Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

СИСТЕМА ИНТЕЛЛЕКТУАЛЬНОГО ПОИСКА

Отчет по дисциплине

«Учебная проектная деятельность»

Студент гр. 431-3

\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.П. Бекиш

(подпись)

« » 2024 г.

(дата)

Руководитель:

Профессор кафедры АСУ,

д.т.н., доц.

\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Захарова

(подпись)

Томск 2024

**Оглавление**

[Введение 3](#_Toc186101450)

[1 АНАЛИЗ ТРЕБОВАНИЙ 5](#_Toc186101451)

[1.1 Проблематика 5](#_Toc186101452)

[1.2 Формулирование требований 5](#_Toc186101453)

[1.3 Обзор аналогов 6](#_Toc186101454)

[2 ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ И СПЕЦИФИКАЦИЯ 8](#_Toc186101455)

[2.1 Выбор модели 8](#_Toc186101456)

[2.2 Выбор инструментов реализации 8](#_Toc186101457)

[2.3 Входные и выходные данные 9](#_Toc186101458)

[2.3.1 Входные данные 9](#_Toc186101459)

[2.3.2 Выходные данные 9](#_Toc186101460)

[3 ПРОЕКТИРОВАНИЕ 10](#_Toc186101461)

[3.1 Диаграмма последовательности 10](#_Toc186101462)

[4 КОДИРОВАНИЕ 11](#_Toc186101463)

[4.1 Конфигурирование инструментов разработки 11](#_Toc186101464)

[4.2 Программная реализация 11](#_Toc186101465)

[5 ТЕСТИРОВАНИЕ 13](#_Toc186101466)

[Заключение 14](#_Toc186101467)

[Список использованных источников 15](#_Toc186101468)

[Приложение А (справочное) Диаграмма последовательности 16](#_Toc186101469)

[Приложение Б (справочное) Тестирование 17](#_Toc186101470)

# Введение

Языковые модели [1] являются важной и активно развивающейся областью, которая за последние несколько лет привлекает большое внимание промышленности, научных и академических кругов.

В современном мире объемы информации с каждым днем растут все больше и больше, поэтому анализировать информацию становится труднее, как и выбрать более корректную. Организациям, у которых есть своя база знаний, из которой они хотят так же получать точную информацию, избегая анализа лишнего материала. Обычному человеку, который большую часть информации впитывает из интернета, где публикуют все возможные данные по той теме, которую он ищет, что становится трудно в анализе как минимум ее достоверности.

Объект разработки представляет собой языковые модели. Предметом разработки является использование методов и технологий языковых моделей в корпоративных базах знаний.

В связи с этим, в качестве основной задачи выступает создание системы, которая позволяет обычному человеку искать ту информацию, которая необходима, а корпорациям внедрять или подключать в свои основные системы, чтобы находить необходимые данные в своей базе знаний.

Целью работы является разработка системы, которая обеспечивает по заданному запросу пользователя выдавать быстрый и корректный ответ для дальнейших действий.

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Анализ необходимых требований;
2. Функциональные возможности и спецификации;
3. Проектирование;
4. Реализация системы;
5. Тестирование.

Результатом выполненной работы является система, которая находит нужную информацию на поставленный запрос. Система представляет собой чат, в который пользователь может задавать вопрос, а сотрудник компании подключать внутреннею базу знаний и искать информацию в ней.

# АНАЛИЗ ТРЕБОВАНИЙ

## **Проблематика**

За последние несколько десятилетий объемы информации очень увеличились, что повлекло за собой очень детальный анализ этого входного потока, что очень замедляет как учебный, так и рабочий. В этих условиях традиционные методы работы с информацией и базами знаний становятся неэффективными и требуют значительных ресурсов для поддержания актуальности и доступности информации. Языковые модели (LLM), такие как ChatGPT, предлагают инновационные решения для автоматизации обработки и анализа данных, улучшения поиска и доступа к знаниям для пользователей и сотрудников компаний, а также поддержки принятия решений. Однако, несмотря на очевидные преимущества, многие пользователи и компании еще не полностью осознали потенциал LLM и сталкиваются с вызовами при их внедрении. Важно понять, почему интеграция LLM в собственную жизнь или корпоративную базу является ключевым фактором успеха в условиях быстро меняющегося цифрового ландшафта и как преодолеть барьеры на пути к успешной реализации таких проектов.

