**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  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  старший преподаватель департамента программной инженерии  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Павлочев  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга**  **Программа и методика испытаний**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.05.10-01 51 01-1-ЛУ** | | | | | |
|  | |  | | | |
| Исполнитель  студент группы БПИ219  \_\_\_\_\_\_\_\_\_ / С. О. Котовский /  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2024 г. | | | |

**Москва 2024**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.05.10-01 51 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга**  **Программа и методика испытаний**  **RU.17701729.05.10-01 51 01-1**  **Листов 22** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2024**

ОГЛАВЛЕНИЕ

[1. ОБЪЕКТ ИСПЫТАНИЙ 4](#_Toc161931806)

[1.1. Наименование программы и объекта, в котором программа используется 4](#_Toc161931807)

[1.2. Краткая характеристика области применения программы и объекта, в котором она используется 4](#_Toc161931808)

[2. ЦЕЛЬ ИСПЫТАНИЙ 5](#_Toc161931809)

[3. ТРЕБОВАНИЯ К ПРОГРАММЕ 6](#_Toc161931810)

[3.1. Требования к функциональным характеристикам 6](#_Toc161931811)

[3.2. Требования к интерфейсу 7](#_Toc161931812)

[3.3. Требования к надёжности 7](#_Toc161931813)

[4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 9](#_Toc161931814)

[4.1. Состав программной документации 9](#_Toc161931815)

[4.2. Специальные требования к программной документации 9](#_Toc161931816)

[5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ 10](#_Toc161931817)

[5.1. Технические средства, используемые во время испытаний 10](#_Toc161931818)

[5.2. Программные средства, используемые во время испытаний 10](#_Toc161931819)

[5.3. Порядок проведения испытаний 10](#_Toc161931820)

[6. МЕТОДЫ ИСПЫТАНИЙ ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ 12](#_Toc161931821)

[6.1. Испытание выполнения требований к программной документации 12](#_Toc161931822)

[6.2. Испытание требований к интерфейсу 12](#_Toc161931823)

[6.3. Испытание выполнения требований к функциональным характеристикам 12](#_Toc161931824)

[6.3.1. Испытание функции авторизации 12](#_Toc161931825)

[6.3.2. Испытание функции отображения страницы контестов 12](#_Toc161931826)

[6.3.3. Испытание функции отображения задач контеста 12](#_Toc161931827)

[6.3.4. Испытание функции отправки решения 13](#_Toc161931828)

[6.3.5. Испытание функции просмотра рейтинга 13](#_Toc161931829)

[6.3.6. Испытание функции просмотра профиля пользователя 13](#_Toc161931830)

[6.3.7. Испытание функции выхода из системы 14](#_Toc161931831)

[6.3.8. Испытание функции управления контестами (для администраторов) 14](#_Toc161931832)

[6.3.9. Испытание функции управления пользователями (для администратора) 14](#_Toc161931833)

[6.4. Испытание выполнения требований к надёжности 15](#_Toc161931834)

[7. МЕТОДЫ ИСПЫТАНИЙ для backend-прослойки 16](#_Toc161931835)

[7.1. Испытание выполнения требований к программной документации 16](#_Toc161931836)

[7.2. Испытание требований к интерфейсу API 16](#_Toc161931837)

[7.3. Испытание выполнения требований к функциональным характеристикам 16](#_Toc161931838)

[7.3.1. Испытание функции "Получение информации о пользователе" (Get user) 16](#_Toc161931839)

[7.3.2. Испытание функции "Получение статуса контеста" (Get contest) 17](#_Toc161931840)

[7.3.3. Испытание функции "Получение условия задачи" (Get problem) 17](#_Toc161931841)

[7.3.4. Испытание функции "Отправка решения на проверку" (Submit a task for checking) 17](#_Toc161931842)

[7.3.5. Испытание функции "Регистрация пользователей из CSV" (Register Users from CSV) 18](#_Toc161931843)

[7.3.6. Испытание функционала "Получение информации о пользователе" (Get user) 18](#_Toc161931844)

[7.3.7. Испытание функционала "Получение статуса контеста" (Get contest) 19](#_Toc161931845)

[7.3.8. Испытание функционала "Получение условия задачи" (Get problem) 19](#_Toc161931846)

[7.3.9. Испытание функционала "Отправка решения и получение результатов" (Submit a task for checking) 20](#_Toc161931847)

[7.3.10. Испытание функционала "Получение информации о посылке" (Get info about the run) 20](#_Toc161931848)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 22](#_Toc161931849)

# ОБЪЕКТ ИСПЫТАНИЙ

## Наименование программы и объекта, в котором программа используется

Наименование темы разработки: «Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга».

