







МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Техническое задание  
на разработку приложения для настольных компьютеров  
«Интеллектуальный помощник для деловых встреч Встречеслав»

Исполнители

	С.И. Бондарева
	П.С. Парамонова
	В.С. Попов
	Н. А. Агафонов
	Н. Н. Андреев
	К.В. Воронецкий

Заказчик

\_\_\_\_\_ В.С. Тарасов

Воронеж 2025

1. Общие сведения.....	11
1.1. Полное наименование системы и ее условное обозначение .....	11
1.2. Наименование исполнителя и заказчика приложения .....	11
1.3. Перечень документов, на основании которых создается приложение .....	11
1.4. Плановый срок начала и окончания работ .....	12
2. Цели и назначение создания приложения .....	13
2.1. Цели создания приложения.....	13
2.2. Назначение приложения.....	13
3. Характеристика объекта автоматизации .....	15
3.1. Краткие сведения об объекте автоматизации .....	15
3.2. Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды .....	15
4. Требования к автоматизированной системе.....	16
4.1. Требования к структуре АС в целом.....	16
4.1.1. Перечень подсистем, их назначение и основные характеристики .....	16
4.2. Требования к функциям (задачам), выполняемыми приложением .....	17
4.2.1. Требования к модулю выделения аудио из видео .....	17
4.2.2. Требования к модулю стенографирования .....	18
4.2.3. Требования к модулю редактирования текста.....	19
4.2.4. Требования к модулю протоколирования .....	19
4.2.5. Требования к модулю хранения информации .....	20
4.3. Требования к видам обеспечения системы .....	20
4.3.1. Требования к лингвистическому обеспечению системы.....	21
4.3.2. Требования к программному обеспечению системы.....	21
4.4. Общие технические требования .....	21
4.4.1. Требования по безопасности .....	22
4.4.2. Требования к эргономике и технической эстетике .....	22
5. Технические риски.....	23
6. Состав и содержание работ по созданию системы.....	24
7. Порядок разработки автоматизированной системы .....	26
7.1. Порядок организации разработки системы.....	26
8. Порядок контроля и приемки системы .....	28

9. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие .....	29
10. Требования к документированию .....	30
10.1. Перечень подлежащих разработке документов.....	30
10.2. Вид представления и количество документов .....	30
11. Источники разработки .....	31
Приложение А. Диаграмма классов .....	32
Приложение Б. Sequence диаграмма .....	37
Приложение В. ER диаграмма базы данных .....	39

## Термины

В этом разделе приведены основные термины и определения, используемые в техническом задании для того, чтобы обеспечить единое понимание их значений. Информация о них приведена в таблице 1.

**Таблица 1 – Термины, используемые в техническом задании**

Термин	Значение
Атрибуты	Свойства или характеристики сущностей на ER диаграмме.
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.

Термин	Значение
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.
Персональный компьютер	Вычислительное устройство, предназначенное для индивидуального использования, включающее процессор, память и устройства ввода/вывода. К видам персональных компьютеров относятся настольные и портативные – ноутбуки.
Простой текст	Текст, который не имеет специальных тегов, специального форматирования.
Протокол встречи	Документ, содержащий краткую сводку обсужденных на встрече проблем и задач, принятые по каждому из проблем решение и общий вывод по итогам встречи.

Термин	Значение
Репозиторий	Централизованное цифровое хранилище, которое разработчики используют для внесения изменений в исходный код приложения и управления ими.
Системы контроля версий (СКВ)	Программный инструмент, предназначенный для управления изменениями в исходном коде, документации. СКВ позволяет отслеживать, сохранять и при необходимости откатывать изменения, обеспечивает совместную работу нескольких разработчиков, предотвращая конфликты при внесении правок.
Стенограмма	Текстовый документ, содержащий дословную запись устной речи, полученную путем расшифровки аудио- или видеозаписи выступления, беседы, совещания.

Термин	Значение
Сущность на ER диаграмме (сущность)	Объекты или понятия, которые существуют в системе и о которых нужно хранить информацию.
Таск-трекер	Программная система, предназначенная для управления задачами в рамках проекта. Таск-трекер позволяет создавать, распределять, отслеживать и контролировать выполнение задач, устанавливать приоритеты, сроки выполнения и ответственных исполнителей.

Термин	Значение
Фоновая обработка	Режим, в котором программа продолжает выполнять свои функции после закрытия основного окна или его сворачивания. В этом режиме приложение остаётся активным, выполняет автоматические процессы (например, выделение аудио составляющей из видео).
ER диаграмма	Графическое представление данных и их взаимосвязей, используемое в процессе проектирования баз данных.
Figma	Облачная платформа для создания макетов дизайна приложений, а также инструмент для организации рабочего процесса и совместной работы команды в реальном времени.
Java	Строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems.



Термин	Значение
Kanban	Метод управления рабочими процессами, основанный на визуализации задач, ограничении незавершенной работы и постепенном улучшении процессов. В Kanban задачи представлены в виде карточек на доске (Kanban-доске) и перемещаются между статусами, что позволяет отслеживать их прогресс и балансировать нагрузку внутри команды.
Sequence диаграмма	Наглядное представление совокупности разных элементов модели системы, изображение того, как и в каком порядке они взаимодействуют.
Solid state drive(SSD)	Устройство хранения данных, которое использует интегральные схемы в качестве памяти для постоянного хранения данных.
Ubuntu	Операционная система на базе ядра Linux, разработанная компанией Canonical и распространяемая с открытым исходным кодом.

