Технический проект для мобильного приложения для курьеров службы доставки

Содержание

[1](#_Toc21182) **[Пользовательский сценарий и интерфейс](#_Toc21182)** [29](#_Toc21182)

[1.1 Организация хранения данных 29](#_Toc28967)

[2](#_Toc25292) **[Бизнес-процессы и потоки данных системы](#_Toc25292)** [31](#_Toc25292)

[2.1 Бизнес-процессы 31](#_Toc17392)

[3](#_Toc26686) **[Структура программы и алгоритмы](#_Toc26686)** [33](#_Toc26686)

[3.1 Структура программы 33](#_Toc9811)

[4](#_Toc9349) **[Выбранная архитектура системы](#_Toc9349)** [34](#_Toc9349)

[5](#_Toc5197) **[Экранные формы приложения](#_Toc5197)** [35](#_Toc5197)

1 Пользовательский сценарий и интерфейс

Для каждой роли пользователя предполагаются свои рабочие пространства, доступные только им. Диаграмма use case представлена на рисунке 1.1

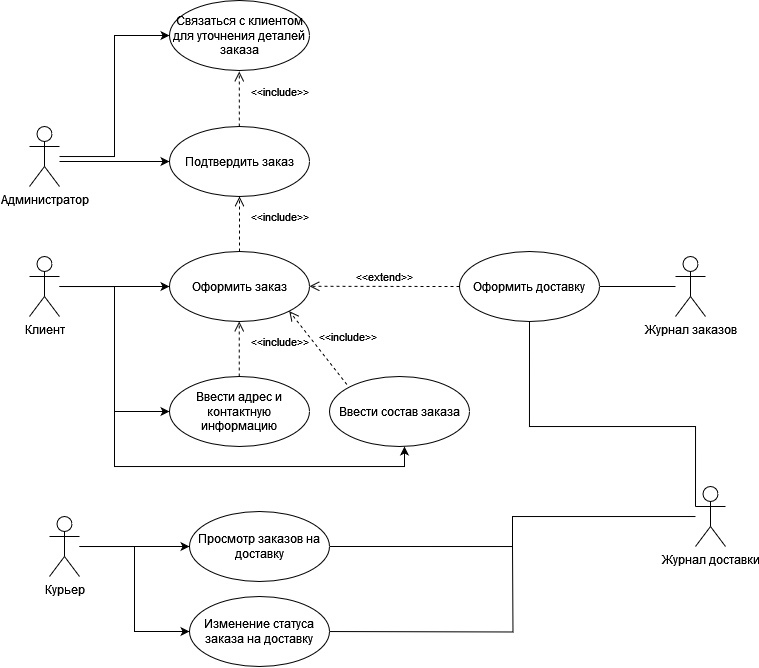


Рисунок 1.1 – Диаграмма use case

Пользовательский сценарий для клиента, предполагает, что он может оформить заказ, а именно выбрать состав заказа и ввести адрес и контактную информацию для оформления доставки. Для этого должен находиться экран с соответствующими полями и перечнем всех доступных продуктов для заказа.

Для администраторов пользовательский сценарий должен представлять собой подтверждение заказа, который оставил клиент и также уточнение у него деталей, если он указал контактную информацию неверно.

Для курьеров пользовательский сценарий представляет собой просмотр заказов на доставку и изменение статуса заказа на доставку, то есть курьер может указать принимает он заказ или нет и так же доставлен заказ или нет.

* 1. Организация хранения данных

Для организации хранения данных выбрана СУБД PostgreSQL, для работы с которой используется библиотека psycopg2.

База данных имеет 6 таблиц, каждая из которых имеет первичных ключ. Так же база имеет промежуточную таблицу order\_composition, чтобы избежать связи многие ко многим.

Физическая модель базы данных представлена на рисунке 1.2.

