МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего образования «Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

Лабораторная работа №0

Тема: Проектирование приложения

Выполнили:

Обучающиеся группы № 6301-010302D

А.Л. Летов

Н.А. Стрельников

Самара 2025

**1. Схема взаимодействия компонентов (сервер, БД, клиент)**

В нашем приложении планируется 4 страницы:

1. авторизация (регистрация);
2. главная страница (страница пользователя), на которой будут отображаться друзья, посты (друзей, а также просмотр и создание своих), фотографии пользователя, также будет возможность добавления друзей и перехода на две другие страницы;
3. страница групп, которая откроется на одной из групп пользователя, справа от которой будет отображаться информация о группе (список участников), а слева переключение между группами;
4. страница сообществ, которая откроется на одном из сообществ; слева планируется вывод информации о сообществе, справа добавление существующего или создание нового;

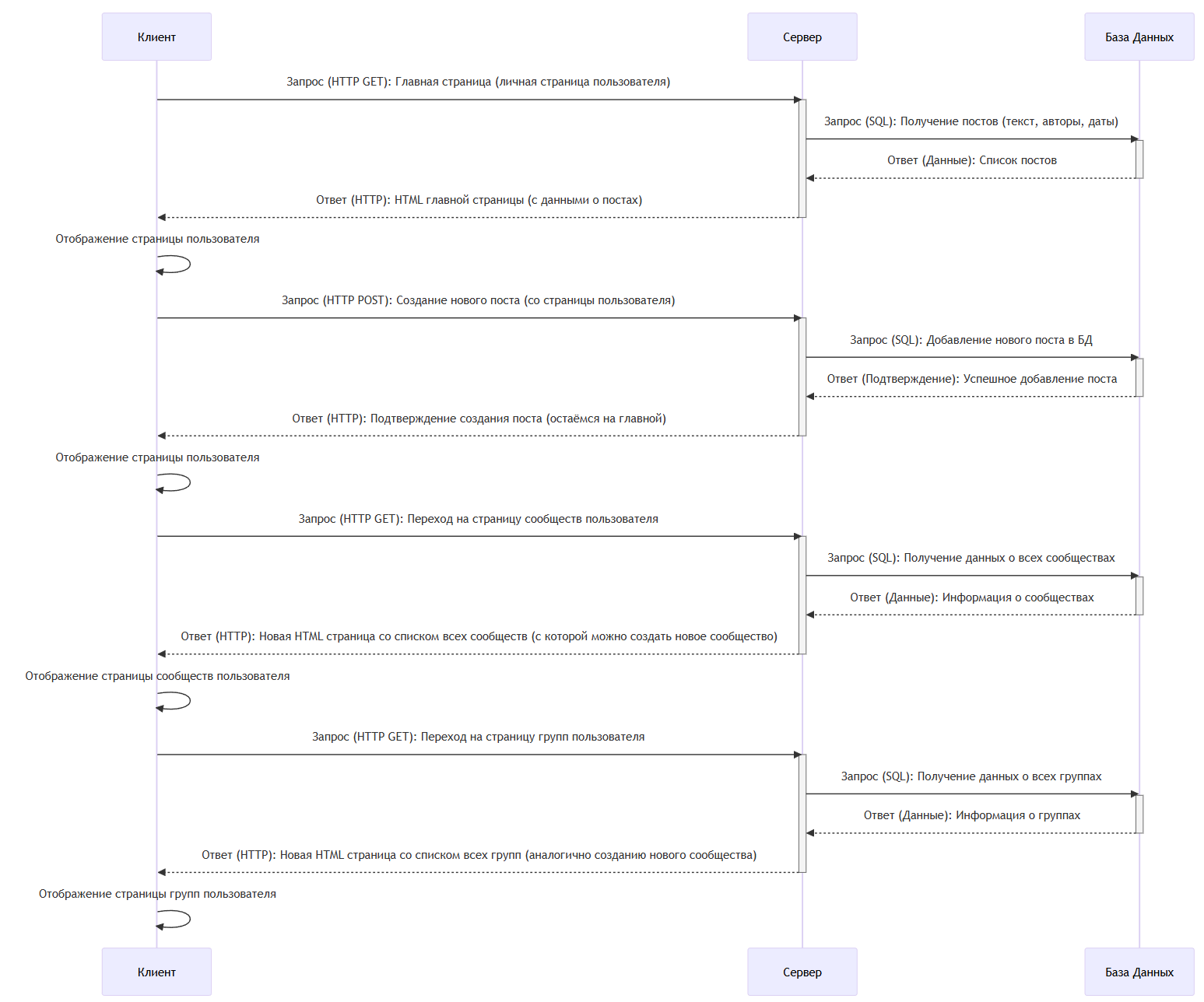
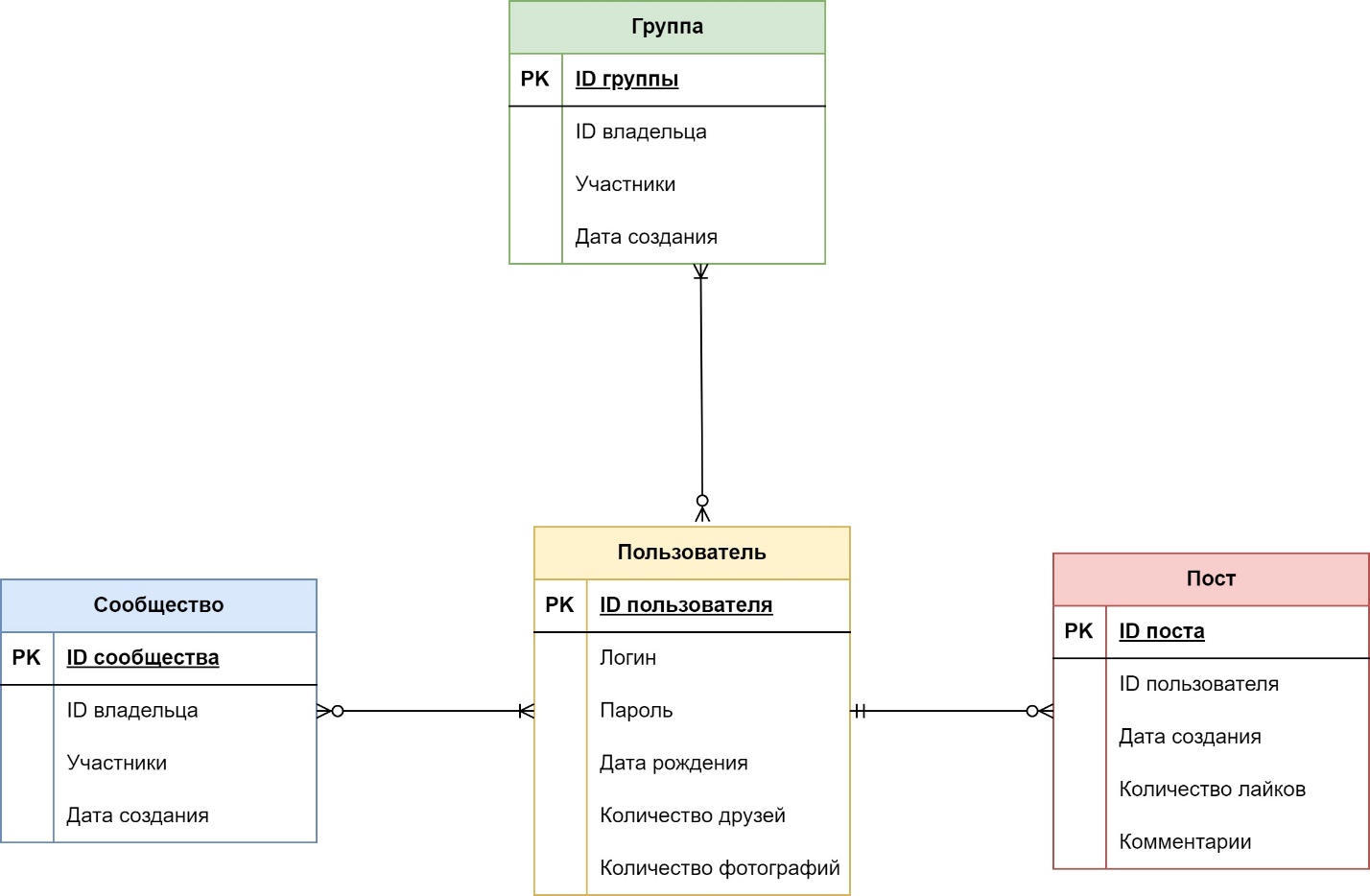


Рисунок 1 – Схема взаимодействия компонентов

**2. Разработка логической схемы базы данных**

В нашей схеме представлено 4 сущности: группа, сообщество, пользователь и пост. Сущности пользователь и сообщество, а также пользователь и группа соединены связью многие ко многим. Между сущностями пользователь и пост связь один ко многим.



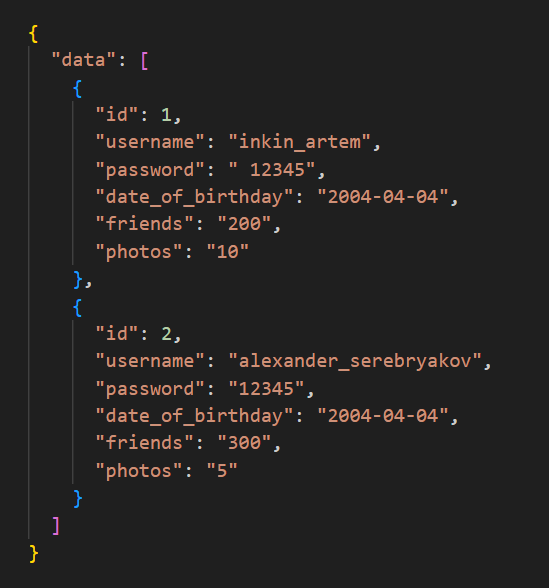
**3. Определение структуры API**

REST API – архитектурный стиль разработки API веб-приложений или компонентов распределённого приложения, используя протокол HTTP. В REST API существуют следующие HTTP методы, используемые для взаимодействия с объектами на серверах: POST (добавление), DELETE (удаление), PUT/PATCH (обновление), GET (получение). Для каждой сущности должен быть определен URL, по которому мы с ней взаимодействуем (/users, /posts, /groups, /communities).

Примеры форматов запросов и ответов:

1. Запрос: GET /users – получение всей информации о пользователях.

Ответ:

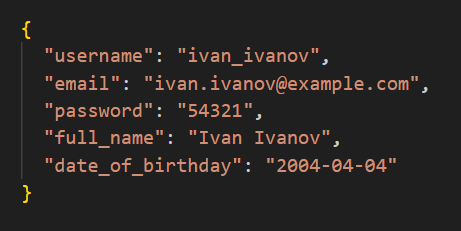


1. Создание нового пользователя.

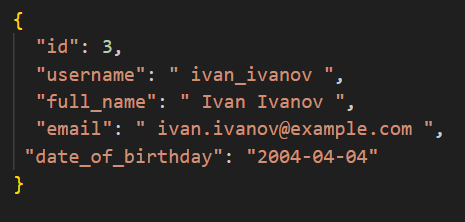
Запрос:

POST /users

Content-Type: application/json



Ответ:

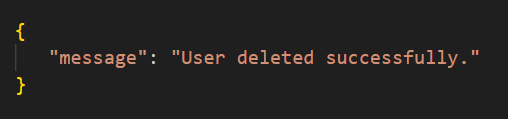


3) Удаление пользователя.

Запрос:

DELETE /users/1

Ответ:

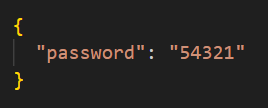


4) Обновить информацию о пользователе.

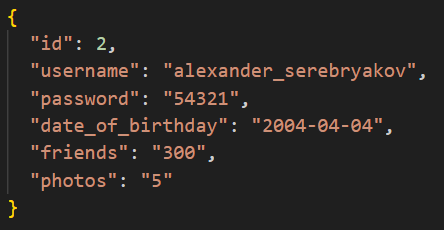
Запрос:

PUT /users/1

Content-Type: application/json



Ответ:

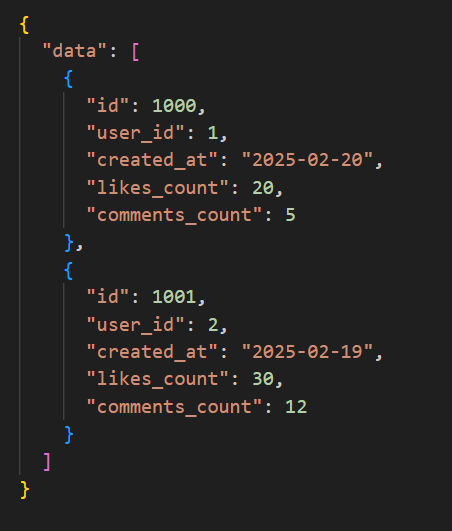


5) Получить список постов.

Запрос:

GET /posts

Ответ:



Еще запросы:

6) Получить пост по его id:

GET /posts/101

7) Получить список друзей пользователя:

GET /users/1/friends