

МИНОБРНАУКИ РОССИИ

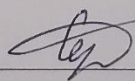
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

КУРСОВАЯ РАБОТА

по дисциплине «Курсовое проектирование»,
«Проектирование программной системы для
формирования стенограмм заседаний Думы города
Владивостока»

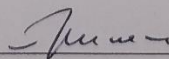
Б-ИН-21-167535.8847-с.17.000. КР

Студент
гр. БИН-21-1



Д. О. Сергеев

Руководитель,
д.т.н., профессор
кафедры ИТС, доцент



В.М. Гриняк

Владивосток 2025

Аннотация

Данная курсовая работа посвящена проектированию программной системы для автоматизации процесса формирования текстовых стенограмм заседаний Думы города Владивостока. Основной целью разработки является упрощение и ускорение процесса преобразования аудиозаписей заседаний в текстовый формат, что позволяет отказаться от ручного труда, минимизировать ошибки и повысить производительность работы.

В работе подробно рассмотрены все этапы проектирования системы: анализ требований и пользователей, разработка функциональности, проектирование интерфейса, структуры базы данных и архитектуры системы. Также описаны основные алгоритмы обработки данных, включая использование нейронной сети для распознавания речи.

В качестве базы данных выбрана MySQL, что обусловлено её надёжностью, удобством интеграции и поддержкой инструментов для визуального проектирования. Проектная архитектура включает разделение на frontend и backend, где для интерфейса используются HTML, CSS и JavaScript, а серверная часть реализуется с использованием Python.

Курсовая работа имеет прикладной характер и закладывает основу для создания реального программного продукта, который способен повысить качество и удобство работы с протоколами заседаний.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВВГУ»)

Институт информационных технологий и анализа данных
Кафедра информационных технологий и систем

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ
на курсовую работу по дисциплине
«Курсовое проектирование»

Студенту гр. БИН-21-01 Сергееву Денису Олеговичу

Задание 1. Описать предметную область.

Задание 2. Описать роли пользователей системы.

Задание 3. Спроектировать интерфейс.

Задание 4. Спроектировать структуру данных в БД.

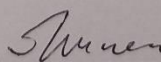
Задание 5. Спроектировать архитектуру системы.

Структура курсовой работы:

- Постановка задачи;
- Описание предметной области;
- Описание ролей пользователей;
- Проектирование функциональности системы;
- Разработка интерфейса;
- Проектирование структуры БД;
- Проектирование архитектуры системы.

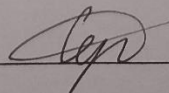
Срок сдачи работы: 16.01.2025 г.

Руководитель,
д.т.н., профессор
кафедры ИТС, доцент



В.М. Гриняк

Задание получил:

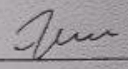
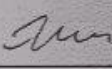


Д. О. Сергеев

КАЛЕНДАРНЫЙ ПЛАН-ГРАФИК
выполнения курсовой работы студента группы БИН-21-1

Студент: Сергеев Денис Олегович

Тема курсовой работы: «Проектирование программной системы для формирования стенограмм заседаний Думы г. Владивостока»

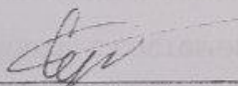
Содержание выполняемых работ по программе	Сроки выполнения		Подпись руководителя
	Начало	Окончание	
Изучение предметной области	11.11.2024	24.11.2024	
Анализ и составление ролей пользователей	24.11.2024	09.12.2024	
Разработка интерфейса	09.12.2024	15.12.2024	
Спроектировать структуру данных в БД	15.12.2024	02.01.2025	
Проектирование архитектуры системы	02.01.2025	11.01.2025	

Согласовано:

Студент

01.12.2024

(Дата)

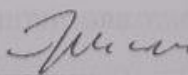

(Подпись)

(Сергеев Денис Олегович)

Руководитель курсовой работы
Д.Т.Н., профессор кафедры
ИТС, доцент

01.12.2024

(Дата)


(Подпись)

(Гриняк Виктор Михайлович)