Термин	Значение
Vosk	<p>ИИ-система для распознавания речи на естественном языке из аудиофайла и формирования на его основе текста на том же языке, взаимодействие с которой может осуществляться с помощью языка программирования Java. В рамках автоматизированной системы применяется модель, которая распознает речь на русском языке, в которой могут присутствовать англицизмы, профессиональная лексика.</p>

## **1. Общие сведения**

В этом разделе приведены общие сведения о разрабатываемой автоматизированной системе.

### **1.1. Полное наименование системы и ее условное обозначение**

Полное наименование системы: «Интеллектуальный помощник для деловых встреч Встречеслав».

Краткое наименование приложения: «Встречеслав».

### **1.2. Наименование исполнителя и заказчика приложения**

Заказчик: Старший преподаватель Тарасов Вячеслав Сергеевич, Воронежский Государственный Университет, Факультет компьютерных наук, кафедра Программирования и Информационных Технологий.

Разработчик: «5» команда группы «9».

Состав команды разработчика:

- Бондарева Софья Игоревна;
- Парамонова Полина Сергеевна;
- Попов Виталий Сергеевич;
- Агафонов Никита Александрович;
- Андреев Никита Николаевич;
- Воронежский Константин Владимирович.

### **1.3. Перечень документов, на основании которых создается приложение**

Данное приложение будет создаваться на основании следующих документов:

- учебный план по программе бакалавриата по направлению 09.03.04 «Программная инженерия» по специальности для 2022 года начала обучения;

- распоряжение от 09.01.2025 № 1600-62 «О сроках текущей и промежуточных аттестаций».

#### **1.4. Плановый срок начала и окончания работ**

Плановый срок начала работ: 31 марта 2025 года.

Плановый срок окончания работ: 10 июня 2025 года.

## **2. Цели и назначение создания приложения**

Этот раздел определяет назначение автоматизированной «Встречеслав» и его цели, включая автоматизацию обработки деловых встреч, снижение затрат времени на организационные задачи и обеспечение безопасности данных.

### **2.1. Цели создания приложения**

Целями выполнения работ по созданию приложения «Встречеслав» являются:

- автоматизация анализа и стенографирования встреч за счет создания системой стенограммы и протокола встречи. Качество составления стенограммы и протокола встречи должно оцениваться как удовлетворительное не менее чем 70 процентами участников тестовой группы из 35 человек;
- релиз на трех основных платформах: Windows, macOS и Linux. После этого не менее 70% пользователей из тестовой группы из 35 человек должны быть удовлетворены работой системы на своих устройствах. Это подтвердится тем, что они оценят свой опыт работы с приложением не менее чем в 7 из 10 баллов в опросе, проведенном командой разработчиков автоматизированной систем.

### **2.2. Назначение приложения**

Разрабатываемое приложение будет решать проблему потери информации при ручном документировании деловых встреч и временных затрат за счет создания стенограммы встречи и её протокола. Особенностью будет являться полная локальная обработка данных на устройстве пользователя без использования облачных сервисов, что даст пользователю контроль за конфиденциальностью результатов работы приложения. Приложение избавит от необходимости вести ручные записи и самостоятельно составлять протоколы встреч. Система может оказаться полезна для руководителей, юристов, госслужащих, работников в

сфере информационных технологии, которые проводят много встреч и для которых важна безопасность данных, а также для студентов или людей, которые имеют дело с относительно большим объёмом – в масштабе одного человека - информации, которую необходимо перевести в текстовый формат.

### **3. Характеристика объекта автоматизации**

Этот раздел описывает объект автоматизации, его функциональные возможности, целевое назначение и технические требования для корректной работы, что позволяет определить границы применения системы.

#### **3.1. Краткие сведения об объекте автоматизации**

«Встречеслав» будет представлять из себя приложение для персональных компьютеров, предназначенное для создания стенограммы по загруженным файлам, составления протокола встречи и создания задач по видеозаписям деловых встреч.

#### **3.2. Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды**

Корректное функционирование приложения для персональных компьютеров должно быть обеспечено для устройств с операционной системой Windows 10 версии 22h2, с macOS версии Sonoma (14) и выше, с Ubuntu версии 24.04. Персональный компьютер, на котором гарантируется корректная работа системы, должен соответствовать следующим требованиям:

- в качестве постоянного запоминающего устройства – SSD со скоростью чтения и записи от 500 мегабит в секунду;
- не менее 12 Гб свободной памяти на постоянном запоминающем устройстве;
- не менее 5 Гб свободной оперативной памяти для размещения приложения на протяжении всей его работы;
- процессор, характеристики производительности которого – тактовая частота, количество ядер и потоков – будут аналогичны характеристикам процессора Intel Core i3 12100. Допускается также, чтобы характеристики производительности процессора были выше.

## **4. Требования к автоматизированной системе**

В данном разделе описываются требования к автоматизированной системе, её функциям и структуре.

### **4.1. Требования к структуре АС в целом**

Система должна представлять из себя приложение для персональных компьютеров, в котором должны быть следующие логические модули: выделения аудио из видео, стенографирования, редактирования текста, протоколирования, хранения информации. Архитектура автоматизированной системы должна быть монолитной, в рамках которой должны быть выделены вышеперечисленные логические модули. Вариант того, как при этом могла бы выглядеть логическая структура модулей в виде классов в рамках кодовой базы системы, приведён на диаграмме классов в приложении А к настоящему техническому заданию. При этом, предполагаемый вариант того, как пользователь мог бы получить протокол встречи в результате взаимодействия с системой, которая будет разработана в соответствии с настоящим техническим заданием, представлен в рамках sequence диаграммы, которая приведена в приложении Б к техническому заданию.

