оставлять отзывы и оценки (при необходимости)
* система уведомлений (пуш-уведомления о событиях или об изменениях в приложении)

**Технические требования:**

* приложение должно работать на устройстве под управлением операционной системы Android (версии не ниже 12.0);
* время загрузки приложения не должно превышать 3 секунд;
* поддержка русского и английского языков на первом этапе разработки.

**Дизайн:**

* оформление должно быть современным, простым и интуитивно понятным для пользователя;
* элементы интерфейса должны быть адаптивными для различных размеров экранов.

Данное техническое задание описывает ключевые аспекты разработки мобильного приложения. Оно служит основой для взаимодействия между заказчиком и командой разработчиков.

## 1.3 Описание предметной области и функции решаемых задач

Разработка мобильных приложений охватывает создание программного обеспечения, которое предназначено для работы на мобильных устройствах, таких как смартфоны и планшеты. Эта область включает в себя проектирование, программирование, тестирование и поддержку программных приложений, которые могут использоваться в различных отраслях, включая, но не ограничиваясь, здравоохранением, образованием, финансами, электронной коммерцией и развлечениями.

Я выбрала разработку мобильного приложения, Разработка мобильного приложения «Экологические советы»

## 1.4 Выбор программных средств для разработки мобильного приложения

Основной целью разрабатываемого мобильного приложения является чтобы ознакомить людей с темой экологии.

В данной курсовой работе я использовала следующие программные средства для разработки мобильного приложения:

* язык программирования – Kotlin;
* инструменты для управления проектами – Git;
* тестирование приложения – фреймворк Espresso;
* создание документации к разрабатываемого мобильному приложению – язык разметки Markdown.

## 1.5 Выбор среды для разработки мобильного приложения

Средой разработки для мобильного приложения к курсовой работе я выбрала Android Studio.

# ГЛАВА 2 РЕАЛИЗАЦИЯ РАЗРАБАТЫВАЕМОГО МОБИЛЬНОГО ПРИЛОЖЕНИЯ

## 2.1 Разработка интерфейса мобильного приложения (frontend-разработка)

Ответ на вопрос: кто будет использовать ваше приложение?

Возраст от 17 до 40. Это может заинтересовать студентов, людей, которых привлекает тема экологии.

Аналогичные приложения выглядят следующим образом:



Рисунок 1 – главное окно приложения «GreenVoice»



Рисунок 2 – главное окно приложения «Jasjl Bolashaq Lite»



Рисунок 3 – главное окно приложения «Журнал «Экология производства»»

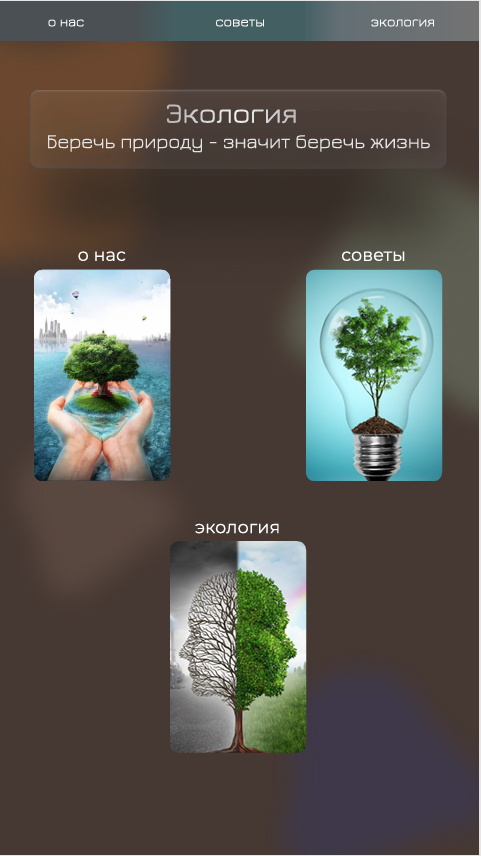


Рисунок 4 – главное окно разрабатываемого приложения «ECO»