## **Формулирование требований**

Обращая внимание на предметную область, были сформулированы определенные функциональные требования к разрабатываемой системе:

* + 1. Пользователь должен иметь возможность авторизоваться для использования системы;
    2. Система должна сформировать токен, с помощью которого пользователь может пользоваться внутренней API;
    3. Пользователь должен иметь возможность формировать запрос;
    4. Элементы интерфейса системы должны включать в себя поле ввода, чтобы пользователь мог сформировать запрос;
    5. Система должна предоставлять возможность пользователю загружать необходимые файлы, если у него стоит задача найти необходимую информацию в них, если же это сотрудник, то внедрить/подключить к внутренней корпоративной базе знаний;
    6. Система должна формировать корректный и понятный ответ пользователю.

## **Обзор аналогов**

Существует множество моделей, которые на данный момент могут удовлетворять условия актуальности и постановки задачи, но более мощные по своей точности, производительности и функционалу подходят следующие:

* + 1. IBM Watson [2] – это мощная платформа искусственного интеллекта, которая включает в себя множество инструментов для обработки естественного языка, машинного обучения и аналитики данных. Она широко используется в корпоративных средах для автоматизации бизнес-процессов, улучшения клиентского обслуживания и оптимизации принятия решений. Watson может интегрироваться с корпоративной базой данных для создания чат-ботов, анализа текстов, прогнозирования и других задач, связанных с обработкой больших объемов данных.

1. Azure Cognitive Services от Microsoft [3] предлагает набор API-интерфейсов для интеграции функций искусственного интеллекта в приложения и системы. Эти сервисы включают обработку текста, речи, изображений и видео. С помощью Azure Cognitive Services можно создавать интеллектуальные приложения, которые взаимодействуют с корпоративными данными, анализируют тексты, распознают речь и многое другое.
2. Google Cloud AI [4] предоставляет инструменты и сервисы для разработки и развертывания моделей машинного обучения, включая Natural Language Processing (NLP). Платформа поддерживает создание и обучение собственных моделей, а также использование готовых решений. Google Cloud AI может быть использована для анализа текстов из корпоративных баз данных, автоматического ответа на запросы клиентов, классификации документов и других NLP-задач.
3. Amazon Lex [5] — это сервис для создания голосовых и текстовых интерфейсов на основе технологии Alexa. Он позволяет разрабатывать чат-боты и другие интерактивные интерфейсы, которые могут взаимодействовать с пользователями через текстовые сообщения или голосовые команды. Lex может быть интегрирован с корпоративной системой для автоматизации взаимодействия с клиентами, сотрудников или партнеров, обеспечивая быстрый доступ к необходимой информации.
4. OpenAI GPT [6] — это семейство языковых моделей, разработанных компанией OpenAI. Они используются для генерации текста, перевода, обобщения и других сложных задач обработки естественного языка. Хотя GPT изначально был создан для открытых систем, он может быть адаптирован и использован в корпоративных приложениях для различных целей, таких как автоматическое составление отчетов, ответы на вопросы пользователей и поддержка принятия решений на основе данных.

Следует отметить, что несмотря на наличие аналогов их использование невозможно в связи с санкциями и ограничением доступа российским пользователям. Но проанализировав отечественные решения, отвечающих требованиям, была выбрана модель GigaChat [7].

# ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ И СПЕЦИФИКАЦИЯ

## **2.1 Выбор модели**

За исходную модель, как и было сказано выше была взята GigaChat.