#### **4.1.1. Перечень подсистем, их назначение и основные характеристики**

Модуль выделения аудио из видео должен предоставлять возможность загрузить запись встречи в видеофайле и выделить из неё только аудио составляющую. При этом, поддерживаются следующие форматы видеофайлов: MP4, MKV, MOV, AVI, WEBM.

Модуль стенографирования должен составить стенограмму встречи по аудио, которое было получено в результате работы модуля выделения аудио из видео.

Модуль редактирования текста должен предоставлять возможность отредактировать текст, который будет получен в результате работы подсистем стенографирования и протоколирования.



Модуль протоколирования предоставлял должен создать протокол встречи по результатам работы модуля стенографирования и при этом учитывал изменения, которые могли были внесены в стенограмму с помощью модуля редактирования текста.

Модуль хранения информации должен предоставлять возможность пользователю внести информацию об участниках встреч, реализовывать хранение данных о стенограммах встреч и предоставлять доступ к данным, расположенным в самом модуле.

#### **4.2. Требования к функциям (задачам), выполняемыми приложением**

К системе выдвигается следующий список требований:

- система должна обеспечивать фоновую обработку в режиме онлайн видеофайлов, текста, который создан в результате работы системы и находится в рамках её базы данных;
- система должна предоставлять пользователю возможности сохранять и редактировать результаты работы модулей стенографирования, протоколирования и редактирования текста и получать к ним доступ в рамках системы.

##### **4.2.1. Требования к модулю выделения аудио из видео**

К подсистеме выделения аудио из видео выдвигаются следующие функциональные требования:

- подсистема должна поддерживать загрузку файлов в форматах MP4, MKV, MOV, AVI, WEBM. Подсистема должна извлекать аудио составляющую из видео, когда файл фактически представляет из себя видео, которое было записано с помощью системы, не являющейся частью текущей автоматизированной системы и имеет один из поддерживаемых системой форматов. В случае же, если файл имеет расширение одного из

поддерживаемых форматов, но его содержимое представляет из себя другой тип медиа, например, звук или изображение, то отсутствует гарантия того, что система сможет составить стенограмму, качество которой будет удовлетворительным для пользователя; система должна уведомить пользователя о том, что он пытается обработать файл, который по своей сути не является видеофайлом;

- подсистема должна извлекать аудио составляющей из загруженного видео. При загрузке видео в систему автоматически выделяется его аудиодорожка.

#### **4.2.2. Требования к модулю стенографирования**

К подсистеме стенографирования выдвигаются следующие функциональные требования:

- подсистема должна реализовывать преобразование аудиофайла, созданного подсистемой выделения аудио из видео, в текст. Этот аудиофайл затем используется для стенографирования — процесса, в котором с помощью специализированной ИИ-системы выделяются слова и фразы и преобразуются в текст. Полученный текст должен передавать содержание речи, если она присутствовала в исходном видео. ИИ-система должна обработать аудиофайл, идентифицировать слова и фразы, а затем преобразовывать их в текстовый формат – с обычным текстом, без стилей, отражая содержание речи. Качество составленной стенограммы будет определяется метрикой количества некорректно распознанных слов. Будем выделять два типа ошибок: мягкая ошибка и грубая ошибка. Мягкая ошибка – это такая ошибка, при которой в слове допущено несколько замен букв. Грубая ошибка – такая, при которой смысл слова не восстановим пользователем, или же при которой слово не было включено в предложение. Сама метрика определяется количеством слов с двумя типами таких ошибок к общему числу слов в стенограмме, при этом общее

число слов определяются с помощью эталонной стенограммы – такой, которая была составлена вручную;

- подсистема должна обеспечивать возможность распределения реплик по условным участникам встречи с присвоением каждому из них уникального идентификатора в виде номера в порядке его первого появления в разговоре. Каждая реплика должна быть представлена в формате «Участник N», где N — порядковый номер говорящего в стенограмме, после чего будет приведена сама реплика участника встречи.

#### **4.2.3. Требования к модулю редактирования текста**

К подсистеме редактирования текста выдвигается следующий список функциональных требований:

- подсистема должна обеспечить возможность редактирования стенограммы встречи: должны быть реализованы ввод и удаление – возможность добавлять и удалять текст в любом месте стенограммы, копирование и вставка – перенос фрагментов текста внутри документа или из внешних источников. Система должна позволить вручную указать идентифицирующие данные для каждого участника (ФИО), заменяя шаблон в формате «Участник N» на конкретное имя;
- подсистема должна обеспечить возможность вставки фрагмента текста в любое место текста, который был создан любой из систем стенографирования или протоколирования, копирование любого фрагмента текста и изменение текста пользователем.

#### **4.2.4. Требования к модулю протоколирования**

К подсистеме протоколирования выдвигаются следующие функциональные требования:

- подсистема должна обеспечить автоматическое формирование протокола встречи на основе стенограммы. Подсистема должна передать на

вход большой языковой модели стенограмму встречи и указать, что нужно сформировать протокол встречи с двумя разделами: тезисного пересказа встречи с выводами, которое были приняты в ходе встречи и формулировок задач. Результат работы большой языковой модели – простой текст, поделенный на две части, каждую из которых пользователь может редактировать – удалять текст, вставить новый или же копировать существующий. Подсистема должна обеспечить возможность пользователю вручную дописать к каждой формулировке задачи срок исполнения задачи и вручную указать ответственного за задачу. Подсистема должна предоставить возможность назначить ответственным как участника конференции, так и человека, данные о котором были ранее внесены в подсистему хранения данных;

#### **4.2.5. Требования к модулю хранения информации**

К разрабатываемой подсистеме хранения информации выдвигаются следующие функциональные требования:

- подсистема должна предоставить доступ к ранее внесённым данным об участниках конференции. Внесение новой информации и редактирование уже существующей будет осуществляться пользователем вручную, и результаты будут сохраняться в локальной базе данных;
- система должна обеспечить возможность пользователю вносить информацию, позволяющую идентифицировать участника конференции – ФИО – с возможностью последующего редактирования внесённых данных, и получения доступа к ранее внесённым данным.

#### **4.3. Требования к видам обеспечения системы**

В данном разделе приведены требования к лингвистическому и программному обеспечению системы.

#### **4.3.1. Требования к лингвистическому обеспечению системы**

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

#### **4.3.2. Требования к программному обеспечению системы**

Для разработки модулей редактирования стенограммы, протоколирования и хранения информации должен быть использован язык программирования Java.

Для реализации модуля хранения данных должен быть использован язык SQL, а в качестве системы управления базами данных должна применяться SQLite. Прототипы того, как можно будет выделить сущности и отношения в рамках разрабатываемой базы данных на концептуальном уровне, представлено в приложении В к текущему техническому заданию.

Для реализации модуля стенографирования должна будет использоваться искусственная интеллектуальная система Vosk, от которой требуется предоставление возможности преобразования речи в текст.

Для реализации модуля протоколирования в приложении будет использоваться большая языковая модель, к которой имеется следующий список требований:

- ИИ-система должна осуществлять разделение входного текста на смысловые блоки;
- ИИ-система должна осуществлять генерацию текста протокола встречи на основе входного текста стенограммы встречи и предоставляла пользователю возможность ознакомиться с результатами обработки.

#### **4.4. Общие технические требования**

В данном разделе описаны требования по безопасности и требования к эргономике и технической эстетике.

#### **4.4.1. Требования по безопасности**

Подсистемы стенографирования и протоколирования должны функционировать без необходимости подключения устройства к сети интернет. Вся обработка данных в рамках приложения в том числе создание стенограммы, протоколирование и их хранение, должно осуществляться локально на устройстве пользователя в режиме офлайн. За счет этого пользователь сможет управлять доступностью данных, а их конфиденциальность может быть дополнительно обеспечена внешними системами, например, системами шифрования текста, которые не являются частью текущей разрабатываемой автоматизированной системы.

#### **4.4.2. Требования к эргономике и технической эстетике**

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

Подсистема редактирования текста должна реализовывать базовые инструменты работы с текстом, а именно: ввод нового текста, удаление существующего, копирование и вставка фрагментов текста. Подсистема должна работать с простым текстом.

## 5. Технические риски

В данном разделе приведено описание технических рисков, и варианты их решения.

В число технических рисков входят:

- риск того, что качество распознавания речи ИИ-системой в подсистеме стенографирования окажется недостаточным для корректной работы подсистемы протоколирования;
- риск того, что за счет недостаточно качественной записи или дефектов речи участников результат стенограмма встречи может быть недостаточно точной, или же отражать те положения, которые на самом деле не обождались на встрече;
- риск того, что большая языковая модель не сможет структурировать информацию так, чтобы получался протокол встречи такого качества, которое пользователь оценит как удовлетворительный;

Вариантами разрешения технических рисков являются:

- других ИИ-систем в соответствующих подсистемах, которые будут менее требовательны к производительности устройства, на котором они будут работать;
- обучение моделей на другом наборе данных, которые будут использоваться в автоматизированной системе. Это может повысить качество результатов работы ИИ-систем.

## 6. Состав и содержание работ по созданию системы

Состав и содержание работ по созданию системы включают в себя следующие этапы: предпроектное исследование, разработка и тестирование разработанного программного обеспечения. Каждый из этапов может быть разбит на промежуточные. Состав и содержание работ по каждому из промежуточных этапов, и сами промежуточные этапы приведены в таблице 2.

**Таблица 2 – состав и содержание работ**

Номер этапа	Содержание работ в рамках этапа	Временные границы
1	Сбор необходимой информации, постановка целей, задач системы, которые в будущем должны быть реализованы	20.02.2025 – 01.03.2025
2	Анализ целевой аудитории и рынка	01.03.2025 – 30.03.2025
3	Анализ предметной области, анализ конкурентов и построение структуры требований, выражающейся в списке функциональности, ведущих к решению поставленных задач и целей	01.03.2025 – 30.03.2025
4	Составление дорожной карты продукта	11.03.2025 – 30.03.2025



Номер этапа	Содержание работ в рамках этапа	Временные границы
5	Подготовка дизайн-макетов в Figma	15.03.2025 – 30.03.2025
6	Составление начальной архитектуры приложения, основной ER-диаграммы, определение основного стека технологий	05.03.2025 – 30.03.2025
7	Построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД	05.03.2025 – 30.03.2025
8	Спецификация дизайн-макетов в Figma	30.03.25 – 15.04.25
9	Разработка рабочего проекта, состоящего из написания кода, отладки и корректировки кода программы	02.04.25 – 01.05.25
10	Проведение тестирования программного обеспечения и исправление найденных ошибок	01.05.25 – 01.06.25

## **7. Порядок разработки автоматизированной системы**

В данном разделе приведена организация разработки системы.

### **7.1. Порядок организации разработки системы**

Разработка системы будет осуществляться с применением гибкой методологии управления проектами Kanban. Данная методология позволяет учитывать разнородность команды, в которой участники обладают специализированными компетенциями. Процесс разработки будет организован таким образом, что можно будет учитывать изменение требований в ходе разработки. Опишем основные черты и процессы, которых характерны для разработки в соответствии с данной методологией:

- формирование потока задач – задачи, из которых состоит проект, будут размещаться в общем рабочем пространстве в таск-трекере и упорядочены по приоритетам. Каждый участник берет в работу те задачи, которые соответствуют его компетенции и находятся в зоне его ответственности;
- ограничение количества задач с определённым статусом – это необходимо для того, чтобы выявлять узкие места в процессах и устранять их. Продвижение задач по статусам формирует конвейер;
- управление приоритетами – в зависимости от изменяющихся требований заказчика, критические задачи могут оперативно подниматься в приоритет без необходимости ожидания завершения фиксированного цикла разработки;
- непрерывное тестирование и интеграция – проверка работоспособности приложения будет выполняться по мере готовности отдельных модулей, что обеспечит раннее выявление ошибок и сокращение времени на корректировку;

- регулярные обзоры состояния проекта – команда будет проводить оперативные встречи для синхронизации работы, обсуждения возникающих проблем и уточнения приоритетов без жесткой привязки к спринтам.

Разработка будет состоять из двух крупных стадий. В рамках первой будут разработаны, протестированы и отлажены следующие функциональные модули:

- подсистема преобразования видео в аудио;
- подсистема преобразования аудио;
- подсистема хранения информации.

После этой стадии будет проведен анализ того, сколько для каждой задачи потребовалось времени, чтобы она прошла все стадии конвейера, сколько в целом было завершено задач за некоторый период времени, по результатам которого будут приняты решения о корректировке внутренних процессов – о настройке количества задач с определенным статусом, настройке в целом самого конвейера с целью улучшения показателей эффективности. В рамках второй стадии будет разработана, протестирована и отлажена подсистема протоколирования. Вся кодовая база будет размещена в системе контроля версий, которая будет расположена на сайте GitHub.

## 8. Порядок контроля и приемки системы

Предварительные отчёты по работе будут проводиться во время рубежных аттестаций. Сведения о них приведены в таблице 3.

**Таблица 3 – план отчетов по работам, которые будут выполнены в рамках каждой из аттестаций**

Номер этапа	Название этапа	Временные границы	Сдаваемые документы
1	Аттестация 1	Середина февраля 2025 – конец марта 2025	Готовое техническое задание по проекту
2	Аттестация 2	Начало апреля 2025-конец апреля 2025	Написана большая часть программного кода приложения, проведено тестирование системы.
3	Аттестация 3	Начало мая 2025 года – конец мая 2025	Курсовой проект, готовая система и сопроводительное письмо

## **9. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие**

В требования по подготовке объекта автоматизации входит составление руководства для пользователя системы, следуя которому пользователю должна быть предоставлена возможность развернуть приложения на персональном компьютере, техническое оснащение которого соответствует условиям эксплуатации системы. Руководство должно поставляться вместе с самой автоматизированной системой.

## **10. Требования к документированию**

В данном разделе описана документация, которая должна быть подготовлена в рамках разработки проекта.

### **10.1. Перечень подлежащих разработке документов**

В перечень подлежащих разработке документов входит курсовой проект.

### **10.2. Вид представления и количество документов**

Документы должны быть представлены в электронном виде и опубликованы на сайте GitHub в репозитории команды разработчика, а также в печатном виде.

## **11. Источники разработки**

Источниками разработки являются следующие документы: Федеральный закон от 27.07.2006 N 152-ФЗ «О персональных данных», учебный план по программе бакалавриата по направлению 09.03.04 «Программная инженерия» по специальности для 2022 года начала обучения, распоряжение от 09.01.2025 № 1600-62 «О сроках текущей и промежуточных аттестаций».

## Приложение А. Диаграмма классов

В данном приложении к техническому заданию описана возможная структура классов в разрабатываемой системе. Она приведена на рисунке А.1.

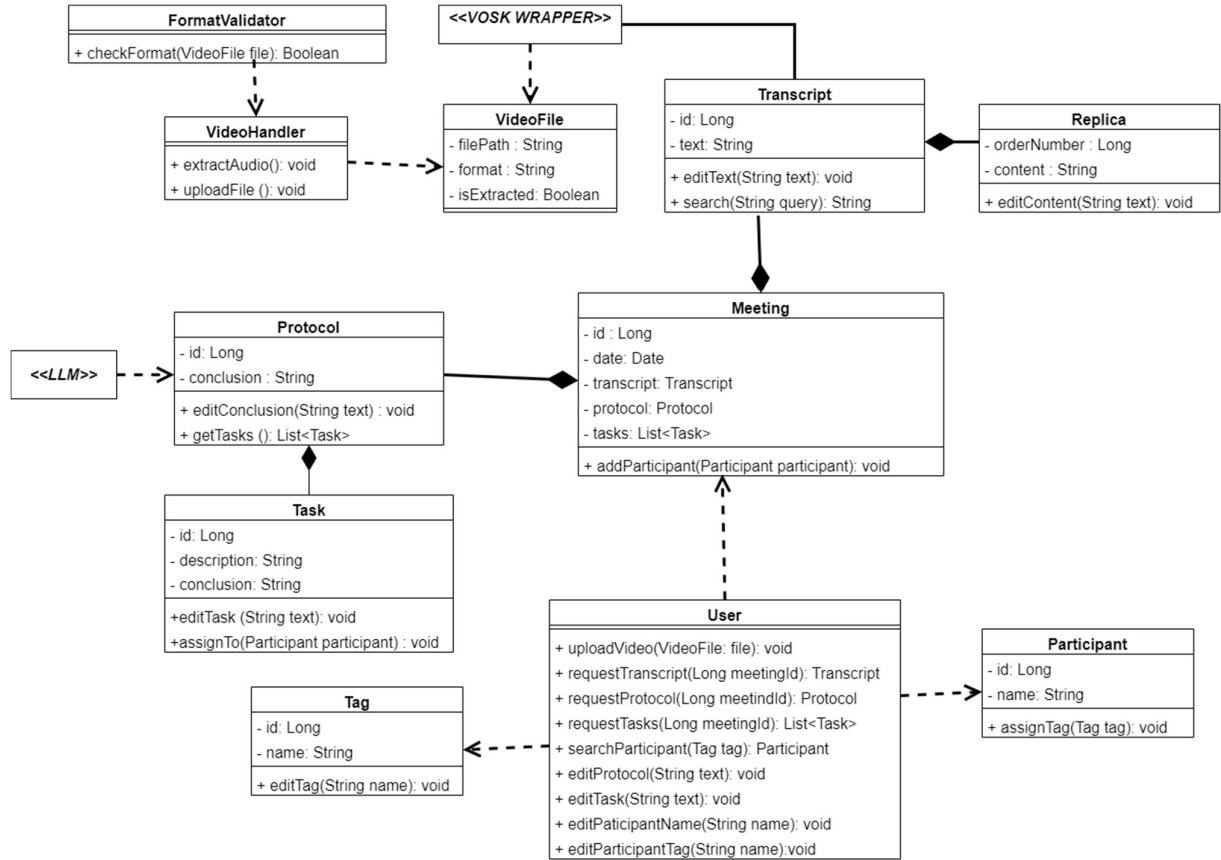


Рисунок А.1 – диаграмма классов

### Классы, их атрибуты и методы:

- User – пользователь

## Методы:

- `uploadVideo(VideoFile file): void` – публичный метод для загрузки видеофайла



- requestTranscript(Long meetingId): Transcript – публичный метод для получения стенограммы встречи
- requestProtocol(Long meetingId): Protocol – публичный метод для получения протокола встречи
- requestTasks(Long meetingId): List<Task> – публичный метод для получения задач встречи
- searchParticipant(Tag tag): Participant – публичный метод для поиска участника встречи по тегу
- editProtocol(String text): void – публичный метод для изменения протокола встречи
- editTask(String text): void – публичный метод для изменения задачи
- editParticipantName(String name) – публичный метод для изменения имени участника встречи
- editParticipantTag(String name) – публичный метод для изменения тега участника встречи

## 2. Meeting – встреча

Атрибуты:

- id: Long – приватный атрибут идентификатора встречи
- date: Date – приватный атрибут даты встречи
- transcript: Transcript – приватный атрибут стенограммы встречи
- protocol: Protocol – приватный атрибут протокола встречи
- tasks: List<Task> - приватный атрибут списка задач встречи

Методы:

- addParticipant(Participant participant): void – публичный метод добавления участника встречи

Tag – тег

Атрибуты:

- id: Long - приватный атрибут идентификатора тега
- name – приватный атрибут названия тега

Методы:

- editTag(String name) : void – публичный метод изменения названия тега

Participant – участник встречи

Атрибуты:

- id: Long - приватный атрибут идентификатора участника
- name – приватный атрибут имени участника встречи

Методы:

- assignTag(Tag tag) : void – публичный метод присвоения тега участнику встречи

5. Protocol – протокол встречи

Атрибуты:

- id: Long - приватный атрибут идентификатора протокола встречи
- conclusion: String – приватный атрибут выводов из встречи

Методы:

- editConclusion(String text) : void – публичный метод изменения итогов встречи
- getTasks() : List<Task> - публичный метод для получения задач

Task – задачи

Атрибуты:

- id: Long - приватный атрибут идентификатора задачи
- description: String – приватный атрибут текста задачи
- conclusion: String – приватный атрибут краткой сводки по задаче

Методы:

- editTask(String text) : void – публичный метод изменения задачи
- assignTo(Participant participant) : void – публичный метод присвоения задачи участнику встречи

<<LLM>> - большая языковая модель

Используется для составления протокола встречи и выделения задач

Transcript – стенограмма

Атрибуты:

- id: Long - приватный атрибут идентификатора стенограммы
- text: String – приватный атрибут текста стенограммы

Методы:

- editText(String text) : void – публичный метод изменения стенограммы
- search(String query) : String – публичный метод поиска по стенограмме

Replica – реплика

Атрибуты:

- orderNumber: Long - приватный атрибут порядкового номера
- content: String – приватный атрибут содержания реплики

Методы:

- editContent(String text) : void – публичный метод изменения содержания реплики

<<VOSK WRAPPER>>

Используется для составления стенограммы встречи и разделения участников по репликам

VideoFile – видеофайл

Атрибуты:

- filePath: String - приватный атрибут пути файла
- format: String – приватный атрибут формата файла
- isExracted: Boolean – приватный атрибут булевого флага для извлечения аудио из файла

VideoHandler – обработчик видео

Методы:

- extractAudio(): void – публичный метод извлечения аудио из видеофайла
- uploadFile(): void – публичный метод загрузки файла

FormatValidator – верификатор формата

Методы:

- checkFormat(VideoFile file): Boolean – публичный метод для проверки формата загруженного видео.

## Приложение Б. Sequence диаграмма

Одним из основных вариантов использования приложения “Встречеслав” является получение протокола встречи, которая была загружена в видеоформате. Сама диаграмма приведена на рисунке Б.1.

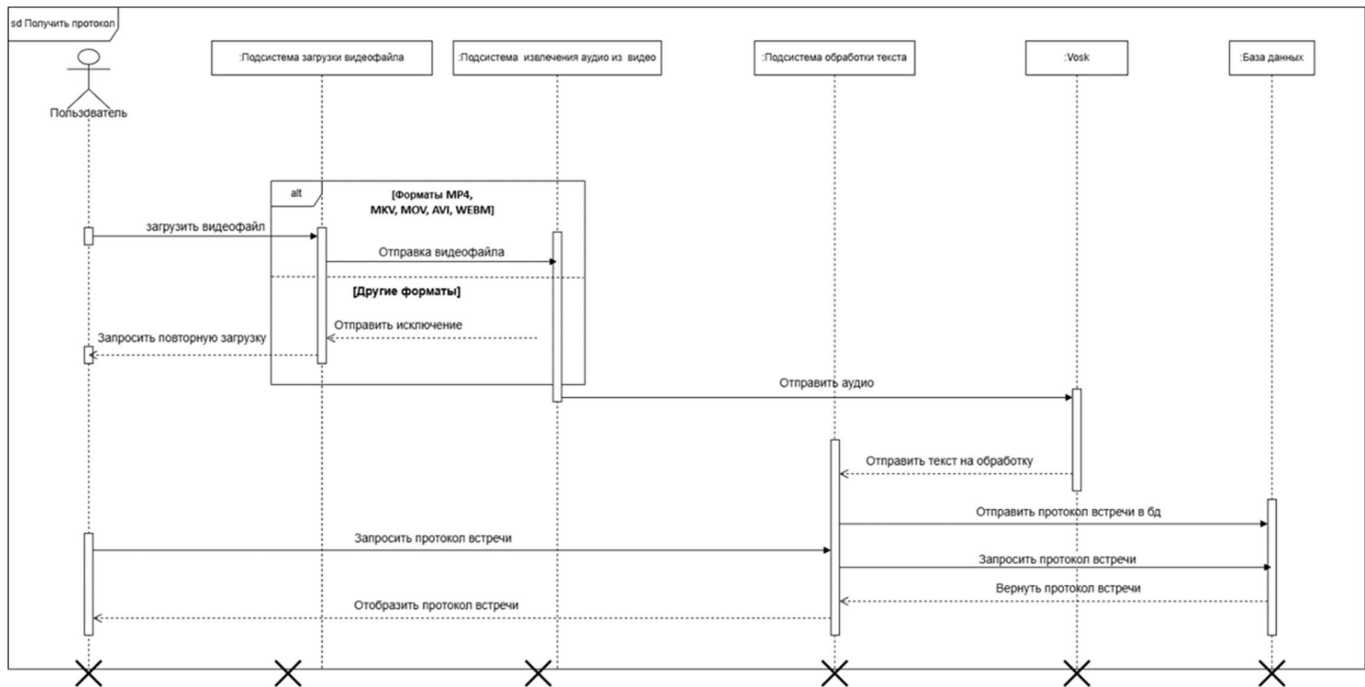


Рисунок Б.1– Sequence диаграмма

Рассматриваем запрос от пользователя на получение протокола.

- Объекты диаграммы последовательности:
- Пользователь;
- Подсистема загрузки видеофайла;
- Подсистема извлечения аудио из видео;
- Подсистема обработки текста;
- Vosk;

— База данных.

Рассмотрим порядок выполнения запроса:

Пользователь загружает видеофайл.

Идёт проверка формата загруженного видеофайла.

Если видеофайл не соответствует форматам: MKV, MOV, AVI, WEBM, то пользователю отправляется запрос на повторную загрузку.

Если видеофайл соответствует форматам: MKV, MOV, AVI, WEBM, то подсистема выделения аудио из видео передаёт аудио, полученное из загруженного видео, для дальнейшей обработки с использованием Vosk.

Vosk формирует дословный текст из аудио и отправляет текст на обработку в подсистему протоколирования.

Подсистема протоколирования формирует протокол встречи и отправляет его в базу данных.

Пользователь запрашивает протокол встречи у подсистемы обработки текста.

Подсистема обработки текста отправляет запрос на получение протокола.

База данных передает протокол встречи в подсистему обработки текста

Подсистема обработки текста отображает протокол встречи для пользователя.

