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

на тему «Разработка мобильного приложения для ресторана»

Студент: Абдуфатиков А.Х

Институт: ИИТэ

Направление: 09.03.03

Группа: ПИ-121

№ зачётной книжки: 121002680

Руководитель: Шутов А.В

Владимир 2024

Оглавление

[Введение 3](#_Toc185206056)

[Глава 1. Анализ предметной области и современные технологии мобильной коммерции 5](#_Toc185206057)

[1.1 Современные тенденции мобильной коммерции 5](#_Toc185206058)

[1.2 Обзор существующих решений 6](#_Toc185206059)

[1.3 Выводы по анализу 8](#_Toc185206060)

[Глава 2. Разработка мобильного приложения для ресторана 9](#_Toc185206061)

[2.1 Функциональные и нефункциональные требования 9](#_Toc185206062)

[2.2 Проектирование архитектуры приложения 11](#_Toc185206063)

[2.3 Реализация функционала 12](#_Toc185206064)

[2.3.1 Авторизация и регистрация 12](#_Toc185206065)

[2.3.2 Отображение каталога товаров 17](#_Toc185206066)

[2.4 Административная панель 21](#_Toc185206067)

[2.4.1 Управление каталогом товаров 22](#_Toc185206068)

[2.4.2 Управление заказами 25](#_Toc185206069)

[Заключение 26](#_Toc185206070)

# **Введение**

Современный мир стремительно движется в сторону цифровизации, и мобильные технологии играют ключевую роль в этом процессе. Мобильные приложения стали незаменимым инструментом как для пользователей, так и для бизнеса, предоставляя доступ к услугам и сервисам в любое время и из любого места. В сфере обслуживания клиентов мобильные приложения занимают особое место, способствуя автоматизации процессов и улучшению взаимодействия с пользователями.

На фоне роста популярности ресторанной индустрии и растущего спроса на удобные и быстрые способы оформления заказов, разработка мобильного приложения для ресторана становится особенно актуальной. Автоматизация приема и обработки заказов, персонализация взаимодействия с клиентами и упрощение внутренних бизнес-процессов являются ключевыми факторами успешного развития ресторанного бизнеса.

Целью данной курсовой работы является создание удобного и функционального мобильного приложения для ресторана, которое включает:

* **Просмотр каталога товаров:** пользователи могут изучать меню ресторана с ценами и изображениями.
* **Добавление товаров в корзину и оформление заказа:** упрощенный процесс покупки с интеграцией платежных систем.
* **Авторизация и регистрация:** возможность создавать личные профили, входить в приложение и сохранять историю заказов.
* **Административная панель:** инструмент для управления контентом приложения, включая добавление новых товаров, редактирование цен и управление заказами.

**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 интуитивно понятный интерфейс для пользователей и администраторов.
3. Реализовать функционал авторизации и регистрации пользователей с безопасным хранением данных.

Разработать административную панель для управления меню и отслеживания заказов.

**Практическая значимость**

Практическая значимость проекта заключается в создании программного продукта, который обеспечивает:

* Удобство выбора и заказа блюд для клиентов.
* Автоматизацию и упрощение управления рестораном для администратора.
* Повышение лояльности клиентов за счет персонализации и современных цифровых решений.

# **Глава 1. Анализ предметной области и современные технологии мобильной коммерции**

## **1.1 Современные тенденции мобильной коммерции**

**Мобильные технологии в сфере ресторанного бизнеса**

Мобильные приложения представляют собой эволюцию цифровых технологий, фокусируясь на упрощении взаимодействия между клиентами и бизнесом. Сегодня они становятся неотъемлемой частью повседневной жизни благодаря широкому распространению смартфонов, высокоскоростному интернету и удобству использования.

