Основной сценарий:

Инициация покупки: Пользователь нажимает кнопку "Купить" или "Добавить в корзину" для выбранного товара.

Добавление в корзину (опционально):

* Если используется корзина, товар добавляется в корзину пользователя.
* Пользователь может продолжить просмотр и добавление других товаров.

Переход к оформлению заказа: Пользователь переходит в корзину или непосредственно к оформлению заказа (если покупка в один клик).

Просмотр корзины (опционально):

* Пользователь видит список товаров в корзине, их количество, цену и общую стоимость.
* Пользователь может изменить количество товаров или удалить товары из корзины.

Оформление заказа:

* Пользователь подтверждает список товаров для покупки.
* Система запрашивает необходимую информацию:
* Адрес доставки: Для физических товаров.
* Способ доставки: Выбор из доступных вариантов (если применимо).
* Способ оплаты: Выбор из доступных вариантов (например, кредитная карта, электронный кошелек и т.д.).
* Платежные данные: Ввод данных для выбранного способа оплаты.
* Контактная информация: Электронная почта и/или номер телефона для уведомлений о заказе.

Подтверждение заказа:

* Система отображает сводку заказа, включая все выбранные товары, общую стоимость, адрес доставки и способ оплаты.
* Пользователь подтверждает заказ.

Обработка платежа:

* Система инициирует процесс обработки платежа через выбранный платежный шлюз.
* Успешный платеж: Система получает подтверждение успешного платежа.
* Неудачный платеж: Система уведомляет пользователя о неудачном платеже и предлагает повторить попытку или выбрать другой способ оплаты.

Создание заказа:

* После успешного платежа система создает запись о заказе в базе данных.
* Заказ включает в себя:
* Идентификатор пользователя.
* Список купленных товаров (с количеством и ценой на момент покупки).
* Общую стоимость заказа.
* Дату и время заказа.
* Адрес доставки.
* Способ доставки.
* Способ оплаты.
* Статус заказа (например, "Ожидает обработки", "Обрабатывается", "Отправлен", "Завершен", "Отменен").

Уменьшение запасов: Система уменьшает количество купленных товаров на складе.

Уведомление пользователя:

* Система отправляет пользователю подтверждение заказа на указанный адрес электронной почты и/или номер телефона.
* Уведомление может содержать детали заказа, номер заказа и информацию о дальнейших шагах.

Завершение процесса: Функция processPurchase завершает свою работу, возвращая статус успешного или неудачного выполнения.

Возможные сценарии:

* **Товар отсутствует на складе: Если товар, который пользователь хочет купить, отсутствует на складе, система должна уведомить пользователя и предложить альтернативы (например, подождать поступления, выбрать другой товар).**
* **Ошибка платежа: Если платеж не удался, система должна предоставить пользователю возможность повторить попытку, выбрать другой способ оплаты или отменить заказ.**
* **Отмена заказа пользователем: Пользователь может иметь возможность отменить заказ до определенного момента (например, до начала обработки заказа). Система должна обрабатывать отмену заказа и возвращать средства (если применимо).**

**Пример кода SQL по хранению информации о покупках пользователя**

**CREATE** **TABLE** Categories (

category\_id **INT** **PRIMARY** KEY AUTO\_INCREMENT,

category\_name **VARCHAR**(255) **NOT** **NULL**,

description TEXT

);

**CREATE** **TABLE** Products (

product\_id **INT** **PRIMARY** KEY AUTO\_INCREMENT,

category\_id **INT**,

product\_name **VARCHAR**(255) **NOT** **NULL**,

description TEXT,

price **DECIMAL**(10, 2) **NOT** **NULL**,

stock\_quantity **INT** **NOT** **NULL**,

image\_url **VARCHAR**(255),

**FOREIGN** KEY (category\_id) **REFERENCES** Categories(category\_id)

);

**CREATE** **TABLE** Users (

user\_id **INT** **PRIMARY** KEY AUTO\_INCREMENT,

username **VARCHAR**(50) **UNIQUE** **NOT** **NULL**,

email **VARCHAR**(255) **UNIQUE** **NOT** **NULL**,

registration\_date **TIMESTAMP** **DEFAULT** CURRENT\_TIMESTAMP,

phone\_number **VARCHAR**(20),

address JSON

*-- ... (другие данные пользователя) - добавьте другие поля пользователя по необходимости*

);

**CREATE** **TABLE** Orders (

order\_id **INT** **PRIMARY** KEY AUTO\_INCREMENT,

user\_id **INT**,

order\_date **TIMESTAMP** **DEFAULT** CURRENT\_TIMESTAMP,

total\_amount **DECIMAL**(10, 2) **NOT** **NULL**,

shipping\_address JSON,

payment\_method **VARCHAR**(255),

order\_status ENUM('Ожидает обработки', 'Обрабатывается', 'Отправлен', 'Завершен', 'Отменен') **NOT** **NULL**,

**FOREIGN** KEY (user\_id) **REFERENCES** Users(user\_id)

*-- ... (детали заказа) - добавьте другие детали заказа по необходимости*

);

**CREATE** **TABLE** Order\_Items (

order\_item\_id **INT** **PRIMARY** KEY AUTO\_INCREMENT,

order\_id **INT**,

product\_id **INT**,

quantity **INT** **NOT** **NULL**,

price\_at\_purchase **DECIMAL**(10, 2) **NOT** **NULL**,

**FOREIGN** KEY (order\_id) **REFERENCES** Orders(order\_id),

**FOREIGN** KEY (product\_id) **REFERENCES** Products(product\_id)

);