

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего
образования**

**«Национальный исследовательский Нижегородский государственный университет им.
Н. И. Лобачевского»
(ННГУ)**

**Институт информационных технологий, математики и механики
Кафедра математического обеспечения и суперкомпьютерных технологий**

УЧЕБНЫЙ КУРС

«Проектирование и архитектура программных систем»

для подготовки по направлению 09.03.04 «Программная инженерия»

СПЕЦИФИКАЦИИ И СЦЕНАРИИ ТЕСТОВ

«QA-система на основе ИИ»

Выполнили студенты группы 3822Б1ПР1
Ворошилов Виталий Александрович, Крылов Михаил Георгиевич,
Моисеев Артём Владимирович, Морозов Егор Алексеевич,
Рамс Сергей Никитич

Проверил к.т.н., доцент
Лебедев Илья Геннадьевич

Нижегород
2025

Содержание

1. Спецификации на входы.....	3
2. Спецификации на выходы.....	4
3. Среда тестирования.....	5
4. Специальные требования	6
5. Взаимозависимости.....	7
6. Тесты	8
6.1. Модель.....	8
6.1.1. EM (Exact Match) & F1 Score	8
6.2. Серверная часть.....	9
6.2.1. Авторизация.....	9
6.2.2. История: получение истории (новый аккаунт).....	9
6.2.3. История: удаление записи	9
6.2.4. История: удаление записи / интеграционный тест	9
6.2.5. Создание запроса (основной случай)	10
6.2.6. Создание запроса (слишком длинный текст)	10
6.2.7. Ошибка на вычислительном узле	10
6.2.8. Таймаут при ожидании ответа от вычислительного узла	10
6.2.9. Запрос отмены обработки запроса.....	10
6.2.10. Сценарии Telegram бота	10
6.3. Клиентская часть / Web UI	11
6.3.1. Загрузка и инициализация страницы	11
6.3.2. Проблемы с сетью	11

1. Спецификации на входы

- Для модульных тестов серверной части потребуются эмуляторы внешних сущностей:
 - Мок базы данных
 - Мок вычислительного узла
- Для тестирования модели, работающей на вычислительном узле:
 - Эталонный текст
- Для интеграционных тестов:
 - Входы серверной части:
 - HTTP-запрос
 - Входы клиентской части (Web UI):
 - HTTP-ответ
 - Входы вычислительных узлов:
 - HTTP-запрос

Для запуска модульных тестов и проведения интеграционных тестов необходимы мышь, клавиатура, компьютер. Для тестов модели необходим компьютер с GPU.

2. Спецификации на выходы

- Выход каждого теста – вердикт: прошел тест, или не прошел.
- Вердикты содержатся в выходных логах среды тестирования.
- Выход тестов модели – метрики.

3. Среда тестирования

Тесты серверной и клиентской части – не считая тестов для вычислительных узлов – не притязательны к аппаратной конфигурации. Помимо способного к запуску тестов окружения (инструменты сборки/среда исполнения, ABI), потребуется

- для серверной части:
 - база данных PostgreSQL
- для вычислительных узлов:
 - способный запустить модель GPU

4. Специальные требования

- Для интеграционных тестов:
 - Доступность необходимых портов
- Другие специальные требования к организации тестирования отсутствуют.

5. Взаимозависимости

Взаимозависимостей между тестами нет.

6. Тесты

6.1. Модель

6.1.1. EM (Exact Match) & F1 Score

Тестирование модели организуется на базе двух метрик: EM (Exact Match) и F1 score на тестовой части набора данных sberQUAD.

Метрика Exact Match (EM) используется для оценки точности совпадения между предсказанным и эталонным текстами. В контексте сравнения текстов она принимает значение 1, если предсказание полностью совпадает с эталоном (включая пунктуацию и регистр), и 0 - в противном случае. Формула для одного примера:

$$EM = \begin{cases} 1, & \text{если } \hat{y} = y \\ 0, & \text{иначе} \end{cases}$$

Рисунок 6.1.1.1 – Формула EM для одного примера

где \hat{y} – предсказанный текст, y – эталонный.

Для набора из N примеров метрика усредняется:

$$EM_{avg} = \frac{1}{N} \sum_{i=1}^N 1(\hat{y}_i = y_i)$$

Рисунок 6.1.1.2 – Формула EM для N примеров

F1 — это метрика, которая сравнивает два текста по совпадающим словам. Она учитывает и то, сколько слов из предсказания есть в эталоне (точность), и то, сколько слов из эталона удалось угадать (полнота). Чем больше пересечение, тем выше F1.

