

Создание ИИ-ассистента для студентов на основе RAG-архитектуры и методологии проектной деятельности

МЛечный путь

1 семестр

4 участника

Waterfall

Заказчик:

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»

Куратор: Ильинский Александр Дмитриевич

ЦЕЛЕВАЯ АУДИТОРИЯ ПРОДУКТА. БОЛЬ КЛИЕНТА

ПРОБЛЕМА

Отсутствие доступной методологической поддержки в проектной деятельности для студентов



ЦЕЛЕВАЯ АУДИТОРИЯ

Студенты, которые участвуют в проектной деятельности, хотят освоить управление проектами, Agile, Scrum, DevOps, и нуждаются в методологической поддержке

ЗАЧЕМ РЕШЕНИЕ?

1. Повышение продуктивности
2. Экономия времени и сил
3. Преподаватели не могут давать индивидуальные консультации каждому студенту
4. Разрозненность источников знаний
5. Студенты испытывают методологические трудности

КАК ПРОБЛЕМА РЕШАЕТСЯ СЕЙЧАС

РУЧНОЙ ПОИСК



КОНСУЛЬТАЦИИ У
ПРЕПОДАВАТЕЛЕЙ / МЕНТОРОВ



ИСПОЛЬЗОВАНИЕ ОБЩИХ LLM
(CHATGPT, DEEPSEEK)

СОСТАВ КОМАНДЫ И ЗОНЫ ОТВЕТСТВЕННОСТИ

Раков Дмитрий Владимирович

Тимлид/Менеджер проекта

- Коммуникация с куратором, приоритизация задач
- Ведение плана работ
- Контроль сроков и качества выполнения задач
- Принятие решений в спорных ситуациях

Яцук Владислав Романович

ML-разработчик

- Выбор, настройка и наполнение векторной базы данных
- Реализация RAG-пайплайна
- Настройка и интеграция LLM
- Оптимизация промтов и качества ответов

Косторной Дмитрий Вадимович

Разработчик (Backend)

- Разработка логики Telegram-бота
- Создание и настройка rag-клиента
- Механизм загрузки и обработки документов
- Интеграция всех систем

Соболев Егор Владимирович

Аналитик/Тестирующий

- Проектирование структуры бд
- Создание и поддержка тестовой базы вопросов
- Функциональное тестирование системы
- Диаграммы базовых процессов

СИНХРОНИЗАЦИЯ, ЦЕПОЧКИ ПЕРЕДАЧИ И КОНТРОЛЬ ПРОГРЕССА

Раков Д. → Команда
(Утверждённые ТЗ и приоритеты)

Соболев Е. → Разработчики
(WFD-диаграмма, диаграмма бд,
отчёты по тестированиям)

Яцук В. → Косторной Д.
(Готовый RAG-пайплайн (работающий API))

Соболев Е. → Разработчики
(Отчёты по тестированиям)

Все → Раков Д.
(Отчёты о выполненных задачах)

Косторной Д. → Раков Д.
(Рабочий функционал бота: ссылка
на бота + чек-лист проверки)

ТОЧКИ СИНХРОНИЗАЦИИ

Еженедельные совещания (30 мин, воскресенье):
обсуждение прогресса по задачам, пересмотр
приоритетов, обсуждение блокеров

Инструмент: Таблица с диаграммой Ганта

Критерий «Готово»: Задача выполнена, код проверен
(pull request), а функционал протестирован и принят
тимлидом

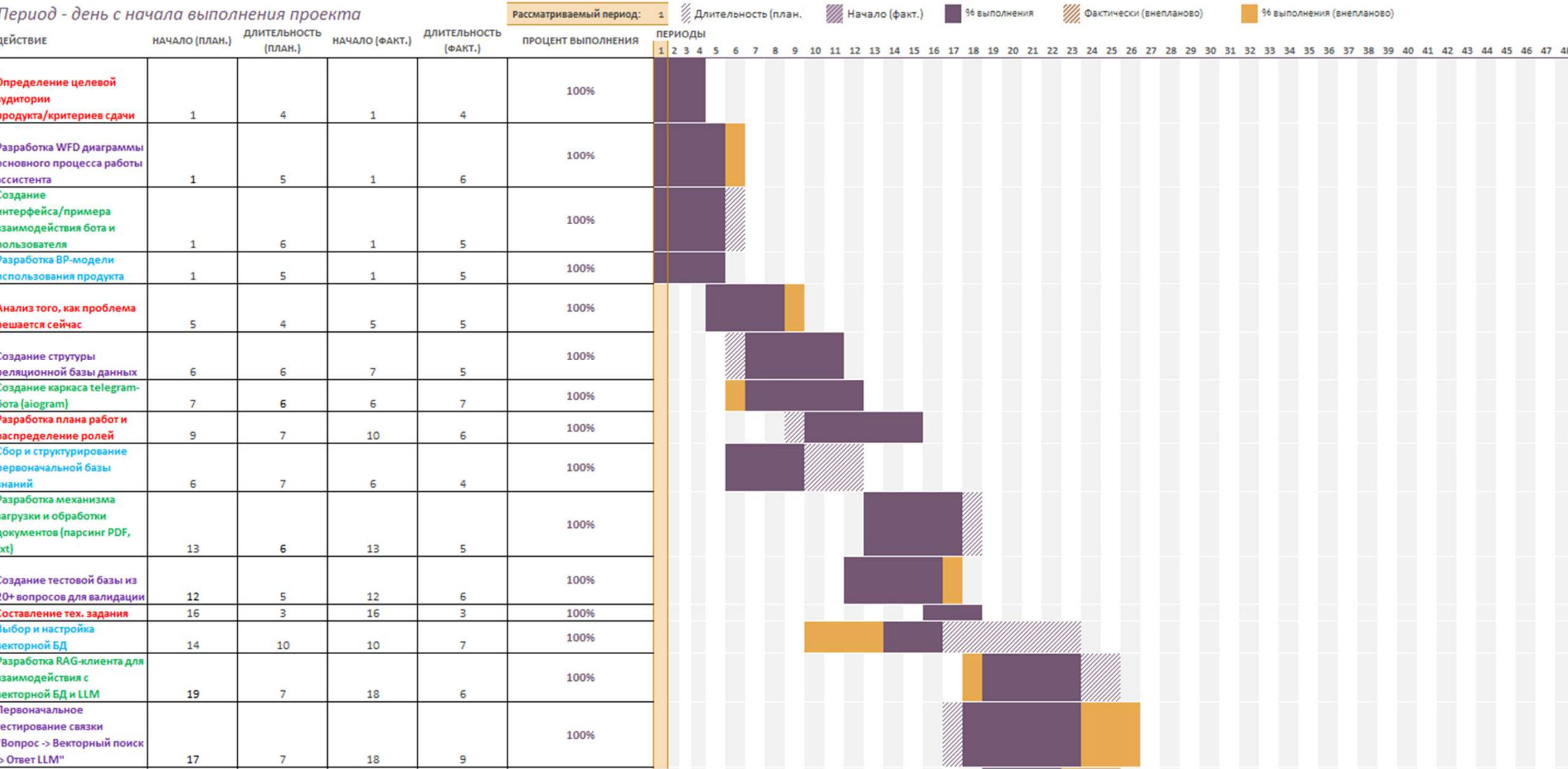
Сигналы проблем: У задачи не меняется %
выполнения > 3 дней, отсутствие актуализации от
разработчика

Планировщик проекта

Период - день с начала выполнения проекта

Косторной Д. В. - зелёные задачи Соболев Е. В. - фиолетовые задачи Яцук В. Р. - синие задачи Раков Д. В. - красные задачи

Начало работы: 23.11.2025



АНАЛИЗ КОНКУРЕНТОВ



Плюсы

- + Даёт советы по запросу
- + Может сформулировать документы (ТЗ, план проекта)

Минусы

- Ответы могут быть неточными или выдуманными.
- Не использует проверенную базу знаний
- Нет гарантии соответствия PMBOK, Scrum Guide, ГОСТам



Плюсы

- + Автоматизирует создание backlog, user stories, составление плана спринта

Минусы

- Нет проверенной базы знаний
- Только генерация плана спринта

АНАЛИЗ КОНКУРЕНТОВ



Плюсы

- + Очень простой интерфейс
- + Подходит для небольшой команды.
- + Быстрый вход без обучения.

Минусы

- Нет поддержки Scrum/Agile как методологий.
- Нет подсказок по распределению ролей, созданию документации или этапам проекта



Плюсы

- + Визуализация процессов
- + Наличие шаблонов

Минусы

- Шаблоны не являются методологическими рекомендациями.
- Отсутствуют стандартизированные обучающие материалы (PMBOK, ГОСТы, Scrum Guide).

РЕШЕНИЕ

Telegram-бот с RAG-архитектурой и открытым исходным кодом, который способен консультировать студентов по вопросам ведения проектов на основе проверенной базы знаний (статьи, книги, ГОСТы).

КРИТЕРИИ СДАЧИ

- Разработать интерфейс взаимодействия с пользователем на основе br-модели и wfd-диаграммы
- Реализовать механизм обработки текстов документов из базы знаний
- Реализовать RAG-пайплайн (векторная БД + LLM)
- Реализовать механизм хранения всех необходимых данных на сервере (определить стандарт общения сервера и клиента, разработать схему запроса)
- Кодовая база и база знаний задокументированы
- Качество ответов оценено как релевантное на наборе из 20+ тестовых вопросов

РЕЗУЛЬТАТЫ ИТЕРАЦИИ

- Определена целевая аудитория продукта/критерии сдачи
- Создана WFD-диаграмма и br-модель основных сценариев использования
- Создана структура базы данных
- Создан интерфейс/пример взаимодействия бота и пользователя
- Собрана и структурирована начальная версия базы знаний
- Разработан механизм обработки и парсинга документов из базы знаний
- Создана тестовая база из 20 вопросов
- Выбрана и настроена векторная БД
- Разработан RAG-клиент для взаимодействия с векторной БД и LLM
- Реализован RAG-пайплайн: индексация документов в векторную БД
- Создан каркас telegram-бота и базовый сценарий «Задайте вопрос»

ЗАДАЧИ НА СЛЕДУЮЩУЮ ИТЕРАЦИЮ

- Интеграция LLM в RAG-пайплайн
- Настройка Docker-окружения
- Тестирование базового ответа на вопросы
- Оптимизация БД
- Дополнение базы знаний
- Базовое логирование запросов и ответов в реляционную БД
- Оптимизация промтов и параметров RAG-пайплайна для улучшения ответов
- Фиксы и доработки бота на основе первичного тестирования
- Проведение юзабилити-тестирования с 3 студентами
- Функциональное тестирование
- Хостинг финальной версии бота
- Сбор метрик производительности системы

БАЗА ЗНАНИЙ

Мой диск > Млечный путь. Создан... > Документы на базу зна... ▾ 👤

Тип ▾

Люди ▾

Изменено ▾

Источник ▾

Название ⬆

PDF 2020-Scrum-Guide-Russian.pdf 👤

PDF ГОСТ-19.701-90-Схемы-алгоритмов-программ-данных-и-систем.pdf 👤

PDF ГОСТ-12207-2010.pdf 👤

PDF Как подготовить бэклог продукта с большим количеством зависимостей и не потратить время впустую.pdf 👤

PDF Как провести ретроспективу, которая реально работает.pdf 👤

PDF Как провести ретроспективу, чтобы команда работала лучше.pdf 👤

PDF Как проводить стендап.pdf 👤

PDF Как составить техническое задание - пошаговый гайд.pdf 👤

PDF Как спланировать двухнедельный спринт.pdf 👤

PDF Командные роли в картинках.pdf 👤

PDF ОЦЕНОЧНЫЕ МАТЕРИАЛЫ.pdf 👤

PDF Планирование спринта - Scrum.pdf 👤

PDF Планирование спринта Scrum ключевые этапы и рекомендации.pdf 👤

PDF ПРОЕКТНОЕ ОБУЧЕНИЕ.pdf 👤

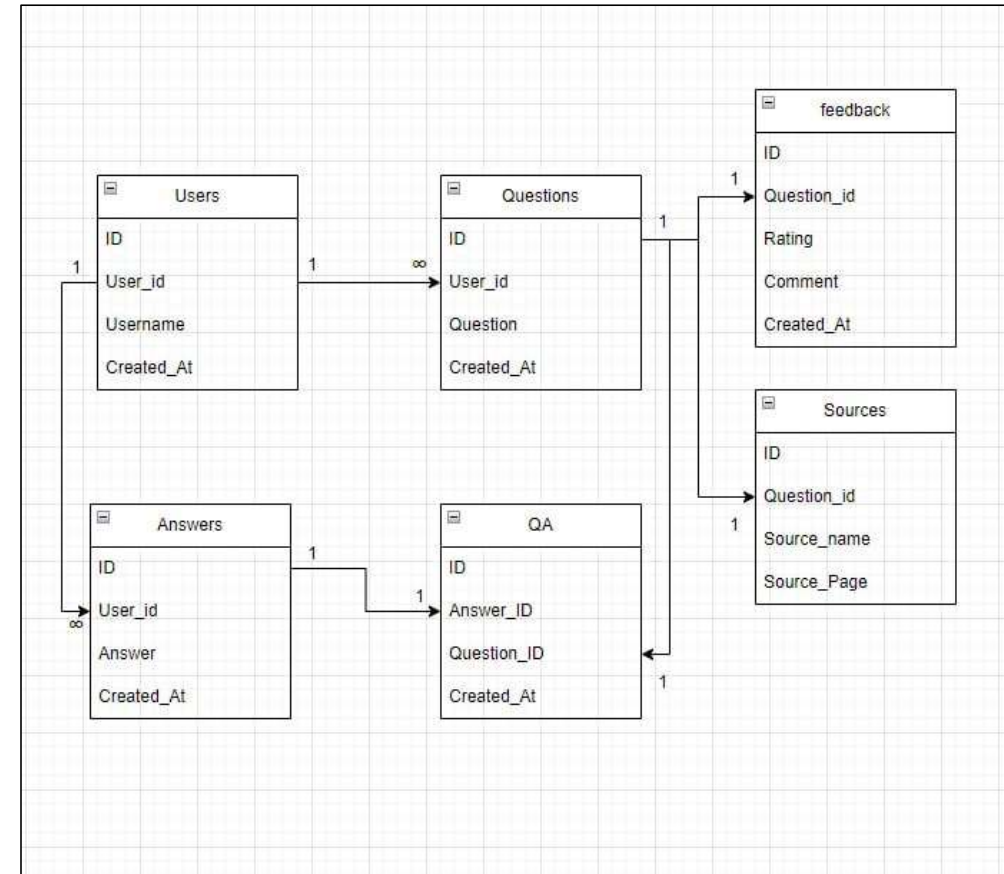
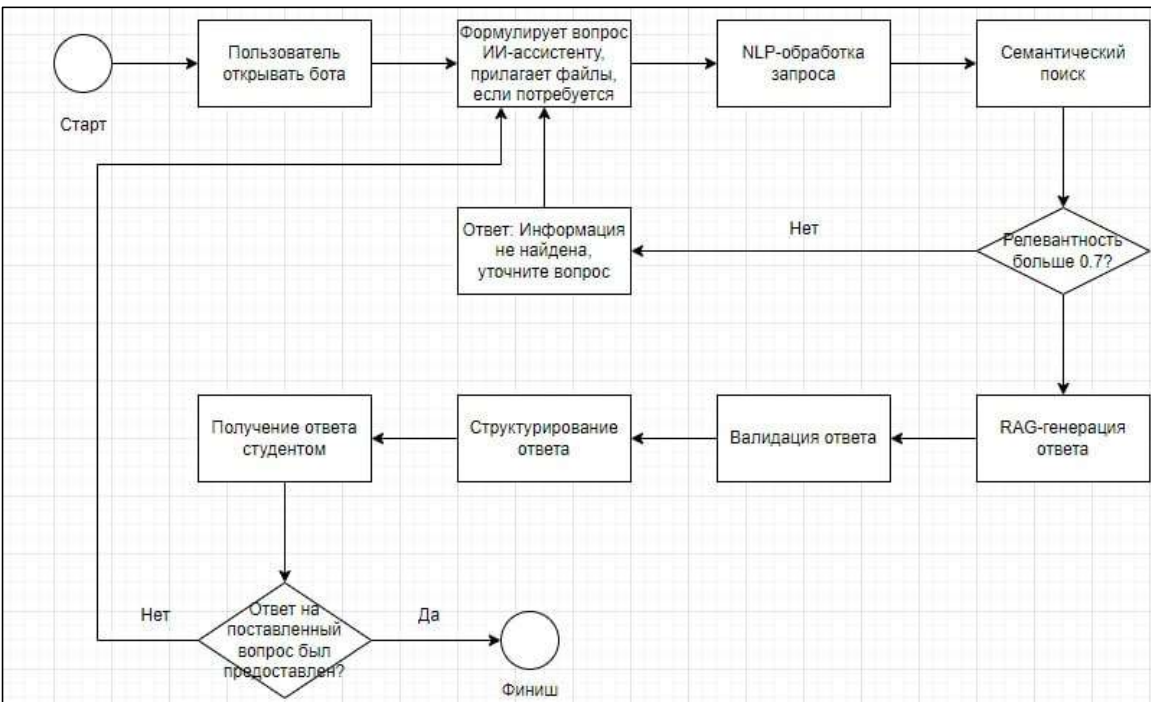
PDF ПРОЕКТНОЕ обучение в урфу.pdf 👤

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Техническое задание по проекту
“Создание ИИ-ассистента для студентов на основе RAG-архитектуры”
1. ОБЩИЕ СВЕДЕНИЯ
Наименование проекта: Создание ИИ-ассистента для студентов на основе RAG-архитектуры и методологии проектной деятельности
Цель проекта: Разработать ИИ-ассистента в виде Telegram-бота, который консультирует студентов по методологиям проектной деятельности (<u>Agile</u> , <u>Scrum</u> , <u>DevOps</u> , составление ТЗ) на основе проверенной базы знаний.
Заказчик: УрФУ, проектный практикум
2. ОПИСАНИЕ СИСТЕМЫ
Система должна состоять из трех основных компонентов:
2.1. Telegram-бот - интерфейс для взаимодействия с пользователем
2.2. RAG-сервис (API) - обрабатывает запросы, выполняет поиск в векторной БД и генерирует ответы с помощью LLM
2.3. Векторная база данных - хранит <u>эмбединги</u> документов из базы знаний
3. ОСНОВНОЙ ФУНКЦИОНАЛ
3.1. Сценарий "Задать вопрос"
– Пользователь задает текстовый вопрос по проектной деятельности
– Бот возвращает ответ, сгенерированный LLM на основе найденных релевантных фрагментов из базы знаний
3.2. Сценарий "Информация о боте"

ОТЧЁТ О ПРОДЕЛАННОЙ РАБОТЕ

Соболев Е. В. Аналитик/Тестировщик



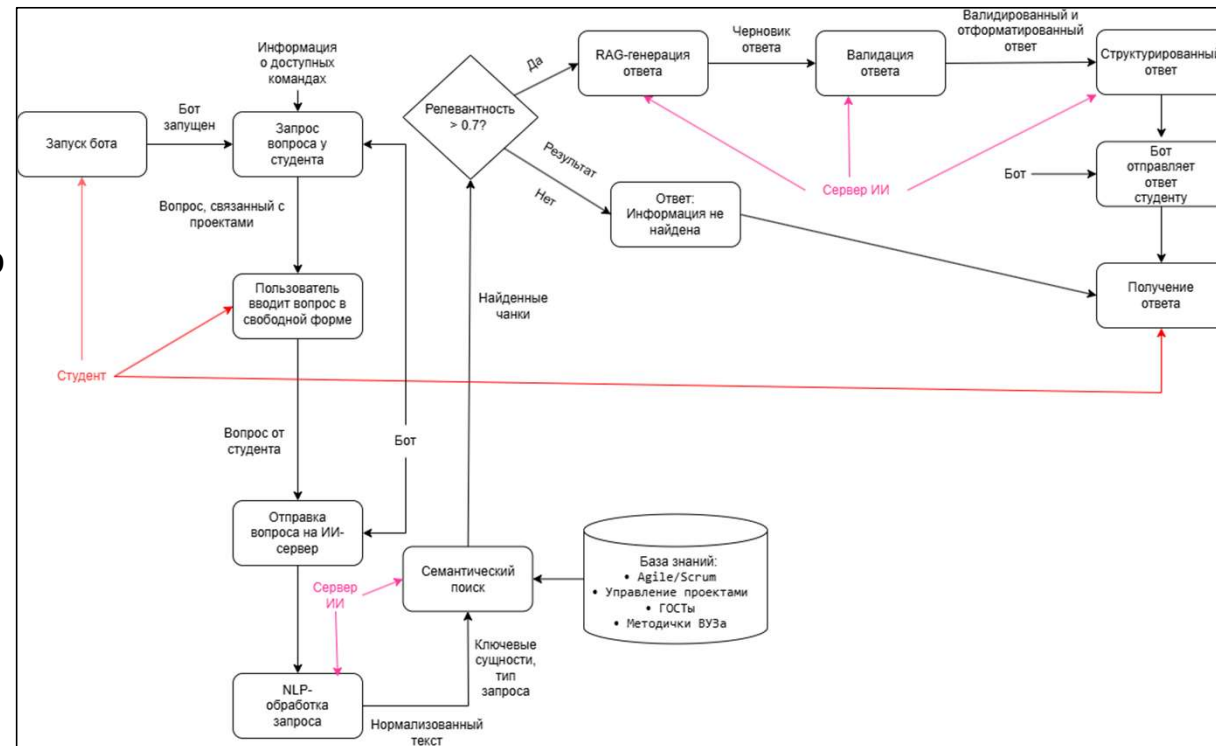
ОТЧЁТ О ПРОДЕЛАННОЙ РАБОТЕ Яцук В. Р. М1-разработчик

ВЕКТОРНАЯ БАЗА ДАННЫХ

- Технология: ChromaDB
- Документов в базе: 150+ чанков
- Размер чанка: 500 символов
- Алгоритм поиска: семантический поиск по векторам
- Модель эмбедингов: встроенная в ChromaDB
- Метрика похожести: cosine similarity

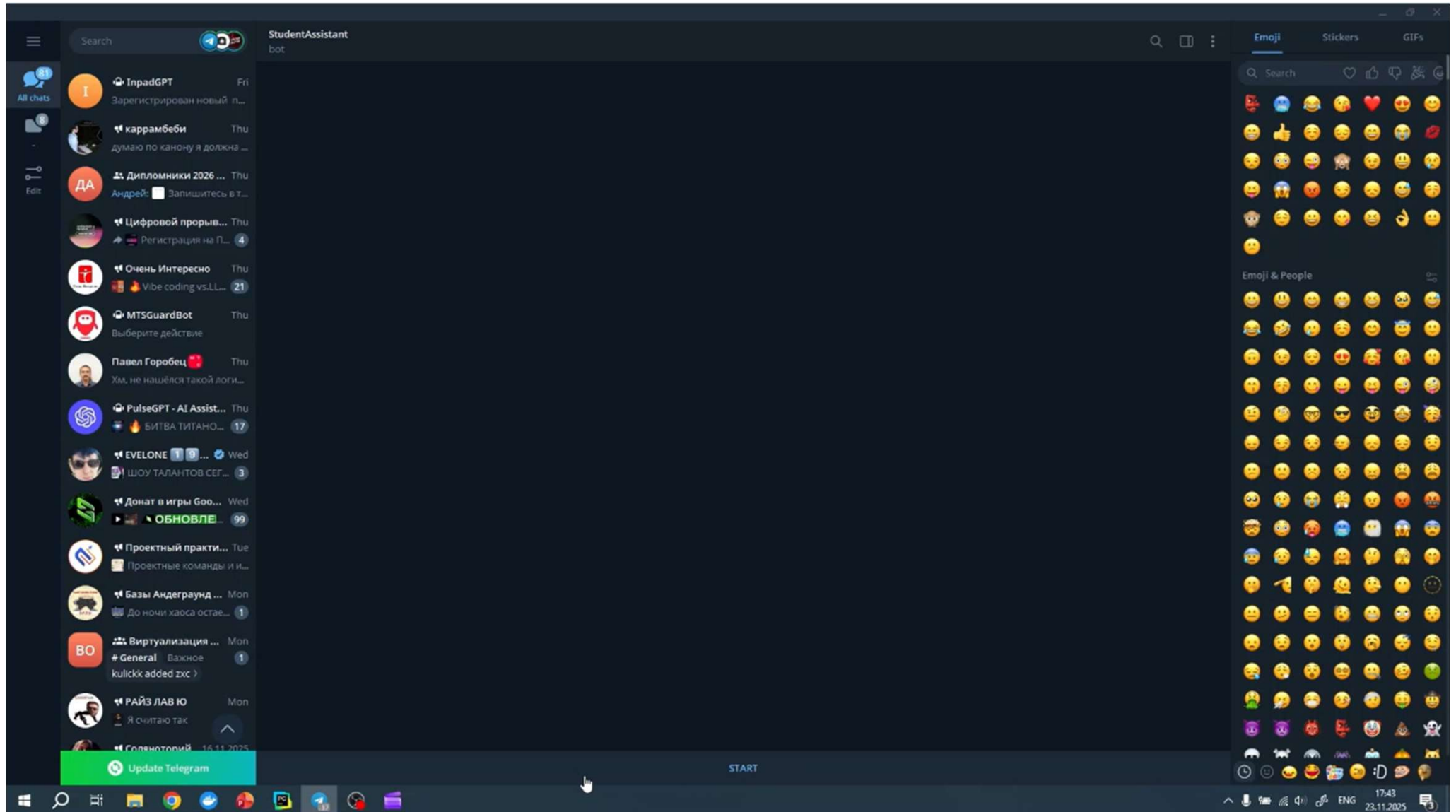


ВР МОДЕЛЬ



ОТЧЁТ О ПРОДЕЛАННОЙ РАБОТЕ

Косторной Д. В. Разработчик (Backend)



RAG-КЛИЕНТ И ИНТЕГРАЦИЯ

Архитектура интеграции

Telegram Bot (aiogram)



RAG API (FastAPI)



ChromaDB + LLM



RAGClient.get_answer()



```
class RAGClient:
    1 usage (1 dynamic)  Косторной Дмитрий Вадимович
    @staticmethod
    async def get_answer(question: str, user_id: int) -> str:
        """Реальный запрос к RAG API"""
        try:
            async with aiohttp.ClientSession() as session:
                async with session.post(
                    f"{RAG_API_URL}/ask",
                    json={"question": question, "user_id": user_id},
                    timeout=30
                ) as response:

                    if response.status == 200:
                        data = await response.json()
                        answer = data.get('answer', 'Не удалось получить ответ')
                        sources = data.get('sources', [])

                        if sources:
                            answer += f"\n\n📖 Источники: {' '.join(sources)}"

                        return answer
                    else:
                        logger.error(f"RAG API error: {response.status}")
                        return "Извините, сервис временно недоступен. Попробуйте

except aiohttp.ClientError as e:
```


РЕФЛЕКСИЯ



Создание ИИ-ассистента для студентов на основе RAG-архитектуры и методологии проектной деятельности

МЛечный путь

1 семестр

4 участника

Waterfall

Заказчик:

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»

Куратор: Ильинский Александр Дмитриевич