Таблица 3 – Необходимые экраны разрабатываемого приложения

|  |  |
| --- | --- |
| Название экрана | Скриншот |
| Регистрация пользователя | 1 скриншот |
| Восстановление пароля | 1 скриншот |
| Главный экран приложения | 1 скриншот |
| Личный кабинет пользователя | 2 скриншота |
|  |
| Новости, Журнал «Экология производства» | 3 скриншота |
| Трекер, онлайн магазин и новости, «Jasjl Bolashaq Lite» |  |
| Эко-мероприятия, новости, фото, тектовые и голосовые заметки, карта, «GreenVoice» |  |
| Экран отзыва (или оценки) | 1 скриншот |
| Экран пуш-уведомления | 1 скриншот |

**Описание регистрации пользователя в приложении**

Чтобы зарегистрироваться в данном приложения необходимо в поле номера телефона - ввести ваш номер телефона, в поле имени - ввести ваше имя, в поле пароля — придумать пароль и в следующем окошке повторить его.

**Описание восстановления пароля пользователя в приложении**

Для того чтобы восстановить пароль, необходимо:

* шаг 1 – нажать на кнопку регистрации
* шаг 2 – заполнить заново свои поля

**Описание главного экрана приложения**

На главном экране находиться:

* кнопки передвижения
* картинки
* текст

**Описание Экология производства**

Открывает книгу про Экологию производства

**Описание News**

Показывает новости

**Описание Заметки**

Дает доступ к написанию заметок или голосовым заметкам.

В разрабатываемом мобильном приложении я планирую использовать преимущественно следующие цвета:

Таблица 4 – Используемые цвета в приложении

|  |  |
| --- | --- |
| Название цвета | HEX |
| Тёмно-серо-коричневый | 463933 |
| Миртовый | 243E22 |
| Серебристо-серый | 889397 |
| Глубокий жёлто-коричневый | 5D320E |
|  |  |

Преимуществами разрабатываемого мобильного приложения с визуальной точки зрения являются:

* Улучшенный дизайн
* Удобное управление
* Интересная информация

## 2.2 Разработка логики работы мобильного приложения (backend-разработка)

Разработка логики работы мобильного приложения «Экологические советы» включает в себя создание серверной части, которая отвечает за обработку данных, взаимодействие с базой данных и обеспечение функциональности приложения. Основные компоненты backend-разработки включают архитектуру сервера, API, управление данными и безопасность.

2.2.1 Архитектура сервера

Для разработки серверной части приложения рекомендуется использовать архитектуру RESTful API, которая обеспечивает возможность взаимодействия между клиентом и сервером. Это позволит создать четкие и понятные маршруты для обработки запросов. Сервер будет обрабатывать следующие основные запросы:

- Регистрация пользователя

- Авторизация пользователя

- Получение экологических советов

- Добавление отзывов и оценок

- Получение информации о пользователе

2.2.2 Выбор технологий

Для реализации серверной части можно использовать следующие технологии:

- Язык программирования: Kotlin или Java, так как они хорошо интегрируются с Android и позволяют создавать высокопроизводительные приложения.

- Фреймворк: Spring Boot, который предоставляет удобные инструменты для создания RESTful API и управления зависимостями.

- База данных: PostgreSQL для хранения данных пользователей, советов и отзывов.

- ORM: Hibernate для управления взаимодействием между приложением и базой данных.

Основные функции backend

1. Регистрация пользователя:

- При регистрации пользователь вводит свои данные (имя, адрес электронной почты, пароль).

- Данные проверяются на уникальность и корректность.

- Пароль хранится в зашифрованном виде с использованием алгоритма bcrypt.

2. Авторизация пользователя:

- Пользователь вводит свои учетные данные (электронная почта и пароль).

- Система проверяет введенные данные и, в случае успешной проверки, возвращает токен доступа для дальнейшего взаимодействия с API.

3. Получение экологических советов:

- Клиент отправляет запрос на получение списка советов.