Содержание

Введение.....	5
1 Описание предметной области.....	6
2 Роли пользователей системы.....	7
2.1 Определение целевых групп пользователей.....	7
2.2 Роли и обязанности пользователей.....	8
2.3 Ожидаемый опыт взаимодействия пользователей с системой.....	12
3 Функциональность системы.....	13
3.1 Основные функции для пользователей.....	13
3.2 Дополнительные функции и их значение.....	14
3.3 Ограничения и предположения.....	15
4 Интерфейс пользователя.....	16
4.1 Принципы проектирования интерфейса.....	16
4.2 Наглядный процесс работы с системой.....	17
4.3 Оценка удобства использования интерфейса.....	19
5 Структура базы данных.....	20
5.1. Обоснование выбора базы данных.....	20
5.2. Описание структуры БД.....	20
5.3. Хранение и обработка результатов работы нейронной сети.....	21
6. Состав программы.....	23
6.1. Общая архитектура системы.....	23
6.2. Диаграммы архитектуры.....	24
6.3. Основные алгоритмы обработки данных.....	24
Заключение.....	26
Список использованных источников.....	27

Введение

Современный мир требует автоматизации многих процессов, особенно тех, которые связаны с обработкой больших объемов данных. Одной из таких задач является создание стенограмм заседаний и совещаний. Традиционно этот процесс выполняется вручную, что требует значительных временных и трудовых затрат, а также человеческих ресурсов. Ручная обработка может быть подвержена ошибкам, что снижает качество конечного результата. Для устранения этих недостатков все чаще применяются автоматизированные системы, которые позволяют существенно повысить точность и скорость работы.

Курсовая работа направлена на разработку программной системы, предназначенной для автоматизации формирования стенограмм заседаний Думы города Владивостока. Основной функционал системы заключается в преобразовании аудиозаписей заседаний в текстовый формат с использованием современных технологий машинного обучения и искусственного интеллекта. Предлагаемая система будет разработана в виде веб-приложения, обеспечивающего удобный и понятный интерфейс для пользователей.

Актуальность разработки обусловлена необходимостью сокращения времени и затрат на обработку аудиозаписей, а также минимизацией человеческого фактора при создании текстовых стенограмм. Система предоставляет возможность загрузки аудиофайлов, их обработки с использованием нейронной сети, а также получения результатов в различных текстовых форматах, таких как DOCX или TXT. Эти функции позволят упростить процесс стенографирования и сделать его более доступным и эффективным.

В рамках работы рассматриваются ключевые этапы разработки системы, включая описание предметной области, анализ целевых пользователей и их требований, проектирование структуры базы данных, создание интерфейса и определение архитектуры системы. Особое внимание уделяется выбору технологии для преобразования речи в текст — нейронной сети Whisper, которая отличается высокой точностью работы и способностью обрабатывать аудиофайлы различного качества.

Разработка программной системы направлена на решение конкретной практической задачи и предполагает последующую интеграцию в рабочий процесс сотрудников Думы города Владивостока. Благодаря автоматизации рутинных процессов система позволит повысить производительность труда, улучшить качество стенограмм и сократить временные издержки.

1 Описание предметной области

Процесс стенографирования заседаний органов власти, таких как парламент, городская Дума, и других государственных учреждений, является важным элементом их деятельности. Стенограмма представляет собой текстовый документ, который фиксирует слово за словом все, что было сказано на заседаниях, включая вопросы, ответы, комментарии и решения. Она служит официальным отчетом о ходе заседания и используется для анализа принятых решений, а также для предоставления информации общественности, средствам массовой информации и заинтересованным сторонам[1].

Традиционно процесс создания стенограмм проводится вручную: сотрудники прослушивают аудиозаписи заседаний и транскрибируют речь в текст. Этот процесс требует значительных временных и трудовых ресурсов, а также высокой точности и внимательности, поскольку ошибки в стенограмме могут исказить информацию и влиять на юридическую значимость документа. В большинстве случаев стенографы сталкиваются с различными проблемами, такими как:

- сложные акценты и диалекты участников заседания;
- быстрая речь;
- наличие шума на аудиозаписях;
- необходимость уточнять или интерпретировать слова участников.

Системы автоматической транскрипции речи в текст могут значительно упростить этот процесс, избавив от необходимости вручную записывать каждое слово. Они позволяют обрабатывать аудиозаписи и преобразовывать их в текст с высокой скоростью, что особенно важно в условиях ограниченного времени. Важным аспектом таких систем является их способность поддерживать точность транскрипции, которая зависит от качества модели машинного обучения, используемой для распознавания речи. Современные технологии, такие как нейронные сети, способны обеспечивать точность, близкую к человеческому восприятию, и обрабатывать аудиофайлы с разной степенью качества.

Основной целью разработки таких систем является автоматизация трудоемкого процесса стенографирования, сокращение времени на подготовку стенограмм и повышение их точности. Важно, чтобы система могла адаптироваться к различным условиям записи и обеспечивать надежность при работе с большими объемами данных.

