ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА СОЗДАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

Техническое задание на создание автоматизированной системы «Интеллектуальный бот-помощник для автоматизации обратной связи в дистанционном обучении»

1. Общие сведения

- **Наименование АС**: Интеллектуальный бот-помощник для курса «Анализ данных».
- **Заказчик**: Цифровая кафедра.
- **Разработчик**: --
- **Сроки разработки**: Начало — 24.02.2025, окончание — 18.06.2025.

2. Цели и назначение создания АС

2.1. Цели создания:

- Снижение нагрузки на преподавателей за счет автоматизации ответов на типовые вопросы.
- Обеспечение студентов мгновенной обратной связью (время ответа ≤ 5 минут).
- Повышение качества обучения через персонализированные рекомендации.

2.2. Назначение АС:

Автоматизация процессов взаимодействия студентов и преподавателей в рамках курса «Анализ данных» через Telegram-бота с функциями:

- Обработка текстовых и графических запросов.
- Генерация ответов на базе LLM.
- Формирование аналитических отчетов.

3. Характеристика объекта автоматизации

Объект автоматизации: Процесс обратной связи в дистанционном обучении.
Условия эксплуатации:

- Количество пользователей: до 10 пользователей, обращающихся единовременно.
- Режим работы: круглосуточный.
- Требования к сети: стабильное интернет-соединение.
- Окружающая среда: работа на локальном сервере заказчика.

4. Требования к автоматизированной системе

4.1. Требования к структуре АС

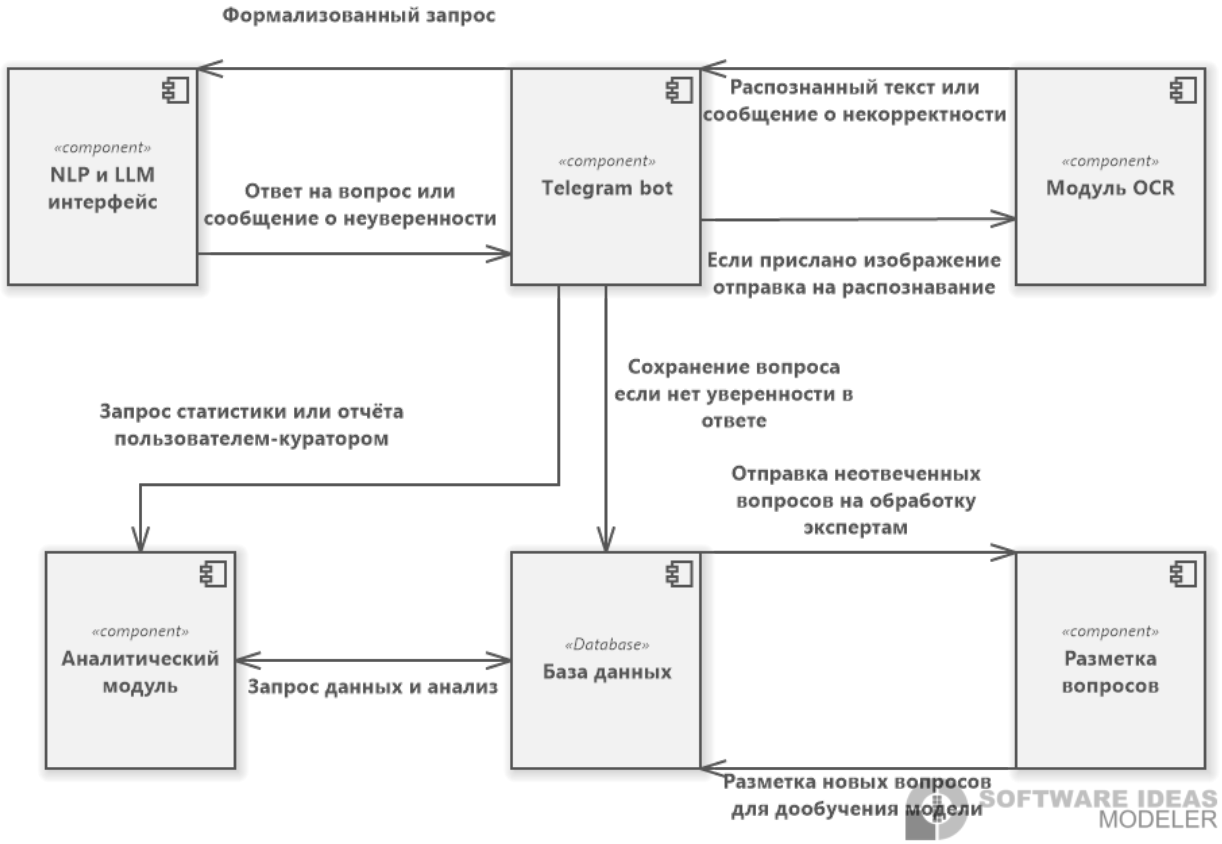

Рис. 1 Диаграмма компонентов АС

Подсистемы:

1. **Модуль NLP (обработка текста):**
   - Необходим для анализа смысла текстовых запросов студентов, классификации интентов и генерации ответов с использованием LLM. Обеспечивает точность и релевантность ответов.
2. **Модуль OCR (распознавание изображений):**
   - Позволяет обрабатывать скриншоты с вопросами (формулы, таблицы), что расширяет функционал бота. Использование Tesseract/EasyOCR обусловлено высокой точностью и интеграцией с Python.
3. **База знаний (PostgreSQL):**
   - Реляционная СУБД выбрана для структурированного хранения получаемых ботом вопросов и даваемых ответов. Обеспечивает быстрый доступ и масштабируемость.
4. **Telegram-интерфейс:**
   - Telegram — популярная платформа для студентов, поддерживает API для интеграции чат-бота, текстовых и графических сообщений.

Характер взаимодействия между подсистемами описан на рисунке 1.

4.2. Требования к функциям:

- Обработка запросов:
  - Текст: анализ интентов, форматирование запросов.

- Изображения: предобработка (бинаризация, шумоподавление), распознавание текста.
- Формирование отчетов: Экспорт статистики (активность студентов, метрики качества ответов).
- Диагностика: Логирование ошибок и автоматическое составление отчета для обслуживающего бота персонала.
- Сохранение поступающих вопросов и отдаваемых ответов в базе данных. Предполагаемое строение базы данных (см. Рис. 2.)

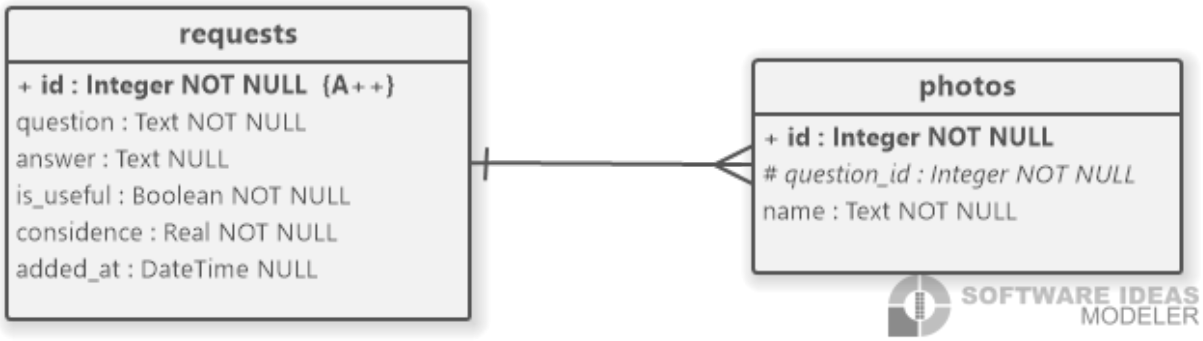

Рис. 2 ERD-диаграмма БД

Таблицы, поля и описания:

1. **Запрос (requests):**
    a. id | serial - идентификатор запроса;
    b. question | text - текстовая формулировка всего запроса;
    c. answer | text - текстовая формулировка ответа от LLM-модели;
    d. is_useful | bool - оценка студентом полезности ответа;
    e. confidence | float - метрика “уверености” LLM в ответе;
    f. added_at | timestamp - дата добавления записи в базу данных.
2. **Фотографии (photos):**
    a. id | serial - идентификатор фотографии;
    b. question_id_FK | int - идентификатор запроса, к которому относится фотография;
    c. name | text - название сохраненного файла.

4.3. Требования к видам обеспечения:

- Программное обеспечение:
    - Язык: Python 3.11+.
    - Библиотеки: PyTorch, Transformers, OpenCV, Aiogram.
    - СУБД: PostgreSQL.
- Техническое обеспечение:
    - Сервер: Ubuntu 22.04 LTS, 8 ГБ RAM, 4 CPU, SSD 128 ГБ.
      Linux ОС необходима для развёртывания окружения и инференса LLM, SSD ускоряет работу с БД, 8 ГБ RAM достаточно для работы LLM и OCR.
- Информационное обеспечение:

    - База знаний: структурированные вопросы по материалам курса с ответами, размеченные экспертами вручную.
- Метрологическое обеспечение:
    - Контроль точности OCR (≥ 90%).
      Допустимый уровень для корректного распознавания текста из изображений.
    - Оценка качества ответов LLM (F1-score ≥ 0.85).
      Гарантирует высокое качество ответов и минимизирует количество ошибок.

4.4. Общие технические требования:

- Надежность: Время доступности — ≥ 90%.
- Время ответа: до 5 минут
- Интерактивность:
    - Интерфейс Telegram-бота с кнопками «Полезно»/«Не полезно» и возможностью оценки полученного ответа от бота.

5. Требования к алгоритмам и математическим моделям

5.1. NLP-модель (обработка текстовых запросов)

- Архитектура:
    - Использование предобученной модели GPT (или аналогичной) для классификации интентов и извлечения сущностей.
    - Дообучение модели на датасете вопросов студентов курса «Анализ данных» для повышения точности в предметной области.
- Метрики качества:
    - F1-score для ответов LLM: ≥ 0.85.
- Обоснование:
    - BERT обеспечивает контекстное понимание текста, что критично для точной интерпретации вопросов.
    - Промежуточное дообучение на собранных обработанных данных минимизирует ошибки в терминологии курса.

5.2. OCR-модуль (распознавание изображений)

- Алгоритмы:
    - Предобработка изображений:
        - Бинаризация (адаптивный порог).
        - Шумоподавление (медианный фильтр).
    - Распознавание текста: Tesseract OCR/EasyOCR с поддержкой многоязычных шрифтов.
    - Классификация существующих шаблонов: С помощью CV будут прикладываемые скриншоты будут классифицироваться среди возможных вариантов для повышения качества считывания и улучшения промпта к LLM (на примере вопросов с единственным и множественным ответом)
- Требования к точности:
    - Точность распознавания текста: ≥ 90%.
    - Максимальное время обработки изображения: ≤ 20 сек.
- Обоснование:
    - Предобработка улучшает читаемость текста на скриншотах.

5.3. Генерация ответов (LLM)

- Модель: Llama-3.2 (или аналогичная).
- Параметры:
  - Температура генерации: 0.7 (для баланса креативности и точности).
  - Максимальная длина ответа: 500 символов.
- Обоснование:
  - Ограничение длины ответа предотвращает ошибку превышения текстового объема для одного сообщения.

6. Требования к пользователям и их профилям

6.1. Студенты

- Профиль:
  - Основные пользователи, задающие вопросы через Telegram-бота.
  - Уровень подготовки: базовые навыки работы с мессенджерами.
- Права доступа:
  - Отправка текстовых запросов и изображений.
  - Оценка полезности ответов («Полезно»/«Не полезно»).

Взаимодействие пользователя и ИС описано на диаграмме последовательностей (рис. 3)

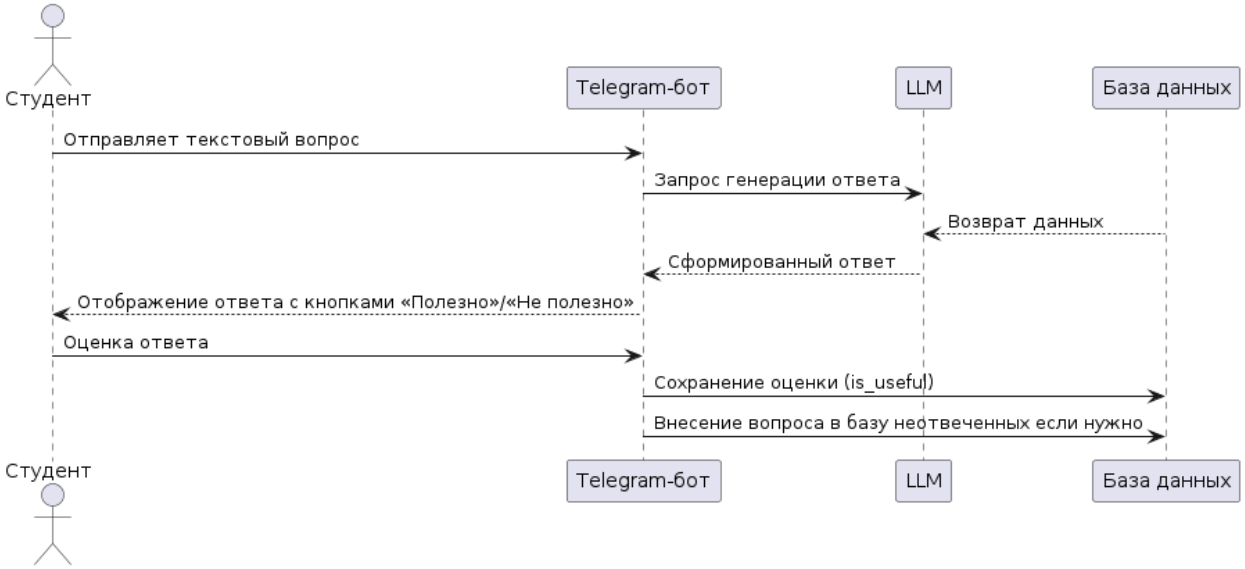

Рис. 3 Диаграмма последовательностей студента в боте

6.2. Преподаватели/Кураторы

- Профиль:
  - Администраторы системы, управляющие базой знаний и аналитикой.
  - Уровень подготовки: опыт работы с CSV/Excel.
- Права доступа:
  - Просмотр статистики использования бота.
  - Экспорт отчетов в форматах CSV/Excel.

- Получение отчетной аналитики.

7. Требования к интерфейсу пользователя и решению в целом

7.1. Telegram-бот

- Основные элементы интерфейса:
  - Кнопки быстрого доступа:
    - /start — приветственное сообщение.
    - /help — инструкция по использованию.
  - Интерактивные кнопки «Полезно»/«Не полезно» под каждым ответом.
- Поддержка форматов:
  - Текст (до 1000 символов).
  - Изображения (JPG/PNG, до 5 МБ).

7.2. Административная панель бота

- Отчетная аналитика:
  - Активность студентов (количество запросов/день).
  - Экспорт данных: CSV/Excel с фильтрами по дате и категориям.

8. Требования к входным и выходным данным

8.1. Входные данные

- Текстовые запросы:
  - Формат: UTF-8, язык — русский/английский.
- Изображения:
  - Разрешение: ≥ 300 DPI.
  - Допустимые форматы: JPG, PNG.
  - Изображения созданы с сайта тестирования (см. Рис. 4)

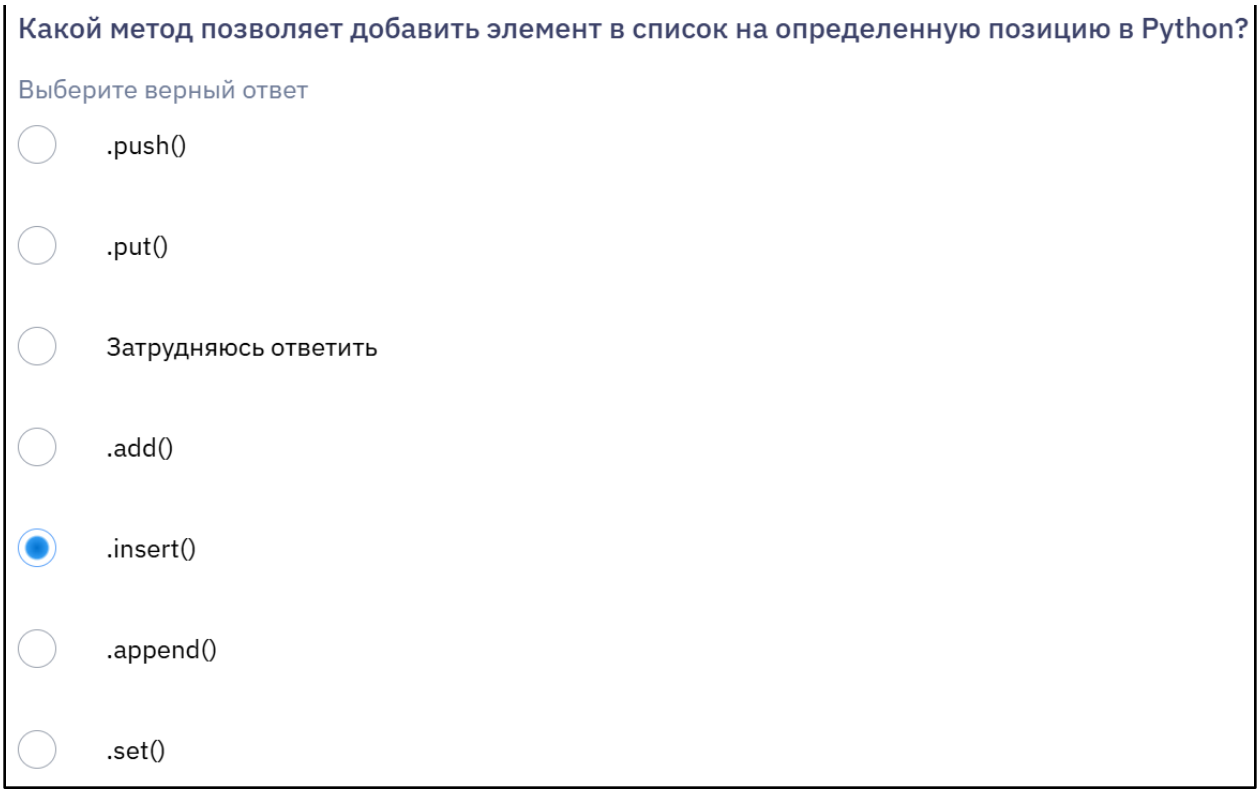

Рис. 4. Скриншот вопроса с сайта тестирования

ешь

9. Требования к документированию

1. **Перечень обязательных документов:**
   - **Техническое задание** (ГОСТ 34.602-2020) — описание целей, задач, требований к системе.
   - **Руководство пользователя** — инструкция по взаимодействию с ботом для студентов и преподавателей.
   - **Техническое описание архитектуры** — схема системы, ER-диаграмма базы данных, описание модулей.
   - **Программная документация** — комментарии в коде, API-спецификации, описание алгоритмов.
2. **Форматы и стандарты:**
   - Документы оформляются в электронном виде (PDF, DOCX).
   - Код сопровождается комментариями в соответствии с PEP8 (для Python).
   - ER-диаграммы создаются в нотации Crow’s Foot.
   - Отчеты включают дату, версию системы и подпись ответственного лица.
3. **Сроки предоставления:**
   - Итоговая документация передается заказчику в течение 7 рабочих дней после завершения разработки.

10. Источники разработки

1. **Нормативные документы:**
    - ГОСТ 34.602-2020 «Техническое задание на создание автоматизированной системы».
2. **Использованные материалы:**
    - **Лейн, Хапке, Ховард.** «Обработка естественного языка в действии» (2020) — методы NLP.
    - **Романов Д., Прохоров Н.** «Глубокое обучение. Компьютерное зрение на Python» (2021) — алгоритмы OCR.
    - **Данные курса «Анализ данных»** — структурированные вопросы студентов.
3. **Программные ресурсы:**
    - Библиотеки: PyTorch, Transformers, Tesseract, Aiogram.
    - API: Telegram Bot API, Llama3.2.
4. **Консультанты и участники:**
    - Кураторы Цифровой кафедры ТюмГУ — требования к функционалу.
    - Команда 3 курса - основные команда разработки продукта.
    - Команда 2 курса - дополнительная команда разработки продукта.
5. **Результаты исследований:**
    - Статистика опроса студентов (актуальность проблемы, ожидания от бота).
    - Анализ аналогов (GPT Tools Chat, Gigachat, Llama).