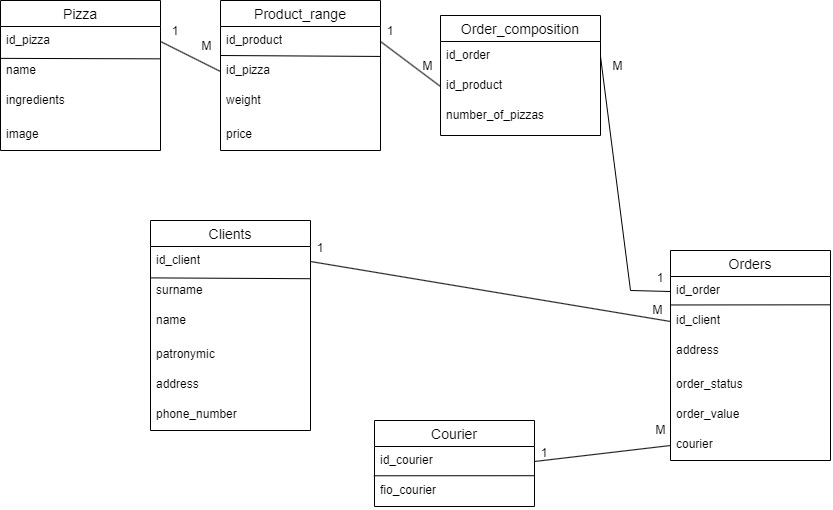


Рисунок 1.2 – Физическая модель базы данных

1. Бизнес-процессы и потоки данных системы
   1. Бизнес-процессы

Бизнес-процесс приложения для курьеров заключается в доставке заказа клиенту. Описание этого процесса представлено на IDF0.

Нулевая декомпозиция IDEF изображена на рисунке 2.1.

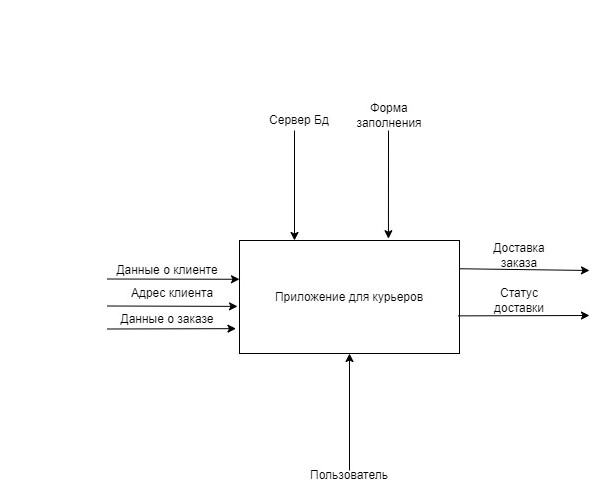


Рисунок 2.1 – Диаграмма IDEF0

Входными данными для проектируемой системы является: данные о клиенте, адрес клиента, данные о заказе.

Управляющими взаимодействием: форма заполнения, сервер БД (правила/условия выполнения запроса).

Механизм управления: пользователь приложения.

Выводимые данные: доставка заказа и статус заказа.

Диаграмма IDF1 представлена на рисунке 2.2.

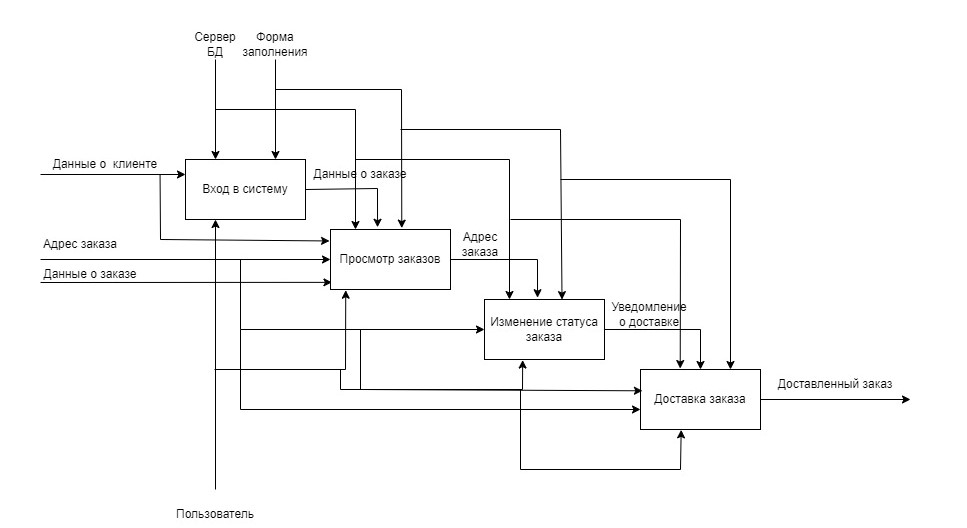


Рисунок 2.2 – Диаграмма IDEF1

1. Структура программы и алгоритмы
   1. Структура программы

Выбранный язык программирования - Dart с использованием библиотеки Flutter. Рефакторинг кода предполагает, что переход по разным модулям происходит посредством перехода по соответствующим кнопкам.

