Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ

Школа бакалавриата

ОТЧЕТ

По проекту

## «Транскрибация аудио-записи совещания и составление его протокола по шаблону»

по дисциплине «Проектный практикум»

|  |  |
| --- | --- |
| Заказчик: Фамилия И.О. | Шестеров М. А. |
| Куратор: Фамилия И.О.  ученая степень, ученое звание, должность | Шестеров М. А. |
| Студенты команды “Прекрасные лица”  Фамилия И.О. | Лемтюгин С. А. |
| Фамилия И.О. | Макушкин М. В. |
| Фамилия И.О. | Егоров Н. Р. |

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1 Основная часть 5

1.1 Отчет работы каждого из участников 5

1.1.1 Разработчки 1 5

1.1.2 Разработчик 2 7

1.1.3 Тимлид/аналитик 9

1.2 Разбор требований к продукту 11

1.3 Анализ и составление аналогов разрабатываемого продукта 14

1.4 Архитектура программного продукта 18

1.5 Описание методологии разработки 23

1.6 Информация о планировании деятельности в ходе разработки и распределении задач между участниками команды разработчиков 26

ЗАКЛЮЧЕНИЕ 28

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30

Приложение A Изображения файлов разработки и тестирование 31

ВВЕДЕНИЕ

## 1. Четкое и полное описание цели и задач проекта.

Проект направлен на автоматизацию процесса составления протоколов совещаний путем трансформации аудиозаписей в текстовые документы. Основная цель заключается в создании работающего сервиса, который преобразует аудиофайлы, полученные с видеозвонков и встреч, в текст, разбитый по спикерам, а также автоматически формирует протокол по заранее установленному шаблону. Задачи проекта включают настройку системы WhisperX для распознавания русскоязычной речи, оптимизацию модели для повышения точности, интеграцию с искусственным интеллектом для заполнения протоколов, разработку сервиса в облаке для удобства пользователей и улучшение качества формирования протоколов.

## 2. Обоснование актуальности и важности проекта.

В современных компаниях создание протоколов совещаний является важным процессом, который зачастую требует значительных временных затрат и может быть подвержен человеческим ошибкам. Актуальность данного проекта обусловлена необходимостью повышения эффективности работы команд и сокращения времени на подготовку отчетности. Автоматизация данного процесса позволит получать протоколы сразу после завершения встречи, что значительно улучшит коммуникацию и оперативность в принятии решений. Проект также отвечает требованиям современного бизнеса, стремящегося к оптимизации процессов и внедрению инновационных технологий.

## 3. Описание области применения программного продукта.

Программный продукт будет востребован в различных сферах, включая корпоративный сектор, образовательные учреждения, а также в государственных организациях, где ведется документирование встреч и совещаний. Сервис будет полезен для специалистов по документированию, секретарей, менеджеров и всех, кто участвует в проведении и организации встреч. Он обеспечит удобный и быстрый способ получения протоколов, что, в свою очередь, повысит продуктивность работы и улучшит качество взаимодействия между участниками.

## 4. Описание ожидаемых результатов и планируемых достижений по завершении проекта.

Ожидаемые результаты включают создание высококачественного сервиса, который эффективно распознает речь и формирует протоколы с минимальным количеством ошибок. К моменту завершения проекта будет реализована функциональность, позволяющая пользователям получать готовые протоколы сразу после завершения встреч. Планируемые достижения также включают успешное тестирование и оптимизацию модели, интеграцию с ИИ для автоматического заполнения протоколов и разработку удобного сервиса. В конечном итоге проект должен существенно упростить процесс документирования встреч и повысить его качество.

1. Основная часть
   1. Отчет работы каждого из участников
      1. Разработчки 1

Отчет по проделанной работе Макушкина Максима Васильевича.

Этап 1: Подбор и тестирование моделей для транскрибации

В начале разработки локально развернул и протестировал различные модели Whisper (включая whisperx) с целью оценки производительности на разных конфигурациях (модели base, medium, large, turbo). Необходимо было протестировать работу моделей на GPU. Измерить время выполнения транскрибации, выравнивания и диаризации на тестовых аудиофайлах продолжительностью от 8 минут до 2 часов, а также сравнить качество двух моделей по соотношению время/точность Whisper и WhisperX.

Этап 2: Подбор и тестирование языковых моделей

Для генерации финального протокола после транскрибации была выбрана языковая модель LLaMA с использованием библиотеки llama-cpp-python. Модель запускалась локально на GPU. В рамках данного этапа были проделаны следующие действия: скачал и протестировал несколько версий модели LLaMA в формате gguf, подобрал оптимальные параметры для запуска модели (например, n\_gpu\_layers, n\_ctx, n\_threads), провел замеры времени генерации ответа и добился снижения времени генерации при сохранении качества, адаптировал формат prompt'а, чтобы модель возвращала краткий, структурированный протокол, настроил сохранение результатов генерации в файл.

