Задача 2.

Описание работы с проектом.

В проекте специально не учитывал возможное масштабирование и специально показал, как можно сделать keep it simple в то же самое время с учетом возможных ситуаций. Специально не писал интерфейсы к сервисам и обработчик исключений, т.к. для примера это сделал в задании 1. Усложненное решение с ElasticSearch не стал делать, т.к. времени было меньше дня, плюс текущая работа…

Для запуска нужно создать базу manage\_requests на сервере PostgreSQL.

Указать параметры соединения к PostgreSQL в application.properties.

Для первого запуска включить, spring.jpa.hibernate.ddl-auto = create и можно отключить ее после создания таблиц хибернейтом, чтобы данные сохранялись в базе при перезапуске. Можно добавить миграцию ликвибэйз по запросу заказчика.

Для работы с сервисом следует использовать следующие запросы, для примера настройки запросов из Postman.

Добавление продукта POST:

<http://localhost:8080/add/folder?folderName=folder3>

Добавление тега POST:

<http://localhost:8080/add/tag?tagName=tag11>

Добавление реквеста POST:

Graphical user interface, text, application, email

Description automatically generated

Назначение тега реквесту:

Graphical user interface, text, application

Description automatically generated

Назначение реквеста папке:

Graphical user interface, text, application, email

Description automatically generated

Выгрузка всех тегов:

<http://localhost:8080/get/tags>

Выгрузка всех реквестов:

<http://localhost:8080/get/requests>

Выгрузка всех папок:

http://localhost:8080/get/folders

Выгрузка всех реквестов по тегу GET:

<http://localhost:8080/get/requests_by_tag?tagId=1>

Выгрузка всех реквестов по папке GET:

<http://localhost:8080/get/requests_by_folder?folderId=2>