2 Роли пользователей системы

2.1 Определение целевых групп пользователей

Целевой аудиторией системы являются различные категории пользователей, которые будут взаимодействовать с системой на различных уровнях в зависимости от их задач и роли в процессе подготовки стенограмм заседаний. Ключевыми пользователями являются сотрудники, которые напрямую занимаются подготовкой и обработкой стенограмм на основе аудиофайлов, а также лица, осуществляющие администрирование системы, её техническую поддержку и мониторинг.

Целевые группы пользователей можно разделить на следующие категории:

1. Стенографы и редакторы стенограмм

Это основная группа пользователей, которая будет активно работать с системой. Они занимаются загрузкой аудиофайлов, проверкой и редактированием текстов, созданных системой на основе нейронной сети. Этим пользователям важно обеспечить точность расшифровки речи и возможность быстрого исправления возможных ошибок.

2. Администраторы системы

Это технические специалисты, которые отвечают за бесперебойную работу системы, настройку параметров, управление правами доступа и обновление программного обеспечения. Они обеспечивают корректную работу всех компонентов системы и отслеживают производительность сервера.

3. Руководители и аналитики

Руководители и аналитики используют готовые стенограммы для анализа заседаний, подготовки отчетности и разработки рекомендаций. Эти пользователи получают обработанные текстовые данные, которые должны быть в удобном формате и без ошибок для дальнейшего использования.

4. Разработчики и тестировщики

Эти пользователи занимаются улучшением системы, устранением ошибок и добавлением новых функциональных возможностей. Тестировщики проверяют работу системы на разных этапах её использования, выявляют возможные баги и следят за тем, чтобы система работала корректно.

2.2 Роли и обязанности пользователей

Каждый из типов пользователей системы выполняет специфические задачи и имеет свои обязанности, которые помогают поддерживать систему в рабочем состоянии и обеспечивать ее функциональность. Рассмотрим более подробно обязанности и роль каждого типа пользователя.

1. Стенографы и редакторы стенограмм

Основная роль этой группы пользователей — взаимодействие с системой для обработки аудиофайлов, получения и корректировки текста.

Обязанности:

- загрузка аудиофайлов в систему для их обработки;
- проверка текста, сгенерированного системой, на точность и соответствие исходному аудио;
- редактирование и корректировка полученной стенограммы;
- сохранение и экспорт готовых стенограмм в требуемом формате (например, DOCX, TXT);
- приведение текста в соответствие с корпоративными или законодательными стандартами.

