**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

Кафедра ВТиСУ

**КУРСОВАЯ РАБОТА**

по дисциплине «Разработка мобильных приложений»

на тему «Разработка мобильных приложений на платформе Android. Часть 2. Разработка приложения для магазина одежды»

Студент: Абрамов Д.А.

Институт: ИИТЭ

Направление: 09.03.03

Группа: ПИ-121

Руководитель: Шутов А.В.

Владимир, 2024 г.

Содержание

[ВВЕДЕНИЕ 3](#_Toc185508355)

[1. Теоретическая часть 4](#_Toc185508356)

[1.1 Понятие и виды мобильных приложений 4](#_Toc185508357)

[1.2 Описание предметной области и задачи на разработку 5](#_Toc185508358)

[2. Практическая часть 7](#_Toc185508359)

[2.1 Анализ и выбор СMS системы для разработки серверной части мобильного приложения 7](#_Toc185508360)

[2.2 Обзор платформ для разработки мобильного приложения 9](#_Toc185508361)

[2.3 Обзор разработанного приложения 12](#_Toc185508362)

[ЗАКЛЮЧЕНИЕ 17](#_Toc185508363)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 18](#_Toc185508364)

# ВВЕДЕНИЕ

В современном мире мобильные технологии играют ключевую роль в жизни общества. С развитием смартфонов и мобильных устройств увеличивается потребность в создании удобных и функциональных приложений, способных удовлетворять разнообразные потребности пользователей. Одной из наиболее востребованных платформ для разработки мобильных приложений является Android, благодаря её открытости, масштабируемости и широкому спектру возможностей для разработчиков.

Целью данной курсовой работы является разработка мобильного приложения для платформы Android, которое будет предоставлять функционал для поиска и приобретения одежды. Данное приложение должно обеспечить пользователя удобным пользовательским интерфейсом, сочетать в себе простоту навигации и возможность оперативного получения необходимой информации.

Для достижения поставленной цели, необходимо решить следующие задачи:

1. Изучить инструменты и технологии, применимые при разработке приложений на платформе Android.
2. Разработать пользовательский интерфейса с учётом требований удобства и интуитивной понятности.
3. Реализовать функционала приложения.
4. Реализация функционала для просмотра активных заказов.

Таким образом, курсовая работа направлена на приобретение теоретических знаний и практических навыков в области разработки мобильных приложений для Android, что соответствует требованиям образовательной программы и способствует профессиональному развитию в сфере информационных технологий.

# 1. Теоретическая часть

## 1.1 Понятие и виды мобильных приложений

Мобильные приложения — это программное обеспечение, специально разработанное для работы на мобильных устройствах, таких как смартфоны и планшеты. Они позволяют пользователям выполнять различные задачи, от общения с другими людьми до получения информации и развлечений.

Мобильные приложения могут быть загружены и установлены на устройства через онлайн-магазины, такие как App Store для iOS и Google Play для Android.

Виды мобильных приложений:

1. Нативные приложения – разрабатываются специально для одной операционной системы (iOS или Android). Используют языки программирования, соответствующие платформе (Swift или Objective-C для iOS, Java или Kotlin для Android).

Обладают высокой производительностью и доступом ко всем функциям

устройства.

1. Гибридные приложения – сочетают элементы нативных и веб

приложений. Разрабатываются с использованием технологий веб-программирования (HTML, CSS, JavaScript) и обертываются в нативный интерфейс.

Позволяют создавать кроссплатформенные приложения, которые работают на разных операционных системах.

1. Веб-приложения – Это сайты, оптимизированные для мобильных устройств и обеспечивающие интерактивный пользовательский опыт. Не требуют установки на устройстве и работают через веб-браузер.

Обновляются на сервере, что позволяет пользователям всегда иметь доступ к последней версии приложения.

1. Прогрессивные веб-приложения (PWA) – это тип веб-приложений, который предлагает функциональность, схожую с нативными приложениями.

Обладают возможностью работы в оффлайн-режиме, отправки уведомлений и установки на устройства. Комбинируют лучшие аспекты веб-приложений и нативных приложений.

1. Игровые приложения – ориентированы на развлечение и взаимодействие с пользователями через игровую механику. Могут быть как нативными, так и гибридными, и часто требуют больших ресурсов для графики и обработки.

Основные черты мобильных приложений:

Доступность: Пользователи могут легко загрузить и установить приложения через магазины приложений.

Интерактивность: Мобильные приложения предлагают интерактивный интерфейс и пользовательский опыт.

Подключение к интернету: Многие приложения требуют постоянного подключения к сети для обновления данных и использования функциональности.

Мобильные приложения становятся все более важной частью нашей жизни, предоставляя пользователям возможности для работы, учебы, отдыха и общения.