Этап 3: Разработка серверного приложения (FastAPI)

На этом этапе перешел к созданию FastAPI-приложения. Реализован эндпоинт /process, принимающий файл, выполняющий транскрибацию, выравнивание, диаризацию и генерацию протокола. Настроил возвращение результата пользователю в виде текстового файла через FileResponse.

Этап 4: Аутентификация и регистрация пользователей

Теперь было необходимо подключить базу данных и настроить регистрацию и аутентификацию пользователей. Реализовал регистрацию и вход пользователей с сохранением хэшированных паролей (использовалась passlib). Добавил генерацию JWT-токенов и их хранение в cookies. Реализовал выход из системы с удалением токена. Обеспечил CSRF-защиту и проверку длины пароля при регистрации.

Этап 5: Создание веб-интерфейса используя Jinja2 шаблоны

Создал страницы:

• / — форма входа;

• /register — форма регистрации;

• /upload — загрузка файла.

Был создан минималистичный дизайн. Также настроил защиту страниц от неавторизованного доступа через токены в cookie.

В результате всех этапов был создан и стабильно работает сервис, позволяющий:

• Пользователю зарегистрироваться и войти;

• Загрузить аудиофайл;

• Получить готовый текст и протокол;

• Работать через веб-интерфейс или Postman.

* + 1. Разработчик 2

Отчет по проделанной работе Егорова Никиты Романовича.

Этап 1: Анализ конкурентов, тестирование моделей

Занимался обсуждением и формированием требований и критериев нашего проекта, а также помощи в составлении отчетов, презентаций и видео к контрольным точкам. Выступал на консультациях с докладом.   
Также провел анализ конкуретов, подбор вариантов реализации нашего проекта и тестирование транскрибирующих моделей. Развернул модель pyannote, которая отвечает за разбиение аудио и видео файлов по спикерам, а также разобрался и установил модель whisper, whisperX к себе на компьютер, для дальнейшего тестирования.

Занимался обсуждением и формированием требований и критериев, необходимых для успешной реализации нашего проекта. В ходе этой работы активно сотрудничал с командой, чтобы определить ключевые показатели эффективности, которые будут служить основой для оценки результатов. Также помогал в составлении отчетов, презентаций и видеоматериалов для контрольных точек, что способствовало лучшему пониманию целей проекта всеми участниками.

В дополнение к этому провел глубокий анализ конкурентов, что позволило выделить сильные и слабые стороны существующих решений на рынке. Исследовал различные варианты реализации нашего проекта, учитывая наилучшие практики и инновационные подходы, применяемые другими командами. В процессе тестирования транскрибирующих моделей развернул модель pyannote, которая отвечает за разбиение аудио и видео файлов по спикерам. Это позволило получить практический опыт работы с алгоритмами обработки речи и их настройкой.

Кроме того, установил и протестировал модели whisper и whisperX на своем компьютере, что дало возможность оценить их производительность и качество транскрипции. Изучил их документацию, чтобы понять, как можно адаптировать их под наши нужды и улучшить результаты транскрибирования.

Этап 2: Реализация транскрибации в реальном времени

Основной задачей на данном этапе было создание системы транскрибации в реальном времени. Это требовало значительного времени и усилий, поскольку необходимо было обеспечить высокую точность и скорость обработки данных. Я занимался тщательным анализом документации, чтобы выявить наиболее эффективные подходы к реализации задуманной задачи.

В процессе работы я исследовал различные технологии и инструменты, которые могут быть использованы для достижения поставленных целей. Я также провел несколько экспериментов с настройками моделей, чтобы оптимизировать их работу в условиях реального времени. Это включало в себя тестирование различных параметров и алгоритмов, а также анализ их влияния на качество итоговой транскрипции.

Я активно взаимодействовал с командой, обсуждая возможные решения и полученные результаты, что способствовало более глубокому пониманию проблемы и выработке совместных стратегий. Мы также проводили регулярные встречи для обсуждения прогресса и корректировки плана действий.

Этап 3: В результате проделанной работы разработка системы транскрибации в реальном времени продолжается. Активно работаем над подбором оптимальных параметров для достижения наилучших результатов. В процессе реализации занимаюсь как предобработкой, так и постобработкой данных, что позволяет улучшить качество транскрипции и минимизировать ошибки.

Предобработка включает в себя очистку и подготовку аудиофайлов, а также настройку моделей для работы в условиях реального времени. Постобработка, в свою очередь, направлена на улучшение итоговых результатов транскрипции, включая корректировку возможных ошибок и форматирование текста для удобства восприятия.

