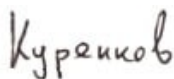


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО

Старший преподаватель департамента
больших данных и информационного
поиска



В.В. Куренков
«10» _____ мая _____ 2023 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук



В.В. Шилов
«10» _____ мая _____ 2023 г.

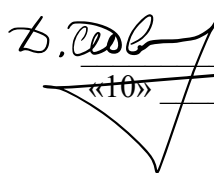
**Система проверки решений по программированию для Лицея НИУ ВШЭ
с автоматизированным подсчётом рейтинга**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.10-01 81 01-1-ЛУ

Исполнитель
студент группы БПИ219



Д.И. Швецов /
«10» _____ мая _____ 2023 г.

Москва 2023

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

УТВЕРЖДЕН
RU.17701729.05.10-01 81 01-1-ЛУ

**Система проверки решений по программированию для Лицея НИУ ВШЭ
с автоматизированным подсчётом рейтинга**

Пояснительная записка

RU.17701729.05.10-01 81 01-1

Листов 31

<i>Подп. и дата</i>	
<i>Инв. № дубл.</i>	
<i>Взам. инв. №</i>	
<i>Подп. и дата</i>	
<i>Инв. № подл</i>	

Москва 2023

ОГЛАВЛЕНИЕ

1.	ВВЕДЕНИЕ	4
1.1.	Наименование программы и объекта, в котором программа используется.....	4
1.2.	Краткая характеристика области применения программы и объекта, в котором она используется	4
2.	ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ	5
3.	НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	6
3.1.	Функциональное назначение.....	6
3.2.	Эксплуатационное назначение.....	6
4.	ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ	7
4.1.	Постановка задачи на разработку программы	7
4.2.	Описание алгоритма и функционирования программы	15
4.2.1.	Описание функционирования регистрации нового пользователя в систему 15	
4.2.2.	Получение информации о профиле учащегося.....	16
4.2.3.	Получение информации о профиле учителя.....	17
4.2.4.	Авторизация пользователя в систему	17
4.2.5.	Добавление нового преподавателя в систему с помощью администраторских прав.....	18
4.2.6.	Получение информации о рейтинге, отсортированном по оценкам пользователя в порядке убывания	19
4.2.7.	Получение информации о рейтинге, отсортированном по количеству задач, пользователей в порядке убывания	19
4.2.8.	Скачивание информации о рейтинге, отсортированном по оценкам, пользователей в порядке убывания в формате .csv	20
4.2.9.	Скачивание информации о рейтинге, отсортированном по количеству задач, пользователей в порядке убывания в формате .csv	20

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

4.2.10. Получение информации о всех посылках (т.е. решениях) определённого пользователя	21
4.2.11. Добавление решения определенной задачи определенного контекста в систему	22
4.2.12. Добавление нового контекста с задачами преподавателем в систему	23
4.2.13. Получение информации о всех контекстах, в которых пользователь принимает участие.....	24
4.2.14. Получение информации о всех контекстах, имеющихся в системе.....	24
4.2.15. Получение информации о всех посылках всех пользователей системы	25
4.2.16. Получение информации о роли пользователя системы	25
4.2.17. Получение информации о контексте по его id в системе.....	26
4.3. Описание и обоснование выбора метода организации входных и выходных данных	26
4.4. Описание и обоснование состава технических и программных средств	27
5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ	28
5.1. Предполагаемая потребность	28
5.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами	28
ПРИЛОЖЕНИЕ 1	30
Глоссарий.....	30
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	31

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

1. ВВЕДЕНИЕ

1.1. Наименование программы и объекта, в котором программа используется

Наименование темы разработки: «Система проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга».

Наименование темы разработки на английском языке: «HSE Lyceum Programming Testing System with Automated Rating Estimation».

Наименование объекта, в котором программа используется: Лицей НИУ ВШЭ.

1.2. Краткая характеристика области применения программы и объекта, в котором она используется

«Система проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга» - набор программных инструментов для оценивания решений задач на разных языках программирования для учащихся Лицея НИУ ВШЭ с рейтингом для каждого учащегося.

Лицей НИУ ВШЭ – общеобразовательное учреждение города Москва.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем тема курсового проекта.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1. Функциональное назначение

Программа содержит в себе систему, тестирующую решения задач по программированию Лицея НИУ ВШЭ и автоматически подсчитывающую рейтинг каждого учащегося, исходя из его оценок и количества решенных задач. Имеет учебно-методическое назначение.