## 1.2 Описание предметной области и задачи на разработку

Предметной областью для разработки мобильного приложения является сфера торговли одеждой. Основной целью разработки является создание удобного и интуитивно понятного мобильного приложения, которое позволит клиентам подобрать удобный, простой, но при этом стильный «outfit». Каждый товар имеет карточку, которая содержит описание изделия, состав, стоимость и информацию по наличию.

Администратор может просматривать информацию о оформленных заказах, также их редактировать и удалять.

Основными пользователями приложения являются сами клиенты и администратор приложения.

Задачей проекта является создание приложения для платформы Android, которое будет обладать функциональными возможностями для взаимодействия пользователями, желающими приобрести одежду в интернет-магазине.

Приложение должно иметь возможность регистрации и авторизации пользователей, добавления товара в корзину, просмотра информации об ценах и описании товара, просмотра оформленных заказов. Все эти функции направлены на повышение удобства для пользователей и автоматизацию процесса продажи.

# 2. Практическая часть

## 2.1 Анализ и выбор СMS системы для разработки серверной части мобильного приложения

Contentful – это облачная headless CMS, которая предоставляет мощные инструменты для создания, управления и доставки контента через API. Она позволяет разработчикам и контент-менеджерам работать отдельно, что значительно упрощает процесс публикации контента.

Преимущества:

1. Позволяет создавать сложные структуры контента с помощью кастомных моделей.

2. Удобный и интуитивный интерфейс для редактирования контента.

3. Поддерживает REST и GraphQL API, что упрощает интеграцию.

4. Обеспечивает хостинг и поддержку, что снижает нагрузку на технический персонал.

Недостатки:

1. Дорогие тарифные планы для масштабируемых проектов.

2. Некоторые ограничения в функционале для работы с большими объемами данных.

3. Невозможно полностью контролировать хостинг и инфраструктуру.

Sanity – это также headless CMS, которая акцентирует внимание на гибкости и кастомизации. Она предлагает уникальные возможности для разработки собственных плагинов и интеграции с другими системами.

Преимущества:

1. Позволяет создавать кастомные типы контента и настройки под конкретные нужды проекта.

2. Поддержка реального времени, что позволяет сразу видеть изменения.

3. Возможность самостийной настройки и расширения системы с помощью Sanity Studio.

4. Легкая интеграция с другими системами благодаря мощным API.

Недостатки:

1. Требует больше времени на первоначальную настройку, особенно для новичков.

2. Может потребоваться время на обучение для команд, не знакомых с инструментом.

Strapi – это open-source headless CMS, которая позволяет разработчикам быстро создать API для управления контентом. Она предлагает средства для создания пользовательских типов контента и имеет встроенную поддержку GraphQL.

Преимущества:

1. Полный контроль над кодом и возможностью хостинга на собственных серверах.

2. Позволяет легко создавать REST и GraphQL API, по сравнению с другими CMS.

3. Удобная панель управления для администраторов и контент-менеджеров.

4. Поддержка плагинов и расширений для кастомизации.

Недостатки:

1. В больших проектах может требовать оптимизации.

2. Требует дополнительных усилий для настройки и развертывания.

3. Меньшая активность сообщества по сравнению с более крупными решениями.

Каждая из этих платформ предлагает уникальные инструменты и подходы к управлению контентом, и выбор между ними зависит от потребностей конкретного проекта и команды.

Для разработки мобильного приложения было принято решение использовать Strapi, по следующим причинам:

1. Strapi — это headless CMS, что означает, что он позволяет вам создавать и управлять контентом без жесткой привязки к конкретному фронтенду. Можно использовать любой фреймворк или библиотеку (React, Vue, Angular и т.д.), что дает большую гибкость в разработке.

2. Strapi автоматически генерирует RESTful и GraphQL API, что позволяет легко интегрировать приложение с другими сервисами и использовать современные подходы к разработке.

3. Strapi поддерживает плагины и механизмы кастомизации, что позволяет разработчикам добавлять новые функции и интегрировать сторонние сервисы согласно специфическим требованиям проекта.

4. Strapi позволяет работать с несколькими базами данных, такими как MongoDB, PostgreSQL, MySQL и SQLite, что позволяет выбрать оптимальное решение для вашего приложения.

5. Strapi предлагает встроенные механизмы управления пользователями и их правами доступа, что обеспечивает защиту данных на уровне API и управления контентом.

6. Strapi позволяет разрабатывать приложения как для веба, так и для мобильных устройств, что делает Strapi идеальным решением для многоплатформенной разработки.

В то время как Contentful и Sanity предлагают свои преимущества, такие как облачные решения и интеграции, Strapi выделяется своей гибкостью, открытостью и возможностью самохостинга, что делает его привлекательным выбором для разработчиков, ищущих контроль и настройку под свои нужды.

2.2 Обзор платформ для разработки мобильного приложения

Для разработки приложения на Android существует огромное множество платформ, рассмотрим некоторые из них:

React Native — это популярная платформа с открытым исходным кодом, разработанная Facebook, которая позволяет разработчикам создавать нативные приложения с использованием JavaScript и React. Благодаря обширному сообществу и постоянным обновлениям React Native стал одной из наиболее широко используемых платформ для разработки кроссплатформенных приложений.

Ключевые особенности React Native включают в себя:

* Единая кодовая база для приложений Android и iOS.
* Использование собственных компонентов для повышения производительности взаимодействия с пользователем.
* Горячая перезагрузка для быстрой разработки и тестирования приложений.
* Интеграция с популярными библиотеками и инструментами React.

React Native доступен для Windows, macOS и Linux, и его можно использовать бесплатно.

Недостатки написания приложения на React Native:

* Может потребоваться написание нативного кода для некоторых сложных задач.
* Производительность немного ниже, чем у нативных приложений или Flutter в некоторых сценариях.

Xamarin, продукт Microsoft, представляет собой кроссплатформенную среду разработки приложений, использующую C# для создания нативных приложений для Android, iOS и Windows. Xamarin позволяет разработчикам совместно использовать код на разных платформах, сохраняя при этом возможность доступа к собственным API и функциям для повышения производительности.

Ключевые особенности Xamarin включают в себя:

* Общая база кода C# для приложений Android, iOS и Windows.
* Интеграция Visual Studio для беспрепятственного опыта разработки.
* Доступ к встроенному API для повышения производительности и возможностей приложений.
* Поддержка популярных библиотек и инструментов .NET.

Xamarin доступен для Windows и macOS и бесплатен для проектов с открытым исходным кодом и небольших команд.

Недостатки написания приложения на Xamarin:

* Большой размер приложения из-за дополнительных библиотек.
* Меньшее сообщество по сравнению с React Native и Flutter, что может затруднить поиск помощи.

Flutter — это пакет разработки пользовательского интерфейса с открытым исходным кодом, созданный Google, который позволяет создавать скомпилированные в собственном коде высокопроизводительные приложения для Android, iOS, Интернета и настольных компьютеров с единой кодовой базой. Flutter, созданный на языке программирования Dart, предлагает отличную производительность и множество готовых виджетов для быстрого и бесперебойного процесса разработки.

Особенности Flutter включают в себя:

* Единая кодовая база для Android, iOS, Интернета и настольных компьютеров.
* Быстрая разработка с горячей перезагрузкой и богатым набором готовых виджетов.
* Производительность, подобная нативной, благодаря заранее скомпилированному коду Dart.
* Настраиваемые и расширяемые компоненты пользовательского интерфейса с помощью пакетов.

Flutter доступен для Windows, macOS и Linux, и его можно использовать бесплатно.

Недостатки написания приложения на Flutter:

* Относительно новый фреймворк, что приводит к меньшему количеству доступных библиотек, особенно для специфичных задач.
* Меньшая поддержка для некоторых платформ по сравнению с нативной разработкой.

С учетом всех перечисленных факторов и характеристик, React Native представляется наиболее удобной платформой для разработки мобильного приложения магазина одежды, так как он будет отлично интегрироваться с выбранной crm системой Strapi.

Это связано с множеством готовых библиотек, а также возможностью быстрой разработки и легкой интеграции с различными API. JavaScript, как язык программирования, обеспечивает гибкость и позволяет разработчикам быстро находить решения для различных задач.

Таким образом, React Native предлагает оптимальный баланс между производительностью и удобством разработки для приложения.

## 2.3 Обзор разработанного приложения

При запуске приложения любой пользователь в первую очередь проходит авторизацию, либо регистрацию в случае отсутствия действующего аккаунта. Для идентификации пользователь использует email и пароль, которые хранятся в Strapi.

Если были введены некорректные данные – выскочит уведомление о том, что вы не зарегистрированы! При авторизации, аналогично.