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

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

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

«Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга» - это комплекс мероприятий по доработке курсового проекта «Система проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга», который, в свою очередь, является набором программных инструментов для оценивания решений задач на разных языках программирования для учащихся Лицея НИУ ВШЭ с рейтингом для каждого учащегося.

Лицей НИУ ВШЭ – подразделение НИУ ВШЭ.

# ЦЕЛЬ ИСПЫТАНИЙ

Цель проведения испытаний – проверить доработанную версию программного продукты на соответствие функциональным требованиям и отдельным требованиям к надежности, изложенных в документе «Техническое задание» к данной программе.

# ТРЕБОВАНИЯ К ПРОГРАММЕ

## Требования к функциональным характеристикам

В рамках обновления и оптимизации проекта «Улучшенная версия веб-приложения на Vue.js с интеграцией backend-прослойки для взаимодействия с системой Ejudge» предполагается доработка существующего функционала и добавление новых возможностей. Изменения затронут как веб-приложение, так и backend-часть, с целью улучшения пользовательского опыта, безопасности, производительности и удобства в использовании.

В веб-приложении планируется реализация следующих функций:

* Улучшенная авторизация пользователя с обновленным визуальным оформлением и дополнительной защитой данных.
* Реорганизация и обновление навигационного меню для обеспечения более интуитивного доступа к основным разделам приложения.
* Доработка страницы профиля пользователя с подробным отображением информации и улучшенным интерфейсом.
* Обновление механизма отображения списка контестов и задач, улучшение визуальной составляющей и функциональности.
* Внедрение усовершенствованного механизма отправки решений задач и просмотра результатов тестирования.
* Удаление нижнего колонтитула для освобождения визуального пространства и повышения удобства использования интерфейса.
* Реализация админ-панели для преподавателей с возможностями управления контестами и аккаунтами студентов.

В backend-части будут реализованы следующие функции:

* Интеграция с API Ejudge для обеспечения работы основных функций веб-приложения без необходимости использования сторонних скриптов.
* Улучшенная модульная архитектура, подробно описанная в пояснительной записке. Переработанная архитектура повышает читаемость, простоту поддержки и масштабируемость кода.

Для каждой из указанных функций предусмотрено тщательное тестирование с целью проверки соответствия требованиям технического задания, безопасности, производительности и удобства использования. Подробные методики испытаний и критерии приемки будут описаны в разделе 6 документации, включая сценарии тестирования и ожидаемые результаты для каждой функции.

## Требования к интерфейсу

Для обеспечения высокого уровня пользовательского опыта и интуитивной навигации интерфейс веб-приложения должен соответствовать следующим требованиям:

* **Ясность действий**: Каждый интерактивный элемент интерфейса, такой как кнопки, ссылки, иконки, должен иметь четкое визуальное или текстовое указание своего назначения. Использование иконок должно быть интуитивно понятным, а текстовые подписи - лаконичными и информативными. Макет проекта доступен в Figma:
* **Цветовая гармония**: Элементы интерфейса одинакового цвета должны быть разделены пространством или рамкой, чтобы избежать слияния и улучшить читаемость. Контраст между текстом и фоном должен быть достаточным для обеспечения хорошей видимости на всех типах устройств.
* **Адаптивность дизайна**: Все элементы интерфейса должны корректно отображаться на различных размерах экрана, включая мобильные устройства и планшеты. Информация не должна обрезаться или выходить за пределы видимой области экрана, обеспечивая комфортное чтение и взаимодействие со всеми функциями приложения без необходимости горизонтальной прокрутки.