3.2. Эксплуатационное назначение

Программа направлена на решение учащимися Лицея НИУ ВШЭ задач по программированию, которые им выдал тот или иной преподаватель, просмотр учащимися своего рейтинга по оценкам и количеству решенных задач среди других учащихся. Также программа направлена на просмотр преподавателями решений задач учащимися, получения их рейтинга в виде файлов.

Конечные пользователи – учащиеся и преподаватели информатики/программирования Лицея НИУ ВШЭ.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

4. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

4.1. Постановка задачи на разработку программы

Бэкенд «Системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга» - API, который должен быть реализован на Java с использованием фреймворка Spring Boot, базы данных PostgreSQL и обеспечивать возможность выполнения следующих задач:

1. Регистрация нового пользователя в систему.

Ограничений доступа к функционалу посредством аутентификации не должно быть.

В состав отправляемых на сервер данных должны входить следующие свойства:

1. Имя
2. Фамилия
3. Отчество (при наличии)
4. Email
5. Роль
6. Пароль
7. Название образовательной программы (департамент)
8. Название образовательной группы

В качестве ответа с сервера должна приходить информация об успешно завершённой регистрации пользователя в системе, либо информация об ошибке, содержащая сообщение возникшей ошибки.

2. Получение информации о профиле учащегося.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должны входить следующие свойства.

1. Айди учащегося в данной системе
2. Имя
3. Фамилия
4. Отчество (при наличии)
5. Email

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

6. Название образовательной программы (департамент)

7. Название образовательной группы

3. Получение информации о профиле учителя.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должны входить следующие свойства:

1. Айди преподавателя в данной системе.
2. Имя
3. Фамилия
4. Отчество (при наличии)
5. Email
6. Название образовательной программы (департамент)
7. Название образовательной группы

4. Авторизация пользователя в систему.

Ограничений доступа к функционалу посредством аутентификации не должно быть.

В состав отправляемых на сервер данных должны входить следующие свойства:

1. Email
2. Пароль

В качестве ответа с сервера должна приходить информация об успешно завершенной авторизации пользователя в систему, содержащая роль пользователя и сгенерированный JWT-токен, либо информация об ошибке, содержащая сообщение возникшей ошибки.

5. Добавление нового преподавателя в систему с помощью администраторских прав.

Должно иметься ограничение доступа к функционалу посредством аутентификации с помощью Basic Auth по заранее выданным администратору логину и паролю.

В состав отправляемых на сервер данных должны входить следующие свойства:

1. Имя
2. Фамилия
3. Отчество (при наличии)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

4. Email
5. Роль
6. Пароль
7. Название образовательной программы (департамент)
8. Название образовательной группы

В качестве ответа с сервера должна приходить информация об успешно завершеном добавлении нового преподавателя в систему, либо информация об ошибке, содержащая сообщение возникшей ошибки.

6. Получение информации о рейтинге, отсортированном по оценкам пользователей в порядке убывания.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должен входить список объектов, содержащих следующие свойства:

1. Айди пользователя.
2. Email
3. Роль
4. Имя
5. Фамилия
6. Отчество (при наличии)
7. Название образовательной программы (департамент)
8. Название образовательной группы

7. Получение информации о рейтинге, отсортированном по количеству задач, пользователей в порядке убывания.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

В состав отправляемых с сервера данных должен входить список объектов, содержащий следующие свойства:

1. Айди пользователя
2. Email
3. Роль
4. Имя
5. Фамилия
6. Отчество (при наличии)
7. Название образовательной программы (департамент)
8. Название образовательной группы

8. Скачивание информации о рейтинге, отсортированном по оценкам, пользователей в порядке убывания в формате .csv.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должен входить файл в формате .csv, который скачивается в указанную по умолчанию директорию и содержит строки с представленной в них информацией в следующем виде:

Rating position;Lastname;First name;Email;Department;Group name

9. Скачивание информации о рейтинге, отсортированном по количеству решенных задач, пользователей в порядке убывания в формате .csv.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должен входить файл в формате .csv, который скачивается в указанную по умолчанию директорию и содержит строки с представленной в них информацией в следующем виде:

Rating position;Lastname;First name;Email;Department;Group name

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

10. Получение информации о всех посылках (т. е. решениях) определённого пользователя.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должен входить список объектов, содержащих следующие свойства:

1. Айди посылки в данной системе
2. Код программы
3. Язык программирования, на котором написано данное решение
4. Статус посылки
5. Затраченная память на исполнение программы
6. Затраченное время на исполнение программы
7. Номер теста, в котором возникла ошибка
8. Название контеста, к которому относится решение
9. Название задачи, к которой относится решение

11. Добавление решения определенной задачи определенного контеста в систему.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

В состав отправляемых с сервера данных должны входить следующие свойства:

1. Код программы
2. Язык программирования, на котором написано данное решение
3. Статус посылки
4. Затраченная память на исполнение программы
5. Затраченное время на исполнение программы
6. Номер теста, в котором возникла ошибка
7. Название контеста, к которому относится решение
8. Название задачи, к которой относится решение

В качестве ответа с сервера должна приходить информация об успешно завершённом добавлении нового решения пользователя в систему, либо информация об ошибке, содержащая сообщение возникшей ошибки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

12. Добавление нового конкурса с задачами преподавателем в систему.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

В состав отправляемых на сервер данных должны входить следующие свойства:

1. Название
2. Айди конкурса в системе тестирования Ejudge
3. Начало конкурса (дата)
4. Завершение конкурса (дата)
5. Список задач, где задача содержит свойства
 1. Название
 2. Условие
 3. Ограничение по памяти
 4. Ограничение по времени
 5. Количество попыток, разрешенное для отправки уникального решения пользователем
6. Классификация задачи

В качестве ответа с сервера должна приходить информация об успешно завершеном добавлении нового конкурса с задачами преподавателем в систему, либо информация об ошибке, содержащая сообщение возникшей ошибки.

13. Получение информации о всех конкурсах, в которых пользователь принимает участие.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должен входить список объектов со следующими свойствами:

1. Айди конкурса в текущей системе
2. Айди конкурса в системе тестирования Ejudge
3. Название
4. Время начала (дата)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

5. Время завершения (дата)

6. Список объектов типа задача, где задача содержит свойства

1. Название
2. Условие
3. Ограничение по памяти
4. Ограничение по времени
5. Количество попыток, разрешенное для отправки уникального решения пользователем
6. Классификация задачи

14. Получение информации о всех контекстах, имеющихся в системе.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должен входить список объектов со следующими свойствами:

1. Айди контекста в данной системе
2. Айди контекста в системе тестирования Ejudge
3. Название
4. Время начала (дата)
5. Время завершения (дата)
6. Список объектов типа задача (по умолчанию не содержит данных)

15. Получение информации о всех посылках всех пользователей системы.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должен входить список объектов со следующими свойствами:

1. Айди решения в данной системе
2. Код программы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

3. Язык программирования, на котором написано данное решение
4. Статус посылки
5. Затраченная память на исполнение программы
6. Затраченное время на исполнение программы
7. Номер теста, в котором возникла ошибка
8. Название контеста, к которому относится решение
9. Название задачи, к которой относится решение

16. Получение информации о роли пользователя системы.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

Для реализации данного функционала не должно требоваться отправление никаких данных на сервер.

В состав отправляемых с сервера данных должна входить строка, содержащая название роли пользователя.

17. Получения информации о контесте по его id в системе.

Должно иметься ограничение доступа к функционалу посредством аутентификации по JWT-токену.

В состав отправляемых на сервер данных должно входить свойство, содержащее айди контеста в данной системе.

В состав отправляемых с сервера данных должен входить список объектов со следующими свойствами:

1. Айди контеста в текущей системе
2. Айди контеста в системе тестирования Ejudge
3. Название
4. Время начала (дата)
5. Время завершения (дата)
6. Список объектов типа задача, где задача содержит свойства
 1. Название
 2. Условие
 3. Ограничение по памяти
 4. Ограничение по времени

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

5. Количество попыток, разрешенное для отправки уникального решения пользователем

6. Классификация задачи

4.2. Описание алгоритма и функционирования программы**4.2.1. Описание функционирования регистрации нового пользователя в систему**

С клиентской части поступает POST запрос с данными для регистрации нового пользователя в системе.

1. Происходит проверка наличия пользователя с указанным email в базе данных. Если пользователь уже существует, функция возвращает на сторону клиента ошибку 400 (неверный запрос) с сообщением "Email address already in use».
2. Выполняется хэширование пароля пользователя.
3. Создается новый объект User, содержащий информацию о новом пользователе.
4. Объект User сохраняется в базе данных.
5. Создается новый объект UserInfo, содержащий персональную информацию о пользователе такую как его имя, фамилия и отчество.
6. Объект UserInfo сохраняется в базе данных.
7. Создается новый объект Department, содержащий информацию об образовательной программе (департаменте).
8. Объект Department сохраняется в базе данных.
9. Создается новый объект DepartmentConn, связывающий пользователя и образовательную программу (департамент).
10. Объект DepartmentConn сохраняется в базе данных.
11. Создается новый объект EdGroup, содержащий информацию об учебной группе.
12. Объект EdGroup сохраняется в базе данных.
13. Создается новый объект EdGroupConn, связывающий учебную группу и образовательную программу (департамент).
14. Объект EdGroupConn сохраняется в базе данных.
15. Происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с сообщением "User was successfully created and registered".

Выполнение регистрации пользователя в систему обеспечивает транзакционность выполняемых операций, что означает, что все операции будут либо выполнены успешно, либо откатятся в случае возникновения ошибки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

4.2.2. Получение информации о профиле учащегося

С клиентской части поступает GET запрос на получение информации о профиле учащегося.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит извлечение информации об учащемся по его email из таблицы "users" в базе данных.
3. Проверяется, существует ли запись об учащемся в таблице "users". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "No user with email: <email> found".
4. Происходит извлечение персональной информации об учащемся из таблицы «users_info» в базе данных.
5. Проверяется, существует ли запись о персональной информации учащегося в таблице «users_info». Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "No user personal info with email: <email> found".
6. Происходит извлечение связи учащегося с образовательной программой (департаментом) из таблицы «departments_conn» в базе данных.
7. Проверяется, существует ли запись о связи учащегося с образовательной программой (департаментом) в таблице «departments_conn». Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "User with email: <email> has no department".
8. Происходит извлечение информации об образовательной программе (департаменте) из таблицы "departments" в базе данных по id.
9. Проверяется, существует ли запись об образовательной программе (департаменте) в таблице "departments". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "User with email: <email> has no department name".
10. Происходит извлечение связи образовательной программы с образовательной группой из таблицы «ed_groups_conn» в базе данных.
11. Проверяется, существует ли запись о связи образовательной программы с образовательной группой в таблице «ed_groups_conn». Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "User with email: <email> has no education group".

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

12. Происходит извлечение информации об образовательной группе из таблицы «ed_groups» в базе данных по id.
13. Проверяется, существует ли запись об образовательной группе в таблице "ed_groups". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "User with email: <email> has no education name".
14. Происходит создание объекта, содержащего всю собранную информацию об учащемся.
15. Происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом учащегося, отправляемым в теле ответа.

4.2.3. Получение информации о профиле учителя

С клиентской части поступает GET запрос на получение информации о профиле учителя.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит извлечение информации об учителе по его email из таблицы "users" в базе данных.
3. Проверяется, существует ли запись об учителе в таблице "users". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "No user with email: <email> found".
4. Происходит извлечение персональной информации об учителе из таблицы «users_info» в базе данных.
5. Проверяется, существует ли запись о персональной информации учителя в таблице «users_info». Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "No user personal info with email: <email> found".
6. Происходит создание объекта, содержащего всю собранную информацию об учителе.
7. Происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом учителя, отправляемым в теле ответа.

4.2.4. Авторизация пользователя в систему

С клиентской части поступает POST запрос на выполнение авторизации в системе.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

1. Происходит попытка авторизации пользователя с помощью полученного со стороны клиента email и пароля. Если авторизация проходит успешно, создается новый объект, содержащий информацию о пользователе.
2. Проверяется, был ли пользователь успешно авторизован. Если да, генерируется JWT-токен и происходит получение роли пользователя.
3. Происходит создание объекта, содержащего сгенерированный токен и роль пользователя.
4. Если авторизация не прошла успешно, на сторону клиента возвращается ответ с кодом 400 (неверный запрос) с сообщением "Invalid user request".
5. Если возникло исключение при выполнении авторизации, на сторону клиента возвращается ответ с кодом 400 (неверный запрос) с сообщением об ошибке, полученным из исключения.
6. Если ошибок не возникло, происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом, отправляемым в теле ответа.

4.2.5. Добавление нового преподавателя в систему с помощью администраторских прав

С клиентской части поступает POST запрос на добавление нового преподавателя в систему.

1. Происходит проверка аутентификации админа на основе Basic Auth. Если переданные данные неверные, на сторону клиента возвращается ответ с кодом 401 (Unauthorized).
2. Проверяется, что преподаватель с указанным адресом электронной почты (переданной в теле запроса) еще не зарегистрирован в системе путем проверки наличия записи с таким email в таблице пользователей в базе данных.
3. Выполняется хэширование пароля преподавателя.
4. Создается новый объект User, содержащий информацию о новом преподавателе.
5. Объект User сохраняется в базе данных.
6. Создается новый объект UserInfo, содержащий персональную информацию о преподавателе, такую как его имя, фамилия и отчество.
7. Объект UserInfo сохраняется в базе данных.
8. Происходит возвращение ответа на сторону клиента с кодом 201 (создано) с сообщением "New teacher was successfully created and registered".

Выполнение добавления нового преподавателя в систему обеспечивает транзакционность выполняемых операций, что означает, что все операции будут либо выполнены успешно, либо откатятся в случае возникновения ошибки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1**4.2.6. Получение информации о рейтинге, отсортированном по оценкам пользователя в порядке убывания**

С клиентской части поступает GET запрос на получение рейтинга пользователей по оценкам.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит получение всех пользователей из репозитория «users».
3. Полученные пользователи сортируются по среднему баллу пользователя по всем контестам в убывающем порядке.
4. Полученные пользователи сортируются по количеству контестов, в которых они приняли участие, в убывающем порядке.
5. Происходит создание нового объекта, содержащего список отсортированных пользователей по оценкам в порядке убывания.
6. Если в ходе создания объекта для ответа возникла ошибка, на сторону клиента возвращается ответ с кодом 400 (неверный запрос) с соответствующим сообщением возникшей ошибки.

Если ошибок не возникло, происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом, отправляемым в теле ответа.

4.2.7. Получение информации о рейтинге, отсортированном по количеству задач, пользователей в порядке убывания

С клиентской части поступает GET запрос на получение рейтинга пользователей по количеству задач.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит получение всех пользователей из репозитория «users».
3. Полученные пользователи сортируются по количеству контестов, в которых они приняли участие, в убывающем порядке.
4. Происходит создание нового объекта, содержащего список отсортированных пользователей по количеству задач в порядке убывания.
5. Если в ходе создания объекта для ответа возникла ошибка, на сторону клиента возвращается ответ с кодом 400 (неверный запрос) с соответствующим сообщением возникшей ошибки.
6. Если ошибок не возникло, происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом, отправляемым в теле ответа.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

4.2.8. Скачивание информации о рейтинге, отсортированном по оценкам, пользователей в порядке убывания в формате .csv

С клиентской части поступает GET запрос на скачивание рейтинга пользователей по оценкам в формате .csv.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит получение всех пользователей из репозитория «users».
3. Полученные пользователи сортируются по среднему баллу пользователя по всем контестам в убывающем порядке.
4. Полученные пользователи сортируются по количеству контестов, в которых они приняли участие, в убывающем порядке.
5. Происходит создание нового объекта, содержащего список отсортированных пользователей по количеству задач в порядке убывания.
6. Если в ходе создания объекта для ответа возникла ошибка, на сторону клиента возвращается ответ с кодом 400 (неверный запрос) с соответствующим сообщением возникшей ошибки.
7. Происходит конвертирование объектов списка и запись в преобразованных данных в файл формата .csv.
8. Создаются заголовки с указанием типа контента (application/octet-stream), чтобы браузер понимал, что файл и содержимое заголовка Content-Disposition указывают на то, что файл должен быть скачан, а не просмотрен в браузере.
9. Создается файловый системный ресурс с файлом .csv, который был создан в пункте 7.
10. На сторону клиента возвращается ответ с кодом 200 (успешный запрос) и файловым потоком в теле ответа запроса с заголовками, созданными в пункте 8.
11. Если во время выполнения происходит исключение, на сторону клиента возвращается ошибка 400 (неверный запрос) с сообщением об ошибке в теле ответа.

4.2.9. Скачивание информации о рейтинге, отсортированном по количеству задач, пользователей в порядке убывания в формате .csv

С клиентской части поступает GET запрос на получение рейтинга пользователей по количеству задач в формате .csv.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

2. Происходит получение всех пользователей из репозитория «users».
3. Полученные пользователи сортируются по количеству контестов, в которых они приняли участие, в убывающем порядке.
4. Происходит создание нового объекта, содержащего список отсортированных пользователей по количеству задач в порядке убывания.
5. Если в ходе создания объекта для ответа возникла ошибка, на сторону клиента возвращается ответ с кодом 400 (неверный запрос) с соответствующим сообщением возникшей ошибки.
6. Происходит конвертирование объектов списка и запись в преобразованных данных в файл формата .csv.
7. Создаются заголовки с указанием типа контента (application/octet-stream), чтобы браузер понимал, что файл и содержимое заголовка Content-Disposition указывают на то, что файл должен быть скачан, а не просмотрен в браузере.
8. Создается файловый системный ресурс с файлом .csv, который был создан в пункте 6.
9. На сторону клиента возвращается ответ с кодом 200 (успешный запрос) и файловым потоком в теле ответа запроса с заголовками, созданными в пункте 7.
10. Если во время выполнения происходит исключение, на сторону клиента возвращается ошибка 400 (неверный запрос) с сообщением об ошибке в теле ответа.

4.2.10. Получение информации о всех посылках (т.е. решениях) определённого пользователя

С клиентской части поступает GET запрос на получение всех посылок определенного пользователя.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит извлечение информации о пользователе по его email из таблицы "users" в базе данных.
3. Проверяется, существует ли запись о пользователе в таблице "users". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "No user with email: <email> found".
4. Происходит получение списка решений пользователя в таблице «user_solutions» в базе данных.
5. Если список пуст, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "User with email: <email> has no solutions".

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

6. Создается объект со списком всех решений пользователя.
7. На сторону клиента возвращается ответ с кодом 200 (успешный запрос) и созданным объектом, отправляемым в теле ответа.
8. Если во время выполнения происходит исключение, на сторону клиента возвращается ошибка 400 (неверный запрос) с сообщением об ошибке в теле ответа.

4.2.11. Добавление решения определенной задачи определенного контекста в систему

С клиентской части поступает POST запрос на добавление решения определенной задачи определенного пользователя.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит извлечение информации о пользователе по его email из таблицы "users" в базе данных.
3. Проверяется, существует ли запись о пользователе в таблице "users". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "No user with email: <email> found".
4. Создается новый объект Language, содержащий информацию о языке программирования, на котором было написано решение пользователя.
5. Объект Language сохраняется в базе данных.
6. Создается новый объект Status, содержащий информацию о статусе, отправляемого решения.
7. Объект Status сохраняется в базе данных.
8. Происходит извлечение информации о контексте по его названию из таблицы "contests" в базе данных.
9. Происходит извлечение информации о задачах данного контекста из таблиц «tasks_of_contest», "tasks" в базе данных.
10. Создается новый объект UniqueContestTask, содержащий информацию о контексте и задаче, к которой она относится.
11. Объект UniqueContestTask сохраняется в базе данных.
12. Создается новый объект Solution, содержащий информацию о решении пользователя.
13. Объект Solution сохраняется в базе данных.
14. Создается новый объект UserSolution, связывающий решение и пользователя.
15. Объект UserSolution сохраняется в базе данных.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

16. Если во время выполнения происходит исключение, на сторону клиента возвращается ошибка 400 (неверный запрос) с сообщением об ошибке в теле ответа.

17. Происходит возвращение ответа на сторону клиента с кодом 201 (создано) с сообщением "New solution was successfully added".

Выполнение добавления нового решения пользователя в систему обеспечивает транзакционность выполняемых операций, что означает, что все операции будут либо выполнены успешно, либо откатятся в случае возникновения ошибки.

4.2.12. Добавление нового контекста с задачами преподавателем в систему

С клиентской части поступает POST запрос на добавление контекста с задачами преподавателем.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит извлечение информации о пользователе по его email из таблицы "users" в базе данных.
3. Проверяется, существует ли запись о пользователе в таблице "users". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "The user with such email does not exist".
4. Если найденный пользователь не имеет роль учителя, на сторону клиента возвращается объект с кодом ошибки 400 (неверный запрос) с сообщением "The user with role: <role> is not permitted to add contest».
5. Создается новый объект Contest, содержащий информацию о добавляемом контексте.
6. Объект Contest сохраняется в базе данных.
7. Для каждой задачи создается новый объект Classification, содержащий информацию о классификации добавляемого контекста.
8. Объект Classification сохраняется в базе данных.
9. Создается новый объект Task, содержащий информацию о добавляемой задаче.
10. Объект Task сохраняется в базе данных.
11. Создается новый объект TaskConn, содержащий информацию о связи контекста и задачи, относящейся к этому контексту.
12. Объект TaskConn сохраняется в базе данных.
13. Происходит возвращение ответа на сторону клиента с кодом 201 (создано) с сообщением "New contest was successfully added to system".

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

Выполнение добавления нового конкурса с задачами преподавателем в систему обеспечивает транзакционность выполняемых операций, что означает, что все операции будут либо выполнены успешно, либо откатятся в случае возникновения ошибки.

4.2.13. Получение информации о всех конкурсах, в которых пользователь принимает участие.

С клиентской части поступает GET запрос на получение информации о конкурсах, в которых пользователь принимает участие.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит извлечение информации о пользователе по его email из таблицы "users" в базе данных.
3. Проверяется, существует ли запись о пользователе в таблице "users". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "The user with such email does not exist».
4. Происходит получение всех связей между найденным пользователем и конкурсами.
5. Создаётся пустой список для хранения объектов, которые будут предоставлены в качестве результата.
6. Происходит проход по всем связям конкурсов, сбор всей необходимой информации, включая задачи данного конкурса, и добавление объектов в результирующий список.
7. Если во время выполнения происходит исключение, на сторону клиента возвращается ошибка 400 (неверный запрос) с сообщением об ошибке в теле ответа.
8. Если ошибок не возникло, происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом, отправляемым в теле ответа.

4.2.14. Получение информации о всех конкурсах, имеющихся в системе

С клиентской части поступает GET запрос на получение информации о всех конкурсах, которые есть в системе.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Создаётся пустой список для хранения объектов, которые будут предоставлены в качестве результата.
3. Происходит извлечение списка всех конкурсов из таблицы "contests" в базе данных.
4. Происходит запись данных о всех конкурсах в список.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

5. Происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом, отправляемым в теле ответа.

4.2.15. Получение информации о всех посылках всех пользователей системы

С клиентской части поступает GET запрос на получение информации о всех посылках всех пользователей системы.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит извлечение информации о пользователе по его email из таблицы "users" в базе данных.
3. Проверяется, существует ли запись о пользователе в таблице "users". Если запись не найдена, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "No user with email: <email> found».
4. Если найденный пользователь не имеет роль учителя, на сторону клиента возвращается объект с кодом ошибки 400 (неверный запрос) с сообщением "User with email: <email> is not a teacher and has role: <role>».
5. Происходит извлечение списка всех решений из таблицы "solutions" в базе данных.
6. Если список решений пуст, на сторону клиента возвращается ответ с кодом ошибки 400 (неверный запрос) с сообщением "There are no solutions in system yet".
7. Создаётся пустой список для хранения объектов, которые будут предоставлены в качестве результата.
8. Происходит проход по всем полученным ранее решениям, сбор всей необходимой информации и добавление объектов в результирующий список.
9. Если во время выполнения происходит исключение, на сторону клиента возвращается ошибка 400 (неверный запрос) с сообщением об ошибке в теле ответа.
10. Если ошибок не возникло, происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом, отправляемым в теле ответа.

4.2.16. Получение информации о роли пользователя системы

С клиентской части поступает GET запрос на получение информации о роли пользователя в системе.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

2. Происходит извлечение информации о роли пользователя по его email из таблицы "users" в базе данных.
3. Если найденный пользователь не имеет роли, на сторону клиента возвращается объект с кодом ошибки 400 (неверный запрос) с сообщением "Role of user with email: <email> was not found».
4. Если ошибок не возникло, происходит возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с ролью пользователя, отправляемой в теле ответа.

4.2.17. Получение информации о контексте по его id в системе

С клиентской части поступает GET запрос на получение информации о контексте по его айди.

1. Происходит валидация JWT-токена. Если токен не валидный, на сторону клиента возвращается ответ с кодом ошибки 403 (доступ запрещен).
2. Происходит извлечение информации о контексте по его айди из таблицы «contests» в базе данных.
3. Если контекст с таким айди не существует, на сторону клиента возвращается объект с кодом ошибки 400 (неверный запрос) с сообщением "The contest with such id does not exist».
4. Происходит сбор информации о всех задачах данного контекста, а также классификации каждой задачи.
5. Если во время выполнения происходит исключение, на сторону клиента возвращается ошибка 400 (неверный запрос) с сообщением об ошибке в теле ответа.
6. Если ошибок не возникло, происходит формирование объекта контекста и возвращение ответа на сторону клиента с кодом 200 (успешный запрос) с созданным объектом, отправляемым в теле ответа.

4.3. Описание и обоснование выбора метода организации входных и выходных данных

Организация входных данных проекта реализована следующим образом. На endpoint-ы API со стороны клиента поступают запросы с заголовками и телом запроса (необязательно) в формате json.

Организация выходных данных проекта реализована следующим образом. Ответ на http запросы, приходящие со стороны клиента, содержит всю необходимую информацию, которая предоставляется в теле ответа (необязательно) в формате json или строки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1**4.4. Описание и обоснование состава технических и программных средств**

Минимальные требования к серверному оборудованию:

1. Сервер приложений.

1.1. Процессор: 4 ядра (8 логических потоков), частота – 2 ГГц и больше.

1.2. Оперативная память: 4 ГБ и больше.

1.3. Свободное дисковое пространство: 50 ГБ (зависит от размера хранимых в системе данных).

1.4. Пропускная способность сетевого интерфейса: 1 Гбит/с.

2. Сервер баз данных.

2.1. Тип накопителя: SSD.

2.2. Процессор: 4 ядра (8 логических потоков), частота – 2 ГГц и больше.

2.3. Оперативная память: 8 ГБ и больше.

2.4. Свободное дисковое пространство: 300 ГБ.

2.5. Пропускная способность сетевого интерфейса: 1 Гбит/с.

На данный момент серверное оборудование имеет следующие характеристики:

description: CPU

product: Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz

vendor: Intel Corp.

physical id: 400

bus info: cpu@0

version: 6.63.2

slot: CPU 0

size: 2GHz

capacity: 2GHz

width: 64 bits

memory:

description: System Memory

physical id: 1000

size: 2GiB

capacity: 2GiB

capabilities: ecc

configuration: errordetection=multi-bit-ecc

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

5. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

5.1. Предполагаемая потребность

Программа будет использоваться учащимися Лицея НИУ ВШЭ для решения задач по программированию и просмотра рейтинга учащихся в режиме реального времени, а также преподавателями Лицея НИУ ВШЭ для отслеживания количества решенных задач учащимися, контроля рейтинга, получения информации об оценках.

5.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

(текст или таблица – Яндекс Контест, Informatics, Timus Online judge, nlogn)

	Timus Online Judge	Яндекс Контест	Informatics	NLOGN	Разрабатываемое решение
Возможность регистрации без приглашения	+	+	+	-	-
Рейтинг по определенному соревнованию	+	+	+	?	+
Общий рейтинг пользователей системы	+	-	+	-	+
Возможность автоматического выставления оценки за соревнование	-	-	-	-	+
Возможность выгрузки таблицы с оценками (рейтингом) всех пользователей за соревнования	?	+	-	-	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

RU.17701729.05.10-01 81 01–1

Наличие архива задач, из которых можно создавать соревнования	+	-	+	-	+
Количество поддерживаемых языков программирования	18	30	16	16	4 (к 10.05.23)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

ПРИЛОЖЕНИЕ 1

Глоссарий

1. Фреймворк - программная платформа, которая предоставляет разработчикам готовые решения для создания приложений. Обычно включает в себя структуру файлов и папок, библиотеки и инструменты для разработки, которые упрощают создание приложений.

2. API (Application Programming Interface) - набор инструкций и структур данных, используемых приложениями для взаимодействия друг с другом. Обычно используется для доступа к определенной функциональности или данным приложения.

3. База данных - структурированное хранилище данных, которое позволяет хранить и организовывать информацию в виде таблиц, строк и столбцов. Часто используется для хранения информации, которую приложения могут использовать для своей работы.

4. Аутентификация - процесс проверки подлинности пользователя. Обычно осуществляется путем проверки его имени и пароля.

5. Авторизация - процесс предоставления пользователю доступа к определенным ресурсам или функциям после того, как он был аутентифицирован. Определяет, какие действия может выполнять пользователь в рамках приложения.

6. JWT-токен - формат токена, который используется для аутентификации и авторизации пользователей. Содержит зашифрованную информацию о пользователе и его правах, которую приложение может использовать для предоставления доступа.

7. Сервер - компьютер или программа, которая предоставляет ресурсы и услуги для других компьютеров или программ (клиентов). Серверы могут предоставлять доступ к файлам, базам данных, приложениям и другим ресурсам.

8. Бэкенд - часть приложения, которая работает на сервере и обрабатывает запросы клиентов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.10-01 81				
Инв. № подл.	Подп. и	Взам. инв.	Инв. №	Подп. и