Навигация осуществлена отслеживанием нажатия на соответствующие кнопки, после чего подгружается тот или иной виджет, который привязан к этой кнопке.

* 1. Алгоритмы решения задач

Входная форма проверяет наличие такого пользователя в базе данных, в случае несовпадения пароля показывает сообщение о некорректности введенных данных, а также сообщает если такого пользователя нет в системе.

Вывод информации в таблицы реализуются SQL-запросами SELECT, после чего полученный результат помещается в переменную и выводится циклом в таблицу. В каждую таблицу выводится и скрывается id записи, чтобы их можно редактировать и удалять запросами UPDATE и DELETE по полю id.

1. Выбранная архитектура системы

Для созданного решения была выбрала двухзвенная архитектура «Клиент-сервер» с базой данных. В основе работы СУБД PostgreSQL лежит серверный процесс базы данных, выполняемый на одном сервере. Доступ из приложений к данным базы PostgreSQL производится с помощью специального процесса базы данных. То есть клиентские программы не могут получать самостоятельный доступ к данным даже в том случае, если они функционируют на том же ПК, на котором осуществляется серверный процесс.

PostgreSQL работает в режиме клиент-сервер. Клиент отправляет запросы на сервер, который в свою очередь получает эти запросы, обрабатывает их и возвращает результаты.

Архитектура приложения представлена на рисунке 4.1.

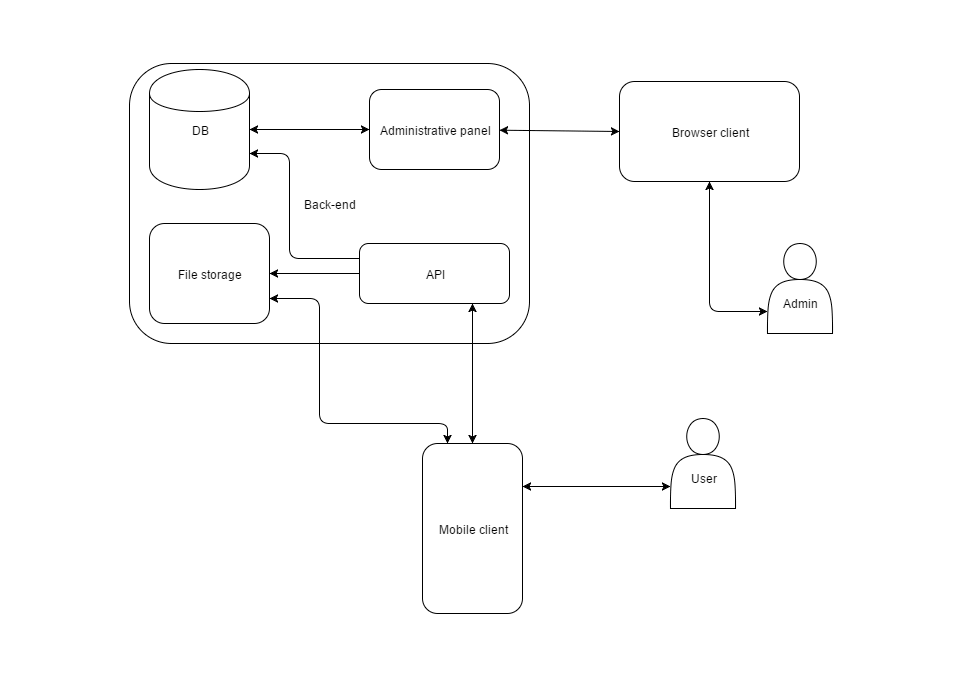


Рисунок 4.1. - Архитектура приложения

1. Экранные формы приложения
2. Главный экран приложения представлен на рисунке 5.1

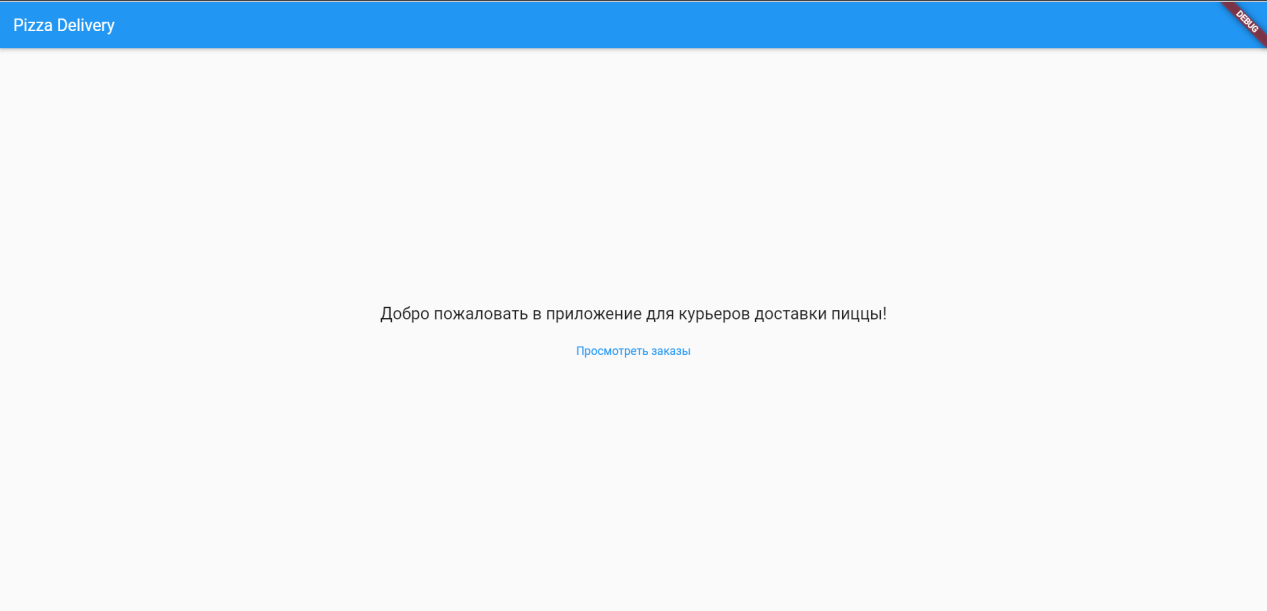


Рисунок 5.1 - Главная страница приложения

По кнопке «Просмотреть заказы» приложение переходит к списку заказов, который представлен на рисунке 5.2.

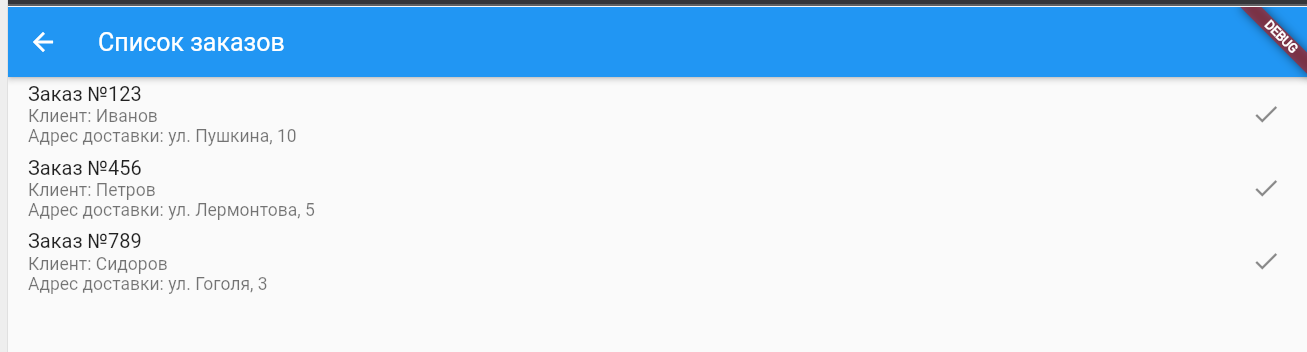
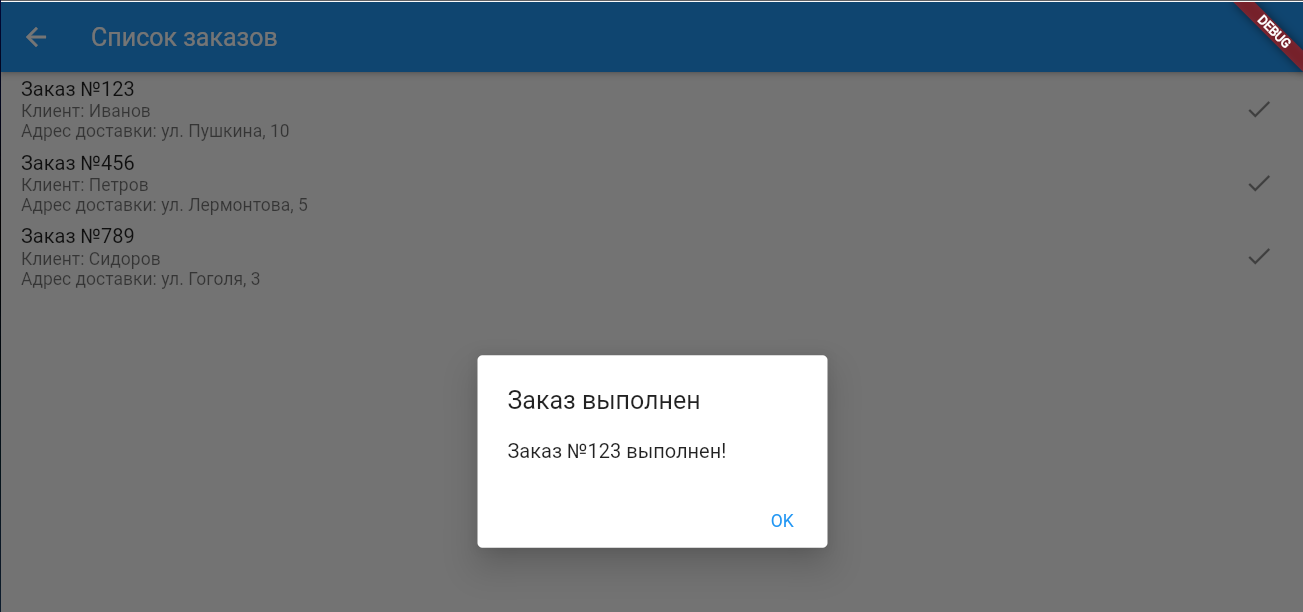


Рисунок 5.2 - Список заказов

При нажатии на кнопку в виде «галочки» должно вывести подтверждение, что заказ выполнен. Это представлено на рисунке 5.3.

  
Рисунок 5.3 - Подтверждение о выполнении

При нажатии на клиента, выводится виджет с картой на которой отображен адрес этого клиента. Это предсталено на рисунке 5.4.

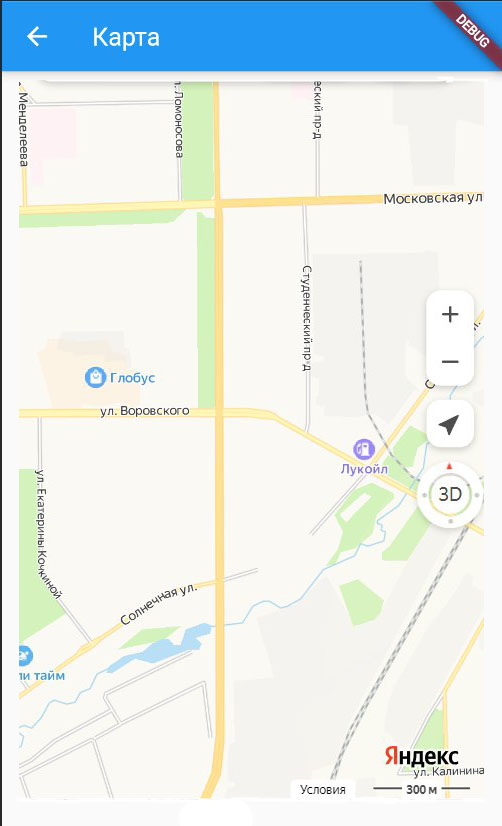


Рисунок 5.4 - Виджет с картой