- Сервер извлекает данные из базы данных и возвращает их в формате JSON.

4. Добавление отзывов и оценок:

- Пользователь может оставить отзыв о полученных советах.

- Сервер принимает данные от клиента, проверяет их и сохраняет в базе данных.

5. Получение информации о пользователе:

- Пользователь может запросить свои данные, такие как имя, история посещений и оставленные отзывы.

- Сервер обрабатывает запрос и возвращает соответствующую информацию.

2.2.4 Безопасность

Для обеспечения безопасности приложения необходимо реализовать следующие меры:

- Аутентификация и авторизация: Использование JWT (JSON Web Tokens) для безопасной передачи данных при авторизации.

- Шифрование данных: Хранение паролей в зашифрованном виде и использование HTTPS для защиты данных при передаче.

- Валидация данных: Проверка всех входящих данных на корректность и защиту от SQL-инъекций и других атак.

2.2.5 Тестирование

Тестирование backend-логики должно включать:

- Юнит-тесты для проверки отдельных функций и методов.

- Интеграционные тесты для проверки взаимодействия между компонентами системы.

- Функциональные тесты для проверки, что приложение выполняет все заявленные функции.

Таким образом, разработка логики работы мобильного приложения «Экологические советы» включает в себя создание надежной и безопасной серверной части, обеспечивающей все необходимые функции для взаимодействия с пользователем.

## 2.3 Разработка подключаемой базы данных (backend-разработка)

Подключение базы данных в приложении «Экологические советы» предполагает использование языка Kotlin для создания бэкенда с помощью фреймворка Spring Boot и взаимодействия с реляционной базой данных PostgreSQL. В этом разделе описывается процесс настройки и реализации подключения к базе данных.

2.3.1 Выбор СУБД

Для данного проекта выбрана реляционная система управления базами данных PostgreSQL. Она обеспечивает высокую производительность, надежность и поддержку сложных запросов.

2.3.2 Настройка базы данных

1. Установка PostgreSQL:

- Установите PostgreSQL на локальный сервер или используйте облачное решение для хостинга базы данных.

- Создайте базу данных и пользователя.

2. Создание структуры базы данных:

- Определите основные таблицы для хранения данных, такие как users, eco\_tips, reviews.

2.3.3 Подключение к базе данных в приложении на Kotlin

1. Добавление зависимостей:

- В build.gradle.kts добавьте необходимые зависимости для работы с PostgreSQL и Spring Data JPA

2. Настройка конфигурации подключения:

- В файле application.properties или application.yml укажите параметры подключения к базе данных

2.3.4 Реализация репозиториев на Kotlin

Для работы с базой данных создайте интерфейсы репозиториев, которые будут использовать Spring Data JPA. Пример реализации репозитория для пользователей

# ЗАКЛЮЧЕНИЕ

В результате выполнения проекта по разработке мобильного приложения я достигла поставленных целей и задач. Приложение было спроектировано с учетом потребностей пользователей, обеспечивая интуитивно понятный интерфейс и высокую производительность.