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Рисунок 6.1.1.3 – Формула F1

где - $\text{precision} = (\text{число совпавших слов}) / (\text{число слов в предсказанном тексте})$, - $\text{recall} = (\text{число совпавших слов}) / (\text{число слов в эталонном тексте})$.

- Ожидаемое поведение: $EM \approx 0.4$, $F1 \approx 0.7$
- Ожидаемая производительность: Практически линейно пропорциональна объему входных данных

- Ожидаемая достоверность: 100%

6.2. Серверная часть

6.2.1. Авторизация

Пользователь инициирует создание ссылки для входа через Telegram-бот.

- Ожидаемое поведение:
 - Ссылка валидна, переход по ссылке инициирует авторизацию в системе
 - Попытка регистрации аккаунта с занятым почтовым адресом должна завершаться ошибкой

6.2.2. История: получение истории (новый аккаунт)

Ответ на запрос на получение истории запросов с только что зарегистрированного аккаунта не должен содержать записей истории.

- Ожидаемое поведение:
 - Ответ на запрос на получение истории запросов с только что зарегистрированного аккаунта не содержит записей истории

6.2.3. История: удаление записи

Запрос на удаление записи из истории запросов должен завершаться успешно.

- Ожидаемое поведение:
 - Запрос на удаление записи из истории завершается успешно. Если в рамках записи в текущий момент времени производится обработка на вычислительном узле, или запрос стоит там в очереди, вычислительному узлу сообщается о том, что запрос можно не обрабатывать.

6.2.4. История: удаление записи / интеграционный тест

Запрос на удаление записи из истории запросов должен завершаться успешно.

- Ожидаемое поведение:
 - Запрос на удаление записи из истории завершается успешно. Если в рамках записи в текущий момент времени производится обработка на вычислительном узле, или запрос стоит там в очереди, вычислительному узлу сообщается о том, что запрос можно не обрабатывать. Если же он уже обрабатывается, сервер не должен сохранять его ответ.

6.2.5. Создание запроса (основной случай)

Создание запроса должно завершаться успешно, порождать новый элемент в истории и добавлять задачу в очередь вычислительного узла.

- Ожидаемое поведение:
 - Создание запроса завершается успешно, порождает новый элемент в истории и добавляет задачу в очередь вычислительного узла, клиент информируется о создании записи в истории для получения результата в будущем

6.2.6. Создание запроса (слишком длинный текст)

Отправка запроса с текстом, превышающим ограничения по длине, должна завершаться ошибкой.

- Ожидаемое поведение:
 - Отправка запроса с текстом, превышающим ограничения по длине, завершается с ошибкой, клиент информируется о невозможности удовлетворить запрос

6.2.7. Ошибка на вычислительном узле

Вычислительный узел сообщил об ошибке при обработке текста.

- Ожидаемое поведение:
 - В запись истории запросов добавляется сообщение об ошибке

6.2.8. Таймаут при ожидании ответа от вычислительного узла

Вычислительный узел не сообщил результат обработки вовремя.

- Ожидаемое поведение:
 - В запись истории запросов добавляется сообщение об ошибке

6.2.9. Запрос отмены обработки запроса

Клиент сообщает о том, что запрос более не актуален.

- Ожидаемое поведение:
 - На вычислительный узел посылается сообщение о том, что запрос более не актуален, если он все еще стоит в очереди, то вычислительный узел убирает запрос из очереди, в противном случае, если запрос уже в обработке, по пришествии ответа от вычислительного узла, сервер должен его отклонить.

6.2.10. Сценарии Telegram бота

Тестируется диспетчеризация внутренних механизмов сервера, приводящихся в действие поступившим от Telegram бота запросом через соответствующий эндпоинт.

- Ожидаемое поведение:
 - Должны вызываться соответствующие запросам процедуры

6.3. Клиентская часть / Web UI

6.3.1. Загрузка и инициализация страницы

Веб-приложение должно быть легковесным, с быстрой загрузкой страниц.

- Ожидаемое поведение:
 - Маршруты загружаются без ошибок и долгой инициализации

6.3.2. Проблемы с сетью

Пользователь должен быть проинформирован о проблемах с сетью, если в данный момент он ожидает отправки запроса, и он еще не был отправлен, запрос должен быть сохранен в локальное хранилище браузера для отправки при возобновлении связи.

- Ожидаемое поведение:
 - Пользователь информируется о проблемах с сетью, если в данный момент он ожидает отправки запроса, и он еще не был отправлен, запрос сохраняется в локальное хранилище браузера