МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Серверная часть E-commerce приложения на языке Dart

09.03.02 Информационные системы и технологии

Программная инженерия в информационных системах

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_ д.ф-м.н, профессор С.Д. Махортов \_\_.\_\_.20\_\_

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Е. Коробов, 3 курс, д/о

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.ф-м.н, доцент Е.А. Михайлюк

Воронеж 2024

СОДЕРЖАНИЕ

[1 Назначение и цель создания приложения 6](#__RefHeading___Toc2658_1070835350)

[1.1 Цели создания приложения 6](#__RefHeading___Toc2664_1070835350)

[1.2 Задачи, решаемые с помощью приложения 6](#__RefHeading___Toc2662_1070835350)

[2 Целевая аудитория 6](#__RefHeading___Toc2656_1070835350)

[3 Обзор конкурентов 6](#__RefHeading___Toc2654_1070835350)

[4 Технологический стек 7](#__RefHeading___Toc2652_1070835350)

[4.1 Требования к программному обеспечению приложения 7](#__RefHeading___Toc2660_1070835350)

[5 Группы пользователей 8](#__RefHeading___Toc2650_1070835350)

Термины и определения

* **Сервер, серверная часть –** компьютер, обслуживающий другие устройства (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач;
* **Клиент, клиентская сторона –** в данном проекте получение пользователем данных по запросу на сервер;
* **Back-end –** программно-аппаратная часть приложения. Отвечает за функционирование внутренней (серверной) части приложения;
* **Пользователь, клиент –** человек, пользующийся функционалом приложения;

Введение

Данная курсовая работа направлена на разработку серверной части E-commerce мобильного приложения.

За последние 10 лет популярность маркетплейсов значительно возросла и продолжает расти каждый день. Так, совокупный объём трат на маркетплейсах в 2023 году почти в 1,5 раза больше, чем в 2022 году, а количество транзакций выросло на 63%. Они стали неотъемлемой частью современной электронной коммерции и предоставляют удобную платформу для покупки и продажи товаров и услуг как на мировом, так и на внутреннем рынке России.

Возможность создания на одном языке клиент-серверное приложение, намного ускоряет и удешевляет процесс разработки. Написание полного стека для мобильного или веб приложения с использованием одной технологии не является новой. Этим путем прошел JavaScript (JS+ React Native + Node JS), Python (Django / Cowasm + Kivy), Go, Java и прочие языки. Но Dart обладает существенными преимуществами:

* Кроссплатформенность, за счет фреймворка Flutter
* Скорость и минимальный бюджет для создания приложения
* Уменьшенный бойлерплейт
* Кодогенерация как клиентской, так и серверной части
* Широкий выбор библиотек, которые решают задачи от управления схемой БД и их миграций, до реализации DI

В качестве серверной части проекта мы будет создавать REST API для CRUD операций над товарами в интернет маркетплейсе. На Dart есть много решений для реализации этой задачи, такие как Shelf, Dart Frog, Conduit. Но самый оптимальный вариант — это Serverpod. Он позволяет совместно с серверной частью создавать клиентский проект на Flutter, сгенерировать API и подключиться к БД. На момент первого квартала 2024 года библиотека вышла в релиз, стабильно поддерживает как iOS, так и Android.

1. Назначение и цель создания приложения
   1. Цели создания приложения

Цель создания приложения:

* реализация серверной части с дальнейшей интеграцией мобильного приложения в условиях одного проекта
  1. Задачи, решаемые с помощью приложения

Разрабатываемое приложение решает следующие задачи:

* оформление заказа;
* интеграция своего локального магазина и товара для оптовой продажи пользователем;
* просмотр истории заказов и отслеживание статус заказа;
* добавление категорий и товаров в каталог

1. Целевая аудитория

Продукт ориентирован на оптовых продавцов и покупателей. Сервис разработан с целью облегчения оптовых сделок и предлагает набор удобств и инструментов, предназначенный на B2B сектор.

Именно сужение целевой аудитории и функционал, ориентированный на данную категорию пользователей, делает приложение уникальным.

1. Обзор конкурентов

Создание приложений, включающее в себя как клиентскую, так и серверную части, с использованием одного языка программирования, значительно ускоряет и удешевляет процесс разработки. Этот подход уже был реализован с помощью таких языков, как JavaScript, Python, Go, Java и других. Однако, язык программирования Dart имеет ряд существенных преимуществ:

* Кроссплатформенность, благодаря фреймворку Flutter, позволяющая разрабатывать приложения для различных платформ.
* Быстрая разработка приложений с минимальными затратами.
* Сокращенный бойлерплейт, упрощающий процесс написания кода.
* Возможность автоматической генерации как клиентской, так и серверной частей приложения.
* Богатый выбор библиотек, позволяющих решать различные задачи, от управления базами данных до реализации внедрения зависимостей.

