**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 3 Разработка приложения для магазина кроссовок»

Студент: Агамирзоев Х.Ф

Институт: ИИТЭ

Направление: 09.03.03

Группа: ПИ-121

Руководитель: Шутов А.В.

Владимир 2024 г.

Оглавление

[Введение 3](#_Toc185346088)

[Глава 1. Теоретическая часть 5](#_Toc185346089)

[1.1 Современные тенденции мобильной коммерции 5](#_Toc185346090)

[1.2 Обзор существующих решений 5](#_Toc185346091)

[1.3 Выводы по анализу 6](#_Toc185346092)

[Глава 2. Практическая часть 7](#_Toc185346093)

[2.1 Функциональные и нефункциональные требования 7](#_Toc185346094)

[2.2 Проектирование архитектуры приложения 7](#_Toc185346095)

[2.3 Реализация функционала 8](#_Toc185346096)

[2.3.1 Авторизация и регистрация 8](#_Toc185346097)

[2.3.2 Отображение каталога товаров 12](#_Toc185346098)

[2.4 Административная панель 15](#_Toc185346099)

[2.4.1 Управление каталогом товаров 16](#_Toc185346100)

[2.4.2 Управление заказами 18](#_Toc185346101)

[Заключение 19](#_Toc185346102)

[Список литературы 20](#_Toc185346103)

# Введение

Современный мир стремительно движется в сторону цифровизации, и мобильные технологии занимают ключевую роль в этом процессе. Мобильные приложения стали незаменимым инструментом как для пользователей, так и для бизнеса, предлагая доступ к сервисам и продуктам в любое время и из любого места. В сфере электронной коммерции (e-commerce) мобильные приложения предоставляют уникальную возможность для покупателей взаимодействовать с товарами, совершать покупки и получать информацию о продукции в удобном и доступном формате.

На фоне глобального интереса к модным и спортивным товарам, кроссовки приобрели статус важного элемента стиля и культуры. Это делает разработку мобильного приложения для продажи кроссовок актуальной и востребованной. Удобство совершения покупок, персонализация предложений и прозрачность взаимодействия с клиентом становятся ключевыми факторами успешной работы любого интернет-магазина.

Целью данной курсовой работы является создание удобного и функционального мобильного приложения для продажи кроссовок.

Функции реализуемого приложения:

Просмотр каталога товаров: пользователи могут изучать ассортимент кроссовок с описаниями и изображениями.

Добавление товаров в корзину и оформление заказа: упрощенный процесс покупки с интеграцией платежных систем.

Авторизация и регистрация: возможность создать учетную запись, войти в систему и восстановить пароль.

Административная панель: возможность администрировать товары, заказы и пользователей.

Задачи работы:

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Провести анализ предметной области, изучить существующие решения и определить потребности целевой аудитории.

2. Разработать удобный интерфейс для пользователей и администраторов, ориентированный на интуитивное взаимодействие.

3. Реализовать авторизацию и регистрацию пользователей с безопасным хранением данных.

4. Разработать функционал административной панели для управления ассортиментом и обработкой заказов.

* Практическая значимость проекта заключается в создании программного продукта, который обеспечивает:
* Простоту покупки для конечного пользователя.
* Удобство управления магазином для администратора.
* Повышение вовлеченности клиентов благодаря персонализации и интеграции современных технологий.

# Глава 1. Теоретическая часть

## 1.1 Современные тенденции мобильной коммерции

Мобильная коммерция (m-commerce) представляет собой эволюцию электронной коммерции, фокусируясь на использовании мобильных устройств для совершения покупок. Сегодня m-commerce становится неотъемлемой частью повседневной жизни пользователей благодаря широкому распространению смартфонов, высокоскоростному интернету и удобству мобильных приложений.

Согласно статистике, более 60% онлайн-покупок совершаются через мобильные устройства. Это связано с несколькими ключевыми тенденциями:

Рост популярности мобильных платежей. Технологии, такие как Apple Pay, Google Pay и QR-коды, делают процесс оплаты удобным и безопасным.

Персонализация пользовательского опыта. Современные приложения используют данные о поведении клиентов для формирования рекомендаций и предложений.

Интеграция с социальными сетями. Пользователи могут совершать покупки напрямую через платформы, такие как Instagram и TikTok.

Мобильные уведомления. Push-уведомления позволяют мгновенно информировать пользователей о новых товарах, акциях и скидках.

Мобильные приложения, особенно для модных товаров, таких как кроссовки, позволяют бизнесу увеличивать лояльность клиентов, предлагать уникальные продукты и оптимизировать внутренние процессы.

## 1.2 Обзор существующих решений

Для успешной реализации проекта важно изучить существующие решения на рынке мобильных приложений для продажи обуви. Наиболее популярные приложения в данной сфере:

Nike App. Приложение предоставляет пользователям полный доступ к каталогу товаров, эксклюзивным коллекциям и программам лояльности.

Adidas. Помимо функционала интернет-магазина, приложение предлагает интеграцию с фитнес-программами.

Lamoda. Универсальное приложение для покупки обуви и одежды с удобным интерфейсом и быстрой доставкой.

Основные преимущества этих приложений:

* Современный и интуитивно понятный дизайн.
* Интеграция с платежными системами.
* Персонализированные предложения для пользователей.

Однако у данных решений также есть ограничения, такие как отсутствие полной локализации для определённых регионов, сложность взаимодействия с технической поддержкой и высокие цены.

## 1.3 Выводы по анализу

1. Проведённый анализ показал, что для создания конкурентоспособного продукта важно учитывать следующие аспекты:
2. Удобство интерфейса: приложение должно быть интуитивным и понятным.
3. Быстродействие: высокая скорость загрузки страниц и товаров играет важную роль в удержании пользователей.
4. Функциональность: необходимо реализовать удобный поиск, фильтры, авторизацию и возможность управления заказами.

Приложение для продажи кроссовок должно сочетать в себе преимущества существующих решений и учитывать их недостатки, предлагая пользователю уникальные возможности. Особое внимание стоит уделить внедрению административной панели для удобного управления ассортиментом и заказами.

# Глава 2. Практическая часть

## 2.1 Функциональные и нефункциональные требования

Для успешной реализации мобильного приложения важно определить функциональные и нефункциональные требования. Они задают рамки разработки и обеспечивают соответствие конечного продукта ожиданиям пользователей.

1. Функциональные требования:
   * Авторизация и регистрация пользователей.
   * Просмотр каталога товаров.
   * Корзина (добавление, удаление товаров, оформление заказа).
   * Административная панель (управление товарным ассортиментом).
2. Нефункциональные требования:
   * Интуитивно понятный интерфейс для пользователей с минимальным опытом.
   * Безопасность: шифрование пользовательских данных и безопасные протоколы передачи.
   * Кроссплатформенность: приложение ориентировано на Android.

## 2.2 Проектирование архитектуры приложения

1. Приложение предоставляет пользовательский интерфейс для взаимодействия с пользователем.
2. Используется библиотека Android Jetpack для управления интерфейсом.
3. **Backend:**
   * Реализует бизнес-логику приложения.
   * Серверная часть создаётся на PHP с использованием REST API для связи с клиентом.
4. **База данных:**
   * Хранение данных о товарах, пользователях и заказах.
   * Используется MySQL с оптимизированными запросами.

**Структура базы данных:**

* Таблица users:
* Поля: id, email, password, role.
* Таблица products:
* Поля: id, name, brand, price, description, image.
* Таблица orders:
* Поля: id, user\_id, total\_price, status.

## 2.3 Реализация функционала

### 2.3.1 Авторизация и регистрация

Фрагмент кода для регистрации пользователя:

// register.php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$email = $\_POST['email'];

$password = password\_hash($\_POST['password'], PASSWORD\_BCRYPT);

$query = "INSERT INTO users (email, password, role) VALUES (?, ?, 'user')";

$stmt = $pdo->prepare($query);

$stmt->execute([$email, $password]);

echo json\_encode(['status' => 'success']);

}

На стороне клиента (Android):

public void registerUser(String email, String password) {

Retrofit retrofit = new Retrofit.Builder()

.baseUrl("https://yourserver.com/api/")

.addConverterFactory(GsonConverterFactory.create())

.build();

ApiService apiService = retrofit.create(ApiService.class);

Call<ResponseBody> call = apiService.register(email, password);

call.enqueue(new Callback<ResponseBody>() {

@Override

public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {

if (response.isSuccessful()) {

Toast.makeText(context, "Регистрация успешна!", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(context, "Ошибка регистрации", Toast.LENGTH\_SHORT).show();

}

}

@Override

public void onFailure(Call<ResponseBody> call, Throwable t) {

Toast.makeText(context, "Ошибка сети", Toast.LENGTH\_SHORT).show();

}

});

}

Файл RegisterActivity.java

Этот файл содержит логику обработки данных, которые пользователь вводит в форму регистрации. Рассмотрим, как он реализует проверки заполнения:

Основные проверки:

1. Пустые поля:

if (username.isEmpty()) { Toast.makeText(RegisterActivity.this, "Введите имя пользователя", Toast.LENGTH\_SHORT).show(); return; }

if (email.isEmpty()) { Toast.makeText(RegisterActivity.this, "Введите адрес электронной почты", Toast.LENGTH\_SHORT).show(); return; }

if (password.isEmpty()) { Toast.makeText(RegisterActivity.this, "Введите пароль", Toast.LENGTH\_SHORT).show(); return; }

if (!password.equals(confirmPassword)) { Toast.makeText(RegisterActivity.this, "Пароли не совпадают", Toast.LENGTH\_SHORT).show(); return; }

1. **Пароль**

if (password.length() < 8) {Toast.makeText(RegisterActivity.this, "Пароль должен быть не менее 8 символов", Toast.LENGTH\_SHORT).show();return;}

1. **Проверка уникальности (взаимодействие с сервером)**:

ApiService apiService = ApiClient.getApiService(); Call<ResponseBody> call = apiService.registerUser(new User(username, email, password)); call.enqueue(new Callback<ResponseBody>() { @Override public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) { if (response.isSuccessful()) { Toast.makeText(RegisterActivity.this, "Регистрация прошла успешно!", Toast.LENGTH\_SHORT).show(); finish(); } else { Toast.makeText(RegisterActivity.this, "Ошибка: этот email уже зарегистрирован", Toast.LENGTH\_SHORT).show(); } } @Override public void onFailure(Call<ResponseBody> call, Throwable t) { Toast.makeText(RegisterActivity.this, "Ошибка соединения: " + t.getMessage(), Toast.LENGTH\_SHORT).show(); }

**Файл** register.php

Этот файл обрабатывает данные на сервере. Он отвечает за проверку уникальности email, а также за добавление пользователя в базу данных.

Основные проверки:

Хеширование пароля и запись в базу данных:

$password = password\_hash($\_POST['password'], PASSWORD\_BCRYPT); $query = $db->prepare("INSERT INTO users (username, email, password) VALUES (?, ?, ?)"); if ($query->execute([$\_POST['username'], $email, $password])) { echo json\_encode(['status' => 'success', 'message' => 'Регистрация успешна']); } else { echo json\_encode(['status' => 'error', 'message' => 'Ошибка регистрации']); }

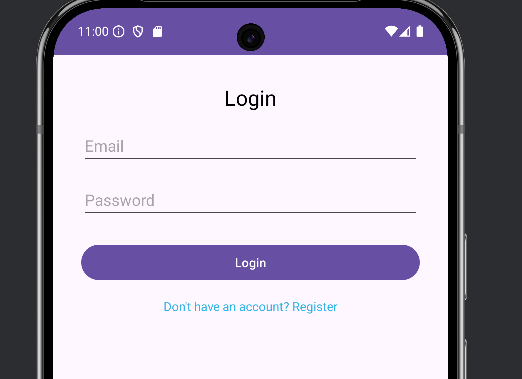


Рисунок 1 - Авторизация в приложении

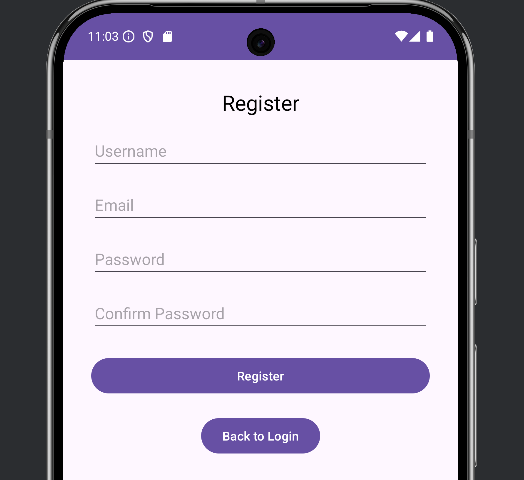


Рисунок 2 - Регистрация

### 2.3.2 Отображение каталога товаров

Фрагмент кода для получения списка товаров:

// get\_products.php

$query = "SELECT \* FROM products";

$stmt = $pdo->query($query);

$products = $stmt->fetchAll(PDO::FETCH\_ASSOC);

echo json\_encode($products);

XML код для отображения 1 продукта в каталоге (См. рисунок 3)

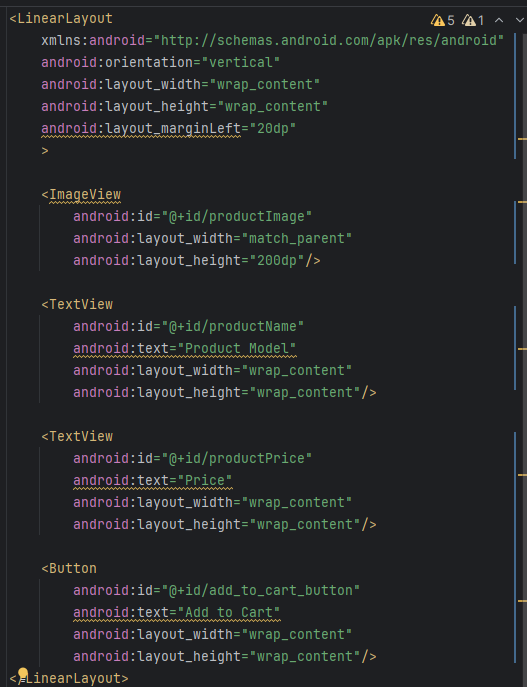


Рисунок 3 - XML код

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 4 - Визуализация XML кода

На стороне Android:

public void fetchProducts() {

Call<List<Product>> call = apiService.getProducts();

call.enqueue(new Callback<List<Product>>() {

@Override

public void onResponse(Call<List<Product>> call, Response<List<Product>> response) {

if (response.isSuccessful()) {

List<Product> products = response.body();

productAdapter.setProducts(products);

}

}

@Override

public void onFailure(Call<List<Product>> call, Throwable t) {

Toast.makeText(context, "Ошибка загрузки каталога", Toast.LENGTH\_SHORT).show();

}

});

}

Изображение выглядит как текст, снимок экрана, Мобильный телефон, Мобильное устройство

Автоматически созданное описание

Рисунок 5 - Каталог

## 2.4 Административная панель

Административная панель предназначена для управления каталогом товаров и заказами. Она реализована на основе серверной части, разработанной на PHP, и предоставляет интерфейс для взаимодействия через веб-браузер.

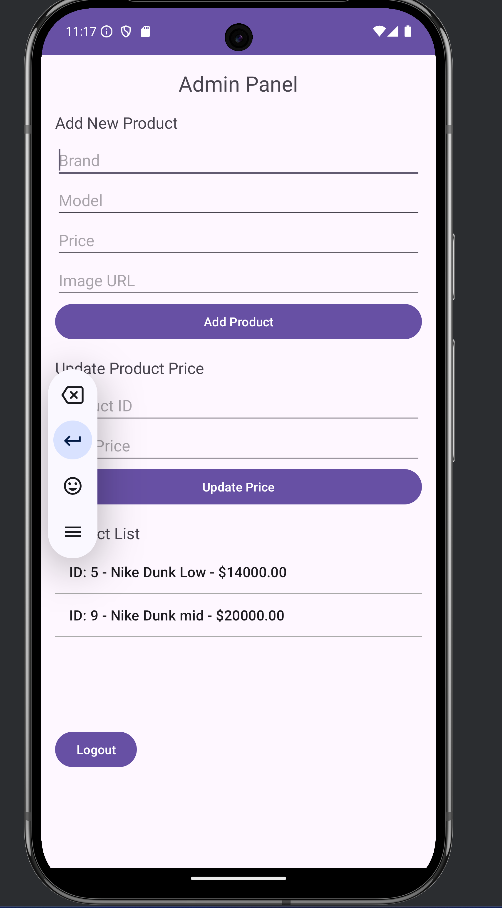


Рисунок 6 - Административная панель управления каталогом

### 2.4.1 Управление каталогом товаров

Администратор может добавлять, удалять и редактировать товары в каталоге. Это достигается с помощью следующих функций:

**Добавление товара**

// add\_product.php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$name = $\_POST['name'];

$brand = $\_POST['brand'];

$price = $\_POST['price'];

$description = $\_POST['description'];

$image = $\_POST['image']; // Здесь передаётся URL или base64 строка

$query = "INSERT INTO products (name, brand, price, description, image) VALUES (?, ?, ?, ?, ?)";

$stmt = $pdo->prepare($query);

$stmt->execute([$name, $brand, $price, $description, $image]);

echo json\_encode(['status' => 'success']);

}

**Удаление товара**

// delete\_product.php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$productId = $\_POST['id'];

$query = "DELETE FROM products WHERE id = ?";

$stmt = $pdo->prepare($query);

$stmt->execute([$productId]);

echo json\_encode(['status' => 'success']);

}

**Редактирование товара**

// update\_product.php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$id = $\_POST['id'];

$name = $\_POST['name'];

$brand = $\_POST['brand'];

$price = $\_POST['price'];

$description = $\_POST['description'];

$image = $\_POST['image'];

$query = "UPDATE products SET name = ?, brand = ?, price = ?, description = ?, image = ? WHERE id = ?";

$stmt = $pdo->prepare($query);

$stmt->execute([$name, $brand, $price, $description, $image, $id]);

echo json\_encode(['status' => 'success']);

}

На клиентской стороне Android приложение предоставляет интерфейс для отображения и управления этими функциями.

### 2.4.2 Управление заказами

Заказы отображаются в административной панели с возможностью изменения их статусов. Администратор может просматривать данные о каждом заказе, включая список товаров и контактную информацию пользователя.

**Фрагмент кода для получения списка заказов:**

// get\_orders.php

$query = "SELECT o.id, o.total\_price, o.status, u.email FROM orders o JOIN users u ON o.user\_id = u.id";

$stmt = $pdo->query($query);

$orders = $stmt->fetchAll(PDO::FETCH\_ASSOC);

echo json\_encode($orders);

**Фрагмент кода для обновления статуса заказа:**

// update\_order\_status.php

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$orderId = $\_POST['id'];

$status = $\_POST['status'];

$query = "UPDATE orders SET status = ? WHERE id = ?";

$stmt = $pdo->prepare($query);

$stmt->execute([$status, $orderId]);

echo json\_encode(['status' => 'success']);

}

# Заключение

В процессе разработки мобильного приложения для продажи кроссовок была выполнена полная реализация проекта, начиная с постановки задач и анализа требований, заканчивая созданием работающего программного продукта. Приложение включает в себя все ключевые функции, необходимые для удобного взаимодействия пользователей с каталогом товаров и оформления заказов.

Были достигнуты следующие результаты:

1. **Анализ предметной области и существующих решений**. На основе исследования современных тенденций мобильной коммерции были выделены основные требования к приложению. Учет актуальных трендов позволил разработать продукт, который соответствует ожиданиям пользователей.
2. **Проектирование и реализация приложения**. Разработаны интерфейсы для работы с каталогом, корзиной, системой авторизации и регистрации пользователей. В приложение также интегрирована административная панель для управления товарами и заказами.
3. **Функциональные особенности**. Пользователи могут просматривать каталог, добавлять товары в корзину, оформлять заказы и управлять своим аккаунтом. Для владельцев бизнеса реализована возможность управления ассортиментом через удобный интерфейс.
4. **Практическая значимость**. Разработанное приложение может быть использовано как готовое решение для небольших и средних магазинов кроссовок. Это позволит оптимизировать процессы торговли, улучшить пользовательский опыт и повысить уровень продаж.

Проект подтвердил значимость мобильных технологий в сфере торговли и продемонстрировал возможности современных инструментов разработки

# Список литературы

1. **Лобанов, Д. В.** Основы разработки мобильных приложений на платформе Android. – Москва: Диалектика, 2021. – 374 с.
2. **Крутиков, А. П.** Проектирование пользовательского интерфейса мобильных приложений. – Санкт-Петербург: Питер, 2019. – 245 с.
3. **Google Developers.** Руководство по разработке на Android [Электронный ресурс]. – URL: <https://developer.android.com> (дата обращения: 15.12.2024).
4. **Официальный сайт библиотеки компонентов Material Design** [Электронный ресурс]. – URL: <https://material.io> (дата обращения: 15.12.2024).
5. **Шилов, Е. Н.** Основы UX/UI-дизайна для мобильных приложений. – Москва: Наука и техника, 2022. – 236 с.