## Требования к надёжности

Надежность является ключевым фактором для обеспечения бесперебойной работы приложения. Для достижения высокой стабильности и безопасности пользовательского опыта, приложение должно удовлетворять следующим критериям:

* **Стабильность работы**: Приложение должно обеспечивать стабильную работу в различных условиях использования, включая пиковые нагрузки и возможные сетевые перебои. Оно не должно завершаться с ошибкой или "падать" при стандартном использовании, включая навигацию по интерфейсу, взаимодействие с элементами управления и выполнение операций с данными.
* **Обработка исключений**: Все возможные исключительные ситуации, такие как ошибки ввода данных пользователями, сбои в сетевых запросах или ошибках доступа к серверным ресурсам, должны быть корректно обработаны. Пользователю следует предоставлять понятные сообщения об ошибках с указанием возможных путей решения проблемы.
* **Резервное копирование и восстановление**: Должны быть реализованы механизмы резервного копирования важных данных и их восстановления в случае аварийных ситуаций или технических сбоев, чтобы минимизировать возможные потери информации и обеспечить непрерывность доступа к функционалу приложения.
* **Тестирование**: Перед выпуском любых обновлений приложения необходимо проводить тщательное тестирование всех функций и компонентов, включая нагрузочное тестирование, для выявления и устранения потенциальных проблем производительности и надежности.

# ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

## Состав программной документации

«Реализация «Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга». Техническое задание (ГОСТ 19.201-78);

«Реализация «Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга». Программа и методика испытаний (ГОСТ 19.301-78);

«Реализация «Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга». Пояснительная записка (ГОСТ 19.404-79);

«Реализация «Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга». Руководство оператора (ГОСТ 19.505-79);

«Реализация «Разработка нового и доработка существующего функционала системы проверки решений по программированию для Лицея НИУ ВШЭ с автоматизированным подсчётом рейтинга». Текст программы (ГОСТ 19.401-78);

## Специальные требования к программной документации

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1.).

Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ».

Техническое задание и пояснительная записка, титульные листы других документов должны быть подписаны руководителем разработки и исполнителем.

Документация и программа сдаётся в электронном виде в формате .pdf или .docx. в архиве формата .zip или .rar.

За три дня до защиты комиссии все материалы курсового проекта: программная документация, программный проект, исполняемый файл, отзыв руководителя, отчёт системы Антиплагиат должны быть загружены одним или несколькими архивами в проект дисциплины «Курсовой проект» в личном кабинете в информационной образовательной среде SmartLMS НИУ ВШЭ.

# СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

## Технические средства, используемые во время испытаний

Для веб-приложения:

**Компьютеры с операционными системами Windows, macOS и Linux** для тестирования кросс-платформенной совместимости и обеспечения работоспособности веб-приложения в различных средах.

**Мобильные устройства с различными версиями Android и iOS** для проверки адаптивности дизайна и функциональности веб-приложения на мобильных устройствах.

**Современные веб-браузеры** (Google Chrome, Mozilla Firefox, Safari, Edge) в последних версиях для тестирования совместимости и производительности веб-приложения.

Для backend-прослойки:

**Серверы** для развертывания и тестирования backend-прослойки, могут быть локальными машинами или облачными серверами, поддерживающими необходимое программное обеспечение.

## Программные средства, используемые во время испытаний

Для веб-приложения:

* **Vue.js Devtools** для отладки и тестирования веб-приложения на Vue.js.
* **Browser Developer Tools** в используемых браузерах для анализа элементов страницы, сетевых запросов и производительности.
* **Postman или другие средства тестирования API** для тестирования и валидации запросов к backend.

Для backend-прослойки:

* **Node.js** и соответствующие библиотеки для запуска и тестирования серверного кода.
* **Postman** для эмуляции запросов к API и проверки ответов от сервера.
* **Docker** для изоляции среды выполнения и обеспечения единообразия тестирования между различными средами.