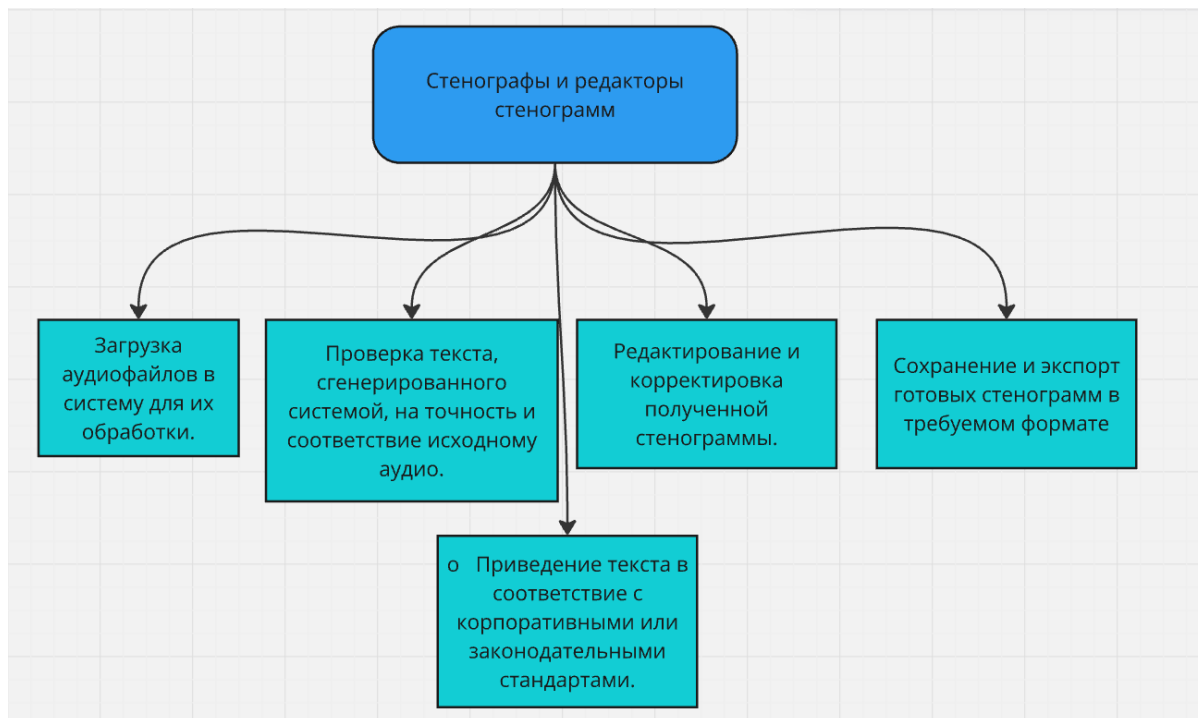


Рисунок 1 – Диаграмма роли стенографов и редакторов стенограмм

Стенографы работают с системой напрямую, взаимодействуя с интерфейсом, загружая файлы и проверяя результат. Их задача — повысить точность распознавания речи и гарантировать, что полученный текст будет соответствовать стандартам качества.(рис.1)

2. Администраторы системы

Администраторы отвечают за настройку и управление всей системой, обеспечивая ее бесперебойную работу. Они имеют доступ к панели управления, где могут настраивать параметры работы системы и контролировать функционирование всех её компонентов.

Обязанности:

- настройка параметров системы и алгоритмов обработки аудио;
- управление правами доступа и ролями пользователей;
- обновление и модернизация программного обеспечения;
- мониторинг работы сервера и базы данных;
- решение технических проблем и обеспечение стабильной работы системы.

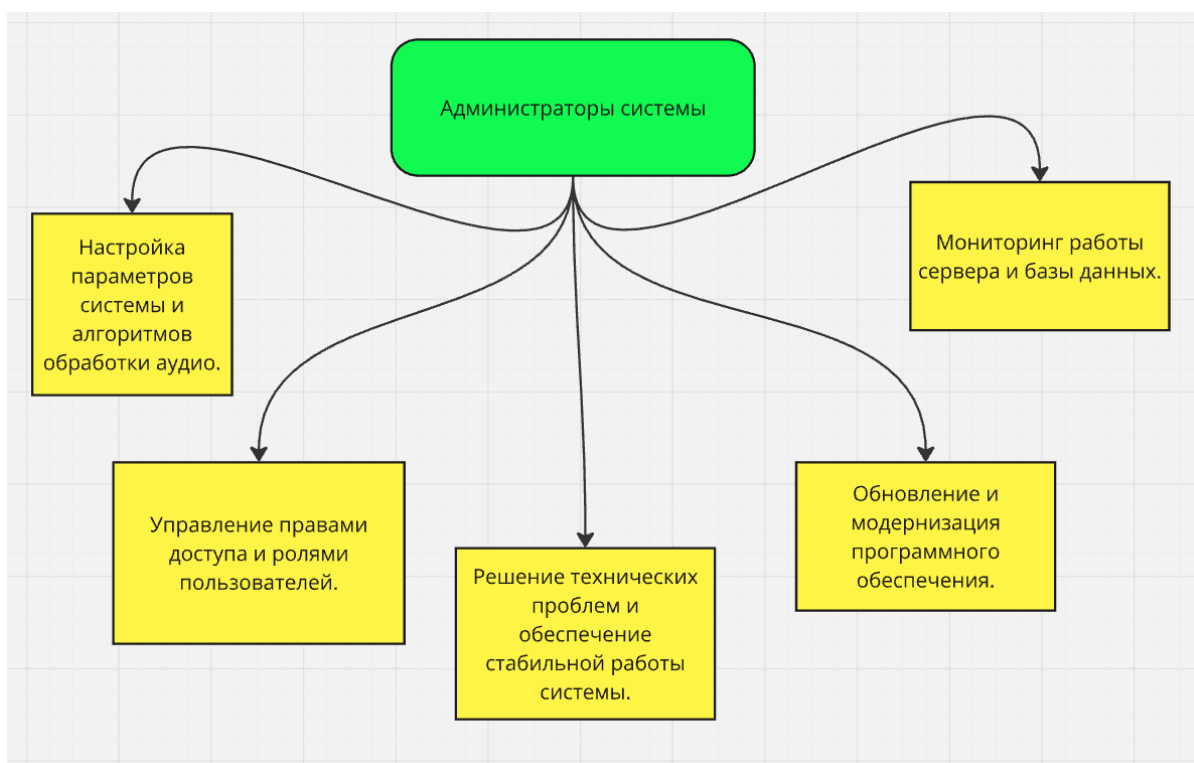


Рисунок 2 – Роли администраторов системы

Администраторы имеют важную роль в поддержке системы, настройке нейронных сетей, управления правами доступа и разрешений пользователей. Эта группа пользователей будет взаимодействовать с системой на техническом уровне, используя административную панель.(рис.2)

3. Руководители и аналитики

Эта категория пользователей ориентирована на анализ готовых результатов работы системы и использование стенограмм для формирования отчетности или принятия управленческих решений.

Обязанности:

- получение готовых стенограмм из системы;

- анализ текстов для выявления важной информации;
- использование стенограмм для подготовки отчетов, аналитических материалов и публичных публикаций.

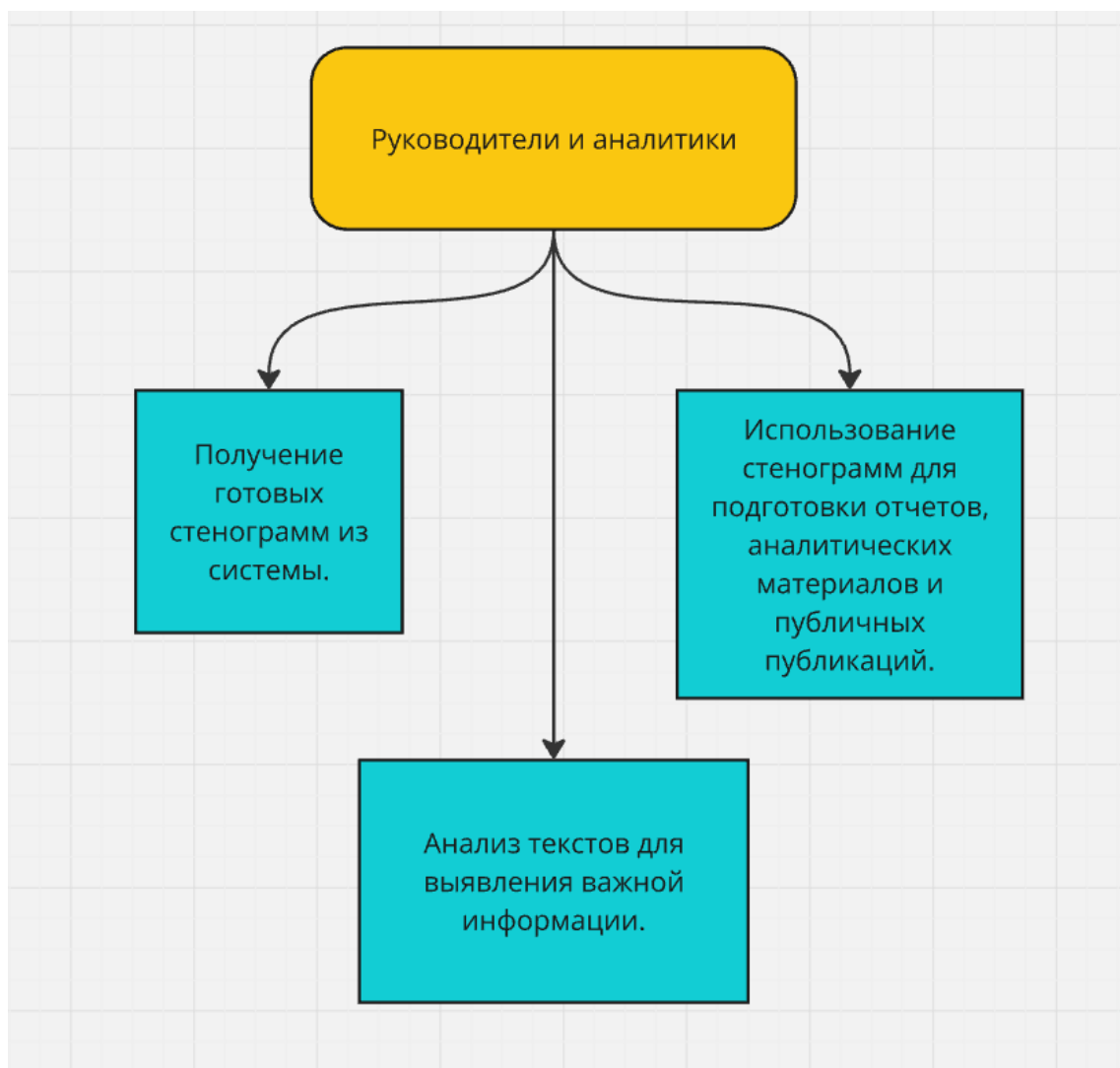


Рисунок 3 – Роли руководителей и аналитиков

Руководители и аналитики редко будут взаимодействовать с системой напрямую, их основная задача — получить готовые текстовые результаты для дальнейшего использования в документах или публичных отчётах.(рис.3)

4. Разработчики и тестировщики

Разработчики отвечают за создание, улучшение и поддержку системы. Тестировщики проводят проверку всех функций системы, выявляют баги и следят за качеством работы.