Данная модель основные требования, которые необходимы для создания система, такие как:

* + - авторизация, для получения токена, которая предоставляет API для взаимодействия с моделью;
    - добавление собственных файлов или подключение собственной базы знаний.

## **2.2 Выбор инструментов реализации**

В ходе анализа, основываясь на вышеопределенной области, был выбран набор используемых инструментов, который обеспечивает эффективную разработку. Также, одним из факторов выбора – популярность, стабильность, простота использования, функциональность. Рассмотрим данный набор инструментов разработки:

1. Jupyter Notebook [8] – это интерактивное веб-приложение для создания и совместного использования документов, содержащих код, визуализации данных, математические формулы и пояснительный текст. Это мощный инструмент для анализа данных, машинного обучения, численного моделирования и других задач, связанных с программированием и наукой о данных.
2. Visual Studio Code [9] – это бесплатный редактор исходного кода от компании Microsoft, который стал одним из самых популярных инструментов среди разработчиков благодаря своей гибкости, расширяемости и поддержке множества языков программирования. VS Code предоставляет мощные функции для редактирования кода, отладки, управления версиями и интеграции с различными системами контроля версий и платформами разработки.
3. Streamlit [10] – это библиотека для Python, которая позволяет быстро создавать веб-приложения, поддерживает различные виджеты, такие как ползунки, выпадающие списки, кнопки и чекбоксы, что позволяет пользователям взаимодействовать с приложением в реальном времени. Она идеально подходит для создания прототипов, чатов, ботов, дашбордов и интерактивных отчетов, позволяя разработчикам сосредоточиться на логике приложения, а не на фронтенде.

Выбранный список инструментов описывает функциональность, удобство, гибкость, скорость и удобство пользования. Обеспечивают полную поддержку для всех этапов разработки, начиная от настройки конфигурации до анализа и редактирования полученного ответа.

## **2.3 Входные и выходные данные**

### **2.3.1 Входные данные**

Для полного функционирования системы, обеспечивающая ответ на запрос, были определены следующие данные, которые поступают в нее:

1. Авторотационные данные, предоставляющие токен для API модели;
2. Сформированный корректный запрос;
3. Дополнительная информация в виде документов или базы знаний при необходимости.

### **2.3.2 Выходные данные**

Выходные данные включают в себя ответ на сформированный запрос, который представляет собой отредактированный текст.

# ПРОЕКТИРОВАНИЕ

## **Диаграмма последовательности**

Диаграмма последовательности, представлена на рисунке А.1 приложения А, была спроектирована для визуализации взаимодействия между различными элементами системы в временных отрезках. В данном подразделе будут рассмотрена диаграмма последовательности, включающая нашу систему и выбранную модель GigaChat.

Для начала пользователь должен пройти авторизацию, чтобы система передала данные модели для получения токена. После отображается системный интерфейс.

После пользователь загружает файлы или подключает базу знаний, в которой ему нужно найти информацию. Отправляет запрос.

Система передает данный запрос в модель, которая ищет и отвечает. Далее, полученный ответ система обрабатывает, отображая в удобном и читабельном виде.

Таким образом диаграмма последовательности демонстрирует внешнее взаимодействие пользователя с системой и выбранной нами модели.

# КОДИРОВАНИЕ

## **4.1 Конфигурирование инструментов разработки**

Вся разработка велась на операционной система Windows, поэтому установка необходимых инструментов, сред и пакетов в основном происходили через ссылки на официальные источники, где находятся установочные устройства.

Установка VS Code была проведена с официального сайта, следую инструкциям.

Как было выше описано – VS Code имеет большое количество внутреннего функционала, то Python можно установить внутри, но версия языка будет фиксирована, поэтому может возникнуть конфликты версии и других компонентов, если они будут не совместимы, поэтому установка будет производится с официального сайта Python.