## Приложение В. ER диаграмма базы данных

В приложении предполагается создание базы данных. Далее рассматривается не окончательный макет того, какие могут быть выделены при этом сущности и связи между ними, который может быть изменен в ходе разработки. Сама диаграмма приведена на рисунке В.1.

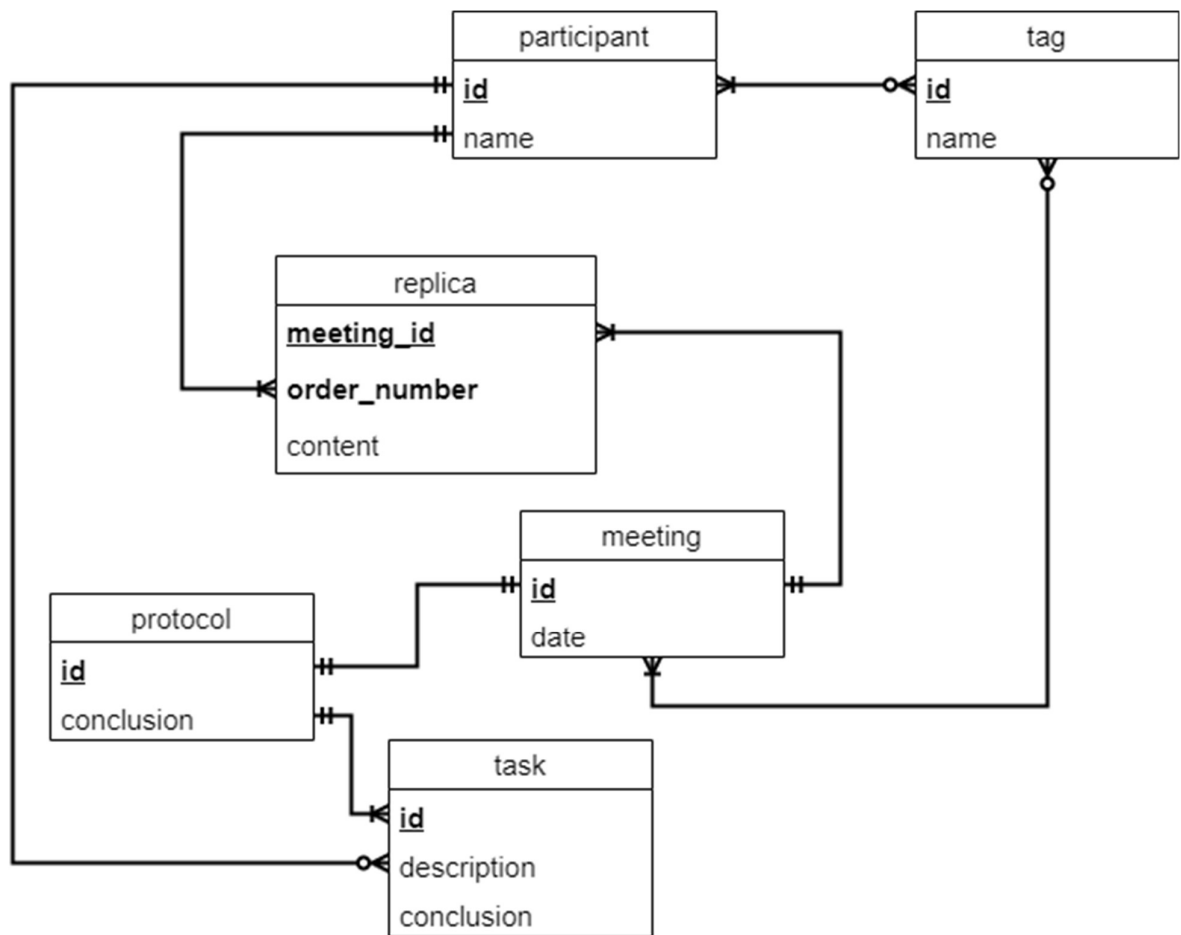


Рисунок В.1 – ER диаграммы базы данных

Сущность meeting (встреча):

- id – уникальный идентификатор встречи;
- date – дата, прошедшей встречи;

Сущность protocol (протокол):

- id – уникальный идентификатор протокола;
- conclusion – расшифровка протокола встречи.

Сущность task (задача):

- id – уникальный идентификатор задачи;
- description – описание задачи;
- conclusion – вывод задачи;

Сущность replica (реплика):

- meeting\_id – уникальный идентификатор встречи;
- order\_number – порядковый номер реплики;
- content – содержание реплики.

Сущность participant (участник встречи):

- id – уникальный идентификатор участника встречи;
- name – имя участника встречи.

Сущность tag (тег):

- id – уникальный номер тега;
- name – названия тега.

### **Типы связей**

Для сущности meeting:

- meeting-replica – один ко многим с обязательной связью;
- meeting-protocol – один к одному с обязательной связью;
- meeting-tag – многие ко многим с обязательной связью.

Для сущности task:



- task-protocol – многие к одному с обязательной связью;
- task-participant – многие к одному с необязательной связью с сущностью task и обязательной связью с сущностью participant.

Для сущности participant:

- participant-replica – один ко многим с обязательной связью;
- participant-tag – многие ко многим с обязательной связью с сущностью participant и необязательной связью с сущностью tag;
- participant-replica – один ко многим с обязательной связью с сущностью participant и необязательной связью с сущностью replica.