Согласно статистике, более 70% пользователей предпочитают оформлять заказы и бронировать столики в ресторанах через мобильные устройства. Это связано с несколькими ключевыми тенденциями:

1. **Рост популярности мобильных платежей.** Технологии, такие как Apple Pay, Google Pay и QR-коды, делают процесс оплаты удобным и безопасным.
2. **Персонализация пользовательского опыта.** Приложения используют данные о предпочтениях клиентов для формирования рекомендаций, предложений и акций.
3. **Интеграция с социальными сетями.** Возможность делиться отзывами, фотографиями блюд и участвовать в акциях через платформы, такие как Instagram и Facebook, усиливает взаимодействие с аудиторией.
4. **Мобильные уведомления.** Push-уведомления мгновенно информируют пользователей о новинках меню, специальных предложениях и скидках.

## Мобильные приложения для ресторанов позволяют бизнесу повышать лояльность клиентов, улучшать качество сервиса и оптимизировать внутренние процессы. Благодаря удобству взаимодействия с клиентами и автоматизации ключевых задач, такие приложения становятся важным инструментом для успешного ведения ресторанного бизнеса.

## **1.2 Анализ существующих решений**

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

* **Yandex.Eda.** Приложение предлагает пользователям возможность заказывать блюда из ресторанов с быстрой доставкой, а также имеет систему персонализированных рекомендаций.
* **Delivery Club.** Помимо стандартного функционала, приложение предоставляет доступ к специальным акциям, программам лояльности и удобной системе отзывов.

**McDonald's.** Официальное приложение сети ресторанов быстрого питания, позволяющее заказывать еду, участвовать в акциях и получать бонусы.

**Основные преимущества этих приложений:**

* Современный и интуитивно понятный дизайн.
* Интеграция с платежными системами.
* Возможность персонализации предложений для пользователей.

Недостатки существующих решений:

* Отсутствие индивидуального подхода для небольших и средних ресторанов.
* Ограниченный функционал для управления внутренними процессами ресторана.
* Зависимость от сторонних служб доставки, что увеличивает расходы для ресторанов.

Анализ этих решений позволяет учесть их сильные стороны и избежать распространённых ограничений, что станет основой для создания удобного и функционального приложения для ресторана.

## **1.3 Выводы по анализу**

Проведённый анализ показал, что для создания конкурентоспособного мобильного приложения для ресторана важно учитывать следующие аспекты:

1. **Удобство интерфейса.** Приложение должно быть интуитивным и понятным как для пользователей, так и для администраторов.
2. **Быстродействие.** Высокая скорость загрузки меню, оформления заказов и обработки данных играет ключевую роль в удержании клиентов.
3. **Функциональность.** Приложение должно включать удобный поиск блюд, фильтры по категориям, возможность авторизации и ведения истории заказов, а также управление текущими заказами.

Приложение для ресторана должно сочетать в себе преимущества существующих решений и учитывать их недостатки, предоставляя клиентам уникальные возможности. Особое внимание следует уделить разработке административной панели, которая упростит управление меню, обработку заказов и анализ ключевых показателей.

# **Глава 2. Разработка мобильного приложения для ресторана**

## **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.

**Функциональные требования:**

1. **Авторизация и регистрация пользователей:**
   * Вход в приложение через электронную почту и пароль.
   * Регистрация новых пользователей с валидацией данных.
   * Сброс пароля через подтверждение по электронной почте.
2. **Просмотр каталога товаров:**
   * Отображение списка блюд с изображениями, названием, ценой и описанием.
   * Возможность фильтрации товаров по цене и популярности.
3. **Корзина:**
   * Добавление товаров в корзину.
   * Удаление товаров из корзины.
   * Отображение итоговой стоимости.
4. **Оформление заказа:**
   * Заполнение данных для доставки.
   * Выбор способа оплаты.
   * Подтверждение заказа с сохранением истории.
5. **Административная панель:**
   * Управление каталогом товаров: добавление, удаление и редактирование.
   * Управление заказами: просмотр статусов и управление выполнением.

**Нефункциональные требования:**

1. Интуитивно понятный интерфейс для пользователей с минимальным опытом.
2. Безопасность: шифрование пользовательских данных и безопасные протоколы передачи.
3. Кроссплатформенность: приложение ориентировано на Android.

