**Функция редактирования заказа**

**1. Название функции:** Редактирование заказа.

**2. Цель функции:**

Предоставить пользователю возможность изменять детали существующего заказа, такие как количество товаров, адрес доставки и способ оплаты, а также добавлять или удалять товары из заказа.

**3. Описание функции:**

Функция редактирования заказа позволяет пользователю вносить изменения в уже созданный заказ. Пользователь может редактировать следующие параметры:

— Товары в заказе:

Добавление нового товара,

Удаление существующего товара,

Изменение количества уже добавленных товаров;

— Адрес доставки:

Изменение адреса, на который должен быть доставлен заказ;

— Способ оплаты:

Изменение способа оплаты заказа (с наличных на банковскую карту).

**4. Пользовательский сценарий:**

- Пользователь открывает приложение и переходит в раздел "Мои заказы".

- Пользователь выбирает заказ, который хочет редактировать.

- В открывшемся окне заказа пользователь видит список товаров.

- Пользователь может:

\* Нажать кнопку "Добавить товар", чтобы открыть каталог и выбрать новый товар.

\* Нажать на существующий товар для изменения количества или его удаления.

- Пользователь на странице редактирования заказа находит поле для ввода адреса доставки.

- Пользователь может изменить адрес.

- Пользователь на странице редактирования заказа выбирает новый способ оплаты из доступных вариантов.

- После внесения изменений пользователь нажимает кнопку "Сохранить изменения".

**5. Требования к интерфейсу:**

Экран редактирования заказа:

- Должен содержать список товаров с возможностью редактирования (изменение количества, удаление).

- Должен иметь кнопку "Добавить товар", открывающую каталог товаров для дальнейшего добавления товара.

- Должен иметь поле для ввода адреса доставки.

- Должен иметь выпадающий список или переключаемые между собой кнопки для выбора способа оплаты.

- Должен содержать кнопку "Сохранить изменения", которая отправляет обновленные данные на сервер.

**6. Ожидаемые результаты:**

- Пользователь успешно редактирует заказ, и изменения сохраняются в системе.

- Обновленный заказ отображается в разделе "Мои заказы" с актуальными данными.

- Пользователь получает уведомление о успешном редактировании заказа.

**8. Тестирование функции:**

- Провести тестирование функции редактирования заказа на различных устройствах и версиях ОС.

- Проверить сценарии добавления, удаления и изменения товаров.

- Проверить корректность изменения адреса и способа оплаты.

- Убедиться в обработке ошибок (если товар недоступен).

**Процесс хранения информации о покупках пользователя**

**1. Структура базы данных:**

Для хранения информации о заказах пользователя следует использовать реляционную базу данных. Основные таблицы включают:

— Users:

Идентификатор пользователя – id:

Имя пользователя - name:

Электронная почта - email:

— Orders:

Идентификатор заказа - id:

Идентификатор пользователя - userId:

Адрес доставки - deliveryAddress:

Способ оплаты - paymentMethod:

Время создания заказа - createdAt:

Время последнего обновления заказа - updatedAt:

— OrderItems:

Идентификатор элемента заказа - id:

Идентификатор заказа - orderId:

Идентификатор товара - productId:

Количество товара - quantity:

**2. Процесс хранения:**

— Создание заказа:

При создании заказа информация о заказе и его элементах сохраняется в таблицах Orders и OrderItems.

— Редактирование заказа:

При редактировании заказа обновляются соответствующие записи в таблицах Orders и OrderItems.

Если пользователь добавляет новый товар, создается новая запись в OrderItems.

Если товар удаляется, соответствующая запись в OrderItems удаляется.

— Запрос информации о заказе:

При запросе информации о заказе происходит выборка данных из таблиц Orders и OrderItems по идентификатору заказа.

**Используемые API методы**

1. **Получение заказа:**

Метод: GET

URL: …/api/orders/id

Параметры: Идентификатор заказа id: (параметр в URL)

1. **Редактирование заказа:**

Метод: PUT

URL: …/api/orders/id

Параметры: Идентификатор заказа id: (параметр в URL)