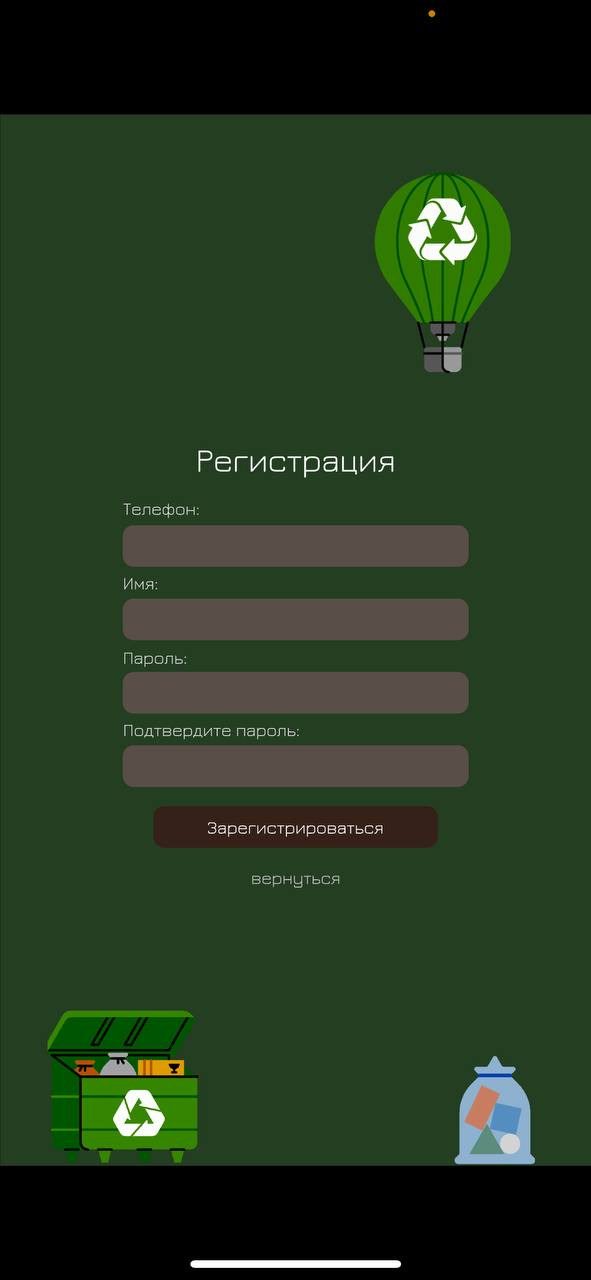
Я успешно справилась с внедрением ключевых функций, таких как возможность регистрации через электронную почту, социальные сети; отображение в личном кабинете информации о пользователе: имени, контактных данных, истории посещений, возможностью оставлять отзывы и оценки, системы уведомлений.

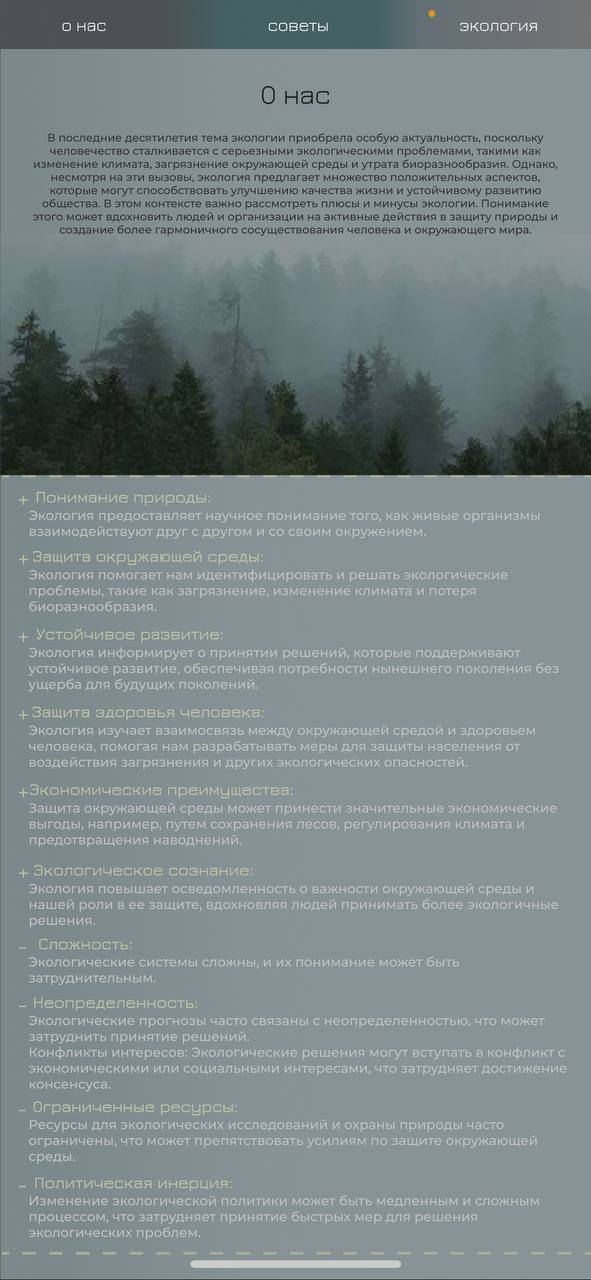
Также стоит отметить, что тестирование приложения выявило и устранило множество потенциальных ошибок, что значительно повысило его надежность.

