Попова Исдо-34  
Аналитика  
  
1. Процесс заказа в мобильном приложении:

[Открыть приложение] --> [Выбрать товары]

[выбрать товары] --> [ Добавить в корзину]

[Добавить в корзину] -> [Перейти к оформлению заказа]

[Перейти к оформлению заказа] -> [ Ввести адрес доставки]

[ввести адрес доставки] --> [Выбрать способ оплаты]

[выбрать способ оплаты] -> [Подтвердить заказ]

[Подтвердить заказ] --> [отправить данные на сервер]

[Отправить данные на сервер] --> [Получить подтверждение заказа]

[ Получить подтверждение заказа] -> [Заказ создан]

2. Процесс синхронизации данных между клиентом и сервером включает в себя следующие API методы:

Создание заказа: POST /orders

Параметры:

customer\_id

items[] (массив товаров с их идентификаторами и количеством)

delivery\_address

payment\_method

Ответ:

order\_id

status

редактирование заказа: PUT /orders/{order\_id}

Параметры:

items[] (обновленный массив товаров)

delivery\_address

payment\_method  
order\_data

Ответ:

status

Отмена заказа: DELETE /orders/{order\_id}

Ответ:

status

изменение персональных данных: PUT /customers/{customer\_id}

Параметры:

name

surname

birth\_date

Ответ:

status

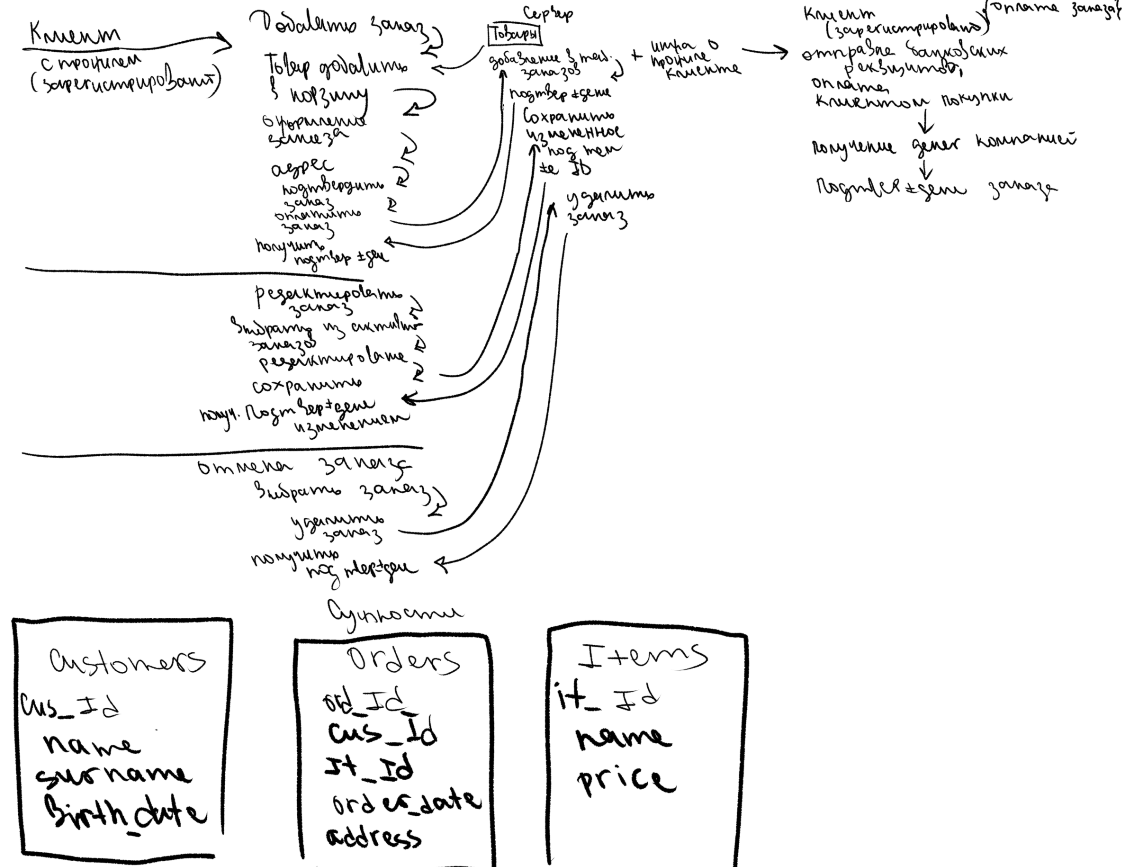
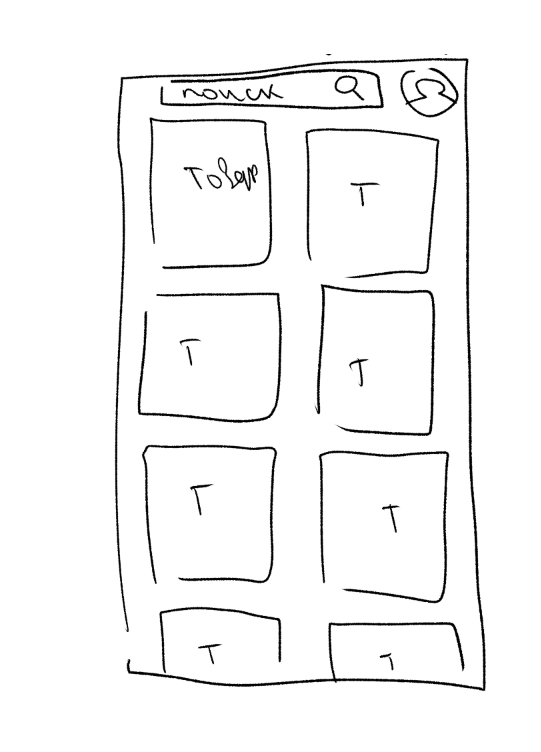
Оплата заказа: POST /payments

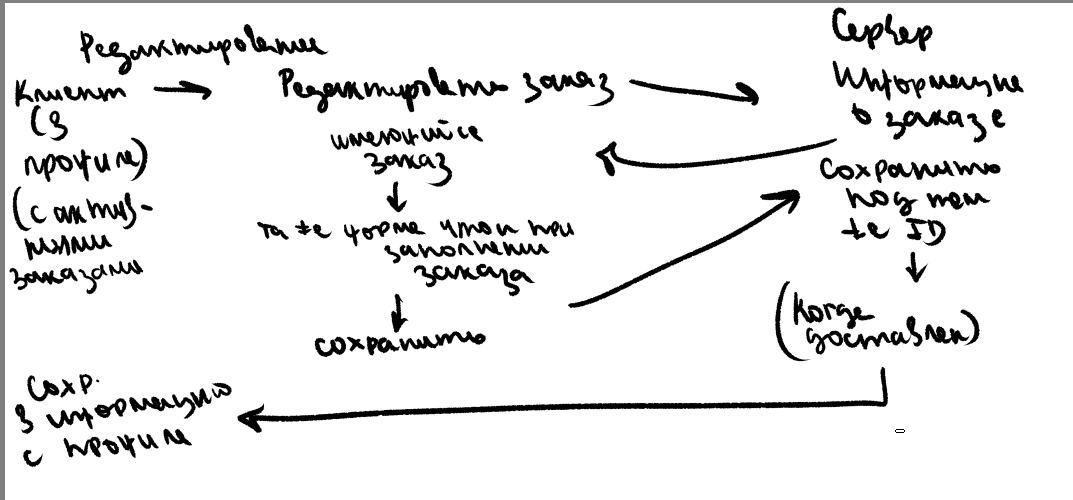
Параметры:

order\_id

payment\_method

Ответ:

payment\_status  
  
  
3. Прототип одного из экранов пользовательского интерфейса   
1. поле ввода ( сверение с name таблиц товаров)   
2. аватарка профиля (при нажатии на нее открывается профиль со списком предыдущих заказов)   
3. заполняющиеся сервером карточки товаров.  


4. Функция редактирования заказа позволяет пользователю изменять детали уже созданного заказа, такие как товары, адрес доставки и способ оплаты.  
  
Используемые API методы

Редактирование заказа: PUT /orders/{order\_id}

Передаваемые параметры

items[]: массив товаров

delivery\_address: новый адрес доставки

Информация о покупках пользователя хранится в его профиле, где каждая запись связывает покупателя с id товара и датой покупки.  
  
5.-- Таблица Customers

CREATE TABLE Customers (

customer\_id INT PRIMARY KEY,

name VARCHAR(50),

surname VARCHAR(50),

birth\_date DATE  
 orderId NEW\_ORDER\_ID,

items JSON\_ARRAY(ITEMS),

deliveryAddress ADDRESS,

);

-- Таблица Items

CREATE TABLE Items (

item\_id INT PRIMARY KEY,

name VARCHAR(50),

price DECIMAL(10, 2)

);

1. SELECT Customers.customerId, Customers.customerName, COUNT(Orders.orderId) AS numberOfPurchases

FROM Customers

LEFT JOIN Orders ON Customers.customerId = Orders.customerId

GROUP BY Customers.customerId, Customers.customerName;

2)SELECT Customers.customerId, Customers.customerName, SUM(OrderItems.price \* OrderItems.quantity) AS totalCost

FROM Customers

INNER JOIN Orders ON Customers.customerId = Orders.customerId

INNER JOIN OrderItems ON Orders.orderId = OrderItems.orderId

GROUP BY Customers.customerId, Customers.customerName

ORDER BY totalCost DESC;  
  
3)SELECT Customers.customerId, Customers.customerName

FROM Customers

INNER JOIN Orders ON Customers.customerId = Orders.customerId

INNER JOIN OrderItems ON Orders.orderId = OrderItems.orderId

GROUP BY Customers.customerId, Customers.customerName

HAVING COUNT(OrderItems.productId) = 1;