* + 1. Тимлид/аналитик

Отчёт по проделанной работе Лемтюгина Степана Андреевича.

Этап 1: Инициирование проекта и постановка задач

На начальном этапе я выступал как инициатор и руководитель разработки сервиса. Сформулировал цели проекта, продумал архитектуру будущей системы и определил ключевые компоненты: транскрибация, генерация протокола, серверная часть, пользовательский интерфейс. Также предложил пути развития сервиса, включая поддержку длинных аудио, улучшение качества вывода и расширение интерфейса.

Этап 2: Управление командой и распределение задач

Выполнял обязанности тимлида: организовывал рабочие созвоны, ставил задачи, распределял обязанности между участниками команды, следил за сроками выполнения. Проводил технические обсуждения и помогал принимать решения по реализации функций и выбору инструментов.

Этап 3: Тестирование качества транскрибации

Лично проводил тесты транскрибации на разных параметрах и моделях Whisper/WhisperX. Анализировал качество распознавания и скорость обработки при различных конфигурациях (модели, длина аудио, параметры вызова функций), подбирал оптимальные значения для получения наилучшего соотношения точность/время выполнения.

Этап 4: Контроль разработки и архитектуры проекта

Следил за соблюдением единой структуры проекта, вносил правки в код и архитектуру при необходимости. Обеспечивал читаемость и поддерживаемость кода, следил за тем, чтобы реализация соответствовала поставленным требованиям и не нарушала логическую целостность сервиса.

Этап 5: Развёртывание и DevOps-задачи

В настоящее время занимаюсь развёртыванием приложения. Подготавливаю окружение, настраиваю сборку и запуск FastAPI-приложения, прорабатываю возможные сценарии масштабирования и интеграции с другими сервисами. Также решаю задачи, связанные с безопасностью и производительностью при развертывании.

Итог:

В результате проделанной работы были заложены прочные основы для стабильной и масштабируемой системы. Мною были инициированы ключевые решения, организована эффективная работа команды, выполнены исследования для повышения качества транскрибации, а также обеспечен контроль за реализацией и подготовка системы к развёртыванию.

* 1. Разбор требований к продукту

Так как проект «Транскрибация аудио-записи совещания и составление его протокола по шаблону» является студенческий, требования и критерии, которые должны быть выполнены для успешной реализации проекта, выдвигались нашей командой, а также учитывались комментарии и советы комиссии на консультациях. Эти требования помогут определить функциональность, производительность и качество конечного продукта:

1. **Функциаональные требования**

Функциональные требования описывают, что именно должен делать продукт. Для нашего сервиса по транскрибации аудиозаписей совещаний можно выделить следующие ключевые функции:

1. **Преобразование аудиофайлов в текст**: сервис должен поддерживать различные форматы аудиофайлов (например, MP3, WAV) и обеспечивать высокую точность распознавания русскоязычной речи.
2. **Разбиение текста по спикерам**: важно, чтобы сервис мог идентифицировать разных спикеров и правильно разделять их реплики в итоговом тексте.
3. **Автоматическое заполнение протокола**: на основе полученного текста сервис должен формировать протокол по заранее установленному шаблону, включая основные моменты встречи, решения и действия.
4. **Интерфейс пользователя**: сервис должен иметь интуитивно понятный интерфейс, что обеспечит легкость в использовании.
5. **Разработка транскрибации в реальном времени**: сервис должен поддерживать функцию онлайн распознавания, позволяя пользователям транскрибировать аудио в реальном времени с микрофона, что повысит удобство использования и эффективность работы.
6. **Нефункциональные требования**

Нефункциональные требования касаются качества и производительности системы. В нашем случае они могут включать:

1. **Производительность**: система должна обрабатывать аудиофайлы и генерировать протоколы в разумные сроки, предпочтительно в режиме реального времени или с минимальными задержками.
2. **Надежность**: сервис должен быть устойчивым к сбоям и обеспечивать высокую доступность. В случае ошибок или сбоев, система должна предоставлять пользователю понятные сообщения об ошибках.
3. **Безопасность**: обработка аудиофайлов и текстов должны соответствовать современным стандартам безопасности, включая защиту данных пользователей и конфиденциальность информации.
4. **Масштабируемость**: система должна быть способна обрабатывать увеличивающееся количество пользователей и аудиофайлов без значительного ухудшения производительности.
5. **Поддержка**: необходимо обеспечить наличие документации и службы поддержки для пользователей, что позволит им быстро решать возникающие вопросы и проблемы.