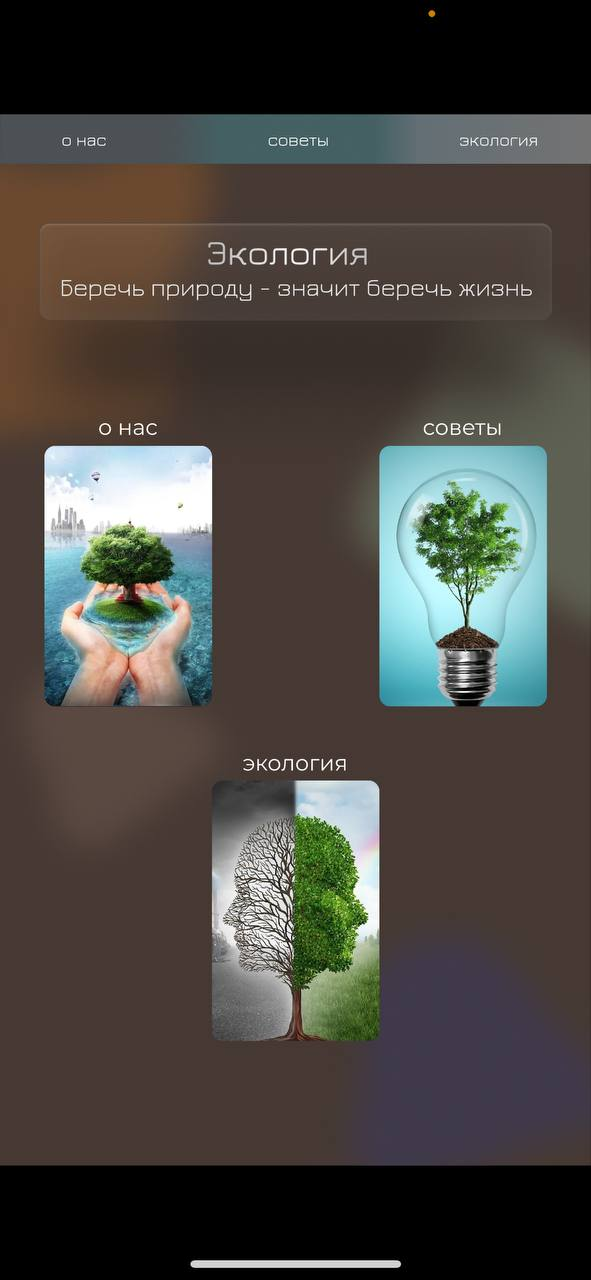
В будущем я планирую продолжать развитие приложения, добавляя новые функции и улучшая его функционал на основе обратной связи от пользователей.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Венгроу, Д. Прикладные структуры данных и алгоритмы. Прокачиваем навыки. / Д. Венгроу. – Санкт-Петербург : Издательство «Питер», 2024. – 512 с.
2. Вроблевски, Л. Сначала мобильные! пер. с англ. П. Миронова. /  
   Л. Вроблевски. – Москва : Издательство «Манн, Иванов и Фербер», 2022. – 176 с.
3. Гаятри, М. Фулстек-тестирование. / М. Гаятри. – Москва : Издательство «Спринт Бук», 2024. – 416 с.
4. Данн, М., Льюис, Ш. Нативная разработка мобильных приложений. Перекрестный справочник для iOS и Android. / М. Данн,  
   Ш. Льюис. – Москва : Издательство «ДМК Пресс», 2020. – 376 с.
5. Петросян, Л.Э., Гусев, К.В. Разработка мобильных приложений на языке Kotlin. / Л.Э. Петросян, К.В. Гусев. – Москва : Издательство «Лань», 2024. – 104 с.
6. Семенчук, В. Мобильное приложение как инструмент бизнеса. /  
   В. Семенчук – Москва : Издательство «Альпина Паблишер», 2017. – 240 с.
7. Скин, Д., Гринхол, Д. Kotlin. Программирование для профессионалов. 2-е изд. / Д. Скин, Д. Гринхол. – Санкт-Петербург : Издательство «Питер», 2023. – 560 с.
8. Соколова, В.В. Разработка мобильных приложений : учебное пособие для среднего профессионального образования / В.В. Соколова. – Москва : Издательство «Юрайт», 2024. – 160 с.
9. Тидвелл, Д., Брюэр, Ч., Валенсия, Э. Разработка интерфейсов. Паттерны проектирования. 3-е изд. / Д. Тидвелл, Ч. Брюэр, Э. Валенсия. – Санкт-Петербург : Издательство «Питер», 2022. – 560 с.
10. Фишерман, Л.В. Git. Практическое руководство. Управление и контроль версий в разработке программного обеспечения. / Л.В. Фишерман. – Санкт-Петербург : Издательство «Наука и техника», 2021. – 304 с.
11. Хортон, Дж. Разработка Android-приложений с нуля. 3-е изд. /  
    Дж. Хортон. – Санкт-Петербург : Издательство «БХВ», 2023. – 576 с.
12. https://developer.android.com/?hl=ru (дата обращения: 22.03.2025).
13. https://www.figma.com/ (дата обращения: 15.03.2025).
14. https://www.github.com/ (дата обращения: 16.03.2025).
15. https://www.reddit.com/r/learnprogramming/?rdt=60398 (дата обращения: 16.03.2025).
16. https://www.stackoverflow.com/ (дата обращения: 15.03.2025).
17. «Mobile App Development for Dummies» – *AppDevelopersIndia,* 13.06.2024. – https://developersappindia.com/blog/mobile-app-development-for-dummies-a-beginners-guide-to-getting-started (дата обращения: 15.10.2024).

ПРИЛОЖЕНИЯ





MainActivity.kt

package com.example.ecoauthapp  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.layout.\*  
import androidx.compose.foundation.text.BasicTextField  
import androidx.compose.material.Button  
import androidx.compose.material.Text  
import androidx.compose.runtime.\*  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.unit.dp  
import androidx.lifecycle.viewmodel.compose.viewModel  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** AuthScreen()  
 **}** }  
}  
  
@Composable  
fun AuthScreen(viewModel: AuthViewModel = viewModel()) {  
 var phone by remember **{** *mutableStateOf*("") **}** var password by remember **{** *mutableStateOf*("") **}** var name by remember **{** *mutableStateOf*("") **}** val result by viewModel.result.collectAsState()  
  
 Column(modifier = Modifier  
 .*fillMaxSize*()  
 .*padding*(16.*dp*)) **{** Text("Добро пожаловать в Eco App!")  
 Spacer(modifier = Modifier.*height*(8.*dp*))  
  
 BasicTextField(  
 value = phone,  
 onValueChange = **{** phone = **it }**,  
 modifier = Modifier.*fillMaxWidth*()  
 )  
 Spacer(modifier = Modifier.*height*(8.*dp*))  
  
 BasicTextField(  
 value = password,  
 onValueChange = **{** password = **it }**,  
 modifier = Modifier.*fillMaxWidth*()  
 )  
 Spacer(modifier = Modifier.*height*(8.*dp*))  
  
 BasicTextField(  
 value = name,  
 onValueChange = **{** name = **it }**,  
 modifier = Modifier.*fillMaxWidth*()  
 )  
 Spacer(modifier = Modifier.*height*(16.*dp*))  
  
 Row **{** Button(onClick = **{** viewModel.register(phone, password, name) **}**) **{** Text("Регистрация")  
 **}** Spacer(modifier = Modifier.*width*(8.*dp*))  
 Button(onClick = **{** viewModel.login(phone, password) **}**) **{** Text("Вход")  
 **}  
 }** Spacer(modifier = Modifier.*height*(16.*dp*))  
 Text("Ответ сервера: $result")  
 **}**}

AuthRepository.kt

package com.example.ecoauthapp  
  
import io.ktor.client.\*  
import io.ktor.client.call.\*  
import io.ktor.client.engine.cio.\*  
import io.ktor.client.plugins.contentnegotiation.\*  
import io.ktor.client.request.\*  
import io.ktor.http.\*  
import io.ktor.serialization.gson.\*  
  
class AuthRepository {  
 private val client = HttpClient(CIO) **{** install(ContentNegotiation) **{** *gson*()  
 **}  
 }** suspend fun register(phone: String, password: String, name: String): String {  
 val response = client.post("http://10.0.2.2:8080/auth/register") **{** *contentType*(ContentType.Application.Json)  
 *setBody*(*mapOf*("phone" *to* phone, "password" *to* password, "name" *to* name))  
 **}** return response.body()  
 }  
  
 suspend fun login(phone: String, password: String): String {  
 val response = client.post("http://10.0.2.2:8080/auth/login") **{** *contentType*(ContentType.Application.Json)  
 *setBody*(*mapOf*("phone" *to* phone, "password" *to* password))  
 **}** return response.body()  
 }  
}

build.gradle.kts(app)

plugins **{** id("com.android.application")  
 id("org.jetbrains.kotlin.android")  
**}***android* **{** namespace = "com.example.ecoauthapp"  
 compileSdk = 34  
  
 defaultConfig **{** applicationId = "com.example.ecoauthapp"  
 minSdk = 24  
 targetSdk = 34  
 versionCode = 1  
 versionName = "1.0"  
 **}** buildFeatures **{** compose = true  
 **}** composeOptions **{** kotlinCompilerExtensionVersion = "1.5.11"  
 **}** compileOptions **{** sourceCompatibility = JavaVersion.*VERSION\_11* targetCompatibility = JavaVersion.*VERSION\_11* **}  
}***dependencies* **{** *implementation*("androidx.compose.ui:ui:1.5.4")  
 *implementation*("androidx.compose.material:material:1.5.4")  
 *implementation*("androidx.compose.ui:ui-tooling-preview:1.5.4")  
 *implementation*("androidx.lifecycle:lifecycle-viewmodel-compose:2.6.2")  
 *implementation*("androidx.activity:activity-compose:1.8.0")  
  
 // Ktor  
 *implementation*("io.ktor:ktor-client-core:2.3.0")  
 *implementation*("io.ktor:ktor-client-cio:2.3.0")  
 *implementation*("io.ktor:ktor-client-content-negotiation:2.3.0")  
 *implementation*("io.ktor:ktor-serialization-gson:2.3.0")  
**}**

build.gradle.kts(project)

plugins **{** *kotlin*("jvm") *version* "1.9.22" *apply* false  
 id("com.android.application") *version* "8.2.2" *apply* false  
**}**allprojects **{  
  
}**

AuthViewModel.kt

package com.example.ecoauthapp  
  
import androidx.lifecycle.ViewModel  
import androidx.lifecycle.*viewModelScope*import kotlinx.coroutines.flow.MutableStateFlow  
import kotlinx.coroutines.flow.StateFlow  
import kotlinx.coroutines.launch  
  
class AuthViewModel : ViewModel() {  
 private val repo = AuthRepository()  
  
 private val \_result = *MutableStateFlow*("")  
 val result: StateFlow<String> = \_result  
  
 fun register(phone: String, password: String, name: String) {  
 *viewModelScope*.*launch* **{** \_result.value = repo.register(phone, password, name)  
 **}** }  
  
 fun login(phone: String, password: String) {  
 *viewModelScope*.*launch* **{** \_result.value = repo.login(phone, password)  
 **}** }  
}