## Порядок проведения испытаний

* **Подготовка среды:** Настройка и подготовка всех необходимых технических и программных средств для испытаний, включая развертывание последних версий веб-приложения и backend-прослойки на тестовых серверах.
* **Функциональное тестирование:** Последовательная проверка всех функций веб-приложения и backend-прослойки с использованием заранее подготовленных тестовых сценариев. Тестирование включает в себя авторизацию, работу с контестами и задачами, отправку и получение решений, а также взаимодействие с системой Ejudge через backend-прослойку.
* **Тестирование интерфейса:** Проверка адаптивности и корректности отображения веб-приложения на различных устройствах и в разных браузерах.
* **Тестирование безопасности:** Проведение проверок на наличие уязвимостей в веб-приложении и backend-прослойке, включая тестирование на SQL-инъекции, XSS-атаки и проверку механизмов аутентификации.

# МЕТОДЫ ИСПЫТАНИЙ ДЛЯ ВЕБ-ПРИЛОЖЕНИЯ

## Испытание выполнения требований к программной документации

В рамках этого испытания проводится тщательная проверка программной документации веб-приложения, включая техническое задание, руководства пользователя и системного администратора, а также другие сопутствующие документы.

## Испытание требований к интерфейсу

Испытание интерфейса веб-приложения включает проверку его соответствия всем заявленным требованиям, включая четкость и понятность всех элементов управления, корректное отображение на различных устройствах и разрешениях экрана, а также общую удобство использования. Важно убедиться, что элементы интерфейса не перекрывают друг друга, имеют четкие обозначения и визуальные подсказки, облегчающие навигацию по приложению.

## Испытание выполнения требований к функциональным характеристикам

### Испытание функции авторизации

**Цель испытания:** Проверить процесс авторизации пользователей, включая ввод логина и пароля, а также обработку ошибок входа.

**Шаги:**

* Попытка входа с корректными учетными данными.
* Попытка входа с некорректными данными для проверки сообщений об ошибках.

**Критерии успеха:** Успешный вход в систему с корректными данными и адекватное информирование пользователя при ошибках.

### Испытание функции отображения страницы контестов

**Цель испытания:** Проверить отображение списка контестов и возможность перехода к детальной информации о каждом из них.

**Шаги:**

* Открытие страницы контестов после авторизации.
* Переход через карточки контестов для проверки переадресации на страницу решения контеста.

**Критерии успеха:** На странице корректно отображаются все доступные контесты, и при нажатии на карточку контеста осуществляется переход к его задачам.

* + 1. Испытание функции отображения задач контеста

**Цель испытания:** Убедиться в корректности отображения списка задач в выбранном контесте и доступности детальной информации по каждой задаче.

**Шаги:**

* Переход на страницу выбранного контеста через нажатие на его карточку на странице контестов.
* Проверка отображения списка задач контеста.
* Проверка наличия детальной информации задачи на карточке задачи.

**Критерии успеха:** Список задач контеста отображается корректно. При нажатии на карточку задачи открывается страница с полным условием задачи, ограничениями и примерами ввода/вывода.

* + 1. Испытание функции отправки решения

**Цель испытания:** Проверить процесс отправки решения задачи, включая выбор языка программирования, ввод или загрузку кода, и получение результата проверки.

**Шаги:**

* Открытие страницы конкретной задачи контеста.
* Выбор языка программирования из предложенного списка.
* Ввод кода решения в текстовое поле или его загрузка через интерфейс приложения.
* Нажатие на кнопку «Отправить решение».

**Критерии успеха:** После отправки решения отображается информационный блок с результатом проверки, включая вердикт системы и, при необходимости, номер теста, на котором решение не прошло проверку.

* + 1. Испытание функции просмотра рейтинга

**Цель испытания:** Проверить корректность отображения страницы рейтинга, а также правильность сортировки пользователей.

**Шаги:**

* Переход на страницу рейтинга по оценкам через навигационное меню.
* Проверка отображения и сортировки списка пользователей.
* Переход на страницу рейтинга по задачам и оценкам, проверка отображения и сортировки списка.

**Критерии успеха:** Рейтинги отображаются корректно, пользователи отсортированы в соответствии с их достижениями.

* + 1. Испытание функции просмотра профиля пользователя

**Цель испытания:** Убедиться, что пользователь может просматривать информацию в своем профиле.

**Шаги:**

* Переход в профиль пользователя через навигационное меню.
* Проверка отображения текущей информации пользователя.

**Критерии успеха:** Информация в профиле отображается.

* + 1. Испытание функции выхода из системы

**Цель испытания:** Проверить, что функция выхода из системы работает корректно, завершая сессию пользователя и переводя его на страницу авторизации.

**Шаги:**

* Авторизоваться в системе с использованием корректных учетных данных.
* Перейти в профиль пользователя.
* Нажать на кнопку «Выход».
* Проверить, что произошел выход из системы и отображается страница авторизации.

**Критерии успеха:** После нажатия на кнопку «Выход», пользователь должен быть перенаправлен на страницу авторизации, и доступ к функциям системы, требующим авторизации, должен быть закрыт.

* + 1. Испытание функции управления контестами (для администраторов)

**Цель испытания:** Проверить возможности администратора по импорту, контестов в систему.

**Шаги:**

* Войти в систему под учетной записью администратора.
* Перейти в “Создание контеста”.
* Импортировать контест, заполнив необходимую информацию.

**Критерии успеха:** Контест успешно импортируется, администратору выводится соответствубщее сообщение.

* + 1. Испытание функции управления пользователями (для администратора)

**Цель испытания:** Проверить возможности администратора по импорту, контестов в систему.

**Шаги:**

* Войти в систему под учетной записью администратора.
* Перейти в “Регистрация пользователей”.
* Пройти процедуру регистрации пользователя из введённых данных, а также csv-файла.

**Критерии успеха:** Пользователи успешно создаются, администратору выводится соответствующее сообщение. Авторизация в системе под указанными данными проходит успешно.

## Испытание выполнения требований к надёжности

Испытание считается пройденным, если после выполнения всех испытаний из пункта 6.3 5 раз подряд веб-приложение показывало стабильно-высокие показатели надежности и времени отклика.

# МЕТОДЫ ИСПЫТАНИЙ для backend-прослойки

## Испытание выполнения требований к программной документации

**Цель испытания:** Проверить полноту и актуальность документации backend-прослойки, включая API документацию, описания архитектуры системы, инструкции по развёртыванию и настройке, а также руководства по безопасности и обработке ошибок.

**Шаги:**

Визуальная проверка наличия всех необходимых документов.

Сверка содержания документов с реальным функционалом и архитектурой backend-прослойки

Проверка актуальности примеров использования API и кодов ответов.

Оценка доступности и понимания документации для новых разработчиков и администраторов системы.

Проверка наличия разделов по безопасности и рекомендации по обеспечению безопасности данных.

**Критерии успеха:** Вся программная документация полностью соответствует текущему состоянию backend-прослойки, легко доступна для конечных пользователей, разработчиков и администраторов системы, и содержит актуальные данные и рекомендации.

## Испытание требований к интерфейсу API

**Цель испытания:** Проверить соответствие API backend-прослойки заявленным требованиям интерфейса, включая четкость и понятность структуры запросов и ответов, а также корректное функционирование всех эндпоинтов.

**Шаги:**

* Проверка соответствия описания эндпоинтов в документации реальному их поведению.
* Тестирование эндпоинтов на корректность ответов при валидных и невалидных запросах.
* Проверка обработки ошибок и корректности кодов ответов для различных сценариев использования API.
* Оценка легкости интеграции с API для внешних разработчиков.

**Критерии успеха:** API предоставляет четкие и понятные интерфейсы для взаимодействия, корректно обрабатывает запросы и ошибки, и его реализация полностью соответствует документации.

## Испытание выполнения требований к функциональным характеристикам

* + 1. Испытание функции "Получение информации о пользователе" (Get user)

**Цель испытания:** Проверить корректность получения информации о пользователе по заданным идентификаторам контеста и пользователя.

**Шаги:**

Отправить запрос на получение информации о пользователе с корректными параметрами contest\_id и other\_user\_id.

Проверить, что ответ содержит корректные данные о пользователе.

**Критерии успеха:** В ответе сервера содержится актуальная информация о пользователе, соответствующая отправленным идентификаторам. Ответ сервера соответствует ожидаемому формату JSON.

* + 1. Испытание функции "Получение статуса контеста" (Get contest)

**Цель испытания:** Убедиться, что информация о статусе контеста возвращается корректно.

**Шаги:**

Отправить запрос на получение статуса конкретного контеста с корректным contest\_id.

Проверить, что ответ содержит детальную информацию о состоянии контеста, включая список задач и доступные языки программирования.

**Критерии успеха:** Ответ сервера содержит полную и точную информацию о запрашиваемом контесте, включая статусы и время его проведения. Данные соответствуют формату JSON и структуре, описанной в документации API.

* + 1. Испытание функции "Получение условия задачи" (Get problem)

**Цель испытания:** Проверить, что условия задачи возвращаются правильно и полностью.

**Шаги:**

Отправить запрос на получение условия задачи, указав contest\_id и номер problem.

Проверить, что ответ содержит полное условие задачи, включая примеры входных и выходных данных.

**Критерии успеха:** Ответ сервера точно соответствует запрошенной задаче, включая её условие, ограничения и примеры. Ответ соответствует ожидаемому формату JSON и содержит все необходимые данные.

* + 1. Испытание функции "Отправка решения на проверку" (Submit a task for checking)

**Цель испытания:** Убедиться в корректности процесса отправки решения на задачу и получении результатов его проверки.

**Шаги:**

Отправить решение задачи, используя указанные contest\_id, problem, lang\_id и файл с решением.

Ожидать ответа с идентификатором посылки.

Используя полученный идентификатор посылки, запросить информацию о результате проверки.

**Критерии успеха:** Сервер корректно принимает решение и возвращает уникальный идентификатор посылки. Далее можно успешно получить информацию о результате проверки, включая статус решения (например, "Принято", "Ошибка компиляции"). Данные соответствуют формату JSON и содержат актуальную информацию о результате проверки.

* + 1. Испытание функции "Регистрация пользователей из CSV" (Register Users from CSV)

**Цель испытания:** Проверить механизм массовой регистрации пользователей через загрузку файла CSV, включая корректное создание аккаунтов для пользователей с уникальными электронными адресами и обработку ошибок для дублирующихся данных.

**Шаги:**

Подготовить два файла CSV: один с уникальными данными пользователей и другой с дублирующимися электронными адресами.

Загрузить первый файл через CURL команду или через интерфейс API, указав тип пользователей (например, студенты).

Проверить в базе данных, что все пользователи из первого файла успешно зарегистрированы.

Загрузить второй файл с дублирующимися данными и отследить логи сервера на наличие ошибок регистрации из-за дублирования электронных адресов.

Проверить ответ API на загрузку второго файла, ожидая сообщения об ошибках для дублирующихся электронных адресов.

**Критерии успеха:** Все пользователи из первого файла должны быть успешно зарегистрированы в системе без ошибок. При попытке загрузки второго файла с дублирующимися электронными адресами система должна корректно обработать ошибки и отразить их в логах сервера или в ответе API, предотвращая регистрацию пользователей с уже существующими в системе электронными адресами.

* + 1. Испытание функционала "Получение информации о пользователе" (Get user)

**Цель испытания:** Проверить корректность работы эндпоинта по получению информации о конкретном пользователе, включая проверку прав доступа и валидность возвращаемых данных.

**Шаги:**

Выполнить запрос к API для получения информации о пользователе, используя корректные идентификаторы контеста и пользователя.

Проверить структуру и содержание ответа, удостоверившись, что возвращаемые данные соответствуют запрошенному пользователю и содержат все необходимые атрибуты (например, ФИО, электронный адрес).

Выполнить запрос с некорректными или несуществующими идентификаторами для проверки обработки ошибок и соответствующих сообщений об ошибках.

**Критерии успеха:** API корректно возвращает информацию о запрошенном пользователе при использовании валидных идентификаторов. При запросах с ошибками или несуществующими данными API возвращает соответствующие сообщения об ошибках.

* + 1. Испытание функционала "Получение статуса контеста" (Get contest)

**Цель испытания:** Проверить точность и актуальность информации, возвращаемой эндпоинтом для получения статуса контеста, включая список задач, доступные языки программирования, временные рамки проведения.

**Шаги:**

Выполнить запрос к API для получения статуса конкретного контеста по его идентификатору.

Проверить, что ответ содержит актуальную информацию о контесте, включая название, статус (начался, идет, закончился), доступные языки программирования, список задач и соответствующие временные рамки.

