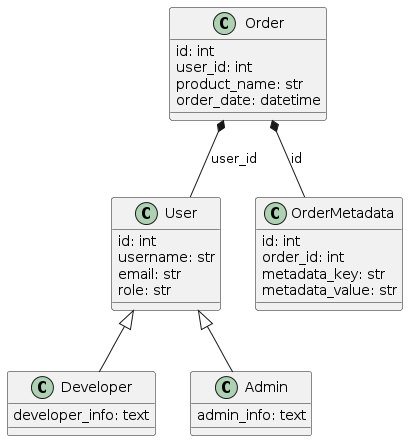
**Все сделано в PlantUML**

**Диаграмма классов**



Коммент:

Конечно, давайте разберем каждый класс на диаграмме:

1. **User**: Этот класс представляет пользователя системы. У него есть атрибуты id, username, email и role, которые характеризуют пользователя.

2. **Developer**: Этот класс является подклассом класса User и добавляет атрибут developer\_info, который, вероятно, содержит информацию о разработчике.

3. **Admin**: Этот класс также является подклассом класса User и имеет атрибут admin\_info, который, вероятно, содержит информацию о администраторе.

4. **Order**: Этот класс представляет заказ пользователя. У него есть атрибуты id, user\_id, product\_name и order\_date, которые описывают заказ.

5. **OrderMetadata**: Этот класс содержит метаданные для заказа. У него есть атрибуты id, order\_id, metadata\_key и metadata\_value, которые позволяют хранить дополнительную информацию о заказе.

На диаграмме также показаны связи между классами:

- Связь между классами User и Developer указывает на то, что класс Developer является подклассом класса User.

- Связь между классами User и Admin указывает на то, что класс Admin также является подклассом класса User.

- Связь между классами Order и User показывает, что у заказа есть связь с пользователем через атрибут user\_id.

- Связь между классами Order и OrderMetadata показывает, что у заказа может быть связанная информация в виде метаданных.

По диаграмме нейросеть сгенерила код, он, по большей части — ОК

class User:

def \_\_init\_\_(self, id, username, email, role):

self.id = id

self.username = username

self.email = email

self.role = role

class Developer(User):

def \_\_init\_\_(self, id, username, email, role, developer\_info):

super().\_\_init\_\_(id, username, email, role)

self.developer\_info = developer\_info

class Admin(User):

def \_\_init\_\_(self, id, username, email, role, admin\_info):

super().\_\_init\_\_(id, username, email, role)

self.admin\_info = admin\_info

class Order:

def \_\_init\_\_(self, id, user\_id, product\_name, order\_date):

self.id = id

self.user\_id = user\_id

self.product\_name = product\_name

self.order\_date = order\_date

class OrderMetadata:

def \_\_init\_\_(self, id, order\_id, metadata\_key, metadata\_value):

self.id = id

self.order\_id = order\_id

self.metadata\_key = metadata\_key

self.metadata\_value = metadata\_value

# Пример использования классов

user1 = User(1, "Alice", "alice@example.com", "user")

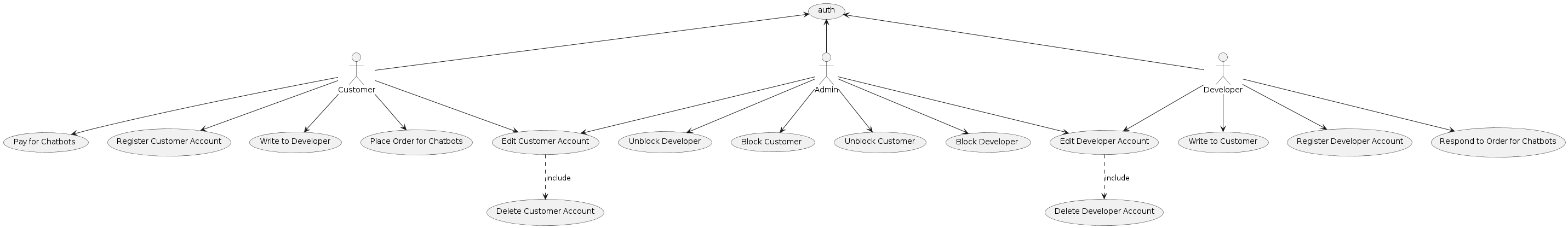
developer1 = Developer(2, "Bob", "bob@example.com", "developer", "Developer Info")

admin1 = Admin(3, "Charlie", "charlie@example.com", "admin", "Admin Info")

order1 = Order(1, user1.id, "Product A", "2022-01-01")

order\_metadata1 = OrderMetadata(1, order1.id, "key1", "value1")

**Диаграмма вариантов использования**

1. Админ, заказчик и девелопер сначала проходят аутентификацию (auth) для доступа к системе.

2. Администратор может выполнять следующие действия:

- Редактировать учетную запись заказчика (Edit Customer Account)

- Блокировать заказчика (Block Customer)

- Разблокировать заказчика (Unblock Customer)

- Редактировать учетную запись девелопера (Edit Developer Account)

- Блокировать девелопера (Block Developer)

- Разблокировать девелопера (Unblock Developer)

3. Заказчик может совершать следующие действия:

- Зарегистрировать учетную запись заказчика (Register Customer Account)

- Редактировать учетную запись заказчика (Edit Customer Account)

- Удалить учетную запись заказчика (Delete Customer Account)

- Написать девелоперу (Write to Developer)

- Разместить заказ на чат-ботов (Place Order for Chatbots)

- Оплатить заказ на чат-ботов (Pay for Chatbots)

4. Девелопер может выполнять следующие действия:

- Зарегистрировать учетную запись девелопера (Register Developer Account)

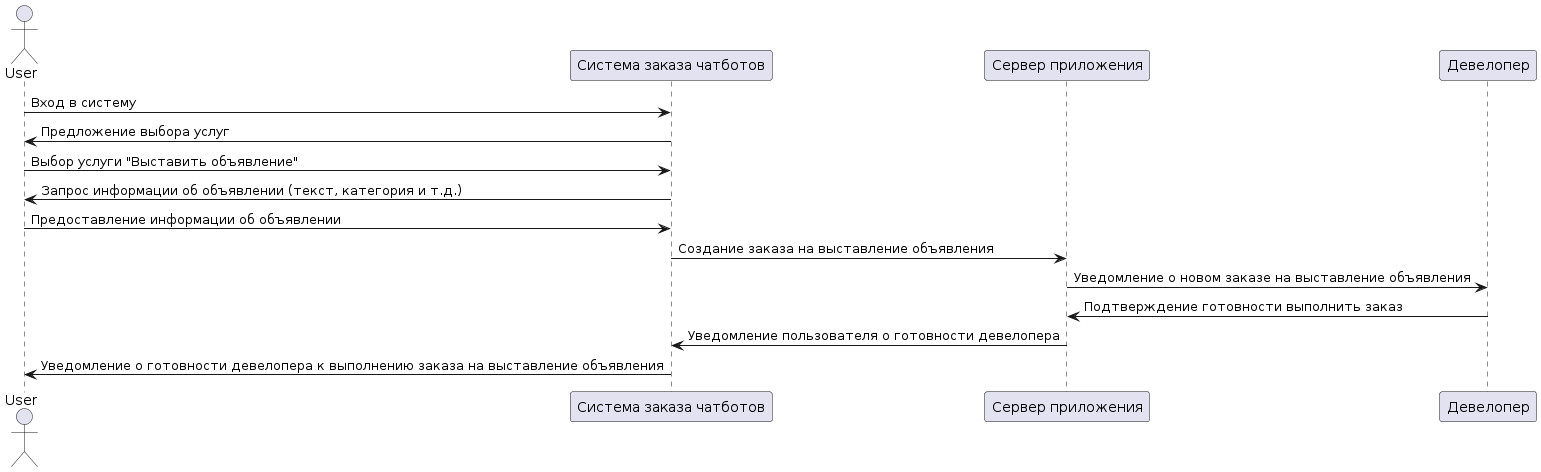
- Редактировать учетную запись девелопера (Edit Developer Account)

- Удалить учетную запись девелопера (Delete Developer Account)

- Откликнуться на заказ на чат-ботов (Respond to Order for Chatbots)

- Написать заказчику (Write to Customer)

**Сиквенс диаграмма**

****

Ну тут нечего объяснять…

**Диаграмма активности подключения к БД**



Просто схема как работает БД (конкретно вытаскивание данных о заказах из БД)