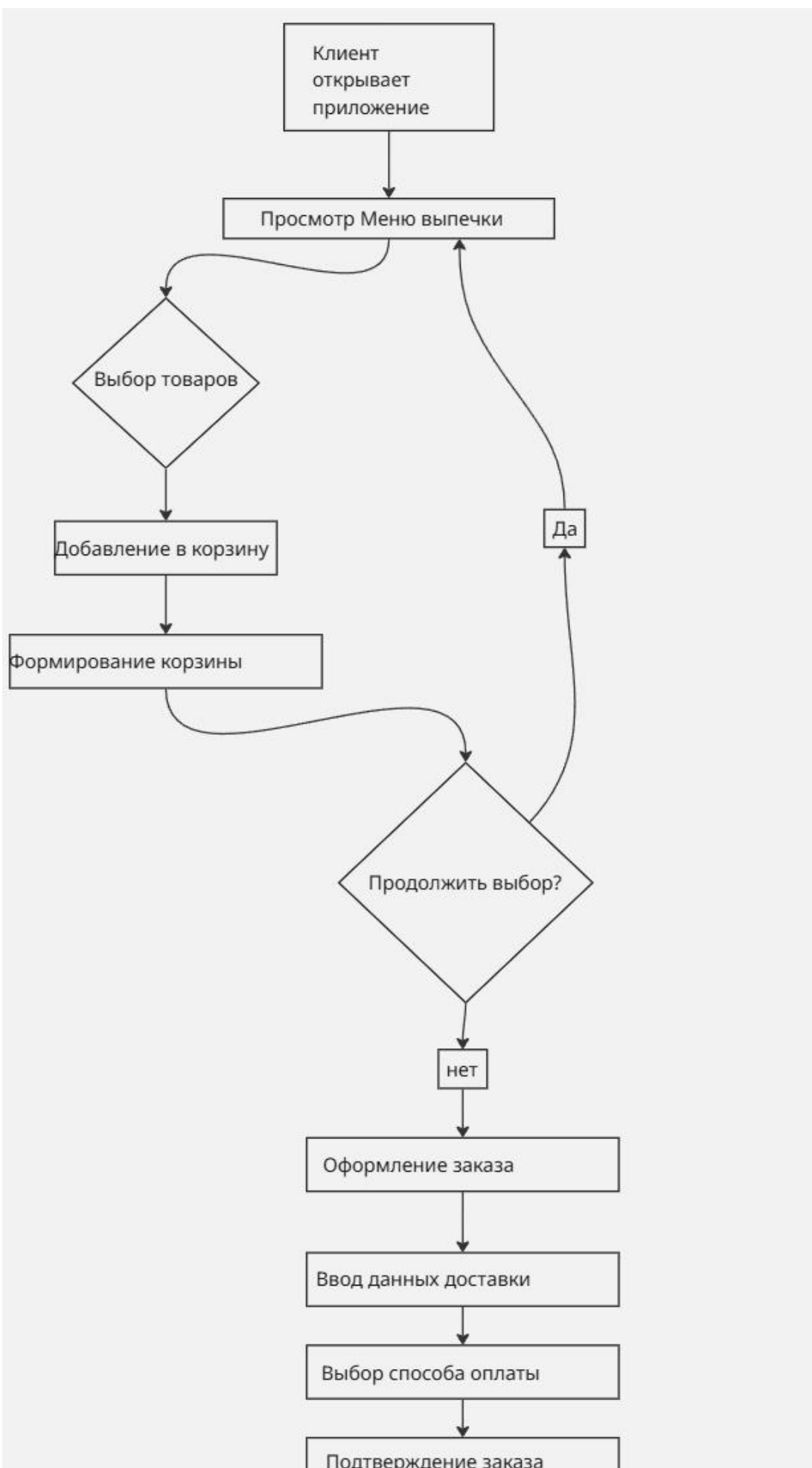


Практика

Задание: Решение задания по мобильному приложению "Частная пекарня"

1. Бизнес-процесс создания заказа



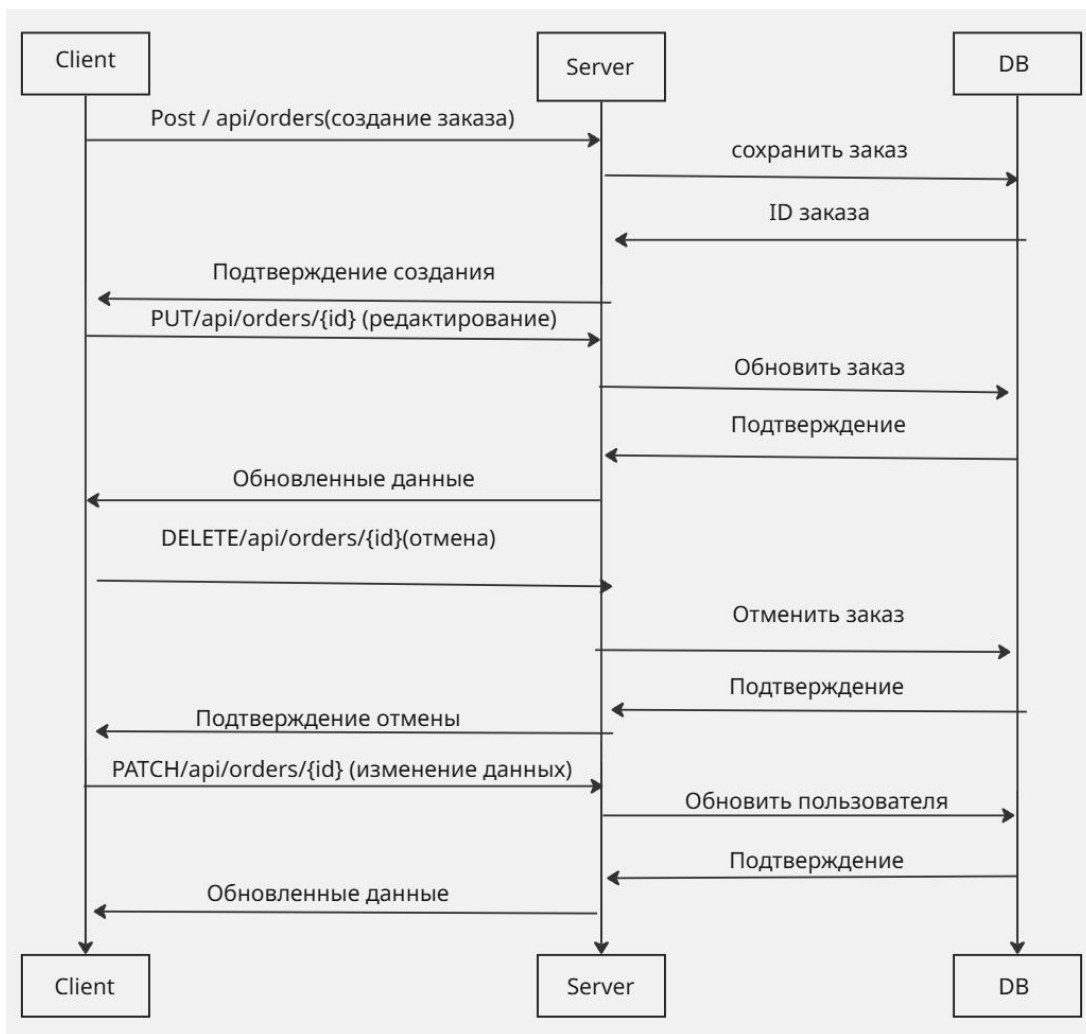
Бизнес-процесс создания заказа — это совокупность последовательных шагов и действий, необходимых для формирования и обработки заказов клиентов от момента подачи заявки до полного исполнения и закрытия сделки. Этот процесс состоит из нескольких ключевых этапов, каждый из которых играет важную роль в обеспечении точности, эффективности и удовлетворенности клиента.

Основные этапы бизнес-процесса создания заказа

2. Процесс синхронизации данных

Процесс синхронизации данных — это процедура обмена информацией между двумя или несколькими источниками данных таким образом, чтобы данные были согласованы и идентичны во всех системах. Этот процесс позволяет поддерживать актуальность и целостность данных в разных устройствах, приложениях и базах данных.

UML диаграмма последовательности синхронизации:



3. Прототип экрана создания заказа

Экран "Оформление заказа":

Верхняя панель:

Название экрана "Оформление заказа"

Кнопка "Назад"

Секция "Ваш заказ":

Список выбранных товаров с изображением, названием, количеством и ценой

Итоговая сумма заказа

Промокод (поле ввода)

Секция "Данные доставки":

Поле "Адрес" (с автозаполнением)

Поле "Комментарий курьеру"

Выбор времени доставки (селектор)

Секция "Оплата":

Варианты оплаты (карта, наличные, онлайн)

Для карты - форма ввода данных

Кнопка "Подтвердить заказ"

4. Функция редактирования заказа

Описание:

Пользователь может редактировать заказ, пока он находится в статусе "В обработке". Доступные действия:

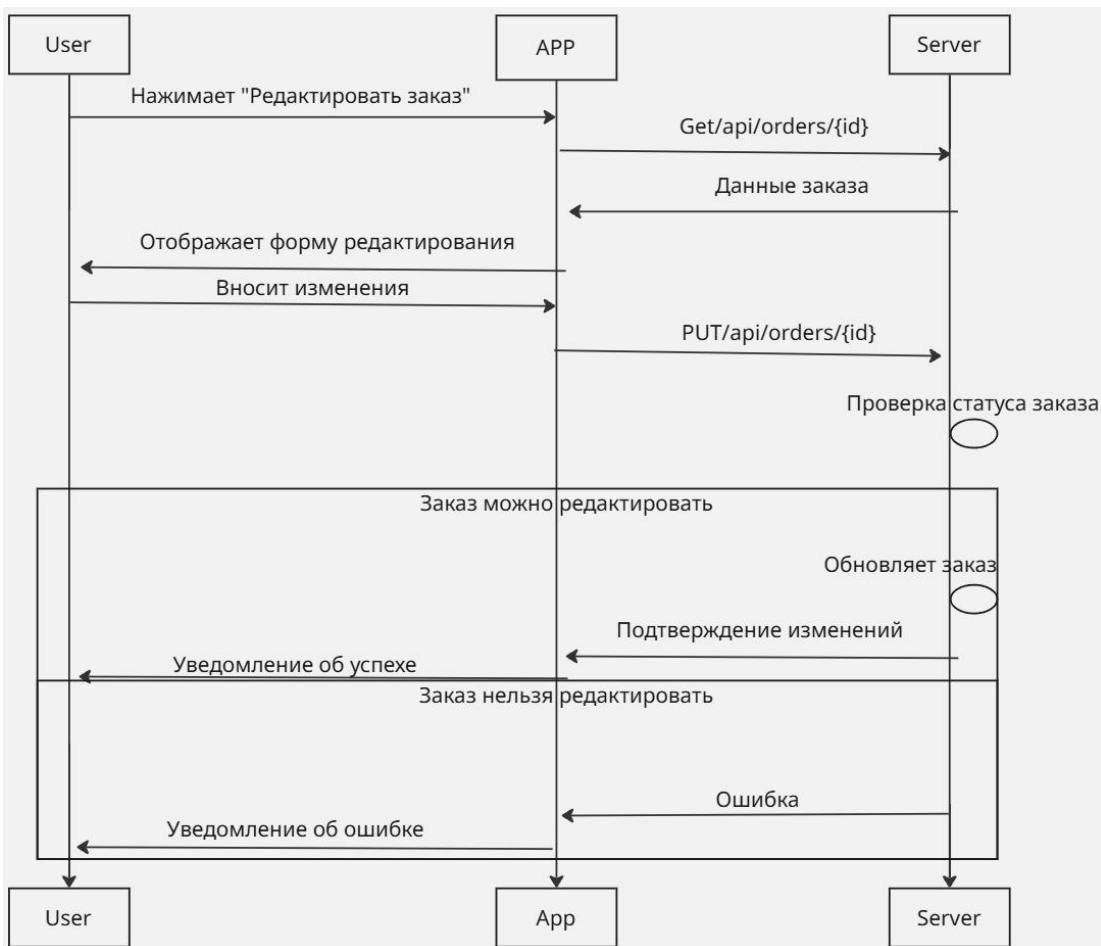
Добавить/удалить товары

Изменить количество товаров

Изменить данные доставки

Изменить способ оплаты

UML диаграмма:



5. SQL-запросы

Покупатели с количеством покупок:

sql

```
SELECT u.id, u.name, COUNT(o.id) as purchase_count
```

```
FROM users u
```

```
LEFT JOIN orders o ON u.id = o.user_id
```

```
GROUP BY u.id, u.name;
```

Общая стоимость товаров по покупателям:

sql

```
SELECT u.id, u.name, SUM(o.total) as total_spent
```

```
FROM users u
```

JOIN orders o ON u.id = o.user_id

GROUP BY u.id, u.name

ORDER BY total_spent DESC;

Покупатели с одной покупкой:

sql

SELECT u.id, u.name

FROM users u

JOIN (

SELECT user_id, COUNT(id) as order_count

FROM orders

GROUP BY user_id

HAVING COUNT(id) = 1

) single_order ON u.id = single_order.user_id;