Проверить обработку запросов для несуществующих контестов, чтобы убедиться, что API возвращает корректные сообщения об ошибках или статусы, указывающие на отсутствие данных.

**Критерии успеха:** Ответ API на запрос о статусе контеста должен точно отражать текущее состояние контеста, включая все задачи, языки программирования и временные рамки. Для несуществующих контестов API должен возвращать четко определенные сообщения об ошибках или пустые наборы данных, указывающие на отсутствие контеста.

* + 1. Испытание функционала "Получение условия задачи" (Get problem)

**Цель испытания:** Проверить точность и полноту информации, предоставляемой эндпоинтом для получения детального условия задачи, включая описание, ограничения и примеры входных/выходных данных.

**Шаги:**

Выполнить запрос к API для получения условий конкретной задачи, используя идентификаторы контеста и задачи.

Проверить, что ответ содержит полное описание задачи, включая ограничения по времени и памяти, а также примеры входных и выходных данных.

Проверить корректность представления форматов входных и выходных данных и соответствие примеров заданной задаче.

**Критерии успеха:** API возвращает полное и точное описание задачи с учетом всех параметров и ограничений. Примеры входных и выходных данных корректно отображаются и соответствуют условиям задачи.

* + 1. Испытание функционала "Отправка решения и получение результатов" (Submit a task for checking)

**Цель испытания:** Проверить процесс отправки решений задач через API, их проверку и получение результатов проверки, включая статус решения и детали выполнения.

**Шаги:**

Отправить решение задачи через API, указав идентификаторы контеста, задачи, язык программирования и путь к файлу с решением.

Ожидать обработки решения и получить результаты проверки, включая статус (например, правильное решение, ошибка компиляции, неправильный ответ) и, при необходимости, детали выполнения (время работы, использованная память).

Проверить соответствие полученных результатов ожидаемым, особенно в случаях неправильных решений, для проверки корректности сообщений об ошибках.

**Критерии успеха:** Система корректно обрабатывает отправленные решения, возвращая точные и понятные результаты проверки. Для решений, не прошедших тестирование, предоставляются четкие указания на ошибки или причины отказа.

* + 1. Испытание функционала "Получение информации о посылке" (Get info about the run)

**Цель испытания:** Проверить возможность получения детальной информации о конкретной посылке, включая статус проверки, время выполнения, использованную память и детали по каждому тесту.

**Шаги:**

Используя API, запросить информацию о ранее отправленной посылке, указав идентификаторы контеста и посылки.

Анализировать полученные данные о посылке, включая статус проверки (например, WA, AC, TLE), время выполнения кода, использованную память, и детали по каждому тесту, если таковые доступны.

**Критерии успеха:** API возвращает точную и полную информацию о посылке, позволяя идентифицировать результаты тестирования для каждой отправленной задачи. Информация о статусе решения, время выполнения и использованная память должны соответствовать фактическим результатам выполнения кода на тестовых данных.

Завершение программы и методики испытаний

В конце программы и методики испытаний для backend-прослойки необходимо провести общую оценку устойчивости и производительности системы при высоких нагрузках, а также проверить корректность обработки параллельных запросов и безопасность данных.

**Дополнительные шаги:**

Провести тестирование на проникновение для идентификации возможных уязвимостей безопасности в API и механизмах аутентификации и авторизации.

Использовать инструменты для нагрузочного тестирования, имитируя высокое количество одновременных запросов к различным эндпоинтам API, чтобы оценить способность системы справляться с пиковыми нагрузками.

Проверить механизмы логирования и мониторинга за активностью системы, чтобы убедиться, что они предоставляют достаточный уровень детализации и актуальности информации для эффективного реагирования на инциденты и ошибки в работе приложения.

**Оценка результатов испытаний:** Все компоненты backend-прослойки должны продемонстрировать высокую степень надежности и безопасности, корректно обрабатывая запросы и обеспечивая актуальность и целостность данных. Успешное прохождение всех испытаний и дополнительных шагов подтвердит готовность системы к эксплуатации и её способность обеспечивать стабильную и безопасную работу веб-приложения.

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |