

ПРИЛОЖЕНИЕ 2

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО

Научный руководитель,
доцент департамента
программной инженерии
факультета компьютерных наук,
канд. физико-математических наук
_____ Г.Н. Жукова
« ____ » _____ 2023 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук
_____ В.В. Шилов
« ____ » _____ 2023 г.

СЕРВЕРНАЯ ЧАСТЬ ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ И УПРАВЛЕНИЯ ИНДИВИДУАЛЬНЫМИ СТУДЕНЧЕСКИМИ ЗАДАНИЯМИ ДЛЯ КУРСА «КОМПЬЮТЕРНЫЙ ПРАКТИКУМ ПО АЛГЕБРЕ НА PYTHON»

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.13-62 ПМИ 01-1

Исполнитель :
Студент группы БПИ196

_____/Сахаров Н.Д.
« ____ » _____ 2023 г.

УТВЕРЖДЁН
RU.17701729.04.13-62 ПМИ 01-1-ЛУ

**СЕРВЕРНАЯ ЧАСТЬ ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ И УПРАВЛЕНИЯ
ИНДИВИДУАЛЬНЫМИ СТУДЕНЧЕСКИМИ ЗАДАНИЯМИ ДЛЯ КУРСА
«КОМПЬЮТЕРНЫЙ ПРАКТИКУМ ПО АЛГЕБРЕ НА PYTHON»**

Программа и методика испытаний

RU.17701729.04.13-62 ПМИ 01-1-ЛУ

Листов 25

Содержание

1	АННОТАЦИЯ	4
2	ОБЪЕКТ ИСПЫТАНИЙ	5
2.1	Наименование	5
2.2	Краткая характеристика области применения	5
3	ЦЕЛЬ ИСПЫТАНИЙ	6
3.1	Цель проведения испытаний	6
4	ТРЕБОВАНИЯ К ПРОГРАММЕ	7
4.1	Требования к функциональным характеристикам	7
4.2	Условия эксплуатации	8
4.3	Требования к составу и параметрам технических средств	8
4.4	Требования к информационной и программной совместимости	8
5	ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ	10
6	СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ	11
6.1	Технические средства, используемые во время испытаний	11
6.2	Программные средства, используемые во время испытаний	11
7	МЕТОДЫ ИСПЫТАНИЙ	12
7.1	Используемые инструменты	12
7.2	Интеграционное тестирование	12
7.2.1	Запуск тестов	12
7.2.2	Общий подход к тестированию	12
7.2.3	Проверка регистрации и авторизации	13
7.2.4	Проверка медеджемента групп	14
7.2.5	Проверка управления лабораторными работами	15
7.2.6	Проверка генерации и раздачи вариантов	16
7.2.7	Проверка управления решениями работ	17
7.2.8	Проверка скачивания файлов	19
7.3	Нагрузочное тестирование	20
7.3.1	Тестирование регистрации и аутентификации	20
7.3.2	Тестирование загрузки и скачивания файлов	21

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7.3.3	Результаты тестирования	21
8	ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ	24
9	ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	25

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1 АННОТАЦИЯ

В данном программном документе представлена программа и методика испытаний программы «Серверная часть приложения для генерации и управления индивидуальными студенческими заданиями для курса «Компьютерный практикум по алгебре на Python»».

В разделе "Объект испытаний" указаны наименование программы, а также краткая характеристика области применения.

В разделе "Цель испытаний" находится цель проведения испытаний программы.

Раздел "Требования к программе" содержит требования, подлежащие проверке в ходе испытаний. В разделе "Требования к программной документации" указан состав программной документации.

В разделе "Средства и порядок испытаний" указаны технические и программные средства, используемые в ходе испытаний, а также порядок проведения испытаний.

Раздел "Методы испытаний" содержит описание используемых методов испытаний, описание проверок с указанием результатов.

В последнем разделе "Приложение. Перечень материалов, использованных в ходе испытаний" содержатся публикации и другие материалы, использованные в ходе испытаний. Оформление данного документа произведено по требованиям ЕСПД:

- ГОСТ 19.102-77 [1];
- ГОСТ 19.103-77 [2];
- ГОСТ 19.104-78 [3];
- ГОСТ 19.105-78 [4];
- ГОСТ 19.106-78 [5];
- ГОСТ 19.301-79 [6].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2 ОБЪЕКТ ИСПЫТАНИЙ

2.1 Наименование

«Серверная часть приложения для генерации и управления индивидуальными студенческими заданиями для курса «Компьютерный практикум по алгебре на Python»»

2.2 Краткая характеристика области применения

Приложение для курса "Компьютерный практикум по алгебре на Python" используется на курсе для автоматизации генерации и раздачи индивидуальных заданий студентам. Это инструмент, созданный для упрощения ручной работы преподавателей. Он обеспечивает удобное управление лабораторными работами, получение работ от студентов и хранение истории лабораторных работ в течение нескольких лет.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3 ЦЕЛЬ ИСПЫТАНИЙ

3.1 Цель проведения испытаний

Целью проведения испытаний является проверка корректности работы сервера, а так же проверка соответствия разработанной программы функциональным требованиям, изложенным в документе "Техническое задание".

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4 ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1 Требования к функциональным характеристикам

Сервер должен поддерживать следующую функциональность:

1. Регистрация пользователей:

- Предоставление формы для регистрации новых пользователей;
- Проверка уникальности имени пользователя и электронной почты;
- Хранение информации о зарегистрированных пользователях в базе данных.

2. Авторизация пользователей:

- Предоставление формы для входа зарегистрированных пользователей в систему;
- Проверка правильности введенного имени пользователя и пароля;
- Предоставление доступа к функционалу приложения только зарегистрированным пользователям.

3. Аутентификация пользователей:

- Проверка подлинности пользователя на основе информации о его учетных данных;
- Предотвращение несанкционированного доступа к данным пользователей.

4. Генерация заданий:

- Предоставление возможности выбора темы задания;
- Генерация заданий на основе выбранных параметров;
- Генерация заданий в формате Jupiter Notebook.

5. Просмотр списка заданий:

- Получение списка заданий студента;
- Получение списка заданий от преподавателя;
- Получение списка всех заданий.

6. Генерация нескольких вариантов заданий на основе типовых заданий:

- Предоставление возможности выбора темы заданий;
- Генерация нескольких вариантов заданий на основе типовых заданий с схожей сложностью;
- Генерация заданий в формате Jupiter Notebook.

7. Раздача заданий группе студентов:

- Автоматическая генерация и раздача нескольких вариантов заданий группе студентов на основе типовых заданий.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

8. Оценка и комментирование решений студентов:

- Предоставление возможности студентам добавлять комментарии к своим решениям заданий;
- Предоставление возможности преподавателям оценивать решение и добавлять комментарии к решениям студентов;
- Хранение результатов в базе данных для дальнейшего просмотра и анализа.

4.2 Условия эксплуатации

Сервер должен иметь доступ к сети, чтобы получать запросы от клиентов, а также доступ к postgresql базе данных

4.3 Требования к составу и параметрам технических средств

- Операционная система: Операционная система должна поддерживать работу Python и менеджера пакетов pipenv.
- Оперативная память: Рекомендуемо 1ГБ оперативной памяти.
- Процессор с тактовой частотой не ниже 2.0ГГц. Рекомендуется использовать как минимум двухядерный процессор.
- Размер дискового пространства зависит от количества пользователей о которых предполагается хранить данные.

4.4 Требования к информационной и программной совместимости

Операционная система Ubuntu 18.04 или любая другая поддерживающая менеджер пакетов pipenv, с установленными пакетами:

- fastapi
- uvicorn
- bcrypt
- sqlalchemy
- asyncpg
- httpx
- fastapi-security
- pydantic
- databases

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- python-dotenv
- python-multipart
- gevent
- psutil
- greenlet
- psycopg2-binary
- pyjwt
- autopep8
- fastapi-cache
- pytest
- testsuite
- pytest-asyncio
- aiosqlite
- flake8
- jupyterlab
- pytz
- black
- python-rocksdb
- numpy
- sympy
- locust

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

- Техническое задание.
- Текст ВКР.
- Руководство программиста.
- Программа и методика испытаний.
- Текст программы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6 СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

6.1 Технические средства, используемые во время испытаний

Во время испытаний был использован компьютер со следующими техническими характеристиками:

- Процессор: Intel Xeon Gold 6338 CPU @ 2.0GHz 2 ядра;
- Оперативная память 12Гб;
- HDD диск объёмом 250Гб.

6.2 Программные средства, используемые во время испытаний

Во время испытаний был использован компьютер с установленной операционной системой Ubuntu 20.04

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7 МЕТОДЫ ИСПЫТАНИЙ

7.1 Используемые инструменты

Для юниттестов используется модуль unittest.

Для создания тестов на запросы поддежанные на сервере используется фреймворк pytest. При помощи системы фикстур и лёгкой конфигурируемости он позволяет модернезировать параметры сервиса перед запуском и создать тестовые данные в базе данных.

7.2 Интеграционное тестирование

7.2.1 Запуск тестов

В проекте предусмотрена тестовая база данных, которая используется для интеграционных тестов. Для автоматического выставления переменных окружения для доступа к тестовой базе данных был разработан скрипт runtests.sh, запуск скрипта из корня проекта позволяет запустить все тесты. Можно запустить тесты только из одного файла, для этого нужно указать имя файла при запуске, как параметр.

7.2.2 Общий подход к тестированию

Каждый запрос имеет набор тестов, который можно описать как позитивные и негативные тесты.

Позитивный тест как правило всего один на каждый запрос, он проверяет корректность выполнения запроса, который отправлен четко по схеме запроса без каких либо проблем. В таких тестах проверяется, что запрос выполняет свою задачу. Пример позитивного теста:

```
async def test_register_student(client, test_student_in):
    response = await client.post(
        "/registration/register/student", json=test_student_in().dict()
    )

    assert response.status_code == 200
    assert "access_token" in response.json()
    assert "token_type" in response.json()
    assert response.json()["token_type"] == "bearer"
    recieved_token = response.json()["access_token"]
    user_id, user_type = decode_access_token(token=recieved_token)
    assert user_id
    assert user_type == UserType.STUDENT
```

Негативные тесты нужны для проверки корректности работы сервера при получении неверных данных. К таким тестам так же относятся проверки запрещенного доступа, неавторизованные запросы, запросы к несуществующим данным.

Пример негативного теста:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

```
@pytest.mark.anyio
async def test_login_not_exists(client: AsyncClient):
    headers = {
        "accept": "application/json",
        "Content-Type": "application/x-www-form-urlencoded",
    }
    data = {
        "username": "not_exists",
        "password": "not_exists",
    }

    response = await client.post("/auth/token", headers=headers, data=data)

    assert response.status_code == 400
    assert "detail" in response.json()
    assert response.json()["detail"] == "Incorrect username or password"
```

Далее будет представлено описание всех тестов. Описание разделено на секции по соответствующим файлам в репозитории проекта и описывают тесты в порядке их следования.

7.2.3 Проверка регистрации и авторизции

Тест для регистрации студента:

Тест проверяет, что при отправке корректных данных для регистрации студента, сервер возвращает код состояния 200 (Успешная регистрация), а также проверяет, что в ответе содержится токен доступа, тип токена равен "bearer" и декодированный токен содержит правильный идентификатор пользователя и тип пользователя равный "student".

Тест для регистрации преподавателя:

Тест проверяет, что при отправке корректных данных для регистрации преподавателя, сервер возвращает код состояния 200 (Успешная регистрация), а также проверяет, что в ответе содержится токен доступа, тип токена равен "bearer" и декодированный токен содержит правильный идентификатор пользователя и тип пользователя равный "tutor".

Тест для получения токена доступа при авторизации:

Тест проверяет, что при отправке корректных данных для авторизации (имя пользователя и пароль), сервер возвращает код состояния 200 (Успешная авторизация), а также проверяет, что в ответе содержится токен доступа, тип токена равен "bearer" и полученный токен соответствует ожиданиям.

Тест для авторизации с несуществующими данными:

Тест проверяет, что при отправке несуществующих данных для авторизации, сервер возвращает код состояния 400 (Некорректное имя пользователя или пароль) и соответствующее сообщение об ошибке.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7.2.4 Проверка медеджмента групп

Тест для создания группы: Тест проверяет, что при отправке корректных данных о создании группы (имя группы и идентификатор преподавателя) сервер возвращает код состояния 201 (Успешное создание), а также проверяет, что созданная группа действительно сохраняется в базе данных.

Тест для создания группы с некорректными данными:

Тест проверяет, что при отправке некорректных данных о создании группы (например, от имени студента) сервер возвращает код состояния 403 (Доступ запрещен).

Тест для получения информации о группе:

Тест проверяет, что при запросе информации о группе по ее идентификатору, сервер возвращает код состояния 200 (Успешный запрос), а также проверяет, что полученная информация соответствует ожиданиям.

Тест для получения информации о несуществующей группе:

Тест проверяет, что при запросе информации о несуществующей группе, сервер возвращает код состояния 404 (Не найдено).

Тест для обновления имени группы:

Тест проверяет, что при отправке запроса на обновление имени группы, сервер возвращает код состояния 204 (Успешное обновление), а также проверяет, что имя группы в базе данных соответствует новому имени.

Тест для обновления преподавателя группы:

Тест проверяет, что при отправке запроса на обновление преподавателя группы, сервер возвращает код состояния 204 (Успешное обновление), а также проверяет, что преподаватель группы в базе данных соответствует новому преподавателю.

Тест для обновления группы с некорректными данными:

Тест проверяет, что при отправке некорректных данных для обновления группы (например, от имени студента или с несуществующим идентификатором группы), сервер возвращает соответствующие коды состояния (403 - Доступ запрещен, 404 - Не найдено).

Тест для добавления студента в группу:

Тест проверяет, что при отправке запроса на добавление студента в группу, сервер возвращает код состояния 200 (Успешное добавление), а также проверяет, что информация о студенте в базе данных обновляется соответствующим образом.

Тест для добавления студента в группу с некорректными данными:

Тест проверяет, что при отправке некорректных данных для добавления студента в группу (например, с несуществующим идентификатором студента), сервер возвращает соответствующие коды состояния (404 - Не найдено).

Тест для добавления студента в группу от имени другого студента:

Тест проверяет, что при отправке запроса на добавление студента в группу от имени другого студента, сервер возвращает код состояния 403 (Доступ запрещен).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7.2.5 Проверка управления лабораторными работами

Тест для создания лабораторной работы:

Тест проверяет, что при отправке запроса на создание лабораторной работы с корректными данными преподавателем, сервер возвращает код состояния 201 (Создано) и в ответе содержится правильный идентификатор созданной лабораторной работы. Проверяется также, что лабораторная работа сохранена в базе данных с правильными данными.

Тест для создания лабораторной работы со студенческим доступом:

Тест проверяет, что при попытке создания лабораторной работы со студенческим доступом, сервер возвращает код состояния 403 (Запрещено).

Тест для создания лабораторной работы с некорректными данными:

Тест проверяет, что при отправке запроса на создание лабораторной работы с некорректными данными (некорректный тип генератора или некорректные даты), сервер возвращает код состояния 422 (Неправильный запрос).

Тест для создания лабораторной работы с несуществующей группой:

Тест проверяет, что при попытке создания лабораторной работы с несуществующей группой, сервер возвращает код состояния 404 (Не найдено).

Тест для получения всех лабораторных работ студента:

Тест проверяет, что при отправке запроса на получение всех лабораторных работ студента с правильным токеном студента, сервер возвращает код состояния 200 (ОК) и в ответе содержатся правильные данные о лабораторной работе. Проверяется, что в ответе содержится только одна лабораторная работа, соответствующая студенту.

Тест для получения всех лабораторных работ студента с негативными данными:

Тест проверяет, что при попытке получить все лабораторные работы студента без правильного доступа (не студент), сервер возвращает код состояния 403 (Запрещено). При использовании некорректного токена доступа, сервер возвращает код состояния 401 (Неавторизовано). Если у студента нет ни одной лабораторной работы, сервер должен возвращать пустой список.

Тест для получения всех лабораторных работ преподавателя:

Тест проверяет, что при отправке запроса на получение всех лабораторных работ преподавателя с правильным токеном преподавателя, сервер возвращает код состояния 200 (ОК) и в ответе содержатся правильные данные о лабораторной работе. Проверяется, что в ответе содержится только одна лабораторная

работа, соответствующая преподавателю.

Тест для получения всех лабораторных работ преподавателя с негативными данными:

Тест проверяет, что при попытке получить все лабораторные работы преподавателя без правильного доступа (не преподаватель), сервер возвращает код состояния 403 (Запрещено). При использовании некорректного токена доступа, сервер возвращает код состояния 401 (Неавторизовано). Если у преподавателя нет ни одной лабораторной работы, сервер должен возвращать пустой список.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7.2.6 Проверка генерации и раздачи вариантов

Тест для генерации вариантов:

Тест проверяет, что при отправке запроса на генерацию вариантов с корректными данными, сервер возвращает код состояния 201 (Создано) и в ответе содержатся правильные данные о сгенерированных вариантах. Для каждого студента из группы должен быть сгенерирован отдельный вариант с правильными данными (идентификатор лабораторной работы, идентификатор студента, номер варианта, имя файла и ключ файла). Проверяется также, что содержимое файла сохранено в хранилище.

Тест для генерации вариантов с пустой группой:

Тест проверяет, что при попытке генерации вариантов для пустой группы, сервер возвращает код состояния 400 (Неправильный запрос).

Тест для генерации вариантов без авторизации:

Тест проверяет, что при попытке генерации вариантов без авторизации, сервер возвращает код состояния 401 (Не авторизован).

Тест для генерации вариантов с неправильным типом пользователя:

Тест проверяет, что при попытке генерации вариантов с неправильным типом пользователя (не преподаватель), сервер возвращает код состояния 403 (Запрещено).

Тест для генерации вариантов с некорректным идентификатором лабораторной работы:

Тест проверяет, что при отправке запроса на генерацию вариантов с некорректным идентификатором лабораторной работы, сервер возвращает код состояния 404 (Не найдено).

Тест для генерации вариантов с некорректным идентификатором группы:

Тест проверяет, что при отправке запроса на генерацию вариантов с некорректным идентификатором группы, сервер возвращает код состояния 404 (Не найдено).

Тест для получения всех вариантов студента:

Тест проверяет, что при отправке запроса на получение всех вариантов студента с корректным идентификатором студента, сервер возвращает код состояния 201 (Создано) и в ответе содержатся правильные данные о варианте. Должен быть возвращен только один вариант, соответствующий студенту, и вариант должен содержать правильные данные (идентификатор лабораторной работы

, идентификатор студента, имя файла и ключ файла). Проверяется также, что содержимое файла сохранено в хранилище.

Тест для получения всех вариантов студента без доступа:

Тест проверяет, что при попытке получить все варианты студента без правильного доступа (не преподаватель или другой студент), сервер возвращает код состояния 403 (Запрещено). При правильном доступе студента должен возвращаться только его собственный вариант.

Тест для получения варианта по идентификатору студента:

Тест проверяет, что при отправке запроса на получение варианта по идентификатору студента с корректным идентификатором студента, сервер возвращает код состояния 201 (Со-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

здано) и в ответе содержатся правильные данные о варианте. Вариант должен содержать правильные данные (идентификатор лабораторной работы, идентификатор студента, имя файла и ключ файла).

Тест для получения варианта по идентификатору студента без доступа:

Тест проверяет, что при попытке получить вариант по идентификатору студента без правильного доступа (не преподаватель или другой студент), сервер возвращает код состояния 403 (Запрещено). При правильном доступе должен возвращаться только вариант соответствующего студента.

Тест для получения вариантов по идентификатору студента преподавателем:

Тест проверяет, что при отправке запроса на получение вариантов по идентификатору студента преподавателем с корректными данными, сервер возвращает код состояния 201 (Создано) и в ответе содержатся правильные данные о вариантах. Должен быть возвращен только один вариант, соответствующий студенту, и вариант должен содержать правильные данные (идентификатор лабораторной работы, идентификатор студента, имя файла и ключ файла). Проверяется также, что содержимое файла сохранено в хранилище.

Тест для получения вариантов по идентификатору студента с негативными данными:

Тест проверяет, что при попытке получить варианты по некорректному идентификатору студента или неправильным данным, сервер возвращает код состояния 403 (Запрещено) или 404 (Не найдено) в зависимости от ситуации.

Тест для получения варианта по идентификатору:

Тест проверяет, что при отправке запроса на получение варианта по идентификатору с корректным идентификатором варианта, сервер возвращает код состояния 201 (Создано) и в ответе содержатся правильные данные о варианте. Вариант должен содержать правильные данные (идентификатор лабораторной работы, идентификатор студента, имя файла и ключ файла).

Тест для получения варианта по идентификатору без доступа:

Тест проверяет, что при попытке получить вариант по идентификатору без правильного доступа (не преподаватель или студент, которому принадлежит вариант), сервер возвращает код состояния 403 (Запрещено). При правильном доступе должен возвращаться только вариант соответствующего студента.

7.2.7 Проверка управления решениями работ

Описание того, что проверяют каждый тест в формате LaTeX:

Тест для загрузки файла решения:

Тест проверяет, что при отправке корректных данных для загрузки файла решения, сервер возвращает код состояния 201 (Создано) и проверяет, что в ответе содержатся правильные данные о загруженном файле (имя файла, идентификатор варианта лабораторной работы) и что содержимое файла сохранено в хранилище.

Тест для загрузки файла решения с доступом только у студента-автора:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Тест проверяет, что при попытке загрузки файла решения студентом, которому не принадлежит вариант лабораторной работы, сервер возвращает код состояния 403 (Запрещено) и соответствующее сообщение об ошибке.

Тест для загрузки файла решения с ошибкой доступа:

Тест проверяет, что при попытке загрузки файла решения, к которому у пользователя нет доступа (вариант лабораторной работы из другой группы), сервер возвращает код состояния 403 (Запрещено) и соответствующее сообщение об ошибке.

Тест для загрузки файла решения с некорректным идентификатором варианта лабораторной работы:

Тест проверяет, что при отправке запроса на загрузку файла решения с некорректным идентификатором варианта лабораторной работы, сервер возвращает код состояния 404 (Не найдено).

Тест для загрузки файла решения с неподдерживаемым типом файла:

Тест проверяет, что при попытке загрузки файла решения с неподдерживаемым типом файла, сервер возвращает код состояния 422 (Неправильный запрос).

Тест для загрузки файла решения без файла:

Тест проверяет, что при отправке запроса на загрузку файла решения без файла, сервер возвращает код состояния 422 (Неправильный запрос).

Тест для получения информации о решении:

Тест проверяет, что при отправке запроса на получение информации о решении с корректным идентификатором решения, сервер возвращает код состояния 200 (Успешный запрос) и в ответе содержатся правильные данные о решении.

Тест для получения информации о решении с некорректным идентификатором решения:

Тест проверяет, что при отправке запроса на получение информации о решении с некорректным идентификатором решения, сервер возвращает код состояния 422 (Неправильный запрос).

Тест для получения информации о решении преподавателем:

Тест проверяет, что при отправке запроса на получение информации о решении преподавателем, сервер возвращает код состояния 200 (Успешный запрос).

Тест для получения информации о решении с недоступным доступом студента:

Тест проверяет, что при отправке запроса на получение информации о решении студентом, который не имеет доступа к данному решению, сервер возвращает код состояния 403 (Запрещено).

Тест для получения информации о решении с отсутствующим идентификатором решения:

Тест проверяет, что при отправке запроса на получение информации о решении без указания идентификатора решения, сервер возвращает код состояния 422 (Неправильный запрос).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Тест для оценки решения:

Тест проверяет, что при отправке корректных данных для оценки решения, сервер возвращает код состояния 202 (Принято) и проверяет, что в ответе содержатся правильные данные об оценке (идентификатор решения, оценка преподавателя, комментарий преподавателя) и что преподаватель добавлен в качестве проверяющего к варианту лабораторной работы.

Тест для оценки решения с недоступным доступом:

Тест проверяет, что при попытке оценки решения без правильного доступа, сервер возвращает код состояния 403 (Запрещено).

Тест для оценки решения с некорректным идентификатором решения:

Тест проверяет, что при отправке запроса на оценку решения с некорректным идентификатором решения, сервер возвращает код состояния 422 (Неправильный запрос).

Тест для оценки решения с некорректной оценкой:

Тест проверяет, что при отправке запроса на оценку решения с некорректной оценкой, сервер возвращает код состояния 422 (Неправильный запрос).

7.2.8 Проверка скачивания файлов**Тест для получения файла варианта лабораторной работы:**

Тест проверяет, что при отправке запроса на получение файла варианта лабораторной работы с корректными данными (ключ файла и имя файла), сервер возвращает код состояния 200 (Успешный запрос), а также проверяет, что полученный файл содержит правильные данные.

Тест для получения файла варианта лабораторной работы с некорректными данными:

Тест проверяет, что при отправке запроса на получение файла варианта лабораторной работы с некорректными данными (несуществующим ключом файла), сервер возвращает код состояния 404 (Не найдено).

Тест для получения файла варианта лабораторной работы без указания ключа файла:

Тест проверяет, что при отправке запроса на получение файла варианта лабораторной работы без указания ключа файла, сервер возвращает код состояния 422 (Неполный запрос).

Тест для получения файла варианта лабораторной работы без указания имени файла:

Тест проверяет, что при отправке запроса на получение файла варианта лабораторной работы без указания имени файла, сервер возвращает код состояния 422 (Неполный запрос).

Тест для получения файла решения лабораторной работы:

Тест проверяет, что при отправке запроса на получение файла решения лабораторной работы с корректными данными (ключ файла и имя файла), сервер возвращает код состояния 200 (Успешный запрос), а также проверяет, что полученный файл содержит правильные данные.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Тест для получения файла решения лабораторной работы преподавателем:

Тест проверяет, что при отправке запроса на получение файла решения лабораторной работы преподавателем с корректными данными (ключ файла и имя файла), сервер возвращает код состояния 200 (Успешный запрос), а также проверяет, что полученный файл содержит правильные данные.

Тест для получения файла решения лабораторной работы студентом, не являющимся автором решения:

Тест проверяет, что при отправке запроса на получение файла решения лабораторной работы студентом, не являющимся автором решения, сервер возвращает код состояния 403 (Доступ запрещен).

Тест для получения файла решения лабораторной работы с некорректными данными:

Тест проверяет, что при отправке запроса на получение файла решения лабораторной работы с некорректными данными (несуществующим ключом файла), сервер возвращает код состояния 404 (Не найдено).

Тест для получения файла решения лабораторной работы без указания ключа файла:

Тест проверяет, что при отправке запроса на получение файла решения лабораторной работы без указания ключа файла, сервер возвращает код состояния 422 (Неполный запрос).

Тест для получения файла решения лабораторной работы без указания имени файла:

Тест проверяет, что при отправке запроса на получение файла решения лабораторной работы без указания имени файла, сервер возвращает код состояния 422 (Неполный запрос).

7.3 Нагрузочное тестирование

Для проведения нагрузочного тестирования в описанных случаях используется инструмент Locust. Locust позволяет создавать виртуальных пользователей, которые могут имитировать различные операции и взаимодействия с сервером. Эти виртуальные пользователи могут быть настроены для выполнения определенных задач с различной интенсивностью и параллелизмом.

Также Locust позволяет определить задачи, которые виртуальные пользователи будут выполнять, и задавать временные интервалы между запросами. В результате, Locust может смоделировать нагрузку на сервер и собирать данные о производительности и надежности системы при различных уровнях нагрузки.

Для тестирования было реализовано два сценария: сценарий регистрации и аутентификации и сценарий загрузки и скачивания файлов.

7.3.1 Тестирование регистрации и аутентификации

- Создается виртуальный пользователь с уникальными данными (имя пользователя, пароль, имя, фамилия, адрес электронной почты) и типом пользователя (студент или пре-

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

подаватель).

- Пользователь регистрируется, отправляя POST-запрос на эндпоинт регистрации с соответствующими данными.
- Пользователь выполняет задачу аутентификации, отправляя POST-запрос на эндпоинт аутентификации с учетными данными.
- Результаты запросов выводятся в консоль для отслеживания статуса и ответа сервера.

7.3.2 Тестирование загрузки и скачивания файлов

- Создается виртуальный пользователь.
- Пользователь находится в режиме загрузки файлов.
- Пользователь выполняет задачу загрузки файла, отправляя POST-запрос на эндпоинт загрузки файла.
- Если загрузка файла прошла успешно, пользователь сохраняет полученный ключ файла.
- Пользователь выполняет задачу скачивания файла, отправляя GET-запрос на эндпоинт скачивания файла с использованием сохраненного ключа файла.
- Результаты запросов выводятся в консоль для отслеживания статуса и ответа сервера.

Каждый виртуальный пользователь выполняет задачи в соответствии с определенными шаблонами (tasks) и интервалами времени между запросами (wait_time), заданными в конфигурации Locust. Результаты тестирования позволяют оценить производительность и стабильность системы при различных уровнях нагрузки.

7.3.3 Результаты тестирования

Тестирование показало что сервер выдерживает нагрузку больше предполагаемой(по количеству пользователей), а так же выявило хорошую отзывчивость сервера при активной нагрузке на операции с файлами.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

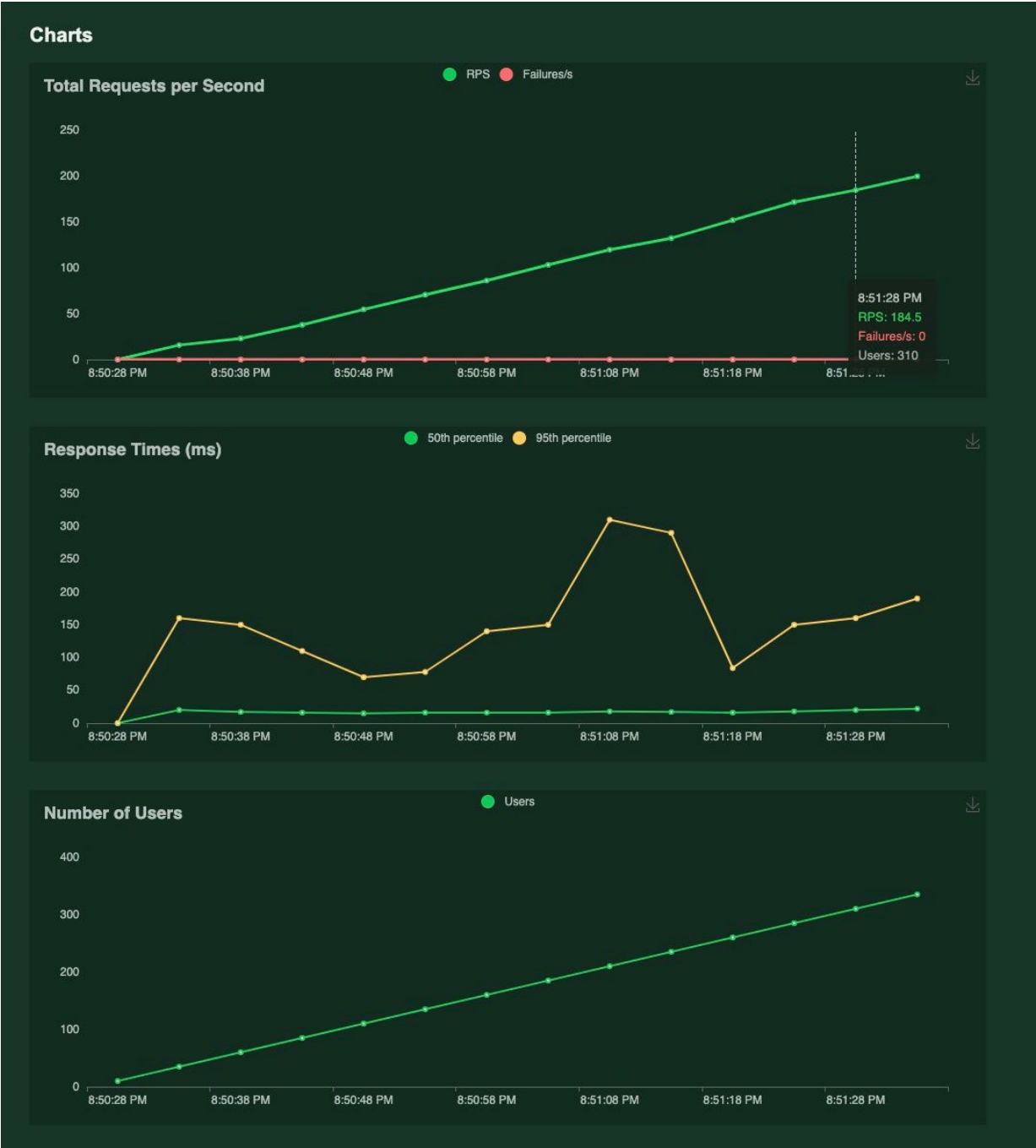


Рис. 1: Графики тестирования регистрации и авторизации

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

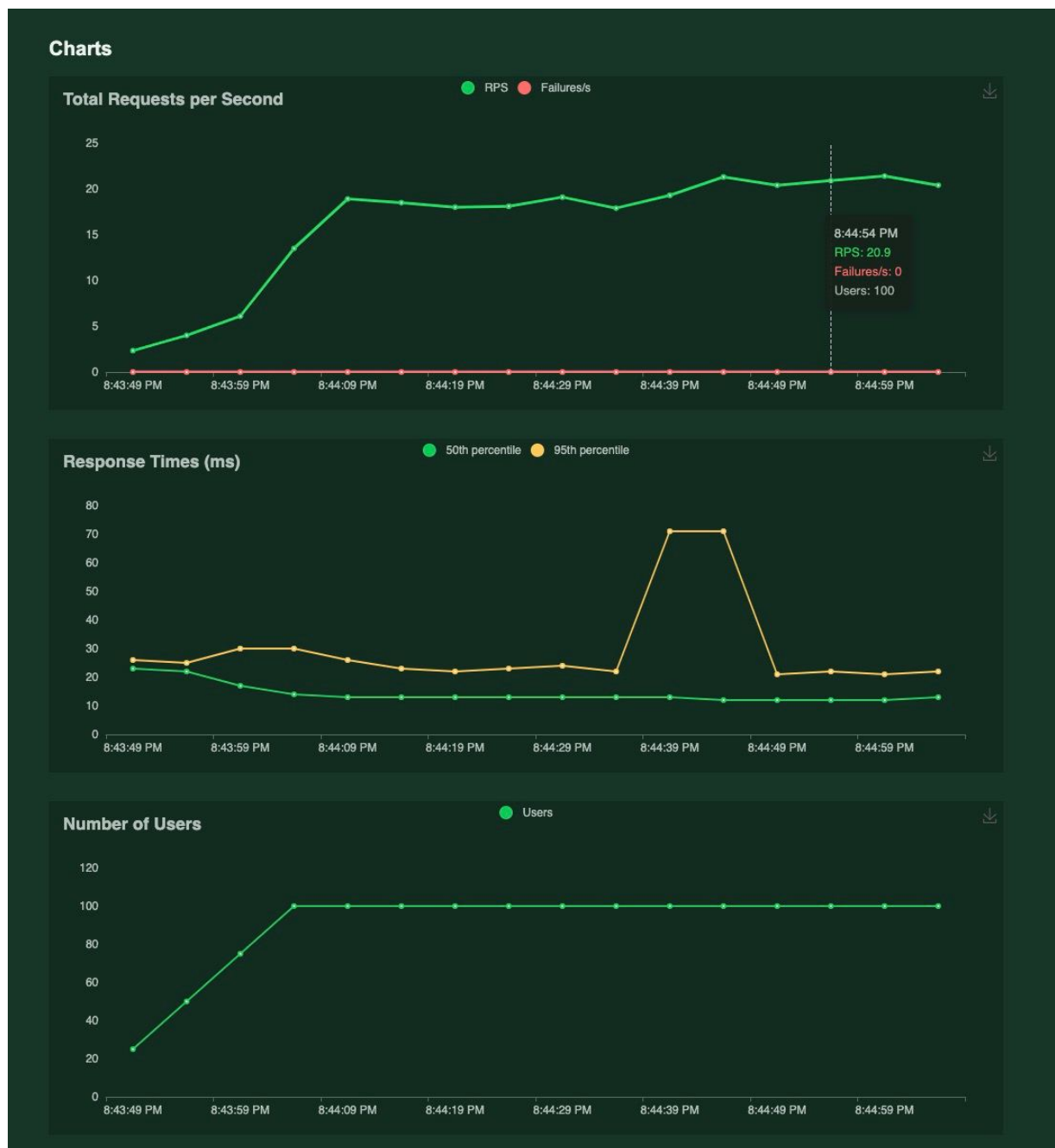


Рис. 2: Графики тестирования операций с файлами

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

8 ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.201-78 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.
8. PostgreSQL [Электронный ресурс]/ Хабр. [2018—2019].-Электрон. текстовые дан. -Режим доступа: <https://habr.com/ru/post/340460/> (дата обращения: 18.04.2022).
9. PostgreSQL [Электронный ресурс]/ Википедия. [2018—2019].-Электрон. текстовые дан. -Режим доступа: <https://ru.wikipedia.org/wiki/PostgreSQL> (дата обращения: 18.04.2022).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

9 ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Номера листов(страниц)					Всего листов (страниц в докум.)	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
Изм.	Измененных	Замененных	Новых	Архивированных					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 ПМИ				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата