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

**Проектирование приложения**

по курсу   
Технологии сетевого программирования

Группа 6304-010302D

Студент С.Д. Мельников

(*подпись*)

Самара 2025

**1. Описание архитектуры**

**Клиент:**

Отправляет запрос на сервер (к примеру, при регистрации, при авторизации, при создании карточки).

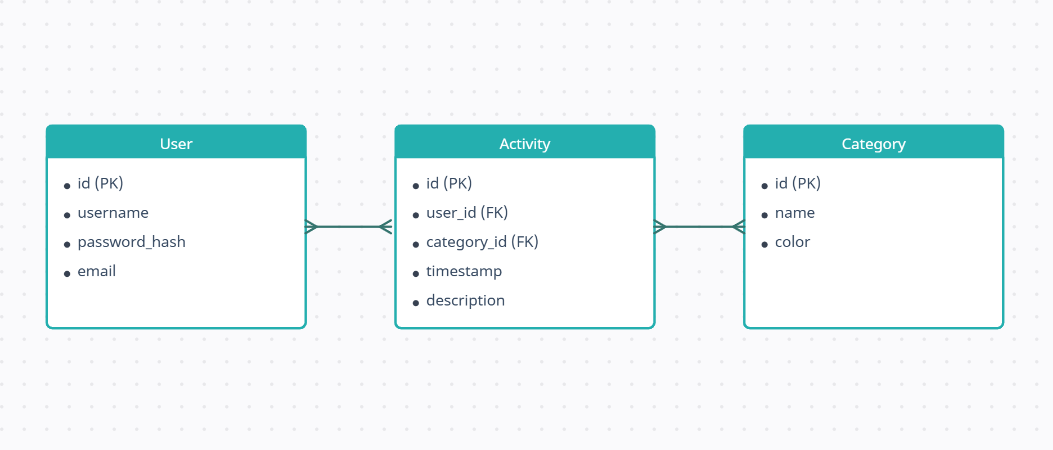
**Сервер:**

Обрабатывает, проверяет данные. Отправляет их в базу-данных (к примеру, добавляет нового пользователя в user), или получает данные из базы данных для отправки их клиенту.

**База данных:**

Хранит данные.

**2. Схема базы данных**



User – Activity (One to Many)

Activity – Category (Many to Many)

**3. Структура API**

**Пользователи**

* POST /api/register – регистрация нового пользователя
* POST /api/login – авторизация пользователя

**Активности**

* POST /api/activities — Добавление активности
* GET /api/activities — Получение всех активностей
* GET /api/activities?date=2000-1-1 — Получение активности за день
* GET /api/productivity?date=2000-1-1 — Расчет продуктивности за день
* GET /api/activities/{id} — Получение активности
* PUT /api/activities/{id} — Добавление активности
* DELETE /api/activities/{id} — удаление активности

**Категории**

* GET /api/activities/{id}/categories — Получение всех категорий активности
* GET /api/categories — Получение всех категорий
* POST /api/categories — Добавление категории
* GET /api/activities/{id} — Получение категории
* DELETE /api/activities/{id} — Удаление категории

**3. Стек технологий**

* **Язык программирования**: Java
* **Система контроля версий**: GitHub
* **База данных**: PostgreSQL
* **Безопасность**: JWT
* **API**: OpenAPI/Swagger
* **Тестирование API**: Postman
* **Шаблонизатор**: Thymeleaf
* **Контейнеризация**: Docker

**4. GitIgnore**

HELP.md

target/

!.mvn/wrapper/maven-wrapper.jar

!\*\*/src/main/\*\*/target/

!\*\*/src/test/\*\*/target/

### STS ###

.apt\_generated

.classpath

.factorypath

.project

.settings

.springBeans

.sts4-cache

### IntelliJ IDEA ###

.idea

\*.iws

\*.iml

\*.ipr

### NetBeans ###

/nbproject/private/

/nbbuild/

/dist/

/nbdist/

/.nb-gradle/

build/

!\*\*/src/main/\*\*/build/

!\*\*/src/test/\*\*/build/

### VS Code ###

.vscode/

**5. Репозиторий GitHub**

https://github.com/JoYfanCode/productivity-tracker