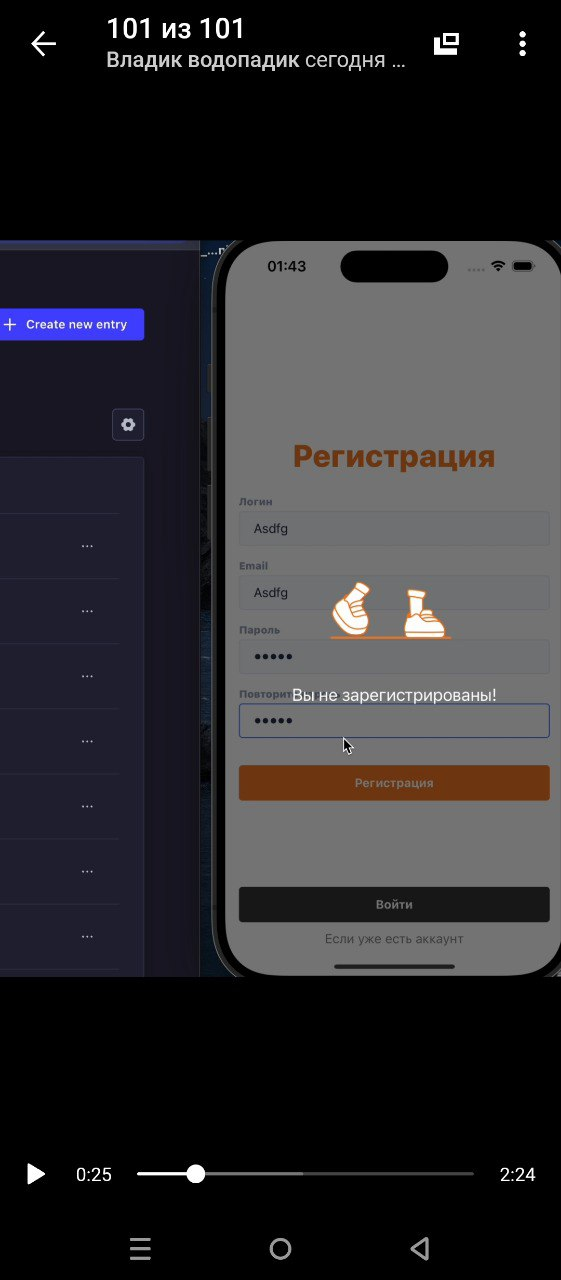


Рисунок 1 – Регистрация

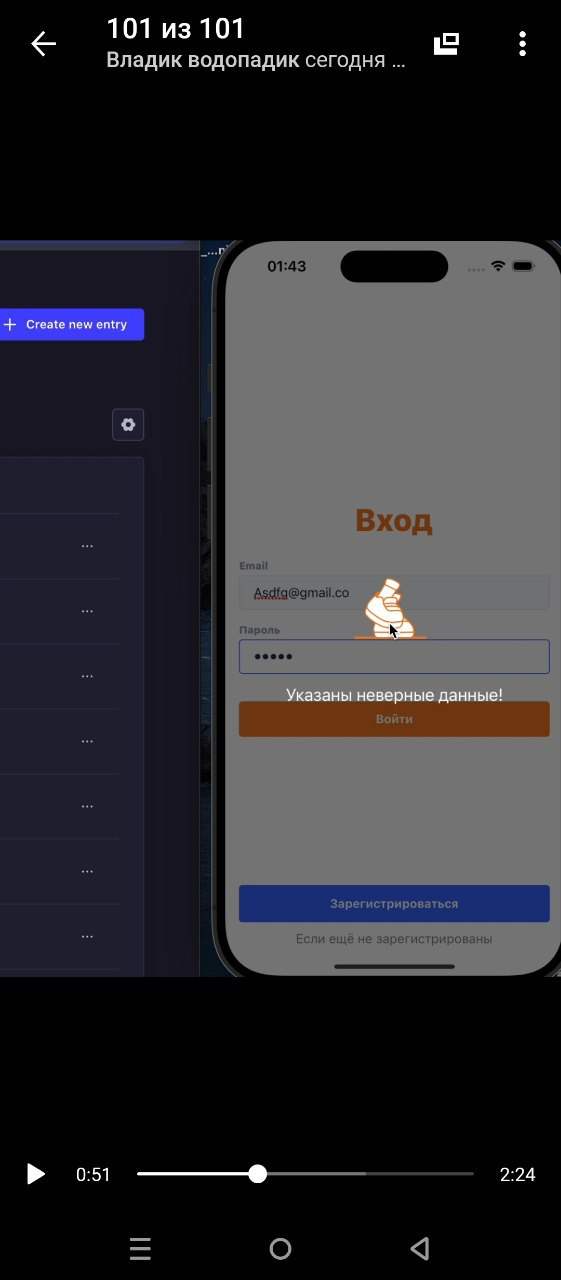


Рисунок 2 – Авторизация

Управление контентом происходит через Strapi «content tipe» – по сути это те же таблицы БД, в которых определяются какие поля будут у сущности, связи и т.п., затем после определения связей можно добавлять эти сущности и выводить в приложении.

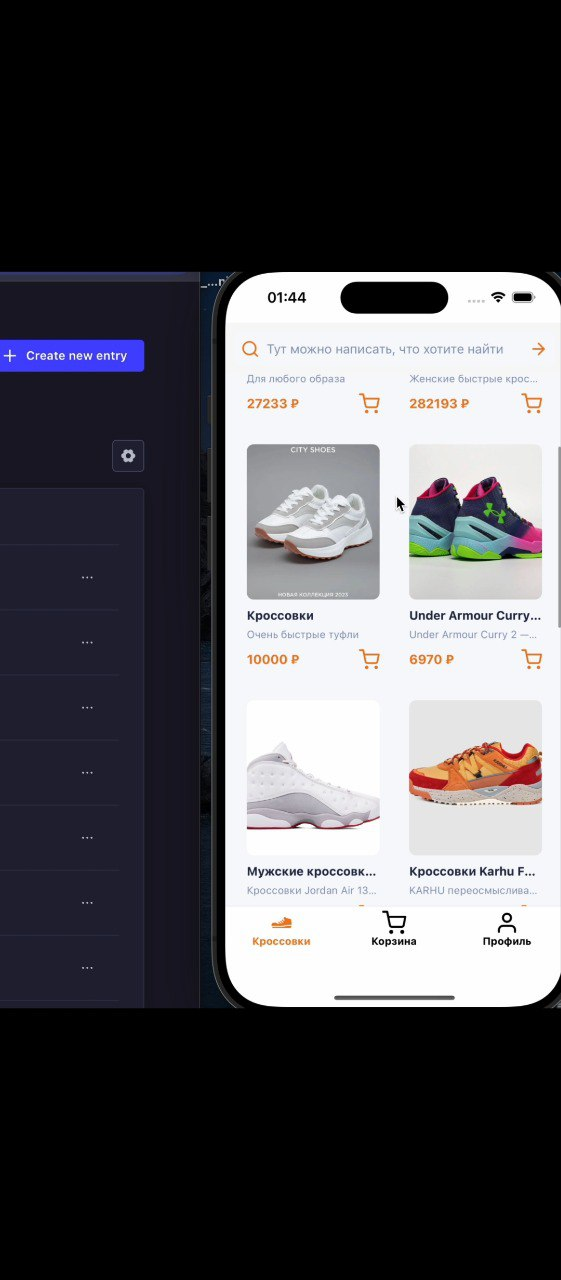


Рисунок 3 – Список товаров

Товар, который опубликован можно добавить в корзину и удалить из корзины. (рис. 4-6)

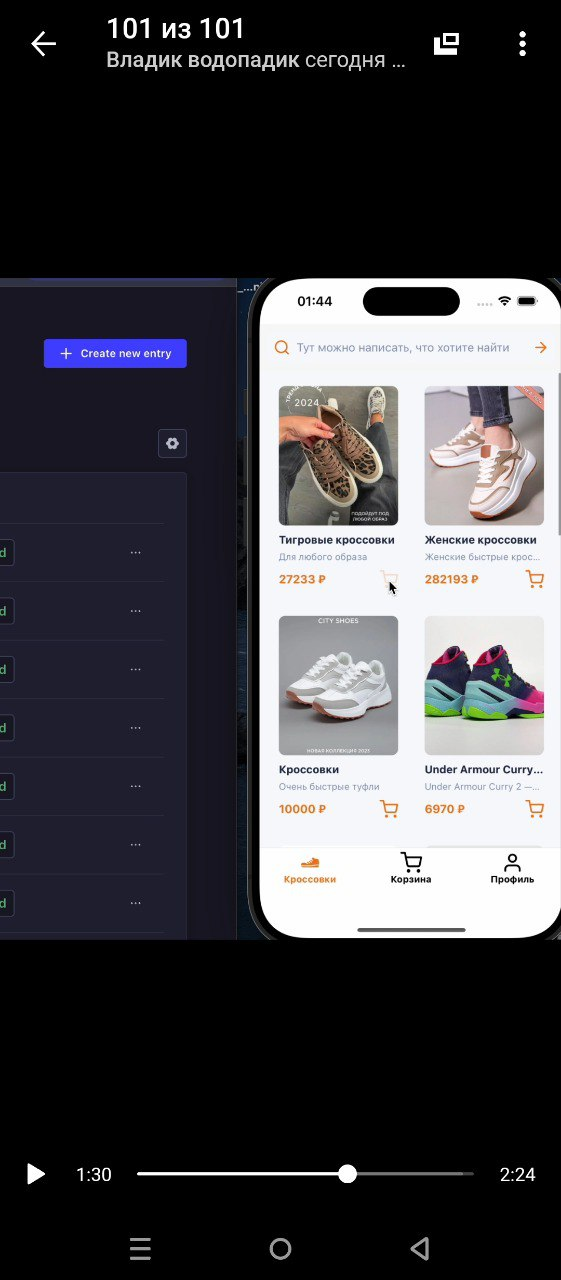


Рисунок 4 – Добавление товара в корзину

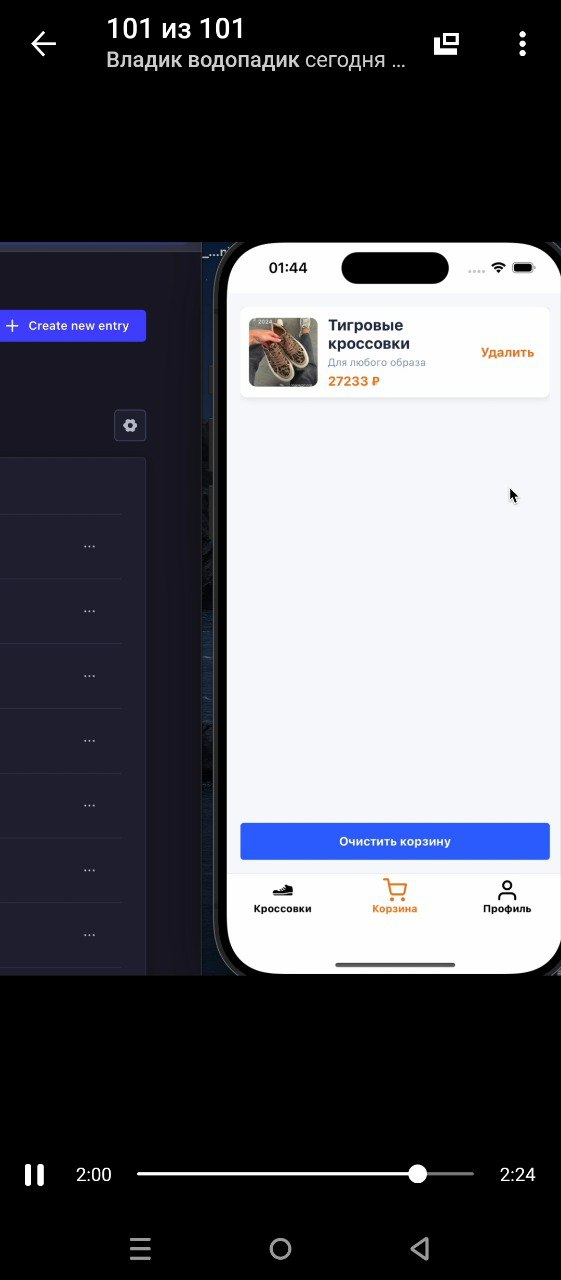


Рисунок 5 – Добавление товара в корзину

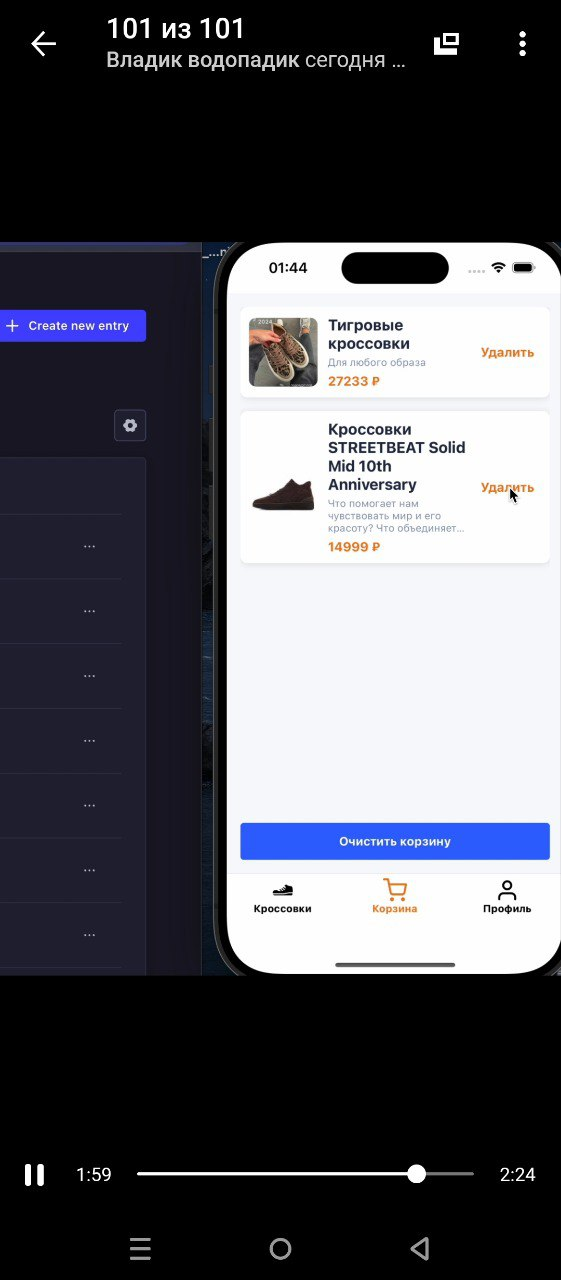


Рисунок 6 – Удаление товара из корзины

Для полного очищения корзины можно воспользоваться кнопкой «Очистить корзину». (рис. 7,8)

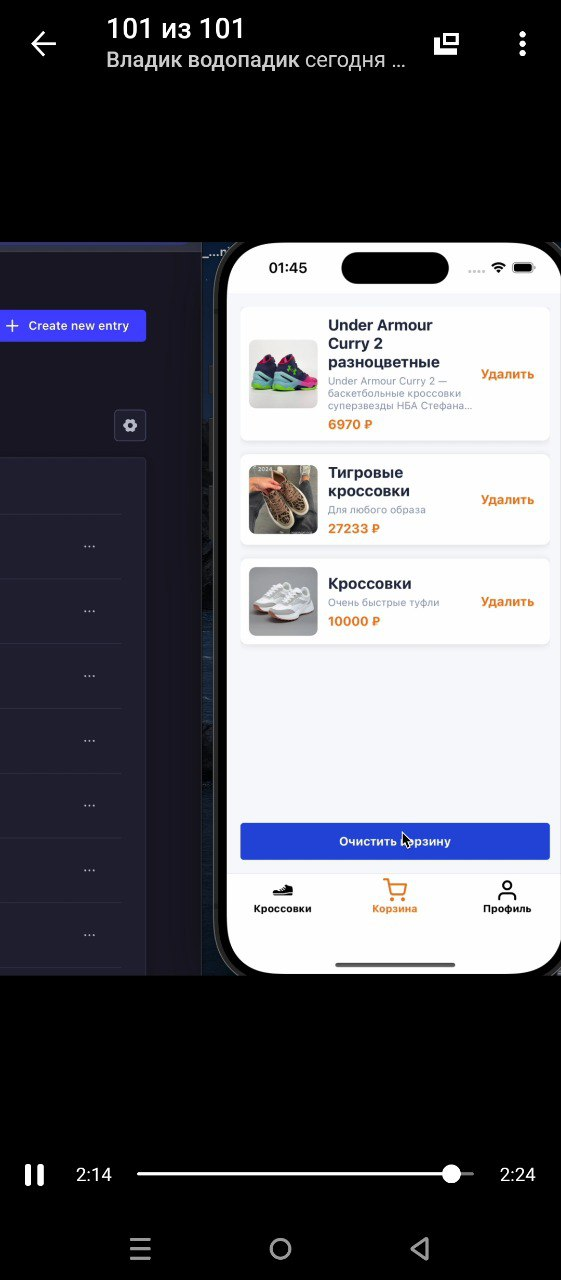


Рисунок 7 – Удаление товара из корзины кнопкой

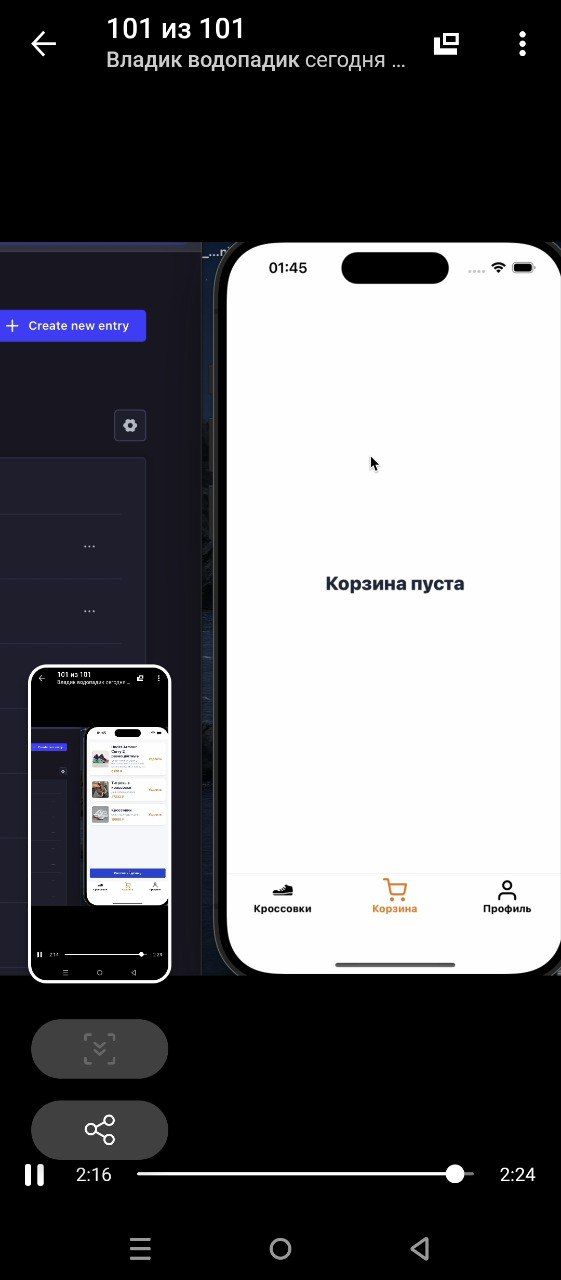


Рисунок 8 – Корзина очищена

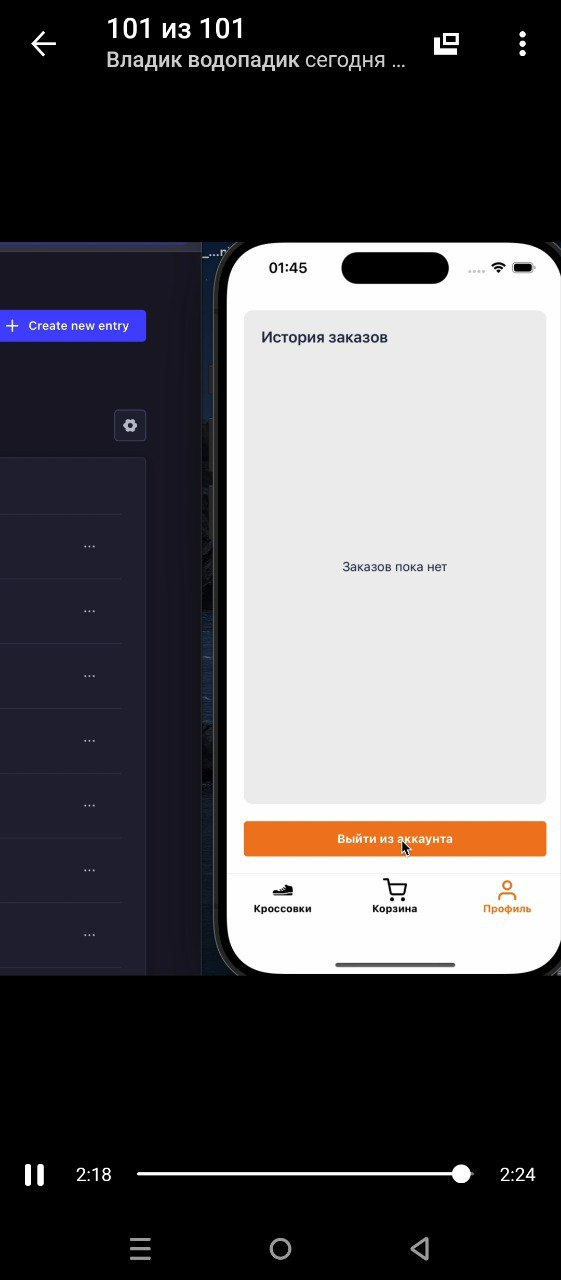


Рисунок 9 – Выйти из аккаунта

# ЗАКЛЮЧЕНИЕ

В рамках курсовой работы было разработано мобильное приложение для платформы Android, предназначенного для автоматизации и упрощения процессов покупки одежды. Проведённый анализ предметной области позволил выявить основные потребности пользователей и позволил определить особенности разработки приложения.

На этапе практической реализации были использованы современные инструменты и технологии, такие как платформа разработки React Native, язык программирования JavaScript для клиентской части, CRM-система Strapi, которая также и является API приложения.

Взаимодействие между клиентской частью и сервером было организовано с использованием REST API, что обеспечило надёжную передачу данных и быструю обработку запросов.

Разработанное приложение позволяет пользователям выполнять ключевые задачи, такие как регистрация, авторизация, добавление в корзину, удаление из корзины, выход из аккаунта. В результате работы был создан продукт, который не только повышает удобство пользования для клиентов магазина одежды, но и способствует оптимизации бизнес-процессов.

Таким образом, поставленные задачи курсовой работы были успешно решены, а разработанное приложение соответствует требованиям современного программного обеспечения, сочетая функциональность, удобство использования и надёжность.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Facebook. React Native. URL: [https://reactnative.dev](https://reactnative.dev/) (дата обращения: 01.10.2023).
2. Ефимова, Н. А. Мобильные приложения на основе React Native и Strapi. Уфа: Глава, 2021. 260 с.
3. Григорьев, С. Б. REST API и GraphQL на Strapi. Ростов-на-Дону: Феникс, 2022. 190 с.
4. Семенова, Л. Е. Мобильная разработка: проекты с React Native. Москва: Восход, 2023. 220 с.
5. Лукашенко, Т. В. Использование Strapi для управления содержимым. Тюмень: Тюменский университет, 2022. 130 с.
6. Strapi. Strapi Documentation. URL: <https://strapi.io/documentation/developer-docs/latest/getting-started/introduction.html> (дата обращения: 01.10.2023).
7. Лебедев, Н. П. Разработка веб-приложений на React / Н. П. Лебедев. — Москва: Издательство Бином. Лаборатория знаний, 2020. — 256 с.
8. Федоров, И. А. Стратегии разработка мобильных приложений / И. А. Федоров. — Санкт-Петербург: Питер, 2021. — 312 с.
9. Степанова, Е. В. Основы работы с Strapi / Е. В. Степанова. — Екатеринбург: Уральский университет, 2022. — 140 с.
10. Кузнецов, А. В. Разработка кроссплатформенных приложений с использованием React Native / А. В. Кузнецов. — Новосибирск: Сибирское университетское издательство, 2023. — 220 с.
11. Овсянников, Д. А. RESTful API и работа с ним / Д. А. Овсянников. — Москва: Наука, 2019. — 184 с.