## **2.2 Проектирование архитектуры приложения**

Архитектура приложения строится на модульной основе, что обеспечивает лёгкость в поддержке и масштабировании. Основные компоненты:

1. **Frontend (UI):**
   * Отвечает за отображение данных и взаимодействие с пользователем.
   * Используется библиотека Android Jetpack для управления интерфейсом.
2. **Backend:**
   * Реализует бизнес-логику приложения.
   * Серверная часть создаётся на PHP с использованием REST API для связи с клиентом.
3. **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.
* Таблица order\_items:
  + Поля: id, order\_id, product\_id, quantity.

## **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:
   1. if(username.isEmpty()){ Toast.makeText(RegisterActivity.this, "Введите имя пользователя", Toast.LENGTH\_SHORT).show(); return; }
   2. if (email.isEmpty()) { Toast.makeText(RegisterActivity.this, "Введите адрес электронной почты", Toast.LENGTH\_SHORT).show(); return; }
   3. if (password.isEmpty()) { Toast.makeText(RegisterActivity.this, "Введите пароль", Toast.LENGTH\_SHORT).show(); return; }
   4. if (!password.equals(confirmPassword)) { Toast.makeText(RegisterActivity.this, "Пароли не совпадают", Toast.LENGTH\_SHORT).show(); return; }
2. **Пароль**
   1. if (password.length() < 8) {Toast.makeText(RegisterActivity.this, "Пароль должен быть не менее 8 символов", Toast.LENGTH\_SHORT).show();return;}
3. **Проверка уникальности (взаимодействие с сервером)**:
   1. 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**:

1. $password = password\_hash($\_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']); }

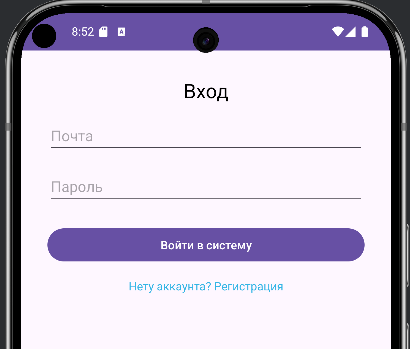


Рисунок 1. Авторизация в приложении

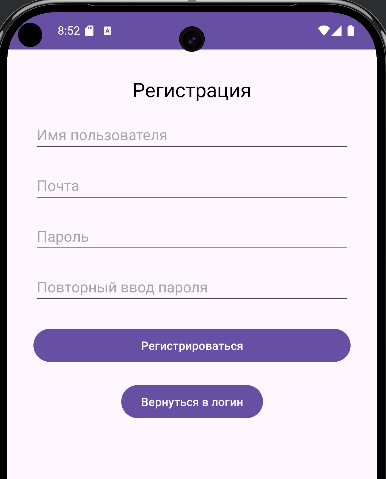


Рисунок 2. Регистрация

### **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



Рисунок 3. XML-код для карточки блюда



Рисунок 4. Визуализация в системе

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();

}

});

}

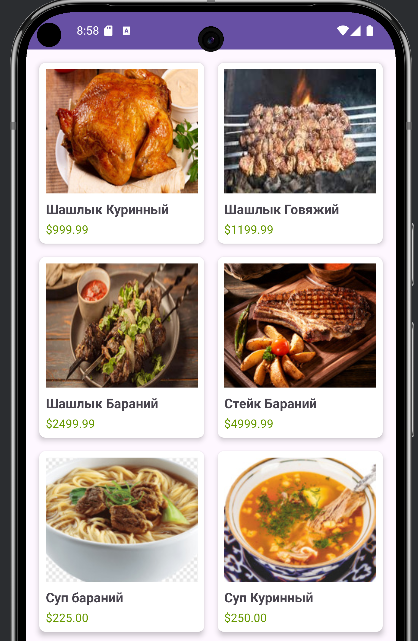


Рисунок 5. Каталог

## **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.

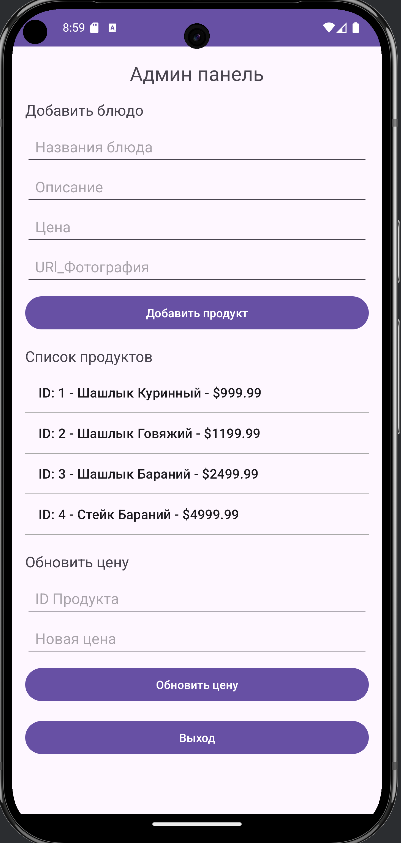


Рисунок 6. Административная панель управления каталогом

### **2.4.1 Управление каталогом товаров**

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

**Добавление товара**

Фрагмент кода для добавления нового 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**

Фрагмент кода для удаления 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**

Фрагмент кода для обновления информации о товаре:

// 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 ресторана 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 меню ресторана, оформления заказов и управления клиентскими данными.

**Достигнуты следующие результаты:**

1. **Анализ предметной области и существующих решений.** На основе исследования современных тенденций в ресторанной индустрии и мобильных технологиях были сформированы основные требования к приложению. Учет актуальных трендов позволил разработать продукт, соответствующий ожиданиям клиентов.
2. **Проектирование и реализация приложения.** Разработаны интерфейсы для работы с меню, оформления заказов, системы авторизации и регистрации пользователей. В приложение также интегрирована административная панель для управления меню, заказами и формирования отчетов.
3. **Функциональные особенности.** Пользователи могут просматривать меню с фотографиями и описаниями блюд, оформлять заказы и управлять личным кабинетом. Для администраторов ресторана реализована возможность управления ассортиментом и мониторинга заказов через удобный интерфейс.
4. **Практическая значимость.** Разработанное приложение может быть использовано как готовое решение для небольших и средних ресторанов. Это позволит автоматизировать процессы приема и обработки заказов, улучшить пользовательский опыт и повысить лояльность клиентов..

Реализация проекта продемонстрировала ключевую роль мобильных технологий в развитии ресторанного бизнеса. Использование мобильных приложений позволяет не только оптимизировать процессы приема и обработки заказов, но и значительно улучшить качество взаимодействия между рестораном и его клиентами. Благодаря современным инструментам разработки удалось создать продукт, который удовлетворяет потребности как пользователей, так и администраторов.

Приложение обеспечивает удобный доступ к меню ресторана, упрощает процесс оформления заказов и оплаты, а также предоставляет возможность персонализации сервиса для клиентов. Административная панель, интегрированная в приложение, позволяет ресторанам эффективно управлять ассортиментом блюд, анализировать данные о заказах и улучшать обслуживание.

Таким образом, проект подтвердил, что мобильные технологии являются неотъемлемой частью успешного ведения бизнеса в условиях высокой конкуренции и растущих ожиданий клиентов. Приложение стало примером того, как современные цифровые решения могут повысить лояльность пользователей, укрепить позиции ресторана на рынке и вывести обслуживание на новый уровень.

# **Список литературы**

1. Иванов А. С. **Разработка мобильных приложений на платформе Android**. — Москва: Альфа-книга, 2020. — 368 с.
2. Смирнов В. К. **Дизайн интерфейсов для мобильных приложений**. — Санкт-Петербург: БХВ-Петербург, 2018. — 290 с.
3. Android Developers. **Документация по разработке приложений для Android** [Электронный ресурс]. — URL: <https://developer.android.com/docs> (дата обращения: 15.12.2024).
4. Официальный сайт Google Material Design. **Material Design Guidelines** [Электронный ресурс]. — URL: https://material.io/design (дата обращения: 15.12.2024).
5. Никифоров М. П. **Основы проектирования пользовательских интерфейсов**. — Екатеринбург: У-Фактория, 2021. — 256 с.