Так же, как и VS Code все установочный компоненты Python [11] были взяты с официального сайта, следуя инструкциям.

Что касается отображаемого интерфейса, мы используем модуль Python – streamlit, установка осуществляется через командную строку с помощью команды:

pip install streamlit

Данные этапы позволяют создать среду для дальней разработки системы.

## **4.2 Программная реализация**

В проекте было реализовано три пакета: app, big\_text, processing.

В пакете app входит интерфейс, который будет отображаться пользователю, чтобы он мог пользоваться системой.

Пакет processing включает в себя такие функции get\_gigachat\_token, send\_prompt, load\_file, get\_files, delete\_all\_files, embeddings, предсталяющие собой различные компоненты и функциональность. Так же существует файл utils.py для вспомогательных функций и реализаций.

И для того, чтобы пользователь мог загружать файлы или подключать базу знаний, был создан пакет big\_text, который обрабатывает большой объем данных и ищет ответ по запросу.

# ТЕСТИРОВАНИЕ

Тестирование – одно из важных этапов разработки, направленное на выявление верного определения входного и выходного списка, который описывает заявленные системные требования. Тестирование производилось вручную. Результаты можно увидеть в приложение Б, на рисунке Б.1.

# Заключение

В ходе выполнения объявленной тематике были выполнены все поставленные цели и задачи. Таким образом, был разработан начальный прототип системы интеллектуального поиска. Дальнейшее развитие проекта будет заключаться в доработке алгоритма получения контента из файлов. Эта система сможет помочь большинству организациям ускорить рабочий процесс.

Был принят к публикации доклад: Бекиш Е.П.. Система интеллектуального поиска// Наука и практика: проектная деятельность – от идеи до внедрения 2024: материалы XIII региональной науч. – практ. конф., Томск, 2024.

# Список использованных источников

1. Языковая модель [Электронный ресурс]: сайт Wikipedia. URL: <https://en.wikipedia.org/wiki/Language_model> (дата обращения: 10.12.2024).
2. IBM Watson [Электронный ресурс]: сайт Wikipedia. URL: <https://en.wikipedia.org/wiki/IBM_Watson> (дата обращения: 10.12.2024).
3. Azure AI services [Электронный ресурс]: сайт Microsoft Learn Challenge. URL: <https://learn.microsoft.com/en-us/azure/ai-services/> (дата обращения: 10.12.2024).
4. Google Cloud AI Platform [Электронный ресурс]: сайт erp-information. URL: <https://www.erp-information.com/google-cloud-ai-platform> (дата обращения: 10.12.2024).
5. Amazon Lex [Электронный ресурс]: сайт Wikipedia. URL: <https://en.wikipedia.org/wiki/Amazon_Lex> (дата обращения: 10.12.2024).
6. GPT-4 [Электронный ресурс]: сайт OpenAI. URL: <https://openai.com/index/gpt-4/> (дата обращения: 10.12.2024).
7. GigaChat [Электронный ресурс]: сайт GigaChat. URL: <https://giga.chat/> (дата обращения: 10.12.2024).
8. Jupyter Notebook [Электронный ресурс]: сайт Jupyter. URL: <https://jupyter.org/> (дата обращения: 10.12.2024).
9. VS Code [Электронный ресурс]: сайт Visual Studio Code. URL: <https://code.visualstudio.com/> (дата обращения: 10.12.2024).
10. Streamlit [Электронный ресурс]: сайт Streamlit. URL: <https://streamlit.io/> (дата обращения: 10.12.2024).
11. Python [Электронный ресурс]: сайт Python. URL: <https://www.python.org/> (дата обращения: 10.12.2024).

# Приложение А (справочное) Диаграмма последовательности

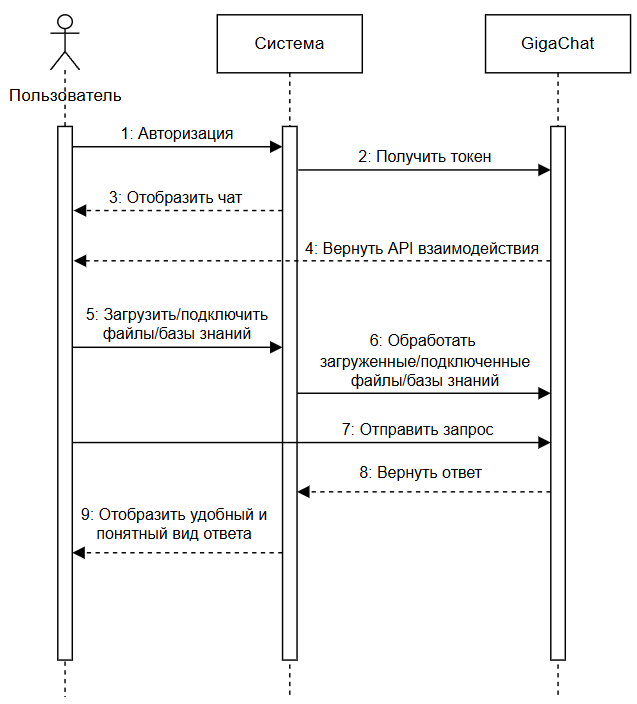


Рисунок А.1 – Диаграмма последовательности

# Приложение Б (справочное) Тестирование

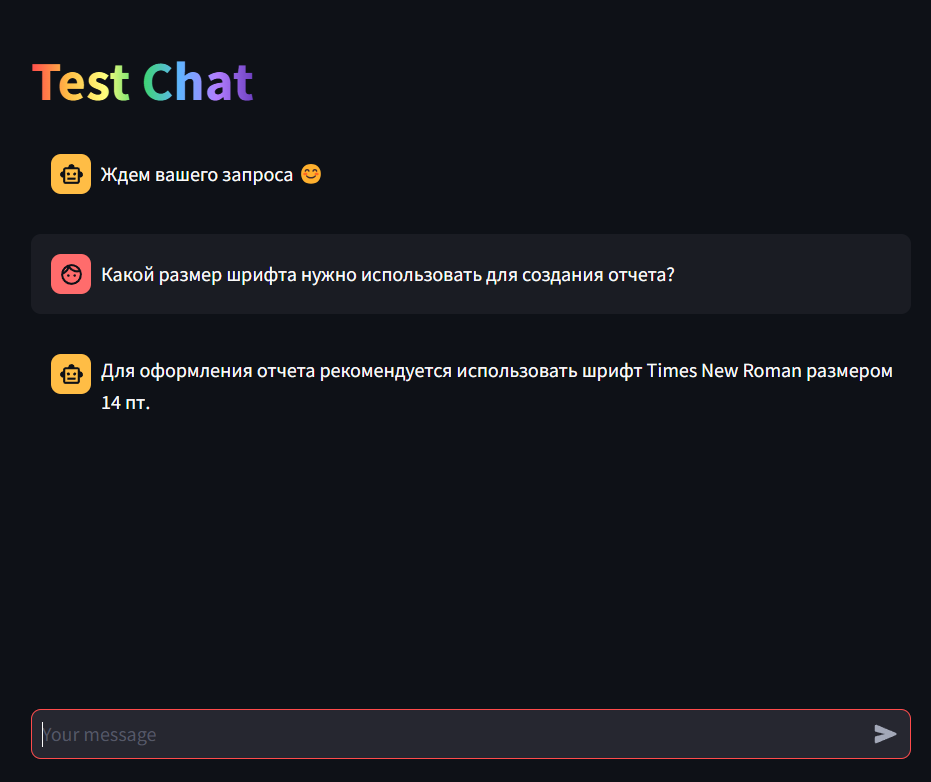


Рисунок Б.1 – Экран системы