**Министерство науки и высшего образования**

**Российской Федерации**

**ФЕДЕРАЛЬНОЕ Государственное Автономное**

**образовательное учреждение высшего образования**

**«новосибирский национальный исследовательский   
государственный университет»**

**ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНОЙ РОБОТОТЕХНИКИ**

Кафедра Интеллектуальных систем теплофизики ИИР

Направление подготовки 15.03.06 Мехатроника и робототехника

Направленность (профиль) Мехатроника и робототехника

**ОТЧЕТ**

**о прохождении производственной практики, технологической (проектно-технологической)**

**практики**

(указывается наименование практики)

**Обучающегося Сыренного Ильи Игоревича группы № 21930** **курса** **4**

**Тема задания**: Разработка интерактивного учебного пособия с ответами на естественном языке на основе Retrieval Augmented Generation

**Место прохождения пратики:** Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Новосибирский национальный исследовательский государственный университет». 630090, Новосибирская область, г. Новосибирск, ул. Пирогова, д. 1

**Сроки прохождения практики:** с 27.02.2025 г. по 17.03.2025 г.

**Руководитель практики от НГУ** Галактионова Юлия Юрьевна, специалист УМОВОИИР /

(Ф.И.О. полностью, должность) (подпись)

**Руководитель ВКР**   Оглезнев Никита Сергеевич, сотрудник КафИСТИИР, ассистент /

(Ф.И.О. полностью, должность) (подпись)

**Оценка по итогам защиты отчета:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(неудовлетворительно, удовлетворительно, хорошо, отлично)

**Отчет заслушан на заседании кафедры** КафИСТИИР

(наименование кафедры)

**протокол \_\_\_\_\_\_\_\_\_от** «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_\_г.

Новосибирск 2025 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc193071044)

[1 ДИЗАЙН-ДОКУМЕНТ 4](#_Toc193071045)

[1.1 Постановка задачи 4](#_Toc193071046)

[1.2 Обзор плана решения задачи 4](#_Toc193071047)

[1.2.1 Этап 1 – построение сервера 4](#_Toc193071048)

[1.2.2 Этап 2 – построение клиента 4](#_Toc193071049)

[1.2.3 Этап 3 – интеграция и тестирование клиент-сервера 4](#_Toc193071050)

[1.2.4 Этап 4 – пилотный запуск 4](#_Toc193071051)

[1.3 Сервер 4](#_Toc193071052)

[1.4 Клиент 5](#_Toc193071053)

[1.5 Пилот 5](#_Toc193071054)

[1.6 Требования к работе системы 5](#_Toc193071055)

[1.6.1 Пропускная способность 6](#_Toc193071056)

[1.6.2 Задержка 6](#_Toc193071057)

[1.6.3 Расчет ресурсов 6](#_Toc193071058)

[1.6.4 Безопасность системы 7](#_Toc193071059)

[2 РЕЗУЛЬТАТЫ 8](#_Toc193071060)

[2.1 Построение сервера 8](#_Toc193071061)

[2.2 Построение клиента 8](#_Toc193071062)

[ЗАКЛЮЧЕНИЕ 9](#_Toc193071063)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10](#_Toc193071064)

ВВЕДЕНИЕ

В рамках практики основное внимание было сосредоточено на проектировании и разработке отдельных частей для пилотного клиент-серверного приложения. Пилотный запуск системы является важным этапом в разработке, представляющим собой тестирование приложения в реальных условиях с ограниченной аудиторией пользователей.

Создание дизайн-документа и проработка архитектурных решений стали ключевыми этапами, которые помогли структурировать проект и подготовить основу для его дальнейшего развития. Данный подход позволил системно подойти к реализации приложения, определив четкие цели и этапы разработки, что значительно облегчает переход к этапу пилотного запуска и последующим улучшениям системы.

1. ДИЗАЙН-ДОКУМЕНТ

В данном случае ML-System-Design-документ представляет собой план по разработке пилотного клиент-серверного приложения. Для успешной реализации проекта мне было важно тщательно проработать ключевые аспекты разработки, чтобы избежать путаницы и обеспечить плавное течение работы на всех этапах. В качестве основы был использован дизайн-документ, предложенный командой Reliable-ML [1]. Я намеренно не буду рассматривать уже пройденные итерации развития проекта, и сосредоточусь на текущей реализации.

* 1. Постановка задачи

На текущей итерации требуется разработка пилотной версии клиент-серверного приложения на основе ранее протестированного бейзлайна RAG. Необходимо создать API-приложение с веб-клиентом для взаимодействия с пользователями.

* 1. Обзор плана решения задачи
     1. Этап 1 – построение сервера

Внедрение бейзлайна в API-приложение с реализацией авторизации, хранением файлов и индексации данных.

* + 1. Этап 2 – построение клиента

Разработка клиентской оболочки, включая функционал для просмотра PDF-файлов, чата с системой и интерфейса загрузки файлов, ориентируясь на ChatPDF и ChatUI.

* + 1. Этап 3 – интеграция и тестирование клиент-сервера

Тестирование взаимодействия серверной и клиентской частей системы, проверка их совместной работы.

* + 1. Этап 4 – пилотный запуск

Запуск приложения для ограниченной аудитории с целью сбора обратной связи и проверки работы в реальных условиях.

* 1. Сервер

В проекте использую наработки с предыдущих итераций. У меня уже есть сервер, совместимый с OpenAI, на FastAPI, однако он требует дополнений. Основные задачи — реализация методов для загрузки документов, авторизации и регистрации пользователей. Для этого продумал схемы хранения данных о пользователях и файлах.

1 Хранение информации о пользователях. Для хранения информации о пользователях я решил использовать SQLite с библиотекой SQLAlchemy. Такой выбор обусловлен необходимостью быстрого прототипирования и масштабируемости системы в будущем. SQLite идеально подходит для проекта на начальной стадии, так как не требует развертывания отдельного сервиса. Использование SQLAlchemy позволит эффективно взаимодействовать с базой данных и упростит расширение системы при необходимости. В будущем можно будет перейти на более сложное решение, если нагрузка на базу данных увеличится.

2 Хранение файлов. Для хранения файлов пользователей буду использовать локальное файловое хранилище. Каждый файл будет храниться как обычный файл на сервере. Такой подход позволяет сэкономить время на интеграцию с облачными хранилищами (например, S3), что подходит для пилотной версии с ограниченной нагрузкой.

3 Хранение индексированных чанков. Для реализации полнотекстового поиска по фрагментам текста я решил использовать модуль FTS5 для SQLite. Это решение подходит для текущих требований проекта и позволит реализовать полнотекстовый поиск без необходимости интегрировать более сложные системы на начальной стадии разработки. В будущем, если потребности в поиске будут увеличиваться, можно будет рассмотреть переход на более мощные решения.

* 1. Клиент

Для реализации клиентской части системы я выбрал ChatUI, написанный на Svelte, который представляет собой гибкий и современный фреймворк для создания чатов с большими языковыми моделями. Я адаптировал ChatUI для интеграции с системой, а также добавил поддержку PDF-документов с использованием открытого PDF-ридера. Это позволяет пользователям удобно просматривать и взаимодействовать с документами в процессе общения с системой.

На текущий момент интеграция PDF-ридера с чатом функционирует, однако необходимо доработать несколько ключевых элементов. В частности, предстоит внедрить полноценный интерфейс для загрузки файлов, что позволит пользователям загружать свои документы в систему для дальнейшей обработки.

* 1. Пилот

В моем проекте пилотная версия нужна для выявления возможных мест для улучшения при использовании в условиях, приближенных к реальным. Для оценки качества пилотной версии я собираюсь собрать мнение пользователей и сформировать дальнейший план развития проекта.

* 1. Требования к работе системы

На данном этапе требования к системе будут рассчитываться исходя из пилотной версии: до 10 пользователей.

* + 1. Пропускная способность

Для расчета пропускной способности системы предположим, что каждый пользователь делает один запрос в систему раз в 15 секунд. Таким образом, максимальная нагрузка на систему составит:

где RPS – количество запросов за секунду. Для обеспечения стабильной работы системы с запасом, включая возможные пики нагрузки, установлю целевую пропускную способность на уровне 1.5 RPS. Это позволит учесть различные сценарии, такие как задержки в сети или повышение активности пользователей.

Для проверки работоспособности системы при заданной пропускной способности планируется использовать нагрузочное тестирование с использованием сценариев в JMeter, чтобы моделировать большое количество запросов в секунду и оценить возможные узкие места в системе.

* + 1. Задержка

Для пилотной версии важно, чтобы пользователи не ощущали больших задержек. Желательная задержка API: 1 секунда.

* + 1. Расчет ресурсов

1 GPU. Для запуска RAG необходима эмбеддинг модель. Она разворачивается локально. Я использую модель deepvk/USER-bge-m3 на 359 миллионов параметров. Согласно открытым источникам [2] только для хранения модели необходимо 1.44гб VRAM-памяти. Кроме того, нужно учесть использование памяти для хранения промежуточных данных и других вычислительных элементов. При обработке запросов пакетами по 32 фрагмента будет необходимо около 4гб VRAM-памяти.

2 CPU. Для системы на 10 пользователей будет достаточно 2 виртуальных ядра на запрос, таким образом, минимальные требования для системы, с запасом: 4 ядра CPU.

3 RAM. Для пилотной версии будет достаточно 2-4гб RAM, т.к. предполагаемая нагрузка на выполнение индексации небольшая.

4 Диск. Для хранения файлов пользователей будет достаточно дискового пространства в 200мб на каждого пользователя. Кроме хранения загруженных файлов необходимо хранить еще и индексированные чанки для каждого документа, а также базу данных для информации о пользователях и другие вспомогательные файлы. Таким образом, для поддержания системы из 10 пользователей понадобится:

* + 1. Безопасность системы

Для обеспечения безопасности доступа к системе в рамках пилотного проекта используется метод предопределенного ключа API, который ограничивает доступ только для заранее определенной группы пользователей. Этот ключ передается в заголовке каждого запроса, и позволяет убедиться, что доступ к системе получают только авторизованные лица.

1. РЕЗУЛЬТАТЫ
   1. Построение сервера

На текущий момент разработана серверная часть, основанная на ранее протестированном бейзлайне RAG. Реализованы ключевые компоненты, такие как система авторизации пользователей и хранение данных. Однако полное внедрение и тестирование этих функциональностей еще предстоит.

* 1. Построение клиента

Клиентская часть находится на стадии разработки. В настоящее время реализован базовый интерфейс с чат-системой и функцией просмотра PDF-документов. Продолжаются работы по расширению функционала и улучшению интерфейса.

ЗАКЛЮЧЕНИЕ

В ходе практики был разработан дизайн-документ, на основе которого осуществляется разработка ключевых компонентов пилотной версии клиент-серверного приложения. На текущий момент завершены этапы проектирования серверной и клиентской частей, продолжаются работы по разработке и интеграции отдельных частей системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ML System Design Doc – Reliable ML [Электронный ресурс]. — 2025. — Режим доступа: <https://github.com/IrinaGoloshchapova/ml_system_design_doc_ru/tree/main> (дата обращения: 15.03.2025).
2. HuggingFace [Электронный ресурс]. — 2025. — Режим доступа: https://huggingface.co/deepvk/USER-bge-m3 (дата обращения: 15.03.2025).