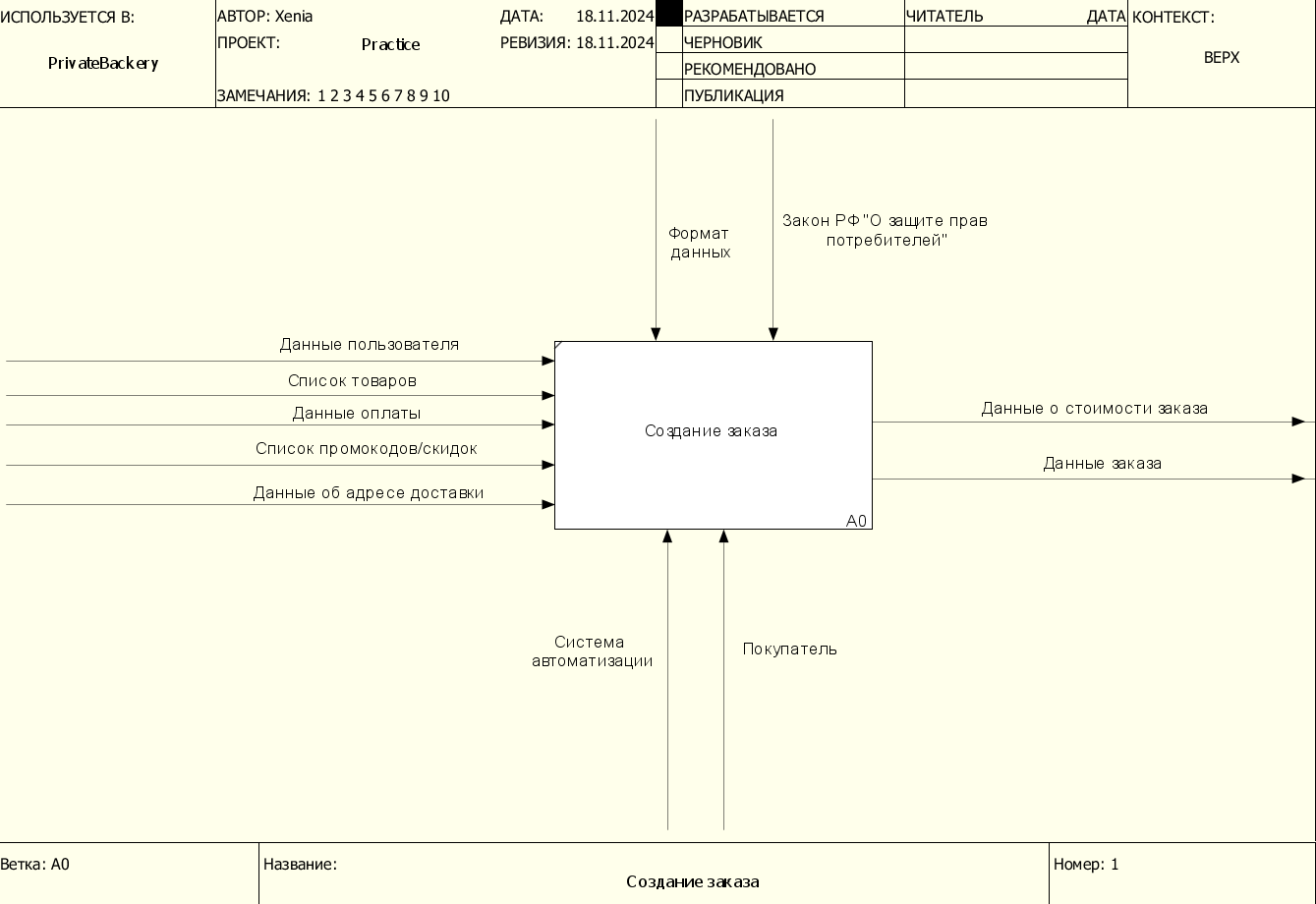
**Неваева Ксения Алексеевна ПИбд-42**

# **Дано:**

Мобильное клиент-серверное приложение "Частная пекарня" (приложение для продажи выпечки)

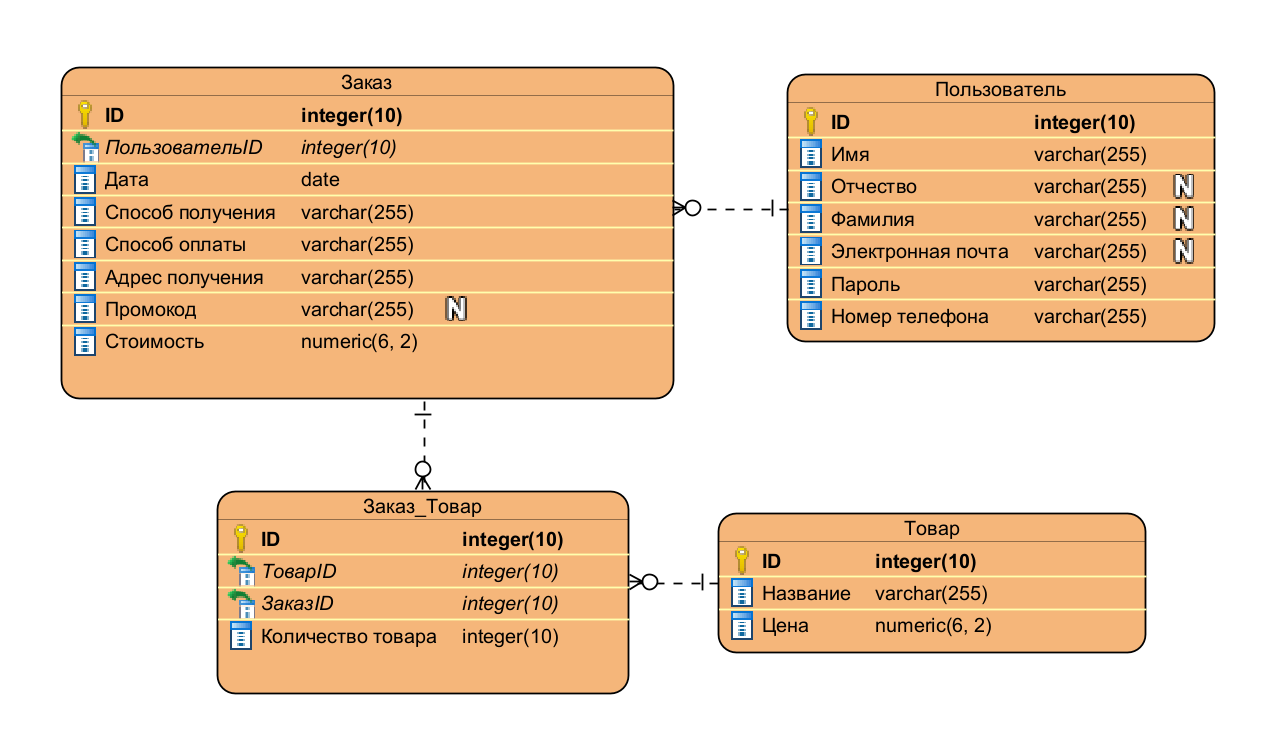
# **Необходимо описать:**

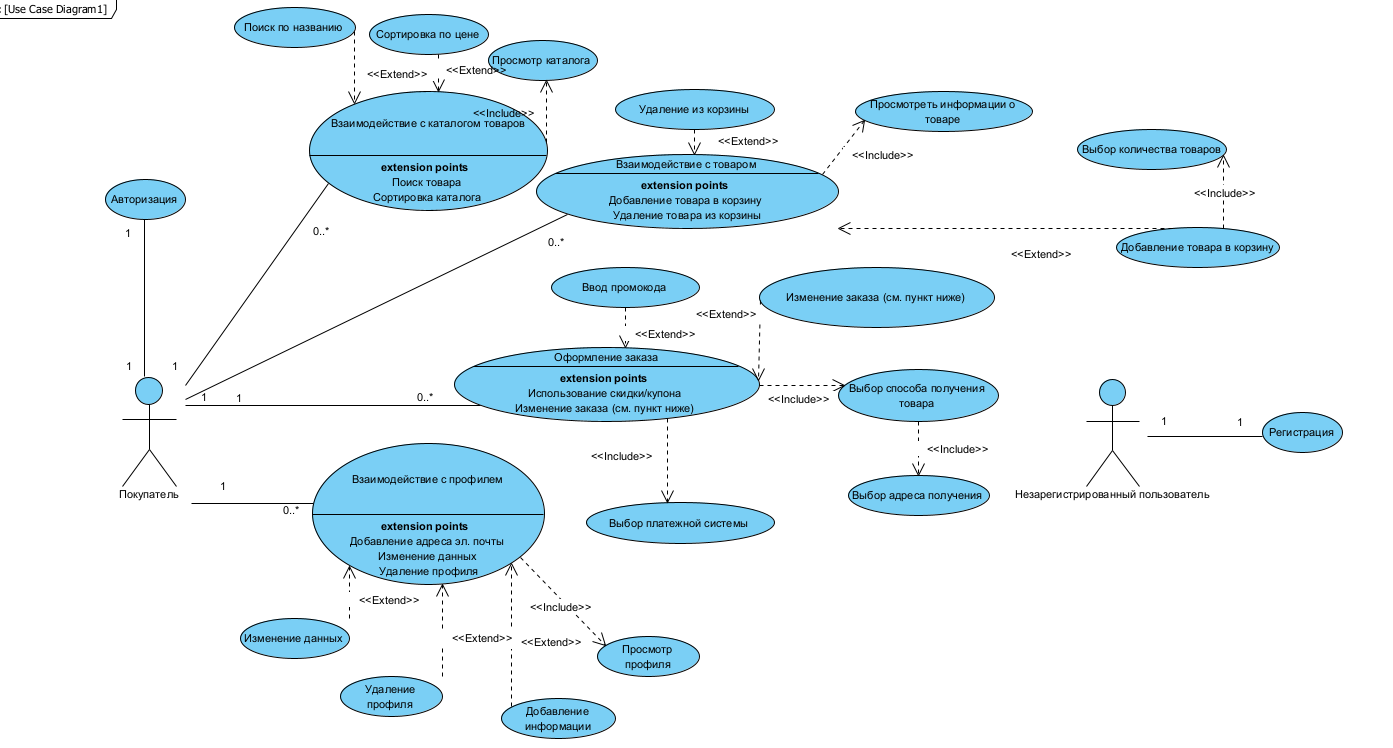
1. Бизнес-процесс создания заказа (использовать любую удобную нотацию моделирования бизнес-процессов).

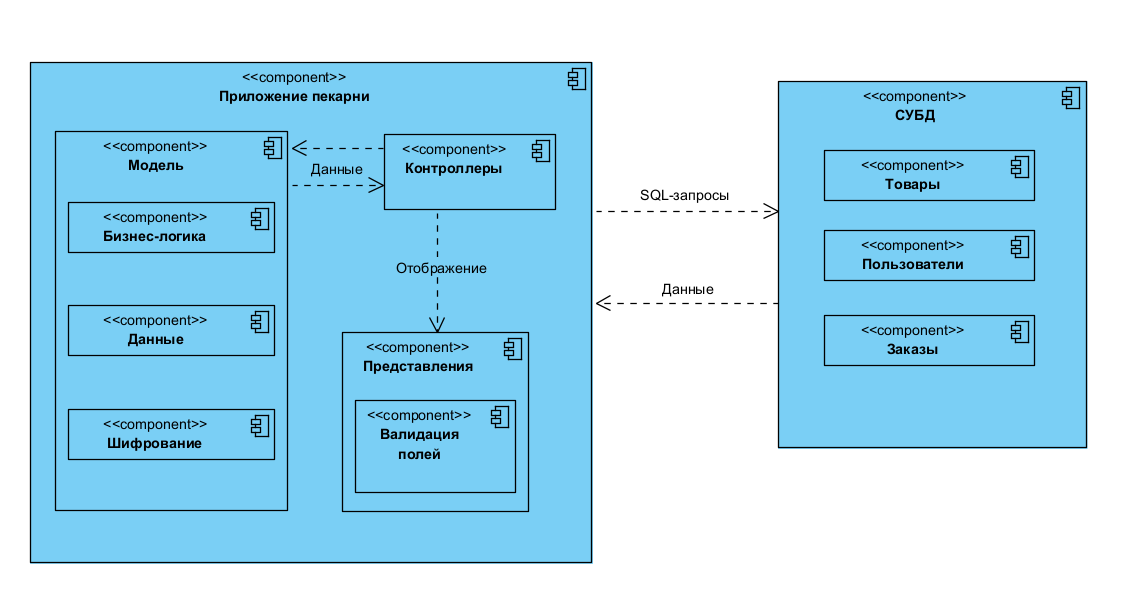


1. Опишите процесс синхронизации данных между клиентом и сервером

(создание, редактирование и отмена заказа, изменение персональных данных, оплата заказа и т.д.). Представить все в диаграммах UML, API методах и других представлениях, также составить ER-диаграмму сущностей.







1. Регистрация

— GET-метод для получения формы регистрации.

— POST-метод для добавления данных пользователя в БД. В параметрах могут передаваться переменные типа string для логина/электронной почты, пароля, номера телефона.

2. Авторизация

— GET-метод для получения формы авторизации.

— POST-метод для создания сессии пользователя. В параметрах могут передаваться переменные типа string для логина/электронной почты/номера телефона и пароля.

3. Взаимодействие с каталогом

— GET-метод для получения списка товаров.

— GET-метод для сортировки списка товаров. Тип сортировки зависит от кнопок/пунктов выпадающего списка с типами сортировки.

— GET-метод для поиска товара. В параметрах можно передать содержимое поля для поиска, которое будет сравниваться с названиями товаров в каталоге.

— POST-методы для добавления товара в корзину или избранное. В обоих случаях передается id пользователя (т.к. корзина и избранные привязаны к пользователю) и id товара.

— GET-метод для перехода на страницу товара. Передается id товара.

4. Заказ

Оформление:

— GET-метод для получения списка покупаемых товаров. Передается id пользователя, id корзины товаров.

— GET-метод для получения данных пользователя. Передается id пользователя.

— GET-метод для получения реквизитов привязанной к аккаунту карты. Передается id пользователя.

— POST-метод для оформления заказа. Передаются id пользователя, список товаров с их стоимостью и количеством, способ доставки, адрес доставки, способ получения, способ оплаты, примененные скидки и промокоды. Номер, дата заказа генерируются автоматически, стоимость заказа представляет собой сумму стоимостей товаров с учетом скидок и акций. Оплата списывается сразу, при условии оплаты картой или СБП.

Редактирование:

— PATCH-метод для редактирования заказа. Передаются номер заказа и информация измененных полей.

5. Профиль

— PATCH-метод для редактирования данных. Параметры: id пользователя, измененные данные.

— GET-метод для получения данных пользователя.

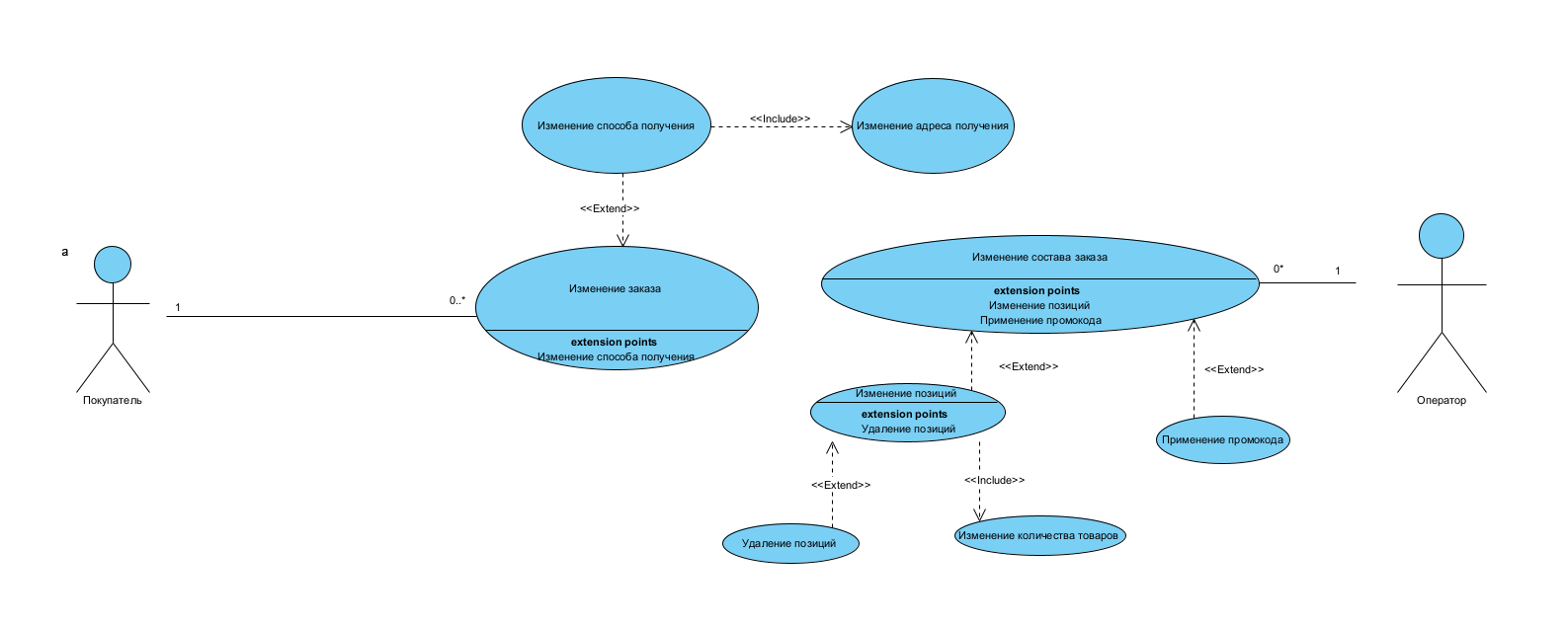
— POST-метод для удаления аккаунта. Параметры: id пользователя.

1. Подготовить прототип одного из экранов данного мобильного приложения и описать пользовательский интерфейс для данного экрана (например, создание заказа).

Прототип доступен по ссылке:

<https://www.figma.com/design/H6Y5sWehgB2O1prbUHTOtw/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D1%82%D0%B8%D0%BF-%D0%BF%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%BA%D0%B0?node-id=0-1&t=K8ep1a3Z2zv5t2Sx-1>

1. Подготовить подробное описание функции редактирования заказа , которую можно было бы использовать в качестве постановки задачи для разработки (помимо текстового описания, использовать UML диаграммы, указать используемые API методы, передаваемые и получаемые параметры, описать процесс хранения информации о покупках пользователя).



Описание:

Можно изменить следующие параметры в заказе: способ получения (самовывоз/доставка) => адрес доставки, промокод (в случае, если покупатель забыл его указать), изменить состав заказа и ввести промокод, в случае если покупатель забыл сделать это в процессе оформления. Целесообразно оставить функционал, связанный с изменением заказа и его цены на оператора поддержки.

Перед началом изменения деталей заказа:  
 GET для получения заказа.

Параметры: id заказа.

Получаем объект класса "Заказ".

1. В изменении заказа покупателем можно поменять адрес и способ доставки.

GET для получения параметра "Способ получения" объекта "Заказ". Параметры: id заказа.

PATCH для обновления поля объекта.Параметры: название способа получения. Получаем сообщение об успехе или неудаче обновления поля.

GET для получения параметра "Адрес получения" объекта "Заказ". Параметры: id заказа. PATCH для обновления поля объекта. Параметры: строка с адресом. Получаем сообщение об успехе или неудаче обновления поля.

1. Работа с промокодами (оператор)

GET для получения списка действующих скидок. В параметрах ничего не передается. Получаем список скидок и промокодов.

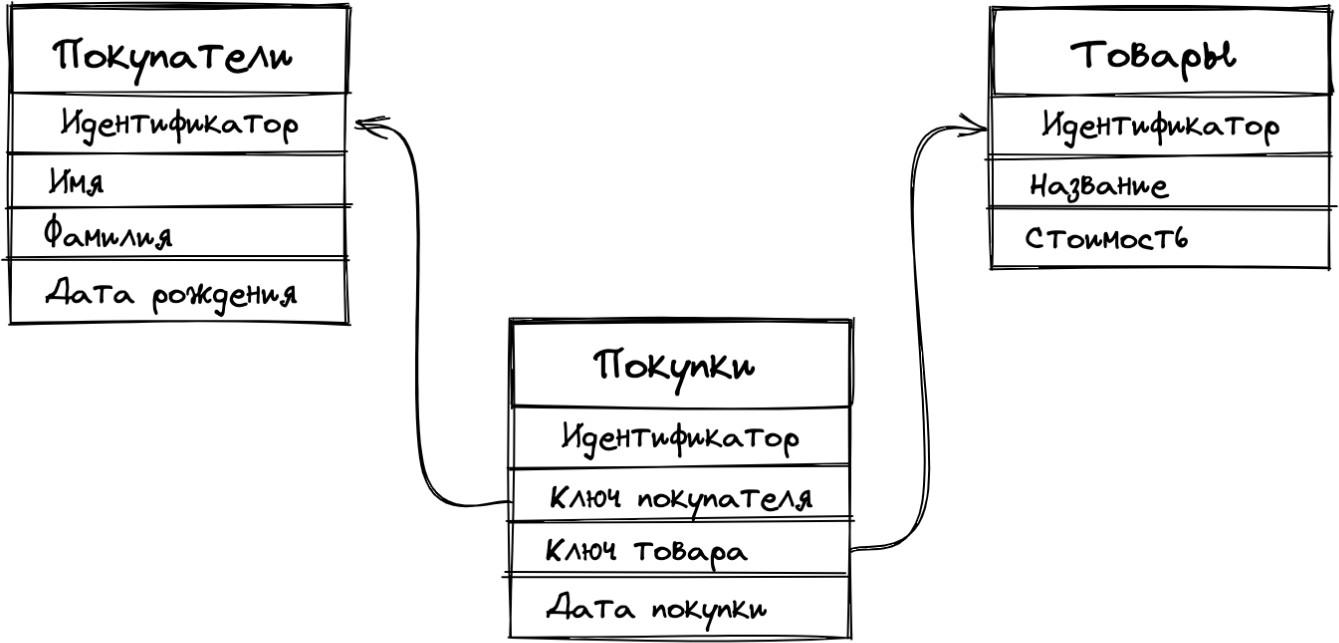
PATCH для обновления поля "Промокод" и поля "Цена" у товаров в заказе и/или общей стоимости заказа. В параметрах передаются список скидок/купонов, строка с введенным купоном/скидкой и список товаров с ценами.Получаем список товаров с обновленными ценами и обновленную стоимость заказа.

1. Изменение состава заказа (например, при отсутствии позиций).

GET для получения списка позиций с количеством. В параметрах id заказа. Получаем список позиций с количеством товаров.

PATCH для обновления списка позиций и/или их количества. В параметрах передаются список позиций с обновленными количествами (в том числе 0, если позиция была удалена. Получаем список позиций с обновленным количеством для каждого товара и обновленную стоимость заказа.

1. SQL-запросы.



1. Вывести покупателей с количеством осуществленных покупок:

SELECT Покупатели.\*,

(

SELECT COUNT(\*)

FROM Покупки

WHERE Покупки.ключ\_покупателя = Покупатель.идентификатор

) as "Количество осуществленных покупок"

FROM Покупатели;

1. Общую стоимость товаров для каждого покупателя и отсортировать результат в порядке убывания:

SELECT Покупатели.\*,

(

SELECT SUM(Товары.стоимость)

FROM Покупки

JOIN Товары ON Покупки.ключ\_товара = Товары.идентификатор

WHERE Покупки.ключ\_покупателя = Покупатели.идентификатор

) as сумма\_стоимостей\_товаров

FROM Покупатели

ORDER BY сумма\_стоимостей\_товаров DESC;

1. Получить покупателей, купивших только один товар:

SELECT Покупатели.\*,

(

SELECT COUNT(\*)

FROM Покупки

WHERE Покупки.ключ\_покупателя = Покупатели.идентификатор

) as количество\_покупок

FROM Покупатели

HAVING количество\_покупок = 1;