Ды МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Приложение для анализа деловых встреч «Встречеслав»

Курсовая работа

Направление: 09.03.04. Программная инженерия

Зав. Кафедрой _____ д. ф.-м. н, доцент С.Д. Махортов

Руководитель _____ ст. преподаватель В.С. Тарасов

Руководитель практики _____ А.В. Москаленко

Обучающийся _____ Н.Н. Агафонов, 3 курс, д/о

Обучающийся _____ Н.Н. Андреев, 3 курс, д/о

Обучающийся _____ С.И. Бондарева, 3 курс, д/о

Обучающийся _____ К.В. Воронецкий, 3 курс, д/о

Обучающийся _____ П.С. Парамонова, 3 курс, д/о

Обучающийся _____ В.С. Попов, 3 курс, д/о

Воронеж 2025

Содержание

Введение

Автоматизация интеллектуального труда становится одной из ключевых тенденций цифровой трансформации. Если раньше технологии применялись преимущественно для автоматизирования рутинных или технических операций, то сегодня всё более активно развиваются решения, способные оказывать поддержку в сферах, связанных с обработкой знаний, коммуникацией и принятием решений. Среди таких решений – интеллектуальные помощники и приложения, способные вести запись, анализ и структурирование информации, полученной в процессе делового общения. Особенно востребованными становятся системы, которые помогают обрабатывать встречи, совещания, переговоры, лекции: от стенографирования сказанного – до генерации итоговых протоколов, выделения ключевых тем и формулирования задач[1].

Внедрение приложений, которые конвертируют аудио и видео в текст, позволяет компаниям и учреждениям повысить прозрачность процессов, сократить человеческие ошибки при оформлении протоколов и снизить нагрузку на сотрудников. Кроме того, автоматизация подобных процедур особенно важна в условиях высокой интенсивности коммуникаций, когда одновременно ведутся десятки встреч и совещаний, а ручная обработка информации становится узким местом в организации деятельности[2]. Такие инструменты находят применение в самых разных отраслях – от ИТ и финансов до государственного управления и образования[3].

Большая часть современных решений в этой области представлена в виде облачных сервисов. Это означает, что для работы таких систем требуется постоянное подключение к интернету, а данные пользователей – включая голос, текст и метаданные – передаются на удалённые серверы. Такой подход может быть приемлем в условиях открытых корпоративных сред, но он представляет серьёзную проблему, когда речь идёт о конфиденциальной информации, внутренней документации или стратегических обсуждениях. Для органи-

заций, работающих с чувствительными данными, требования к информационной безопасности ставятся во главу угла. Нарушение конфиденциальности, утечка информации или даже сам факт её передачи за пределы локальной инфраструктуры может иметь серьёзные последствия[4].

Кроме того, в текущей геополитической и экономической ситуации важную роль играет фактор импортозамещения. Многие зарубежные сервисы стали недоступны или ограничены для использования в России, а зависимость от внешней инфраструктуры может повлечь за собой риски недоступности или отключения ключевых функций в критические моменты. Поэтому всё большую актуальность приобретают решения, разработанные с учётом требований локального развёртывания, независимости от зарубежных поставщиков и возможности полной автономной работы[5].

В рамках данной курсовой работы рассматривается разработка приложения для настольных компьютеров, предназначенного для автоматизированной обработки деловых встреч. Программа позволяет загружать или записывать аудиофайлы, производить расшифровку сказанного, анализировать содержание и формировать удобные структурированные протоколы с разбивкой по темам, выводами и списками задач. Принципиальным отличием данной разработки от многих аналогов является её полностью локальный характер. Все процессы – от распознавания речи до генерации текста – выполняются на компьютере пользователя, без отправки данных в облако и без необходимости подключения к интернету.

Такой подход позволяет использовать приложение в защищённых средах, внутри корпоративных или ведомственных сетей, где подключение к внешним сервисам ограничено или запрещено. Это делает разработку особенно ценной для компаний, предъявляющих высокие требования к защите информации, а также для государственных, научных и образовательных учреждений.

Таким образом, создаваемое приложение отвечает сразу нескольким актуальным вызовам: повышению эффективности делового взаимодействия, обеспечению информационной безопасности и выполнению задач импортозамещения. Далее в работе будут рассмотрены: архитектура системы, её ключевые компоненты, этапы разработки.

Отдельное внимание в процессе разработки было уделено вопросам конфиденциальности и безопасности пользовательских данных. В условиях усиливающихся требований к защите информации, а также с учётом ограничений на использование облачных зарубежных сервисов, проект реализован в виде десктопного приложения, способного функционировать полностью в офлайн-режиме. Все компоненты, включая модули с нейронными сетями для распознавания и анализа речи, работают локально, без необходимости отправки данных во внешние облака или на сторонние серверы. Это не только повышает уровень защиты информации, но и делает систему независимой от внешней инфраструктуры.

1 Постановка задачи

1.1 Цель создания системы

Целью курсового проекта является создание настольного приложения для автоматизации обработки деловых встреч путем автоматического формирования стенограмм по загруженным в приложение видео, структурирования обсуждаемой информации и формирования итогового протокола с целью снижения временных затрат на выполнение организационных задач и обеспечения полной локальности обработки данных без передачи информации во внешние сети.

1.2 Задачи приложения

Приложение решает следующие задачи:

— автоматизация анализа и стенографирования встреч за счет создания системой стенограммы и протокола встречи;

— обеспечение конфиденциальности обработки данных и предоставление пользователю полного контроля над доступностью приложения.

1.3 Требования к разрабатываемой системе

1.3.1 Функциональные требования

К разрабатываемой системе выдвигаются следующие функциональные требования:

– приложение должно обеспечивать фоновую обработку в режиме офлайн видеофайлов, текста, который создан в результате работы системы и находится в рамках её базы данных;

– приложение должно предоставлять пользователю возможность сохранять и редактировать результаты работы модулей стенографирования, протоколирования и редактирования текста и получать к ним доступ в рамках системы;

– приложение должно предоставлять возможность создавать протокол встречи, в котором будет содержаться важные положения встречи.

1.3.2 Требования к лингвистическому обеспечению системы

Графический пользовательский интерфейс приложения должен быть на русском языке, а обработка загруженных записей встреч должна предусматривать работу с видео на русском языке, с учетом наличия профессиональной лексики и англицизмов в речи части пользователей.

1.3.3 Требования к эргономике и технической эстетике приложения

Дизайн приложения должен быть выполнен в едином стиле. Система должна уведомлять пользователя об окончании процесса обработки встречи, исключая необходимость постоянного контроля процесса.

1.4 Задачи, решаемые в процессе разработки

Разработка системы включала следующие задачи:

— анализ предметной области;

— проектирование системы, составление UML диаграмм;

— написание технического задания;

— создание дизайна приложения;

— разработка приложения в соответствии с техническим заданием.

2 Анализ предметной области

2.1 Терминология (глоссарий) предметной области

В настоящей курсовой работе применяются следующие термины с соответствующими определениями.

Таблица 1 - Термины, используемые в тексте курсовой работы

Термин	Значение
Архитектура	Организация компонентов приложения и определение их взаимодействий для эффективного выполнения задач.
Большая языковая модель (LLM, Large Language Model)	Класс искусственных интеллектуальных систем, предназначенных для обработки, генерации и анализа текстовой информации на естественном языке.
Искусственная интеллектуальная система (ИИ, ИИ-система)	Программно-аппаратный комплекс, способный выполнять задачи, такие как анализ данных, обработка естественного языка, прогнозирование и принятие решений. Эти системы основаны на методах машинного обучения, нейронных сетях и алгоритмах обработки информации, что позволяет им адаптироваться к новым условиям и обучаться на основе накопленного опыта.

Термин	Значение
Протокол встречи	Документ, содержащий краткую сводку обсужденных на встрече проблем и задач, принятые по каждому из проблем решение и общий вывод по итогам встречи.
Стенограмма	Текстовый документ, содержащий дословную запись устной речи, полученную путем расшифровки аудио- или видеозаписи выступления, беседы, совещания.
Фреймворк	Структура, предоставляющая базовый функционал для разработки программного обеспечения. Он представляет собой набор библиотек, инструментов и рекомендаций, которые помогают разработчикам создавать приложения, не начиная с нуля.
Эмбеддинг	Способ преобразования чего-то абстрактного, например, слов в набор чисел и векторов.
Непрерывная интеграция(Continuous Integrantion, CI)	Методология разработки, которая подразумевает сборку и тестирование приложения без участия человека направлена на выявление дефектов на наиболее ранних этапах разработки.

2.2 Актуальность

Интеллектуальный помощник для деловых встреч – это современное решение для фиксации и обработки информации, получаемой в ходе переговоров и рабочих встреч. В условиях высокой динамики бизнеса и растущей цифровизации процессов, потребность в удобных, надёжных и безопасных инструментах для документирования деловой коммуникации возрастает с каждым годом.

Одним из ключевых преимуществ данного приложения является автоматическое создание стенограммы и протокола встречи. Это позволяет избежать ручного документирования, значительно экономя время участников и снижая риск потери важной информации. В отличие от облачных сервисов, вся обработка данных происходит локально на устройстве пользователя, что обеспечивает полный контроль над конфиденциальностью и безопасностью информации.

Дополнительное удобство заключается в простоте использования: приложение избавляет пользователя от необходимости делать записи вручную или самостоятельно формировать стенограммы и протоколы. Это делает его особенно полезным для тех, кто работает с большим объёмом устной информации – от руководителей и специалистов в области права до студентов и представителей государственных структур.

Надёжность, безопасность и автоматизация ключевых процессов делают интеллектуального помощника востребованным инструментом в современном деловом и образовательном контексте. Его использование открывает новые возможности для повышения продуктивности и качества коммуникации без необходимости вовлечения сторонних сервисов или дополнительного оборудования.

2.3 Анализ существующих решений

Анализ существующих решений представляет собой важный этап разработки разрабатываемого программного продукта. Проведение такого анализа позволяет определить ключевые потребности целевой аудитории, выявить устоявшиеся практики и стандарты в реализации основных функций, а также зафиксировать потенциальные проблемные зоны, которые остаются нерешёнными в текущих аналогах.

Результаты анализа способствуют более точной формулировке требований к функциональности и пользовательскому опыту, а также позволяют обоснованно приоритизировать направления дальнейшего развития приложения. Кроме того, изучение конкурентных решений позволяет выявить возможности для повышения конкурентоспособности за счёт устранения недостатков, имеющихся у существующих продуктов.

В рамках исследования было рассмотрено несколько решений, представляющих как прямую, так и косвенную конкуренцию. На основе полученных данных была составлена сводная таблица, позволяющая провести сравнительный анализ ключевых характеристик и выделить направления, по которым разрабатываемый продукт может предложить пользователю дополнительные преимущества. В таблице 2 приведены результаты анализа конкурентов.

Таблица 2 - Анализ сервисов приложений-конкурентов

Функция/Приложение(сервис)-аналог	TL;DV	Read.ai	IVA One	WizWhsip
Составление стенограммы и протокола	да	да	да	только стенограммы
Интеграция с таск-трекерами	да	да	потенциально да	нет

Функция/Приложение(сервис)-аналог	TL;DV	Read.ai	IVA One	WizWhsip
Анализ загруженных пользователем файлов	да	нет	нет	да
Локальное хранение данных	нет	нет	да	да
Возможность локально обработать видео	нет	нет	нет	да

2.3.1 TL;DV

TL;DV – это сервис, основанный на искусственном интеллекте, который помогает записывать, транскрибировать, анализировать совещания, онлайн-встречи общение с клиентами.

Зарегистрированный пользователь может не только получить стенограмму по тому материалу, что был записан непосредственно с помощью самого сервиса, но и по тому файлу, который он сам загрузит в систему. Приложение предоставляет набор инструментов для анализа видеофайлов, предоставляемых пользователем.

Ниже на рисунке 1 приведён экран, на котором приведена стенограмма встречи, созданная с помощью TL;DV.

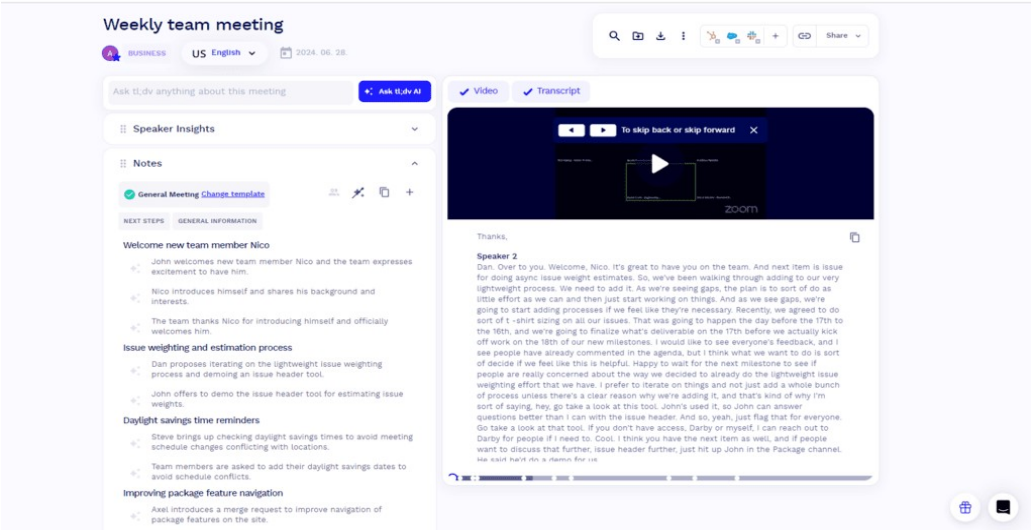

Рисунок 1 – Экран, со стенограммой файла, загруженного пользователем в систему, который был записан с помощью сервисов, не являющихся частью TL;DV

Помимо возможности составления стенограммы, сервис предоставляет пользователю возможность автоматического составления заметок, в которых описываются ключевые положения прошедшей онлайн-встречи.

Пример того, как могут выглядеть заметки по встрече, созданной с помощью сервиса, приведен на рисунке 3.

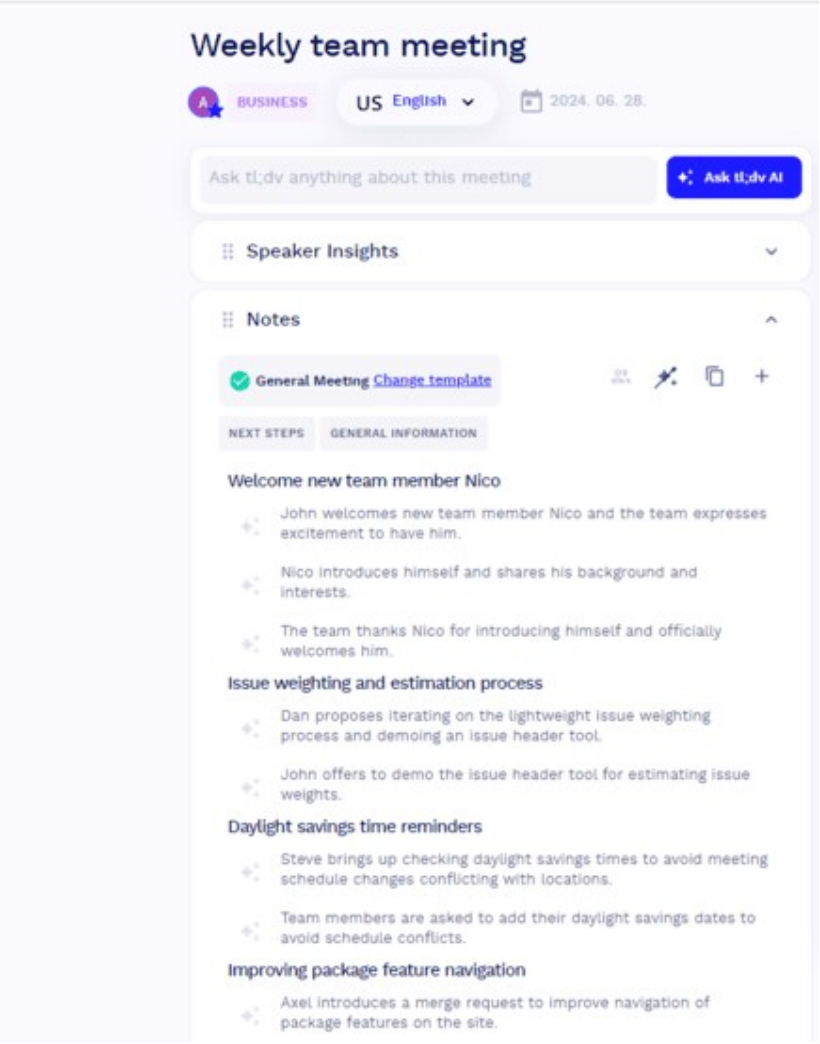

Рисунок 2 – Экран с заметками по онлайн-встрече

Преимущества TL;DV:

— интеграция с такими сервисами, как Google Meet, Zoom, Notion и Slack, что делает его удобным для использования в рамках существующих рабочих процессов;

— возможность использовать шаблоны и автоматически формировать краткие сводки по встречам с помощью встроенного искусственного интеллекта;

— возможность анализировать встречи за определённый период – пользователь можно искать и просматривать записки по предыдущим встречам, фильтровать их по дате и теме.

Недостатки TL;DV:

— требование платной подписки для доступа к ключевым функциям, в том числе к таким, как: неограниченные AI-конспекты встреч, глобальный поиск по стенограммам;

— поддержка только Zoom и Google Meet из платформ для видеоконференций, что ограничивает возможность интеграции в рабочие процессы при использовании других платформ для видеоконференций.

2.3.2 Read.ai

Read.ai – сервис, основанный на искусственном интеллекте, который помогает записывать, транскрибировать, анализировать совещания, онлайн-встречи общение с клиентами.

На рисунке 3 приведен экран со стенограммой, составленной по записи онлайн-встречи.

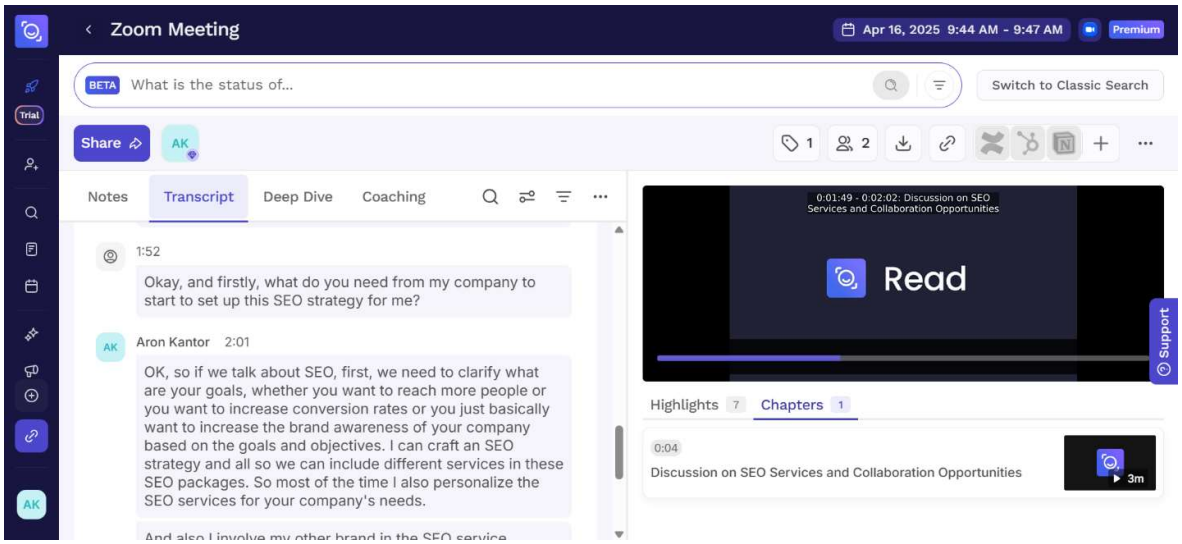

Рисунок 3 – Экран со стенограммой встречи

На рисунке 4 приведен экран с кратким пересказом встречи.

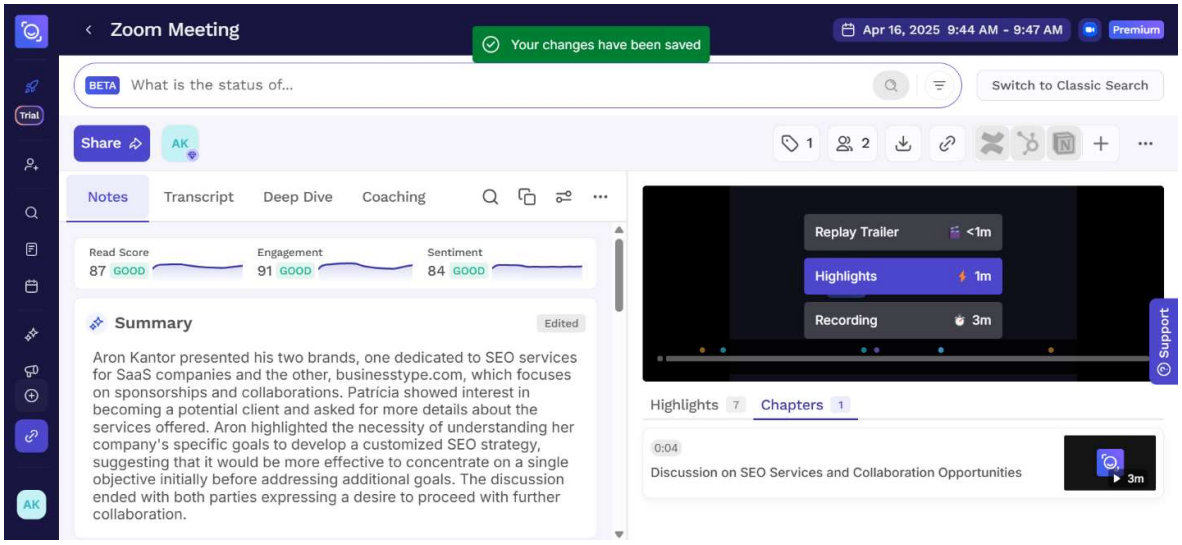

Рисунок 4 – Экран с краткой с пересказом встречи и моментами из неё

Преимущества Read.ai:

— предоставление возможности оптимизировать расписание встреч, учитывая предпочтения и занятость участников.

— анализ вербальных и невербальных сигналов (например, тон речи, мимику) для оценки эмоционального фона и уровня участия каждого участника, что позволяет объективно оценить эффективность встречи и выявить области для улучшения;

— предоставление рекомендаций по улучшению коммуникативных навыков на основе анализа поведения пользователя во время встреч. Это включает советы по повышению вовлечённости, управлению временем выступления и улучшению взаимодействия с аудиторией;

— интеграции с популярными платформами, такими как Zoom, Microsoft Teams, Google Meet, Slack, Notion и другими, что упрощает внедрение в существующие рабочие процессы.

Недостатки Read.ai:

— отсутствие опции для полного локального хранения записей и транскриптов встреч;

— после предоставления доступа к календарю, Read.ai может автоматически присоединяться ко всем встречам, включая те, где его присутствие нежелательно, что может нарушить конфиденциальность, особенно при обсуждении чувствительных тем.

2.3.3 IVA One

IVA One – это единая платформа для корпоративных коммуникаций и совместной работы, разрабатываемая компанией IVA Technologies. Модуль IVA GPT, являющийся частью платформы, реализует создание стенограмм и протоколов встреч.

На рисунке 5 приведен экран с видеоконференцией, проводимой в рамках платформы с включенным стенографированием в реальном времени

Рисунок 5 – Стенографирование встречи в реальном времени в IVA One

Преимущества IVA One:

— платформа интегрирует мессенджер, видеоконференции, календарь, электронную почту и другие инструменты, обеспечивая удобство и эффективность работы;

— поддержка on-premise-развертывания позволяет компаниям контролировать свои данные и обеспечивать высокий уровень информационной безопасности[6].

Недостатки IVA One:

— необходимость развертывания на отдельном сервере с достаточно высокими системными требованиями (8 виртуальных ядер, 16 ГБ ОЗУ, от 1 ТБ дискового пространства[7].

2.4 WizWisp

WizWisp – приложение для стенографирования видео и аудио файлов, которое может работать полностью локально.

На рисунке 6 приведен экран со стенограммой, которая была создана с помощью приложения

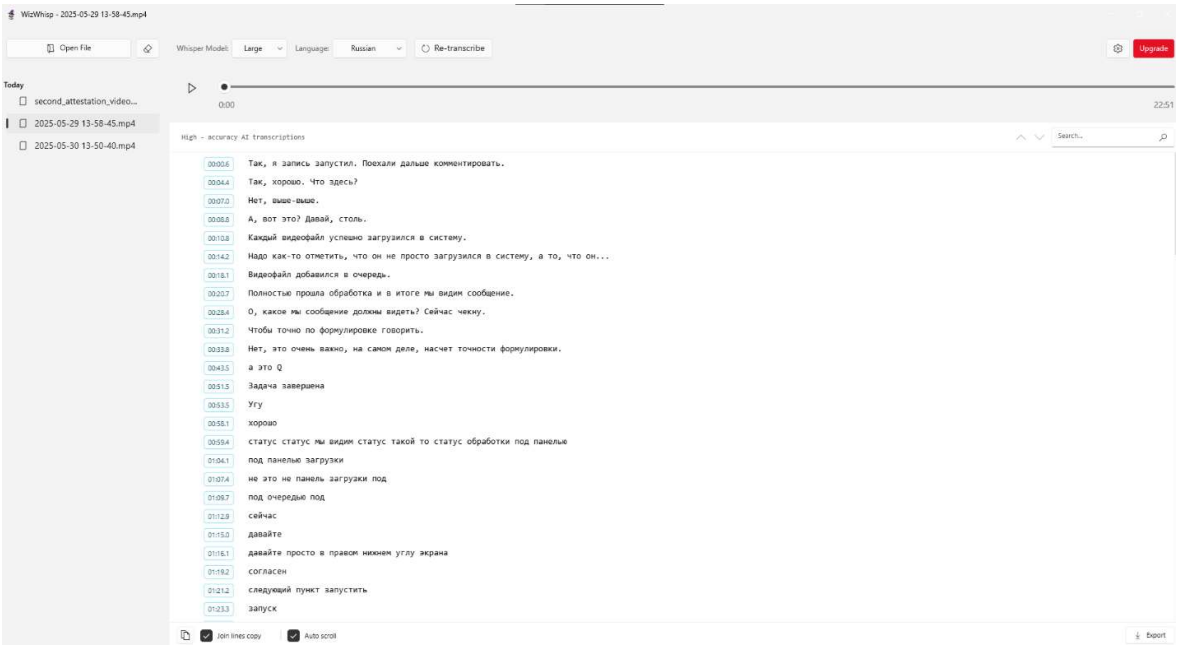

Рисунок 6 – Стенограмма, созданная с помощью WizWisp

Преимущества WispWiz:

— возможность локальной обработки видео и аудио для составления стенограммы;

— поддержка составления стенограмм для 90 языков, в том числе русского;

— относительно малые системные требования[8].

Недостатки WispWiz:

— отсутствие возможности составления протокола по составленной стенограмме;

— наличие интерфейса только на английском языке.

2.4.1 Выводы из анализа существующих решений

В результате анализа существующих решений были выявлены следующие особенности, которые необходимо учесть при разработке приложения:

—обеспечение локальной обработки данных без необходимости подключения к серверной инфраструктуре;

—ориентация на индивидуальное использование, без необходимости какого-либо взаимодействия с внешними сервисами;

—повышенное внимание к конфиденциальности, исключающее передачу пользовательских данных сторонним серверам;

—автономность работы приложения и независимость от состояния облачных платформ.

На основании анализа конкурентов было принято реализовать настольное приложение, которое может работать в фоне с полной локализацией обработки данных. Это позволит устранить зависимость от облачной инфраструктуры, повысить уровень защиты информации, упростить установку.

У двух рассмотренных решений основным недостатком является невозможность организации локальной обработки данных.

Решение IVA One может быть неудобно для личного использования – оно требует настройки и поддержки инфраструктуры. В отличие от таких решений, разрабатываемое настольное приложение не нуждается в сервере и работает полностью локально, обеспечивая простоту и конфиденциальность. Например, в случае IVA One организация локальной обработки затруднена форматом самого приложения: оно должно работать на выделенном сервере.

Приложение WispWiz может составлять стенограммы, но при этом в нем отсутствуют какие-либо инструменты для анализа составленных стенограмм: приложение не предоставляет пользователю, например, составить протокол встречи, или проанализировать вовлеченность участников. Так же его графический интерфейс не адаптирован под русскоязычную аудиторию.

Чтобы устранить ключевые недостатки существующих аналогов – необходимость в серверной инфраструктуре и невозможность локальной обработки данных – было принято решение разработать настольное приложение. Такой подход позволяет обеспечить индивидуальное использование, повысить конфиденциальность и улучшить доступность. Благодаря полной локализации все данные обрабатываются на машине пользователя, исключая передачу информации сторонним сервисам. Кроме того, доступность возрастает за счёт того, что пользователь сам контролирует состояние своей системы: в случае неполадок он может самостоятельно восстановить работоспособность приложения, в отличие от ситуаций с облачной инфраструктурой.

2.5 Целевая аудитория

Приложение ориентировано на следующие группы пользователей:

— сотрудники организаций, работающих с конфиденциальной информацией и заинтересованные в безопасной локальной обработке данных;

— люди, которые регулярно участвуют в онлайн-встречах, формируют решения и ответственны за подготовку документов или отчетов, основанных на результатах этих обсуждений;

— люди, работающие с большим количеством видеоматериала, включая специалистов, студентов, слушателей онлайн-курсов и всех, кому необходимо систематизировать и анализировать содержание видеозаписей – например, лекций, обучающих видео или личных заметок.

3 Реализация

3.1 Структура приложения

Система имеет монолитную архитектуру, в которой можно выделить следующие логические блоки: выделения аудио из видео, стенографирования, редактирования текста, протоколирования, хранения информации, переключения моделей. Так же у приложения реализован графический интерфейс. Решение о том, то система должна иметь монолитную архитектуру было принято потому как в рамках данной системы использование микросервисов было бы избыточным. Разделение компонентов на отдельные сервисы потребовало бы дополнительных ресурсов для организации их взаимодействия, что усложнило бы разработку и сопровождение. Кроме того, приложение не генерирует сетевого трафика, который следовало бы распределять между отдельными сервисами. Поскольку приложение работает полностью локально и не использует сервер, отсутствует необходимость в масштабировании и балансировке нагрузки, что делает монолитную архитектуру наиболее рациональным выбором. Модули в приложении являются чисто логическими сущностями, не формирующих отдельные сервисы.

Для связи графического интерфейса и бизнес-логики в системе используется реализация паттерна Наблюдатель. Данный паттерн был выбран, поскольку он обеспечивает слабую связанность между компонентами и позволяет автоматически синхронизировать состояние пользовательского интерфейса с изменениями во внутренней логике приложения. При использовании паттерна Наблюдатель объекты логики (наблюдаемые) уведомляют подписанные на них компоненты интерфейса (наблюдателей) об изменениях состояния, благодаря чему интерфейс всегда остаётся актуальным без необходимости жёсткого связывания с бизнес-логикой. Такой подход упрощает отладку, повышает масштабируемость системы и облегчает поддержку в долгосрочной перспективе.

Дополнительно для организации взаимодействия между логикой и представлением используются специальные классы-контроллеры. Это решение способствует повышению читаемости кода и архитектурной прозрачности, поскольку разметка и логика разделены по ответственности: визуальная часть хранится в отдельных файлах, а обработка пользовательских действий и бизнес-правила сосредоточены в контроллерах.

В проекте существует можно выделить три блока: это директории, вложенные в директорию dynamic-resources, директории, которые вложены в директорию main и директории, вложенные в директорию test. В main находится реализация бизнес-логики и графического интерфейса приложения. В test находятся тесты, проверяющие корректность работы логики приложения.

Приведем более подробное описание некоторых директорий:

— logic – в этой директории расположены логические модули приложения, в них содержится реализация логики модулей приложения, а так же реализация работы с БД;

— ui – в этой директории расположены все компоненты, связанные с графическим интерфейсом приложения;

— dynamic-resources – в ней расположены модели нейронных сетей, которые используются в приложении, а также там расположена БД, в которую заносится информация о созданных протоколах и стенограммах;

На рисунке 7 приведена структура директории logic.

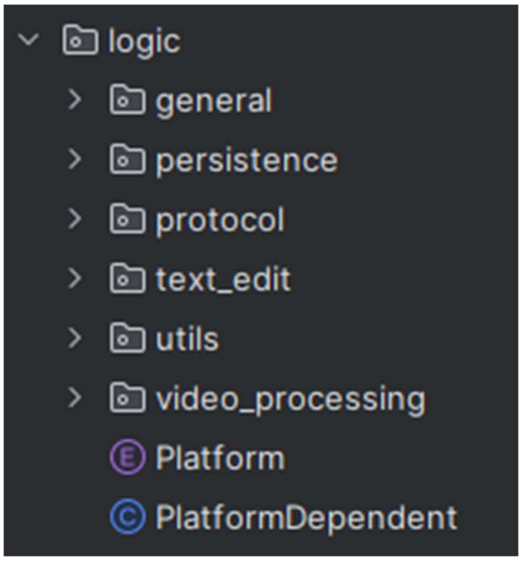

Рисунок 7 – Структура директории logic

Приведем более детально описание того, что расположено в каждой директории в ней:

— utils – содержит вспомогательные утилиты для работы с данными: форматирование времени, сериализация изображений, экспорт сущностей, загрузка лекций и работа с JSON. Это набор инструментов для обработки различных типов данных;

— persistence - отвечает за работу с базой данных: содержит менеджер БД, инициализатор и DAO (Data Access Objects). Реализует слой доступа к данным и управление соединениями с базой данных;

— general - содержит основные бизнес-сущности приложения: протоколы, транскрипты, задачи, теги, участник встреч и материалы встреч. Это ядро бизнес-логики, представляющее основные объекты предметной области;

— video_processing - реализует логику обработки видео и аудио: извлечение аудио, распознавание речи (Vosk), управление очередями обработки. Отвечает за преобразование медиа-контента в текстовый формат;

— protocol - содержит логику работы с языковыми моделями (LLM): сервис для взаимодействия с LLM и обертку для работы с ними. Отвечает за интеллектуальную обработку текста;

— text_edit - реализует функциональность редактирования текста: поиск, буферизация реплик, история изменений. Отвечает за манипуляции с текстовым контентом после его получения из видео/аудио.

3.2 Средства реализации

В качестве средств реализации были использованы следующие технологии:

— язык программирования Java;

— фреймворк для создания графических интерфейсов JavaFX;

— набор библиотек для обработки аудио и видеофайлов FFmpeg;

— инструмент для автоматизации сборки проектов Gradle – инструмент для автоматизации сборки проектов;

— фреймворк для написания тестов JUnit;

— модели vosk-model-ru-0.42 и vosk-model-spk-0.4 из библиотеки для распознавания речи Vosk;

— инструмент для реализации CI GitHub Actions

— модель Triangle104/gemma-3-4b-it-Q6_K-GGUFct из семейства LLM моделей Gemma;

— СУБД SQLite.

Преимуществами языка программирования Java являются его платформенная независимость, что помимо облегчения процесса разработки, так же сокращает время и ресурсы, необходимые для адаптации к различным платформам.

Фреймворк JavaFX был выбран потому, что с его помощью очень удобно интегрировать в графический интерфейс бизнес-логику, написанную на Java:

он позволяет разделить структуру, стили и логику за счёт использования FXML для описания интерфейса, CSS для стилизации и Java для бизнес-логики. Такой подход упрощает сопровождение кода и ускоряет разработку: готовые UI-компоненты позволяют быстро создавать интерфейсы без необходимости в сторонних инструментах.

Библиотеки для извлечения аудио из видео из набора FFmpeg были выбраны потому, что они не зависят от платформы, и потому, что они реализуют работу со всеми видами видеофайлов, которые можно загрузить в приложение. При этом, для обработки файлов нет необходимости в дополнительной конверсии или установке сторонних кодеков. Кроме того, FFmpeg отличается производительностью и активно поддерживается сообществом, что делает его надёжным выбором для интеграции в настольное приложение.

В качестве инструмента автоматизации сборки был выбран Gradle благодаря его гибкости и практическому удобству применения. В отличие от Maven, он требует меньше конфигурационного кода, что ускоряет процесс настройки проекта и упрощает поддержку. Одним из ключевых факторов при выборе Gradle стало его взаимодействие с утилитой jpackage, которая используется для формирования установочных пакетов. Благодаря этому связка Gradle и jpackage позволяет собирать установщики для разных операционных систем с уже включённой JRE и всеми зависимостями. Это особенно важно для настольных приложений, поскольку обеспечивает удобное и понятное распространение продукта для пользователей**.**

В качестве основного инструмента модульного тестирования в приложении выбран фреймворк JUnit. Такой выбор бы сделан по тому, что JUnit поддерживается большинством популярных интегрированных сред разработки (в том числе IntelliJ IDEA), системами сборки (в том числе Gradle), а также средствами автоматизации и непрерывной интеграции (в том числе и с GitHub Actions)[9]. Это существенно упрощает интеграцию тестов в процесс сборки и позволяет автоматизировать их выполнение на всех этапах жизненного цикла

приложения. Также, фреймворк предоставляет простой синтаксис для написания модульных тестов, включая аннотации, которые позволяют гибко управлять тестовой средой.

Для реализации распознавания речи в приложении были выбраны модели из библиотеки Vosk, поскольку они открыты и легковесны. Распространение моделей под лицензией Apache 2.0 исключает юридические ограничения на коммерческое использование, что особенно важно для проектов с открытым исходным кодом и ограниченным бюджетом. В отличие от многих аналогов, Vosk предоставляет компактные модели (например, используемая в приложении модель для разделения реплик по участникам модель весит 15 мегабайт), что снижает требования к оперативной памяти и дисковому пространству, упрощая интеграцию в локальные приложения. Ещё одним преимуществом является возможность полностью автономной работы без подключения к интернету – модели разворачиваются локально, и обращение к внешним API не требуется, что особенно важно для обеспечения конфиденциальности обработки пользовательских данных. Помимо этого, Vosk поддерживает широкий набор языков распознавания и предоставляет обвязки для нескольких языков программирования, включая Java[10], что значительно облегчает встраивание моделей в приложение. Также система позволяет адаптировать словарь под предметную область и поддерживает работу с потоковым аудио[11], что повышает точность и универсальность решения.

В качестве инструмента для CI в проекте выбран GitHub Actions. Основными причинами такого выбора стали высокая гибкость настройки, широкие возможности расширения и глубокая интеграция с экосистемой GitHub. GitHub Actions позволяет описывать рабочие процессы в виде декларативных YAML-файлов, в которых можно точно задать последовательность шагов, условия их выполнения и зависимости между ними. Это обеспечивает полный контроль над процессами сборки, тестирования, анализа кода и развёртывания. Кроме того, GitHub Actions тесно интегрирован с GitHub: статусы сборок,

логи, артефакты и другие результаты CI-процессов доступны прямо в интерфейсе pull request'ов, коммитов и релизов. Это делает разработку более прозрачной и облегчает командную работу. Таким образом, GitHub Actions представляет собой мощный, гибкий и интегрированный инструмент, который обеспечивает автоматизацию CI-процессов в проектах, размещённых на GitHub.

Модель Triangle104/gemma-3-4b-it-Q6_K-GGUFct из семейства Gemma была выбрана в результате практической оценки нескольких LLM, удовлетворяющих требованию ограниченного потребления оперативной памяти – не более 5 ГБ. В ходе тестирования рассматривались модели с различными уровнями сжатия и архитектурами. По результатам испытаний именно данная модель показала наилучший баланс между качеством генерации протоколов встреч и стабильностью работы: она демонстрировала минимальное количество искажений и высокую точность в передаче сути обсуждений. Формат GGUF упрощает интеграцию модели как зависимости в проект, поскольку представляет собой единый файл, который легко добавить и описать в системе сборки, например, в Gradle. Это повышает удобство использования модели в приложении и упрощает процесс её внедрения и обновления. Это делает её наиболее подходящим вариантом для локального использования в приложении, ориентированном на преобразование стенограмм в протоколы встреч.

SQLite был выбран по нескольким причинам, которые хорошо подходят для настольного приложения, работающего полностью локально. Во-первых, эта база данных легковесная и не требует отдельного серверного процесса, как, например, PostgreSQL. Это упрощает установку и использование приложения, так как всё работает внутри одной программы без дополнительной настройки. Во-вторых, несмотря на простоту, SQLite поддерживает язык SQL и позволяет выполнять сложные запросы – например, с объединением таблиц с помощью JOIN. Это даёт возможность работать с данными на полноценном уровне, сохраняя структуру и логику, привычную для реляционных баз. Таким образом,

SQLite стал удобным и надёжным решением, которое сочетает простоту локального использования с возможностями полноценной СУБД.

3.3 Диаграмма последовательности для получения стенограммы

На рисунке 8 изображена диаграмма последовательности для создания стенограммы.

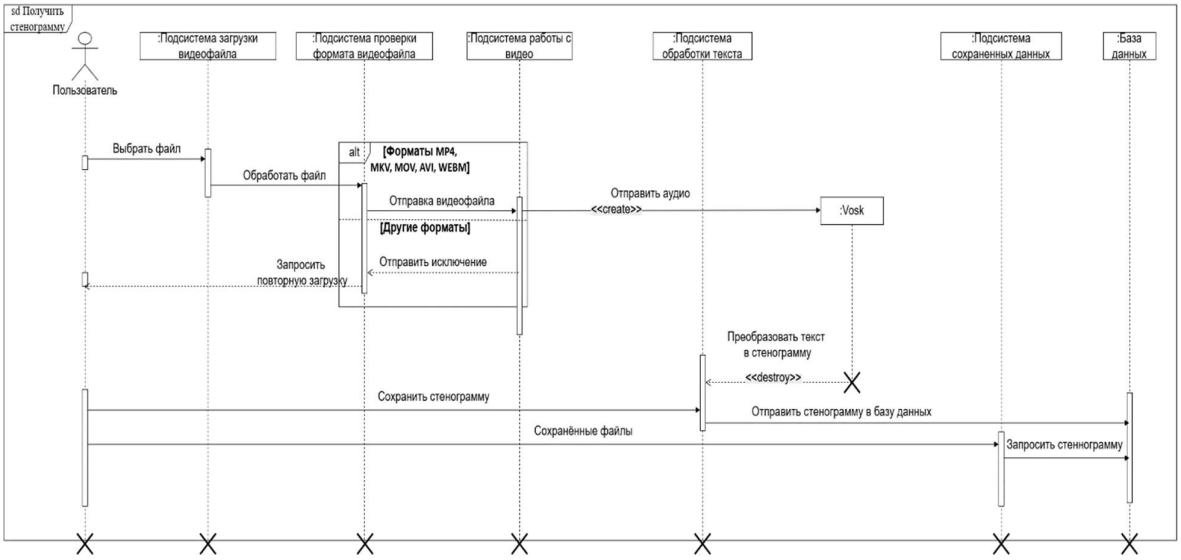

Рисунок 8 – Диаграмма последовательностей для получения стенограммы

Приведем детальное описание диаграммы. Пользователь инициирует процесс, выбирая видеофайл, который затем обрабатывается подсистемой загрузки. После этого файл передается в подсистему проверки формата, где происходит определение его соответствия поддерживаемым типам (MP4, MKV, MOV, AVI, WEBM). Если формат допустим, файл отправляется дальше в подсистему работы с видео. Если формат не поддерживается, формируется исключение, и пользователю отправляется запрос на повторную загрузку.

В случае успешной проверки подсистема работы с видео извлекает аудиодорожку из файла и передает ее в подсистему обработки текста. Далее аудио направляется в компонент с меткой :Vosk, который представляет собой модели из одноименной библиотеки для распознавания речи. На диаграмме :Vosk расположен таким образом, чтобы отразить то, что он не является постоянно активным участником системы, а создается в момент обращения, поскольку

сами модели библиотеки подгружаются в оперативную память только по мере необходимости.

После получения аудио :Vosk преобразует его в текстовую стенограмму и отправляет результат обратно в подсистему обработки текста, где она отображается пользователю. Затем стенограмма сохраняется в подсистеме хранения данных, откуда передается в базу данных для долговременного хранения. Таким образом, диаграмма последовательно отображает весь поток сообщений между участниками системы, необходимый для получения стенограммы из загруженного видео.

3.4 Физическая схема базы данных

На рисунке 9 представлена физическая схема базы данных приложения.

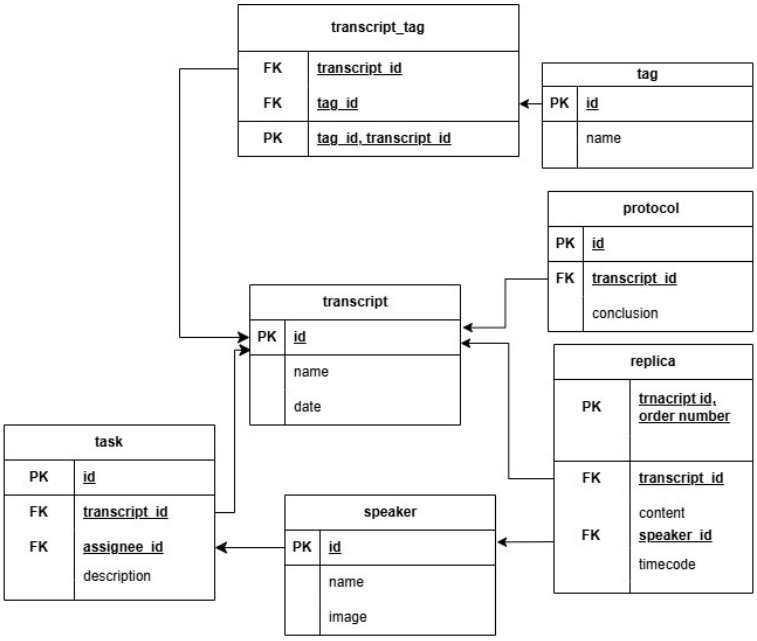

Рисунок 9 – Физическая схема базы данных в приложении

Приведём описание имеющихся таблиц/

Таблица speaker предназначена для хранения информации об участниках обсуждений. Каждая запись содержит уникальный идентификатор, имя участника и его изображение.

Таблица tag представляет собой справочник тегов, используемых для классификации стенограмм. Она содержит уникальные идентификаторы и названия тегов.

Таблица transcript содержит информацию о стенограммах. В неё входят уникальный идентификатор, название стенограммы и дата проведения встречи.

Таблица replica используется для хранения отдельных реплик в рамках стенограммы. Каждая реплика связана с определённой стенограммой и участником встречи, имеет порядковый номер, текст и временную метку.

Таблица protocol фиксирует итоговое заключение по конкретной стенограмме. Она содержит текст вывода и однозначно ссылается на соответствующую стенограмму.

Таблица task предназначена для хранения задач, сформулированных по результатам обсуждений. Каждая задача включает описание, привязана к определённой стенограмме и содержит информацию об ответственном исполнителе.

Таблица transcript_tag реализует связь многие-ко-многим между стенограммами и тегами. Она позволяет одной стенограмме иметь несколько тегов и, наоборот, одному тегу соответствовать нескольким стенограммам.

3.5 Модуль выделения аудио из видео

Модуль выделения аудио из видео предназначен для предварительной обработки записей встреч и других видеоматериалов, позволяя извлекать из них аудиодорожку в удобном для последующего анализа виде. Он поддерживает широкий спектр форматов видеофайлов, включая MP4, MKV, MOV, AVI и WEBM, что делает его универсальным решением для различных источников данных. В качестве базового инструмента для реализации модуля используется библиотека FFmpeg, с которой осуществляется взаимодействие через вызовы внешнего процесса средствами стандартной библиотеки Java (ProcessBuilder), позволяющими запускать команды FFmpeg с нужными аргументами и получать выходной поток.

Принцип работы модуля основан на потоковой обработке: FFmpeg запускается в режиме трансляции аудиоданных – он последовательно читает видеофайл, извлекая из него фрагменты аудио объёмом по 2 килобайта, что соответствует 1024 семплам. Эти фрагменты по мере поступления передаются в речевую модель vosk-model-ru-0.42 для последующего распознавания. Благодаря такой схеме снижается нагрузка на оперативную память, поскольку не требуется полного декодирования файла заранее.

Дополнительно в рамках работы модуля реализована система мониторинга прогресса обработки. Поскольку поток обработки может быть очень интенсивным (более 3000 событий в секунду), применяется специальная стратегия уведомлений: внешним компонентам, в частности пользовательскому интерфейсу, отправляются только два основных сигнала – о начале и завершении работы. При этом передаётся ссылка на объект обработчика, с которым фронт может асинхронно взаимодействовать, периодически запрашивая текущий процент завершённой обработки. Такая архитектура обеспечивает баланс между информативностью и производительностью при взаимодействии модулей.

3.6 Модуль стенографирования

Модуль стенографирования осуществляет построение полной стенограммы встречи на основе аудиоданных, извлечённых из видеозаписи с помощью отдельного модуля выделения аудио. Для выполнения задачи распознавания речи используются модели vosk-model-ru-0.42 и vosk-model-spk-0.4, где первая обеспечивает базовое распознавание речевых фрагментов, а вторая — предварительное разделение реплик по участникам. Каждая реплика, согласно данным модели, маркируется как принадлежащая условному «Участнику N», где N — уникальный идентификатор, присвоенный предполагаемому говорящему.

При распознавании речи особое внимание уделяется формированию временных меток каждой реплики. Этот процесс реализуется в рамках потоковой обработки аудиоданных: FFmpeg транслирует аудиопоток в режиме реального времени, в то время как библиотека Vosk осуществляет обработку этого потока. Само определение временных границ каждой реплики происходит по мере поступления фрагментов аудио и анализу их акустических и речевых признаков. Таким образом, Vosk не только фиксирует текст реплики, но и

точно определяет её начальный и конечный временной штамп, на основе которых формируется тайминг всей стенограммы. Фактически, переход к новой реплике инициируется самой моделью Vosk, которая на основе речевых пауз и акустических признаков делит поток на логические фрагменты, соответствующие репликам.

С целью оптимизации времени запуска процесса стенографирования из модели vosk-model-ru-0.42 был исключён LLM-модуль. Экспериментально было установлено, что его влияние на точность распознавания речи незначительно, тогда как исключение модуля позволило сократить время инициализации процесса с пяти минут до примерно двадцати секунд. Это изменение было признано оправданным, поскольку повышение скорости обработки имеет более высокий приоритет в контексте предполагаемых сценариев использования системы.

Для повышения точности идентификации участников встречи реализован дополнительный алгоритм, основанный на сравнении голосовых эмбеддингов — компактных векторных представлений, извлекаемых из аудиосигнала. Эти векторы, размещённые в многомерном признаковом пространстве (50 измерений), позволяют оценивать акустическое сходство между фрагментами речи разных пользователей. В качестве основной меры используется косинусное сходство: чем ближе значение к единице, тем выше вероятность принадлежности фрагментов одному говорящему.

Алгоритм динамически агрегирует голосовые признаки и обновляет векторные профили участников, адаптируясь к акустическим вариациям, возникающим в результате изменения интонации, эмоционального состояния или технических особенностей записи. В случае обнаружения сходства между профилями различных участников производится слияние соответствующих наборов в один, что позволяет избежать ошибочного дублирования говорящих. Также реализована эвристика, устраняющая статистически незначимые реплики, если они приписаны гипотетическим участникам с менее чем тремя уникальными высказываниями.

Таким образом, встроенная в приложение технология стенографирования не только обеспечивает точное распознавание текста с временными метками, но и адаптируется к изменчивости речевых данных, обеспечивая устойчивую и достоверную сегментацию по участникам видеоконференции.

3.7 Модуль редактирования текста

Модуль редактирования текста предоставляет пользователю расширенные возможности по работе с содержимым стенограмм и протоколов встреч. Помимо базового редактирования текста отдельных реплик, система поддерживает операции удаления и добавления реплик, вставки и удаления текста, редактирования временных меток, а также управления списком участников. Для всех операций действует система истории изменений, что позволяет отменять и повторно применять ранее выполненные действия. Для реализации этой функциональности используется архитектурный паттерн Команда. Каждое действие пользователя инкапсулируется в объект, реализующий унифицированный интерфейс с методами применения и отката изменений. Эти объекты последовательно накапливаются в структуре истории, что обеспечивает целостность и воспроизводимость пользовательских действий, независимо от их типа.

Данная архитектура обладает высокой степенью расширяемости: поддержка новых видов операций редактирования осуществляется путём добавления соответствующих реализаций интерфейса команды, без необходимости модификации существующего механизма управления историей. Таким образом, модуль редактирования остаётся легко масштабируемым и адаптируемым к добавлению новых задач.

После завершения обработки видеоконференции модулем стенографирования, каждая из полученных реплик (за исключением статистически незначимых, автоматически отфильтрованных системой) маркируется как принадлежащая условному абстрактному участнику – «Участнику N». Пользователь

имеет возможность заменить абстрактного участника на конкретного, выбрав его из существующей базы данных или добавив нового. После замены все реплики, изначально ассоциированные с данным абстрактным участником, автоматически обновляются и получают имя указанного конкретного участника.

Дополнительно предусмотрена возможность индивидуального изменения участника для каждой реплики. Это позволяет пользователю уточнять принадлежность реплик вручную, при этом изменение касается только выбранной реплики, не затрагивая остальные.

Также реализована система удобной навигации и взаимодействия с участниками встречи. При назначении участника можно воспользоваться функцией поиска: клавиша Enter выбирает первого отфильтрованного участника, при его отсутствии – создаёт нового. Нажатие Escape отменяет выбор участника без сохранения изменений.

Система также включает поддержку набора горячих клавиш, повышающих эффективность работы пользователя:

— Ctrl + Z — отмена последнего действия;

— Ctrl + Y — повтор отменённого действия;

— Ctrl + X — вырезать текст;

— Ctrl + C — скопировать текст;

— Ctrl + V — вставить текст;

— Ctrl + N — создать новую реплику в начале стенограммы;

— Delete — удалить все выбранные реплики;

— Enter (при выборе участника) — назначить первого отфильтрованного или создать нового;

— Escape — отменить выбор участника.

Таким образом, модуль редактирования обеспечивает гибкость, надёжность и высокую степень интерактивности при работе с содержанием стенограмм и протоколов, оставаясь при этом технически устойчивым и легко расширяемым решением.

3.8 Модуль протоколирования

Модуль протоколирования в приложении предназначен для интеллектуального анализа стенограммы встречи. Его основная задача – на основе предоставленного текста сформулировать краткие выводы и выделить из него задачи, которые подлежат дальнейшему исполнению. Для этого в системе используется интеграция с языковой моделью, обеспечивающей генерацию текстов в заданном контексте и адаптацию результатов под формат, удобный для последующей обработки.

В архитектуре модуля применяется паттерн фасад, который позволяет упростить взаимодействие со сложной подсистемой генерации. Пользователь системы или другой компонент приложения работает с единым высокоуровневым интерфейсом, через который происходит обращение к языковой модели. Такой подход избавляет от необходимости взаимодействовать с многочисленными параметрами генерации, токенами или внутренней логикой обработки текста – все это скрыто за фасадом. Вместо этого система оперирует удобными бизнес-объектами, такими как Transcript, Task, Protocol, получаемыми в результате обработки.

Фасад не только абстрагирует сложность, но и централизует управление жизненным циклом подсистемы. Он отвечает за инициализацию необходимых ресурсов, запуск обработки, а также за корректное завершение работы с моделью, включая освобождение памяти. Дополнительным преимуществом такой архитектуры является гибкость – возможно легко адаптировать поведение модуля под конкретные задачи путём настройки входных параметров, не нарушая основной структуры.

Таким образом, модуль реализует эффективную, расширяемую и удобную в использовании схему обработки стенограмм, в которой применённые паттерны проектирования позволяют обеспечить читаемость архитектуры, поддержку масштабирования и изоляцию бизнес-логики от технических деталей взаимодействия с моделью.

3.9 Модуль хранения информации

Созданные с помощью приложения стенограммы, протоколы и участники встреч хранятся в рамках приложения во встроенной в приложение БД. Для организации доступа к БД в проекте используется DAO-подход (Data Access Object). Вместо автоматического связывания моделей с таблицами через ORM, вся работа с данными осуществляется вручную через классы-обёртки, отвечающие за конкретные таблицы и запросы к ним. Такой подход был выбран с учётом особенностей приложения: система предназначена для использования одним пользователем на локальном компьютере, без серверной части и масштабной многопользовательской нагрузки. Количество таблиц в базы данных невелико – всего 8, а структура сравнительно проста, поэтому использование полнофункционального ORM-инструмента представляется избыточным.

Применение ORM, такого как Hibernate в режиме полной декларативной модели, в данном случае повлекло бы за собой ряд ненужных расходов: автоматическая генерация и отслеживание схемы, инициализация большого количества внутренних объектов, дополнительное потребление оперативной памяти, а также усложнение процесса отладки и запуска приложения. В то время как ручная реализация SQL-запросов позволяет сократить время выполнения операций, минимизировать зависимость от сторонних библиотек и при необходимости реализовать более гибкую и точную выборку данных.

Кроме того, DAO-подход обеспечивает большую прозрачность в логике работы с базой и в целом соответствует требованиям к компактности и контролируемости, особенно актуальным в условиях полной локальной работы приложения без подключения к интернету.

При этом, на уровне логики реализованы сущности, которые реализуют логику создания базы данных, срабатывающую при развертывании приложения локально, логику управления соединения с базой данных, управляет доступом к DAO, CRUD операции.

3.10 Модуль переключения моделей

В приложении реализован механизм очереди задач, обеспечивающий последовательное выполнение заданий пользователя. Такая архитектура позволяет пользователю задать сразу несколько задач, например — обработку видеозаписей с целью получения стенограмм, а также генерацию итоговых протоколов на основе уже расшифрованного текста.

Очередь обрабатывается последовательно, с учётом типа первой поступившей задачи, что определяет активную модель на данном этапе обработки:

— если первой задачей является генерация протокола, активируется языковая модель (LLM), и все задачи на составление протоколов, находящиеся в очереди, выполняются последовательно до полного завершения данного этапа;

— если в процессе выполнения таких задач в очередь добавляются задания на составление стенограмм, они остаются в очереди и обрабатываются только после завершения всех задач на составление протоколов. После этого LLM выгружается из оперативной памяти, и вместо неё загружаются модели Vosk, необходимые для обработки аудио и преобразования его в текст и разделения текста по репликам;

— если первой задачей в очереди является формирование стенограммы, то приоритет получат модели Vosk, и все задачи данного типа обрабатываются последовательно. В этом случае задачи на формирование протоколов, добавленные позже, будут поставлены в конец очереди и выполнены только после завершения всех задач на расшифровку аудио. После завершения стенографирования модель Vosk выгружается, и загружается LLM для выполнения оставшихся задач.

Такой подход позволяет минимизировать количество переключений между различными моделями, снижая затраты на их загрузку и выгрузку, что, в свою очередь, способствует более эффективному использованию оперативной памяти и улучшению пользовательского опыта.

3.11 Реализация интерфейса

При запуске приложения пользователю предоставляется возможность взаимодействовать с приветственным экраном, который изображен на рисунке 10. На нем представлены последние нововведения в приложении, лозунг приложения и принципы приложения.

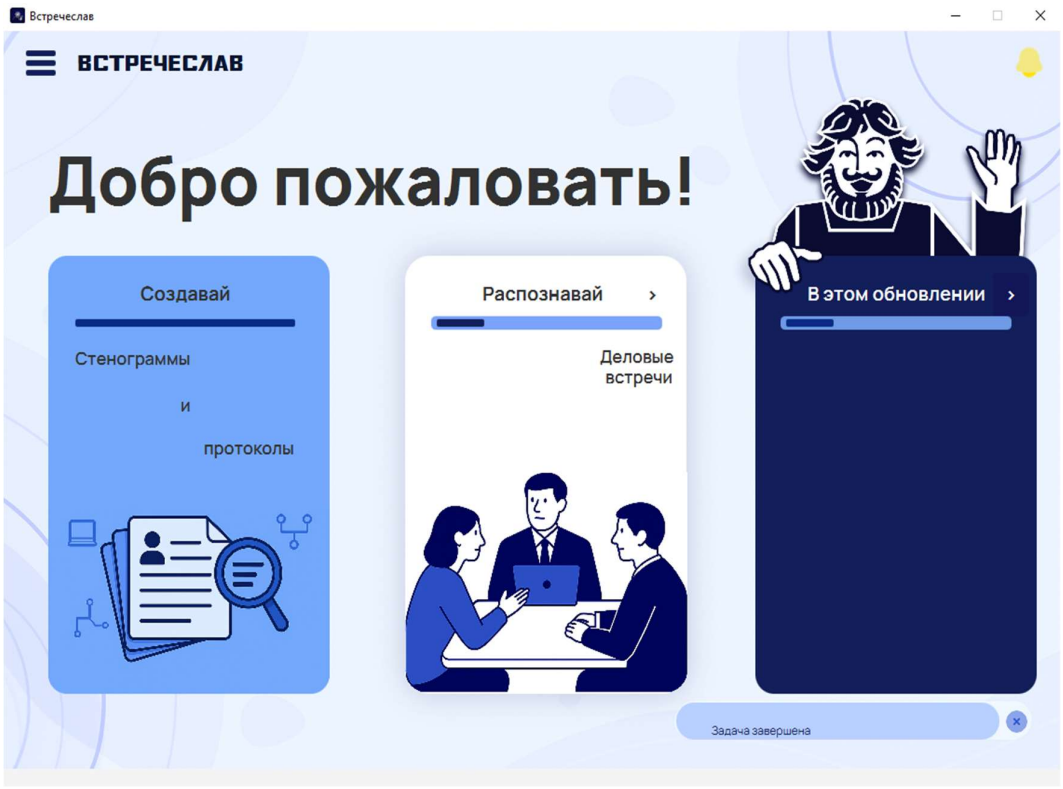

Рисунок 10 – Приветственный экран приложения

При открытии бокового меню с навигацией по приложению, пользователю предоставляется возможность перейти в любой из указанных в нем разделов.

Приветственный экран с открытым навигационным меню представлен на рисунке 11.

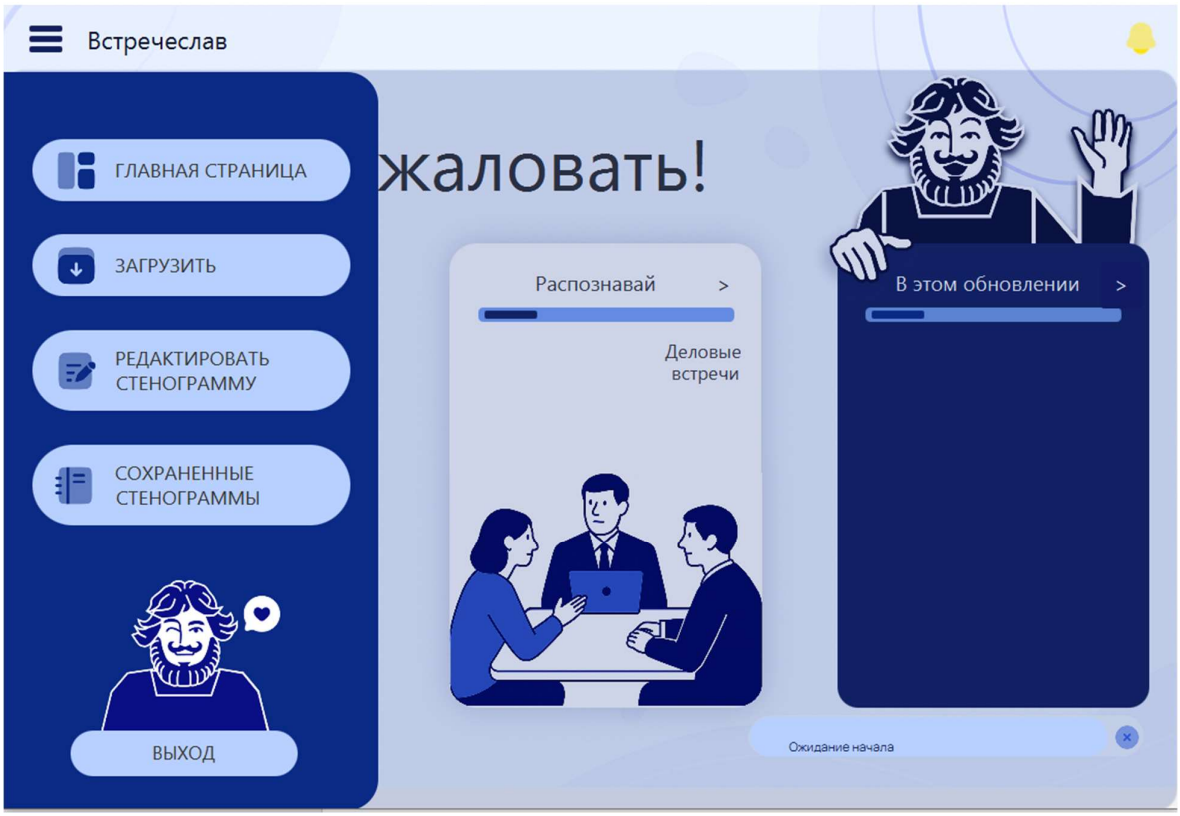

Рисунок 11 – Приветственный экран приложения с открытым навигационным меню

В навигационном меню можно перейти на любой из экранов приложения: приветственный экран, экран загрузки видео, экран редактирования текста, экран материалов встречи.

При выборе пользователем пункта «Загрузить» открывается экран, предоставляющий пользователю возможность загрузить в систему видео с его локального хранилища, либо по ссылке с bbb.edu.vsu.ru. Сам экран представлен на рисунке 12.

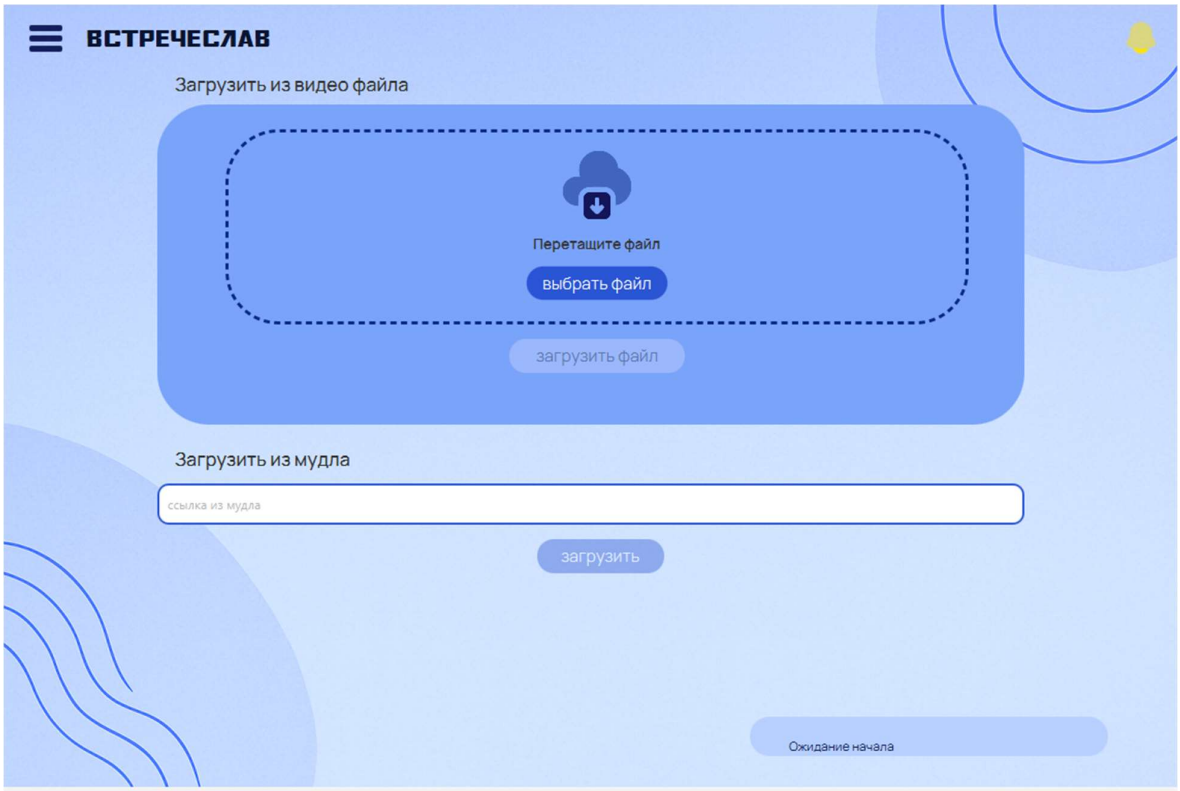

Рисунок 12 – Экран загрузки видео

Опишем экран подробнее. Файл с записью видеоконференции можно загрузить, как нажав на кнопку с выбором файла, так и перетащив его в отмеченную пунктиром область. Для того, чтобы обработать лекцию с bbb.edu.vsu.ru, которая будет представлена в виде URL, требуется ввести URL в поле, которое находится под надписью «Загрузить из мудла». Для того, чтобы начать обработку, в первом случае необходимо нажать кнопку «загрузить файл», во втором – кнопку с надписью «загрузить».

При выборе файлов, для которых нужно составить стенограмму, они добавляются в очередь задач, просмотреть которую можно с любого из других экранов приложения. Пользователю предоставляется возможность ознакомиться с содержимом очереди, и с тем порядком, в котором будут обработаны загруженные им файлы. Экран загрузки с развернутой очередью задач представлен на рисунке

Рисунок 13 – Экран загрузки с очередью.

При завершении задачи в очереди указывается, что задача была завершена. Возможность ознакомиться с результатом работы системы – с составленной по видео стенограмме или же с составленным по стенограмме протоколом – предоставляет экран редактирования.

На рисунке 13 приведен экран редактирования стенограммы.

Рисунок 14 – Экран редактирования текста с содержимым стенограммы

Пользователь может редактировать текст составленных приложением стенограмм и протокола, сохранять их, а также экспортировать их в формате файла с простым текстом.

После составления стенограммы, пользователю предоставляется возможность составить протокол встречи по ней. Составленный протокол встречи приведён на рисунке

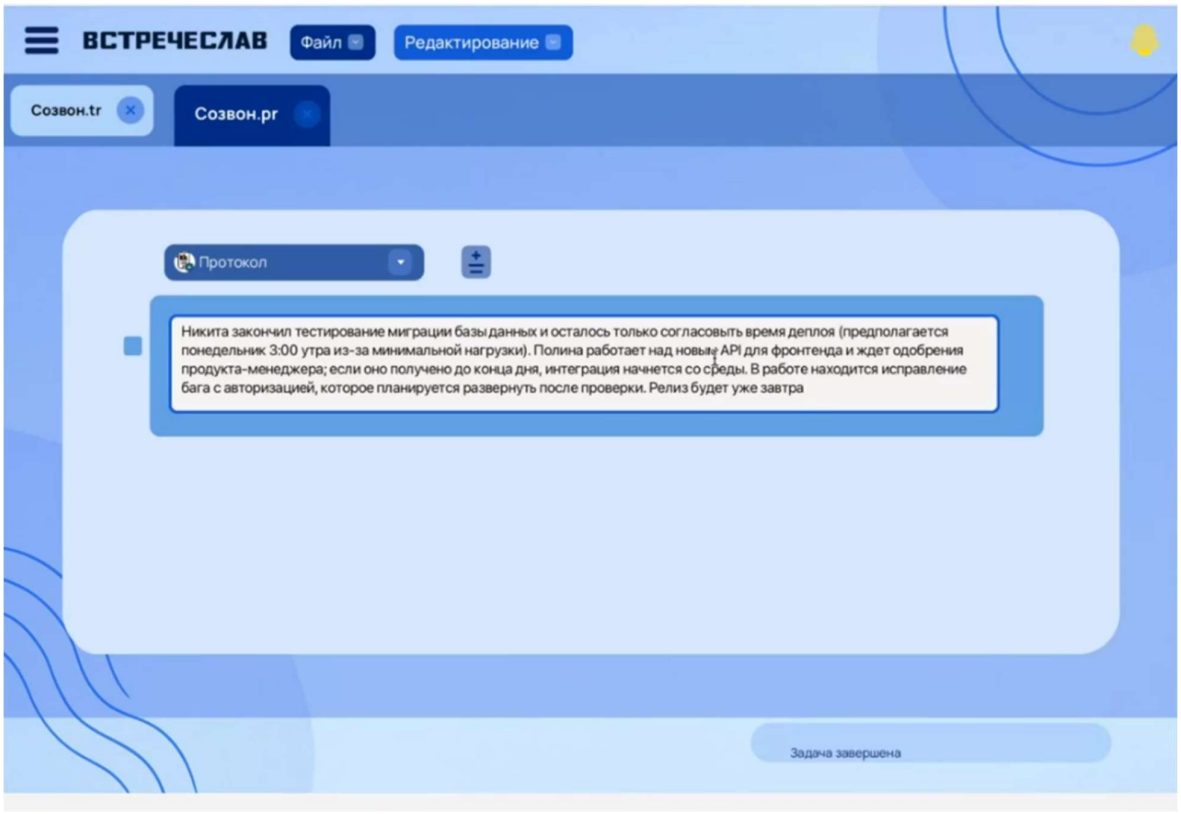

Рисунок 15 – Экран редактирования текста с содержимым протокола встречи

Опишем экран подробнее. На нем представлен текст с протоколом встречи, в котором содержатся выводы и задачи по видеоконференции, для которой предварительно была составлена стенограмма. Его можно сохранить в БД, а так же экспортировать в формат txt.

После сохранения файла в БД, он будет отображаться на экране с материалами встреч. Этот экран представлен на рисунке 16.

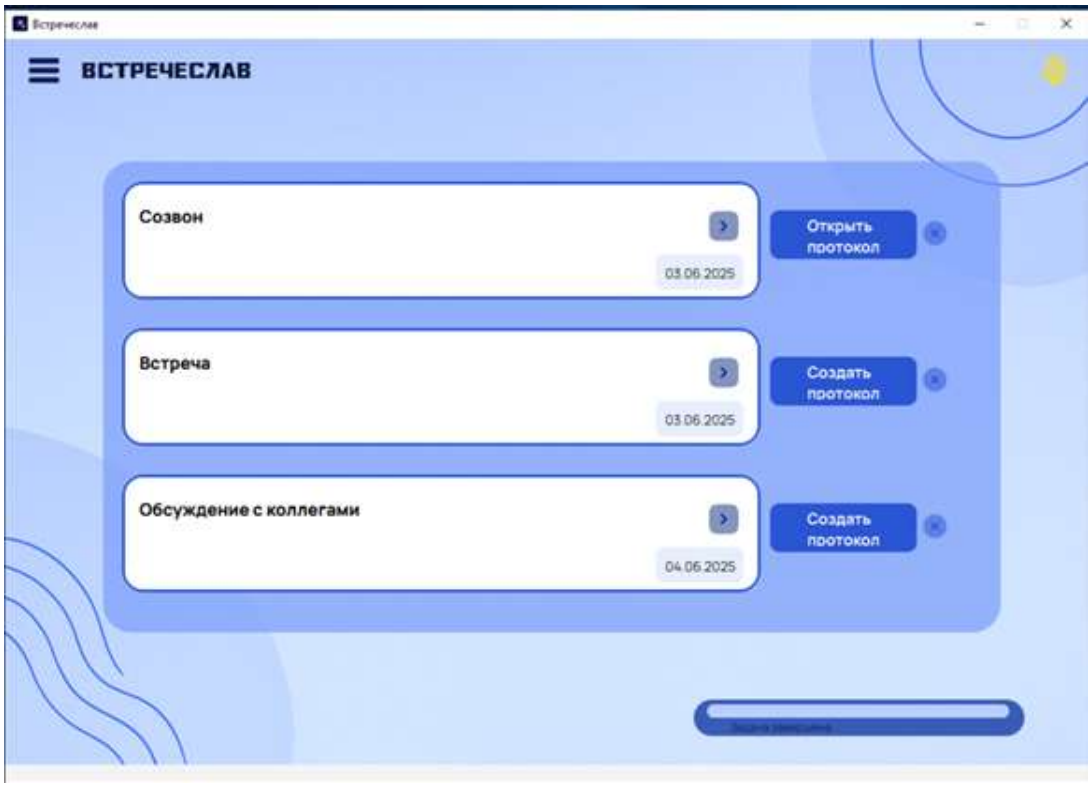

Рисунок 16 – Экран с материалами встреч

Таким образом, были описаны экраны и основные взаимодействия с ними.

Заключение

В ходе выполнения данного курсового проекта были выполнены все поставленные цели:

— автоматизация анализа и стенографирования встреч за счет создания системой стенограммы и протокола встречи;

— обеспечение конфиденциальности обработки данных и предоставление пользователю полного контроля над доступностью приложения.

Список использованных источников

1. Будущее работы: как цифровые технологии определяют новые стандарты производительности и благополучия сотрудников [Электронный ресурс]. – пиво Режим доступа:URL: https://www.novostiitkanala.ru/news/detail.php?ID=181668 – Заглавие с экрана (дата обращения 20.05.2025).

2. Как записывать протоколы собраний: лучшие практики и преимущества [Электронный ресурс] / Sonix. – Загл. с экрана. – Режим доступа: URL: https://shorturl.at/eAd3l/ (дата обращения: 20.06.2025).

3. Транскрибация – все, что нужно знать о преобразовании речи из аудио и видео в текст [Электронный ресурс] / Таймлист. – Заглавие с экрана. – Режим доступа: URL: https://timelist.ru/blog/transcribation (дата обращения: 21.05.2025).

4. Главные угрозы безопасности в облаке [Электронный ресурс] / TAdviser. – Загл. с экрана. – Режим доступа: URL: https://www.tadviser.ru/index.php/Статья:Главные_угрозы_безопасности_в_облаке (дата обращения: 21.05.2025).

5. Импортозамещение программного обеспечения в России: что это и как реализуется [Электронный ресурс] / Bercut. – Загл. с экрана. – Режим доступа: URL: https://bercut.com/blog/technologies/importozameshchenie-programmnogo-obespecheniya/ (дата обращения: 22.05.2025).

6. IVA Technologies представляет IVA One — новую платформу для бизнес-коммуникаций [Электронный ресурс] / IVA Technologies. – Загл. с экрана. – Режим доступа: URL: https://iva.ru/ru/news/iva-technologies-predstavlyaet-iva-one-novuyu-platformu-dlya-biznes-kommunikaczij/ (дата обращения: 21.05.2025).

7. Как это сделать: настройка, ресурсы и время на установку IVA MCU [Электронный ресурс] / IVA Technologies. – Загл. с экрана. – Режим доступа: URL: https://habr.com/ru/companies/iva_tech/articles/879310/ (дата обращения: 20.05.2025).

8. WizWhisp – офлайн-приложение для транскрибации аудио и видео [Электронный ресурс] / NowSmart. – Загл. с экрана. – Режим доступа: URL: http://apps.microsoft.com/detail/9pgq3h6jxl4c?hl=ru-RU&gl=RU (дата обращения: 21.05.2025).

9. Junit 5 User Guide [Электронный ресурс] / JUnit. – Режим доступа: URL: https://junit.org/junit5/docs/current/user-guide/ (дата обращения: 20.05.2025).

10. Vosk API [Электронный ресурс] / Alpha Cephei Inc. – Режим доступа: URL: https://github.com/alphacep/vosk-api (дата обращения: 22.05.2025).

11. Vosk Language Model Documentation [Электронный ресурс] / Alpha Cephei Inc. – Режим доступа: URL: https://alphacephei.com/vosk/lm (дата обращения: 22.05.2025).