**Критерии успешности проекта**

Для оценки успешности проекта можно использовать следующие критерии:

1. **Точность распознавания речи**: успешный проект должен обеспечивать точность распознавания речи не менее 85–90%. Это критически важно для формирования качественного протокола.
2. **Скорость обработки**: Время, необходимое для преобразования аудиофайла в текст и формирования протокола, должно составлять не более 10–15 минут для стандартного совещания продолжительностью до 1 часа.
3. **Уровень удовлетворенности пользователей**: оценка удовлетворенности пользователей, проведенная через опросы и отзывы, должна составлять не менее 80%. Это поможет понять, насколько сервис отвечает потребностям пользователей.
4. **Количество выявленных ошибок**: в процессе тестирования и эксплуатации сервиса количество ошибок и недочетов должно оставаться на минимальном уровне, что свидетельствует о высоком качестве разработки.

Backlog:

1. Настройка WhisperX для распознавания речи.
2. Интеграция с ИИ (Llama) для автоматического заполнения протоколов.
3. Разработка пользовательского интерфейса.
4. Тестирование и оптимизация модели.
5. Создание сервиса в облаке.
6. Подготовка документации для пользователей.
7. **Разработка функционала для транскрибации в реальном времени**.
   1. Анализ и составление аналогов разрабатываемого продукта

На рынке существует несколько аналогичных решений, таких как:

1. **Otter.ai** - облачный сервис для автоматической транскрибации аудио и видео. Он поддерживает множество языков, включая английский, и предлагает функции, такие как распознавание спикеров, создание заметок и интеграцию с видеозвонками (например, Zoom).

**Преимущества**:

1. **Интуитивно понятный интерфейс**: пользователи могут легко загружать аудиофайлы и получать текстовые расшифровки.
2. **Распознавание спикеров**: otter.ai автоматически идентифицирует и различает голос каждого участника, что облегчает понимание контекста обсуждений.
3. **Интеграция с другими сервисами**: поддержка интеграции с Zoom и другими платформами для видеозвонков.

**Недостатки:**

1. **Ограниченная поддержка русского языка**: хотя Otter.ai поддерживает несколько языков, качество распознавания русскоязычной речи может быть недостаточным.
2. **Проблемы с точностью**: в зависимости от качества аудиозаписи, могут возникать ошибки в транскрибации, что требует ручного редактирования.
3. **Платные функции**: некоторые функции доступны только в платной версии.
4. **Sonix.ai** - сервис, который предлагает услуги по транскрибации аудио и видео с использованием как автоматизированных, так и ручных методов. Пользователи могут выбрать между автоматической транскрибацией (с меньшей точностью) и услугами профессиональных транскрибаторов.

**Преимущества**:

1. **Высокая точность ручной транскрибации**: профессиональные транскрибаторы обеспечивают высокую точность, что особенно важно для важных встреч и интервью.
2. **Разнообразие услуг**: rev предлагает не только транскрибацию, но и субтитры, переводы и другие услуги.

**Недостатки:**

1. **Длительное время ожидания**: ручная транскрибация занимает больше времени, чем автоматическая.
2. **Высокая стоимость**: услуги профессиональных транскрибаторов стоят дороже, чем автоматические решения.
3. **Ограниченная автоматизация**: для получения качественной транскрибации пользователям необходимо выбирать между автоматическим и ручным подходом.
4. **Sonix** - облачный сервис для автоматической транскрибации аудио и видео файлов. Он поддерживает множество языков и предлагает функции, такие как создание субтитров, редактирование текста и интеграцию с различными платформами.

**Преимущества**:

1. **Многоязычность**: sonix поддерживает более 30 языков, что делает его универсальным инструментом для пользователей по всему миру.
2. **Интуитивно понятный интерфейс**: пользователи могут легко загружать аудиофайлы, редактировать текстовые расшифровки и экспортировать их в различных форматах.
3. **Интеграция**: sonix предлагает интеграцию с такими сервисами, как Zapier, что позволяет автоматизировать рабочие процессы и улучшать взаимодействие с другими приложениями.
4. **Создание субтитров**: Сервис автоматически генерирует субтитры, что полезно для создания видео-контента.

**Недостатки**:

1. **Точность распознавания**: хотя Sonix показывает хорошие результаты, качество распознавания может варьироваться в зависимости от качества аудиозаписи и акцентов говорящих.
2. **Проблемы с русским языком**: подобно другим аналогам, Sonix может иметь ограничения в точности распознавания русскоязычной речи, что делает его менее подходящим для пользователей, работающих исключительно с русским контентом.
3. **Платные функции**: некоторые функции, такие как экспорт в определенные форматы или использование дополнительных инструментов, доступны только в платных версиях.
4. Статья на Habr (https://habr.com/ru/articles/804329/)

В статье на Habr описывается примеры для создания голосового ассистента используя WhisperX и интеграции его в telegram-бота. В ней рассматриваются различные архитектуры, такие как трансформеры, которые применяются для распознавания речи. Основное внимание уделяется тому, как эти технологии могут быть использованы для повышения точности и эффективности транскрибации.

Основные моменты статьи:

1. Технологии распознавания речи: обсуждаются различные алгоритмы, включая RNN (рекуррентные нейронные сети) и их современные аналоги, такие как трансформеры, которые показывают более высокую точность в распознавании речи.
2. Обучение моделей: рассматриваются методы обучения моделей на больших объемах данных, что позволяет улучшить качество распознавания.
3. Применение в реальных задачах: статья подчеркивает важность использования таких технологий в коммерческих продуктах, где точность и скорость обработки данных имеют критическое значение.
4. Проблемы и вызовы: обсуждаются сложности, связанные с различными акцентами, шумами в фоновом режиме и другими факторами, которые могут влиять на качество распознавания.
5. GitHub-проект (https://github.com/trashchenkov/gigachat\_tutorials/blob/main/%D0%93%D0%BE%D0%BB%D0%BE%D1%81%D0%BE%D0%B2%D0%BE%D0%B9\_%D0%B1%D0%BE%D1%82.ipynb)

Проект на GitHub, связанный с SaluteSpeech и GigaChat, предлагает примеры и руководства по разработке решений для обработки аудио и видео. Он включает в себя использование различных библиотек и технологий для создания собственных систем распознавания речи. Проект является хорошим примером того, как можно использовать существующие библиотеки для создания голосовых интерфейсов. Он предоставляет базу для разработчиков, которые хотят создать свои собственные голосовые приложения, и может быть расширен для более сложных задач.

* 1. Архитектура программного продукта

В данном разделе мы подробно рассмотрим архитектуру нашего программного продукта, который включает в себя возможность распознавания как загруженных аудиофайлов, так и онлайн распознавания с микрофона. Проект включает в себя использование FastAPI в качестве фронтенда, модель WhisperX для распознавания речи, модель Llama для автоматического заполнения шаблона протокола и веб-интерфейс, реализованный с помощью Jinja2.

**Основные компоненты архитектуры**

1. **Frontend (FastAPI с Jinja2)**:

**Описание**: FastAPI — это современный веб-фреймворк для создания API на Python, который будет использоваться вместе с Jinja2 для рендеринга HTML-шаблонов. Jinja2 позволяет динамически генерировать веб-страницы, что улучшает взаимодействие пользователей с приложением.

**Функции**:

* Обработка HTTP-запросов от пользователей (загрузка аудиофайлов).
* Генерация веб-страниц с формами для загрузки файлов и отображения результатов.
* Взаимодействие с пользователем через RESTful API и рендеринг страниц с помощью Jinja2.
* **Поддержка онлайн распознавания**, возможность транскрибировать аудио в реальном времени с использованием микрофона.

1. **Модель WhisperX**:

**Описание**: WhisperX — это усовершенствованная версия модели Whisper, оптимизированная для быстрого и точного распознавания аудиозаписей на русском языке.

**Функции**:

* Преобразование загруженных аудиофайлов в текстовый формат.
* Разделение текста по спикерам для удобства формирования протокола.
* Обработка различных форматов аудиозаписей (например, MP3, WAV).
* **Онлайн распознавание**: поддержка распознавания речи в реальном времени с микрофона для мгновенного получения текста.

1. **Модель Llama**:

**Описание**: Llama — это мощная языковая модель, предназначенная для генерации текста и выполнения задач обработки естественного языка. Она будет использоваться для автоматического заполнения шаблона протокола на основе распознанного текста.

**Функции**:

* Генерация структурированного текста на основе входных данных.
* Заполнение шаблона протокола с учетом логики и содержания встречи.
* Поддержка различных форматов шаблонов для адаптации под требования пользователей.

**Связи между компонентами**

1. **Frontend (FastAPI с Jinja2) ↔ Модель WhisperX**:

* Пользователь загружает аудиофайл через веб-интерфейс, который рендерится с помощью Jinja2.
* FastAPI обрабатывает запрос на загрузку и отправляет аудиофайл на обработку в модель WhisperX через асинхронный запрос.
* Модель WhisperX возвращает распознанный текст вместе с разбиением по спикерам обратно в FastAPI.
* При онлайн распознавании FastAPI устанавливает соединение с клиентом через WebSocket, чтобы передавать аудиоданные в реальном времени в модель WhisperX, которая возвращает текст немедленно.

1. **FastAPI ↔ Модель Llama**:

* После получения текста от WhisperX, FastAPI формирует запрос к модели Llama для заполнения протокола.
* Модель Llama обрабатывает текст и генерирует заполненный протокол, который возвращается в FastAPI.

1. **FastAPI ↔ Пользователь**:

* FastAPI рендерит веб-страницу с результатами, включая готовый протокол, и отправляет её пользователю через интерфейс Jinja2.

**Обоснование выбора архитектурного решения**

1. Использование FastAPI с Jinja2

* **Производительность**: FastAPI обеспечивает высокую скорость обработки запросов благодаря асинхронной архитектуре, что критично для работы с аудиофайлами и длительными операциями распознавания.
* **Генерация HTML**: Jinja2 позволяет динамически генерировать HTML-страницы, что упрощает создание интерфейса и улучшает пользовательский опыт.
* **Легкость интеграции**: FastAPI легко интегрируется с другими библиотеками Python, что упрощает взаимодействие с моделями WhisperX и Llama.

2. Модель WhisperX

* Высокая точность распознавания речи: Whisper демонстрирует выдающуюся точность при распознавании речи, что критически важно для создания качественных транскрипций. Модель обучена на разнообразных данных и способна эффективно различать слова и фразы в различных контекстах, что минимизирует ошибки и повышает надежность итогового текста.
* Поддержка множества языков: Whisper поддерживает несколько языков, включая русский, что делает её универсальным инструментом для международных команд и многоязычных совещаний.
* Скорость обработки: Whisper обладает высокой скоростью обработки аудиоданных, что позволяет быстро получать транскрипции. Это особенно важно для бизнес-среды, где время является критическим фактором.
* Гибкость и адаптивность: Whisper предоставляет возможность настройки под специфические требования пользователей. Модель можно адаптировать для работы с различными акцентами, стилями речи и условиями записи, что делает её подходящей для широкого спектра сценариев использования.
* Открытый код и возможность настройки: Whisper является проектом с открытым исходным кодом, что позволяет разработчикам модифицировать и настраивать модель в соответствии с их потребностями.
* Поддержка сообщества: Whisper имеет активное сообщество разработчиков и пользователей, что обеспечивает постоянное обновление и улучшение модели.
* Экономическая эффективность: Использование Whisper позволяет снизить затраты на разработку и внедрение системы транскрибации.

3. Модель Llama

Генерация текста: Модель Llama обеспечивает качественную генерацию текста, что позволяет создавать протоколы, соответствующие стандартам и требованиям пользователей. Она способна создавать связные и логически структурированные тексты, что улучшает качество итоговых документов.

Адаптивность: Возможность настройки шаблонов позволяет подстраивать продукт под конкретные нужды бизнеса и пользователей. Модель может адаптироваться к различным стилям и форматам, что делает её универсальным инструментом для различных сценариев использования.

Контекстуальное понимание: Llama обладает способностью учитывать контекст обсуждаемой темы, что позволяет ей генерировать более релевантный и целенаправленный текст. Это особенно важно для создания протоколов, которые должны отражать суть обсуждения и ключевые моменты встречи.

Поддержка многоязычности: Модель Llama способна работать с несколькими языками, что делает её подходящей для многоязычных команд и международных проектов.

Интеграция с другими системами: Llama легко интегрируется с другими инструментами и системами, что позволяет создавать комплексные решения для автоматизации документооборота.

* 1. Описание методологии разработки

Проект реализован с использованием **методологии Agile**, что позволит гибко реагировать на изменения и улучшения в процессе разработки. Agile подходит для нашего проекта, так как он требует постоянного взаимодействия с пользователями и быстрой адаптации к их потребностям. Основные этапы разработки включабт следующие ключевые элементы:

1. **Планирование спринтов**

На начальном этапе каждого спринта проводиться планирование, в ходе которого определяются задачи, которые необходимо выполнить в течение следующего цикла разработки. Задачи формируются на основе **backlog** — списка всех требуемых функций и улучшений. Это позволило команде сосредоточиться на наиболее важных аспектах проекта и обеспечить их реализацию в кратчайшие сроки.

Примеры поставленных задач:

* Реализация функционала загрузки аудиофайлов.
* Интеграция модели Whisper для распознавания речи.
* Разработка пользовательского интерфейса с использованием FastAPI и Jinja2.

2. **Разработка и тестирование**

В рамках Agile-методологии параллельно с разработкой проводиться тестирование. Это позволит быстро выявлять и исправлять ошибки, что существенно повысит качество продукта.

Примеры тестирования:

* Тесты на точность распознавания речи с использованием модели Whisper.
* Тесты на корректность генерации протоколов с помощью модели Llama.

3. **Регулярные встречи**

Команда проводила регулярные встречи, по пятницам и воскересеньям, на которых обсуждается прогресс работы и выявляются проблемы. Это обеспечит прозрачность процессов и позволит всем участникам команды быть в курсе текущего состояния проекта.

4. **Отчет о результатах тестирования на промежуточных этапах**

На каждом промежуточном этапе разработки проводиться тесты. Это поможет команде отслеживать прогресс и выявлять области для улучшения.

Примеры результатов тестирования:

* **Тесты на точность распознавания речи**: Проведены тесты с использованием различных аудиофайлов, и получены положительные результаты, подтверждающие высокую точность модели Whisper.

Пример результатов промежуточного тестирования модели (изображения-доказательства есть в приложении):

* Время если не отчищать Сборщиком муссора(gc) память видеокарты:

Время выполнения с диаризацией: 1048.8008 секунд

Время выполнения без диаризации: 196.6503 секунд

* 7-ми минутный без gc:

Время выполнения с диаризацией: 66.3311 секунд

Время выполнения без диаризации: 15.2027 секунд

* 7-ми минутный с gc:

Время выполнения с диаризацией: 42.9891 секунд

Время выполнения без диаризации: 15.6764 секунд

* 30-минутный без gc:

Время выполнения с диаризацией: 282.7992 секунд

Время выполнения без диаризации: 48.1264 секунд

* 30-минутный с gc:

Время выполнения с диаризацией: 136.0283 секунд

Время выполнения без диаризации: 48.0855 секунд

Вывод: 8gb не хватает на 3 модели. Для ускорения надо отчищать видеопамять.

* **Выявление ошибок в логике заполнения протокола**: в ходе тестирования были обнаружены ошибки в алгоритме заполнения протокола, которые были исправлены в следующих итерациях. Это подтверждает важность регулярного тестирования и обратной связи на каждом этапе разработки.
  1. Информация о планировании деятельности в ходе разработки и распределении задач между участниками команды разработчиков

Планирование в нашем проекте осуществляется на основе методологии **Scrum**, что позволяет эффективно управлять процессом разработки и обеспечивать высокую степень гибкости. В рамках этой методологии каждая задача разбивается на более мелкие подзадачи, что упрощает их выполнение и позволяет команде сосредоточиться на конкретных аспектах проекта.

**Процесс планирования**

Каждую неделю команда проводит **ревью**, на которых обсуждаются выполненные задачи, анализируются результаты, и корректируются планы на следующий спринт. Это позволяет не только отслеживать прогресс, но и адаптироваться к изменениям, возникающим в ходе разработки. На этих встречах участники команды делятся своими успехами, обсуждают возникшие проблемы и планируют дальнейшие шаги.

**Этапы планирования**:

1. **Определение задач**: На основе backlog формируются задачи для текущего спринта.
2. **Разделение задач на подзадачи**: Каждая задача разбивается на более мелкие подзадачи, что делает их более управляемыми.
3. **Распределение задач**: Подзадачи распределяются между участниками команды с учетом их компетенций и загруженности.

**Распределение задач**

Для оптимизации работы команды задачи распределяются следующим образом:

**Тимлид**: Занимался координацией, поставлением задач, отслеживанию процесса и другими задачами, которые помогают эффективно построить процесс разработки проекта.

отвечает за настройки модели и тестирование. Он занимается конфигурацией модели Whisper для распознавания речи, а также проводит тесты на точность и производительность, чтобы обеспечить высокое качество работы.

**Разработчик 1**: занимается настройкой WhisperX и LLaMA а также написанием серверной части проекта, чтобы иметь полное представление о структуре проекта.

**Разработчик 2**: занимается разработкой транскрибации в реальном времени и исследованиями данной темы, чтобы иметь большее фокусирование на исследовательской задаче нашего проекта.

**Также каждый участник команды помогает свои коллегам, чтобы иметь экспертизу со стороны, для улучшения эффективности процесса. Это помогает иметь основные задачи, а также включаться и помогать в разработке смежных задач проекта.**

ЗАКЛЮЧЕНИЕ

В процессе разработки программного продукта мы стремились обеспечить полное соответствие поставленным требованиям заказчика и пользователей. Основные аспекты, на которые мы ориентировались, включают функциональность, удобство использования и производительность системы.

**Оценка соответствия требованиям**

Программный продукт продемонстрировал высокую степень соответствия требованиям. Мы провели тщательный анализ выполненных требований, выявив, что ключевые функции, такие как распознавание речи и генерация протоколов, работают на удовлетворяющем уровне, соответствующем ожиданиям пользователей. Мы успешно реализовали сервис, который позволяет пользователям эффективно взаимодействовать и организовывать онлайн-встречи.

**Оценка качества программного продукта**

Качество программного продукта было оценено на основе результатов тестирования, которое охватывало функциональное, нагрузочное и пользовательское тестирование. В процессе тестирования были выявлены несколько дефектов и ошибок, которые, хотя и не критичны, все же влияют на общую работоспособность системы. Например, некоторые ошибки в логике заполнения протокола приводили к некорректному отображению данных.

**Предложения по улучшению и развитию продукта**

Для дальнейшего улучшения продукта и его успешного развития мы рекомендуем следующие шаги:

* **Расширение функциональности**: Включение дополнительных настроек для пользователей, таких как возможность изменения параметров распознавания, что позволит адаптировать продукт под конкретные нужды пользователей.
* **Улучшение пользовательского интерфейса**: Проведение пользовательских исследований для выявления узких мест в интерфейсе и его доработка на основе полученных данных. Это повысит удобство использования и удовлетворенность пользователей.
* **Регулярные обновления и поддержка**: Создание плана по регулярным обновлениям продукта, что позволит не только исправлять ошибки, но и добавлять новые функции, соответствующие изменяющимся требованиям рынка.
* **Обратная связь от пользователей**: Установление систематического сбора отзывов от пользователей для выявления их потребностей и проблем, с которыми они сталкиваются при работе с продуктом.
* Интеграция со сторонними сервисами является важным аспектом, который может значительно повысить функциональность нашего продукта и расширить его возможности. Взаимодействие с продуктами, смежными нашей теме, позволит не только улучшить пользовательский опыт, но и создать более комплексное решение для организации онлайн-встреч и звонков.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. **Habr.** Распознование речи, генерация субтитров и изучение языков при помощи whisper. [Электронный ресурс]. – URL: <https://habr.com/ru/companies/ods/articles/692246/> (дата обращения: 15.03.2025).
2. **Habr.** Выжимаем максимум из моделец Whisper на Apple Silicon. [Электронный ресурс]. – URL: <https://habr.com/ru/articles/884992/> (дата обращения: 15.03.2025).
3. **OpenAI Cookbook.** Whisper prompting guide. [Электронный ресурс]. – URL: <https://cookbook.openai.com/examples/whisper_prompting_guide> (дата обращения: 15.03.2025).
4. **GitHub.** Распознавание в реальном времени. [Электронный ресурс]. – URL: <https://github.com/davabase/whisper_real_time> (дата обращения: 15.03.2025).
5. **GitHub.** RealTimeWhisper. [Электронный ресурс]. – URL: <https://github.com/baileyarzate/RealTimeWhisper> (дата обращения: 27.05.2025).

**YouTube.** SUPER Fast AI Real Time Speech to Text Transcribtion - Faster Whisper / Python. [Электронный ресурс]. – URL: <https://youtu.be/k6nIxWGdrS4?si=K6OENi4BRt4Q02De> (дата обращения: 15.03.2025).

1. **Lablab.** OpenAI Whisper tutorial: how to create speaker identification app. [Электронный ресурс]. – URL: <https://lablab.ai/t/whisper-transcription-and-speaker-identification> (дата обращения: 15.03.2025).
2. **GitHub.** Whisper's transcription plus Pyannote's Diarization. [Электронный ресурс]. – URL: <https://github.com/openai/whisper/discussions/264> (дата обращения: 27.05.2025).
3. GitHub. Conversation transcriptor. [Электронный ресурс]. – URL: <https://github.com/cgroll/conversation-transcriptor> (дата обращения: 27.05.2025).
4. Изображения файлов разработки и тестирование

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 – тестирование WhisperX

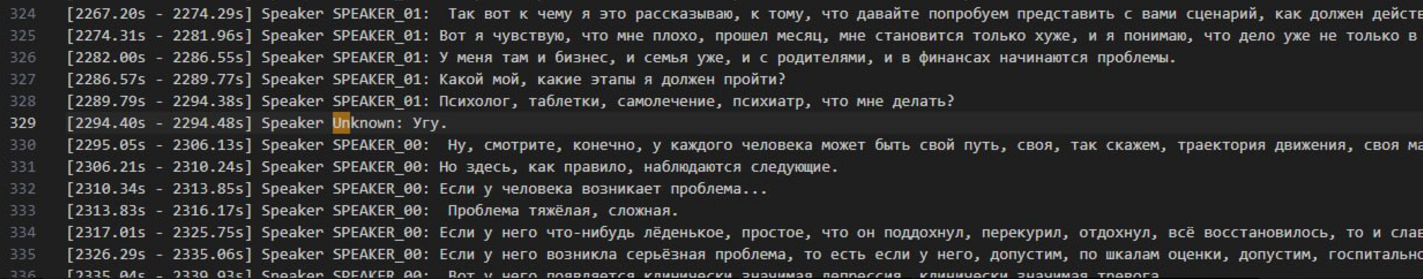


Рисунок 2 – тестирование дериализации

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 3 – тестирование whisperX на 7-ми минутном видео

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 – тестирование whisperX на 30-ми минутном видео

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Графическое программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 – тестирование whisper на 7-ми минутном видео

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 – тестирование whisper на 30-ми минутном видео