В качестве серверной части для нашего проекта - интернет-магазина, необходимо реализовать REST API для выполнения операций CRUD с товарами. Для этого на Dart доступно несколько решений, таких как Shelf, Dart Frog, Conduit. Но наиболее оптимальным вариантом является Serverpod. Эта библиотека позволяет совместно разрабатывать серверную и клиентскую части на Flutter, автоматически генерирует API и обеспечивает подключение к базе данных.

1. Технологический стек
   1. Требования к программному обеспечению приложения

Для реализации серверной части приложения будут использоваться следующие средства:

* Dart 3.3.1
* Serverpod 1.2.7
* Docker;
* СУБД PostgreSQL;

Данные технологии выбраны исходя возможностей, которые они дают для решения задач. В качестве преимуществ выбранных технологий можно отметить:

Для PostgreSQL:

* наличие широкого функционала для горизонтального и вертикального масштабирования;
* поддержка множества типов данных.

Для Dart:

* кроссплатформенность;
* быстрота и лёгкость проектирования мобильных приложений;
* понятная и полная документация.

1. Группы пользователей

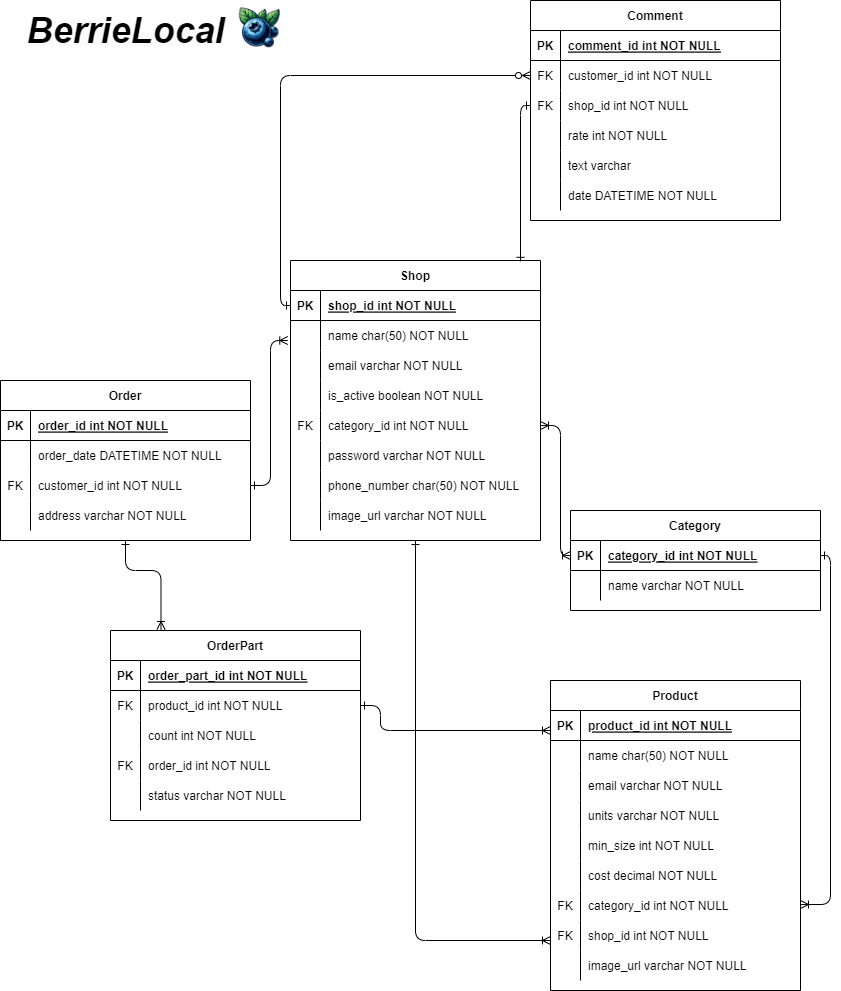
Для взаимодействия с разрабатываемой системой выделяются следующие типы пользователей:

* авторизованный пользователь;
* неавторизованный пользователь.

Авторизованный пользователь имеет доступ к основным функциям приложения.

Заключение

ПРИЛОЖЕНИЕ Д



1. ER-диаграмма