Обязанности:

- разработка и поддержка системы (фронтенда, бэкенда, базы данных);
- интеграция новых функциональных возможностей;
- тестирование системы на различных этапах её разработки;

- поиск и устранение ошибок.

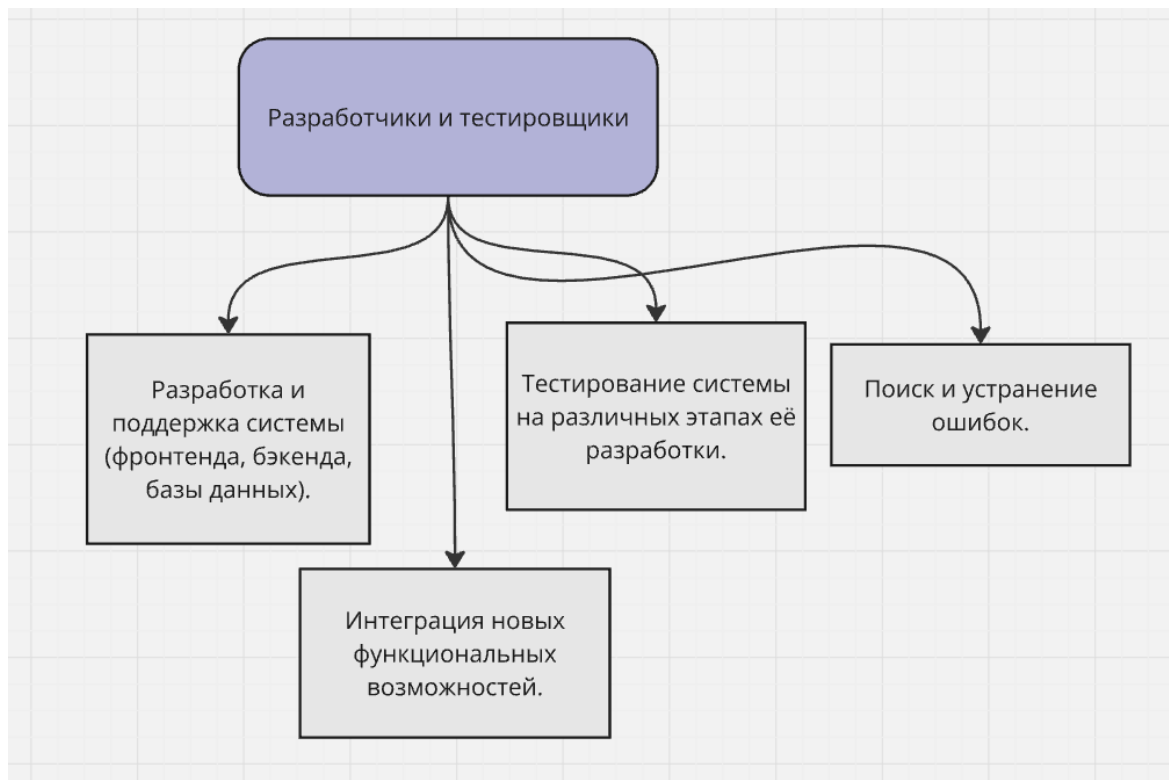


Рисунок 4 – Роли разработчиков и тестировщиков

Для разработчиков и тестировщиков важным аспектом работы будет обеспечение стабильности системы, устранение уязвимостей и улучшение производительности.(рис.4)

2.3 Ожидаемый опыт взаимодействия пользователей с системой

Для каждого типа пользователей важно, чтобы взаимодействие с системой было максимально простым и интуитивно понятным. Ожидаемые пользователи системы будут работать с ней в разных контекстах, что определяет их опыт взаимодействия с системой.

1. Для стенографов и редакторов стенограмм:

Стенографы должны получать возможность без труда загрузить аудиофайл в систему, дождаться обработки и получить текстовый результат, который будет подвержен корректировке. Важным аспектом их взаимодействия будет возможность просматривать текст и вносить правки в случае, если нейронная сеть не смогла корректно распознать отдельные фрагменты речи.

Опыт работы должен быть простым и быстрым: кнопка загрузки, окно предпросмотра и возможность скачать готовый результат. Важно, чтобы стенографы могли работать без лишних шагов и сложных процедур.

2. Для администраторов:

Администраторы будут взаимодействовать с системой на техническом уровне. Их задача — мониторинг работы системы, настройка параметров и управление правами пользователей. Опыт взаимодействия с системой для администраторов будет заключаться в удобной и эффективной панели управления, которая позволяет отслеживать работу нейронной сети, проверять логи и настройки.

3. Для руководителей и аналитиков:

Руководители и аналитики в основном будут работать с уже готовыми стенограммами. Их задача — получить качественные и точные текстовые документы. Ожидается, что для этой категории пользователей процесс будет максимально автоматизирован. После получения стенограмм они смогут легко использовать их в дальнейшей работе. Ключевым моментом является получение готовых материалов с минимальными усилиями.

4. Для разработчиков и тестировщиков:

Разработчики и тестировщики взаимодействуют с системой с целью её улучшения и выявления возможных проблем. Для них важен полный контроль над функциональностью системы и доступ к инструментам тестирования. Это позволит проводить необходимые эксперименты, тестировать различные сценарии работы и устранять ошибки.

3 Функциональность системы

3.1 Основные функции для пользователей

Основная задача системы заключается в преобразовании аудиофайлов с заседаний Думы г. Владивостока в текстовые стенограммы. Весь функционал системы ориентирован на простоту и удобство взаимодействия с пользователем, а также на точность преобразования речи в текст. Рассмотрим основные функции, которые будут доступны пользователям системы:

1. Загрузка аудиофайлов

Одной из первых и наиболее важной функции системы является возможность загрузки аудиофайлов с заседаний. Пользователь может загрузить файл в поддерживаемом формате (например, WAV, MP3, OGG), и система начнёт процесс его обработки. Это позволяет автоматически перенести аудиозапись в текстовый формат без необходимости вручную расшифровывать каждое слово.

2. Выбор формата вывода

После того как аудиофайл был обработан системой, пользователь может выбрать нужный формат для получения готовой стенограммы. На данный момент поддерживаются текстовые форматы, такие как DOCX и TXT. Это позволяет пользователю выбирать удобный для дальнейшей работы формат и получать готовый текст, который можно использовать для создания отчётов или официальных документов.

3. Обработка аудиофайлов с помощью нейронной сети

Важнейшей частью функционала является использование нейронной сети для распознавания речи. Эта модель, интегрированная в бэкенд системы, автоматически обрабатывает загруженные аудиофайлы и преобразует их в текст. При этом система должна учитывать различные акценты, шумы в записи и технические особенности речи, чтобы достичь наибольшей точности распознавания[1].

4. Предпросмотр стенограммы

После обработки аудиофайла пользователю предоставляется возможность просмотреть стенограмму в специальном окне предпросмотра. Это позволяет быстро убедиться в качестве преобразования речи в текст и оценить, насколько точно система справилась с задачей.

5. Скачивание готовой стенограммы

После того как стенограмма готова и пользователь убедился в её качестве, можно скачать файл на своё устройство. Это завершает процесс обработки аудиофайла и позволяет

использовать готовый текст в дальнейшем. Возможность скачать результат в различных форматах облегчает дальнейшую работу и обеспечивает гибкость.

Эти функции позволяют пользователю взаимодействовать с системой с минимальными усилиями, что особенно важно в профессиональной среде, где время имеет значение. Каждая функция ориентирована на повышение эффективности и удобства работы с системой, обеспечивая при этом максимальную точность и функциональность.

3.2 Дополнительные функции и их значение

В будущем планируется расширить функциональность с целью улучшения работы системы и удовлетворения дополнительных потребностей пользователей. Некоторые из этих функций могут быть добавлены в следующих версиях системы:

1. Распознавание акторов (определение говорящего)

Одной из планируемых функций является добавление возможности нейронной сети различать участников заседания и определять, кто из них в данный момент говорит. Это позволит системе автоматически помечать текст в стенограмме в зависимости от того, кто произносит реплику. Подобная функция значительно упростит дальнейшее использование стенограммы, так как пользователи смогут легко идентифицировать каждого участника беседы без необходимости вручную помечать их реплики. Данная функция значительно повысит точность и полезность итогового документа;

2. Поддержка различных языков и диалектов

В будущем можно добавить поддержку нескольких языков и диалектов для улучшения точности распознавания речи, особенно если заседания проводятся с участием людей, говорящих на разных языках. Это позволит системе обслуживать более широкую аудиторию и повысить её универсальность;

3. Редактирование стенограммы

В будущем возможно добавление функции редактирования текста непосредственно в интерфейсе системы. Это даст пользователям возможность вносить коррективы в уже обработанную стенограмму без необходимости переходить в сторонние текстовые редакторы. Такой подход улучшит удобство работы и ускорит процесс окончательной обработки материалов;

4. Многозадачность при обработке нескольких файлов

В будущих версиях системы также планируется возможность параллельной обработки нескольких аудиофайлов. Это значительно повысит производительность, особенно в случае, если необходимо обработать большое количество аудиозаписей, что может быть полезным для тех пользователей, которые работают с большим объёмом информации.

Все эти функции могут постепенно внедряться и тестироваться на разных этапах развития системы, чтобы улучшить её функциональность и обеспечить пользователю более широкие возможности для работы с аудиофайлами и текстовыми стенограммами.

3.3 Ограничения и предположения

Как и любая технологическая система, эта разработка имеет свои ограничения и предположения, которые необходимо учитывать при эксплуатации и дальнейшей доработке.

1. Ограничения точности распознавания речи

Несмотря на использование мощной нейронной сети, система может сталкиваться с проблемами точности распознавания речи в условиях сильных фоновых шумов, акцентов, быстрого или нечеткого произнесения слов. Это может привести к небольшим ошибкам в тексте, которые потребуют вмешательства пользователей для корректировки.

2. Поддержка форматов файлов

Система ограничена определённым набором форматов для загрузки аудиофайлов (например, MP3, WAV, OGG). Это может ограничить возможности пользователей, если они имеют записи в других, менее распространённых форматах. В будущем можно будет расширить поддержку дополнительных форматов, но на данный момент такие ограничения неизбежны.

3. Зависимость от качества исходных аудиофайлов

Качество распознавания речи напрямую зависит от качества самих аудиофайлов. Плохое качество записи, присутствие посторонних шумов, перекрытие речи разных людей и прочие технические особенности могут снизить точность преобразования речи в текст.

4. Ресурсоёмкость системы

Использование нейронных сетей для обработки аудиофайлов требует значительных вычислительных ресурсов, что может ограничивать возможности системы для обработки большого объёма данных за короткое время. Важно учитывать, что при большом количестве пользователей или при необходимости обрабатывать большое количество файлов одновременно, потребуется более мощное оборудование или распределение нагрузки.

Таким образом, ограничения системы в основном связаны с техническими аспектами обработки данных и качеством исходных материалов. Однако, с учетом постоянных улучшений и обновлений, эти ограничения могут быть уменьшены или устранены в будущих версиях системы.

4 Интерфейс пользователя

4.1 Принципы проектирования интерфейса

Проектирование интерфейса играет ключевую роль в разработке программных систем, особенно если их целевая аудитория состоит из пользователей с разным уровнем технической подготовки. Интерфейс системы для формирования стенограмм заседаний Думы Владивостока ориентирован на максимальную простоту и удобство, чтобы сотрудники могли эффективно использовать его без необходимости длительного обучения. [2]

Для создания комфортной рабочей среды были выбраны синие тона, которые традиционно ассоциируются с профессионализмом, стабильностью и доверием. Однако, чтобы придать интерфейсу уникальность и технологическую глубину, в качестве фонового изображения было использовано изображение, сгенерированное с помощью нейронной сети. Это решение позволяет избежать рисков, связанных с нарушением авторских прав, и подчеркивает инновационную природу проекта[3].

Ключевыми принципами проектирования интерфейса стали:

- Минимализм: исключение лишних элементов, чтобы избежать перегрузки экрана;
- Интуитивная навигация: расположение всех функциональных элементов так, чтобы пользователь мог легко найти нужный инструмент;
- Доступность: учёт потребностей пользователей с ограниченной технической подготовкой.

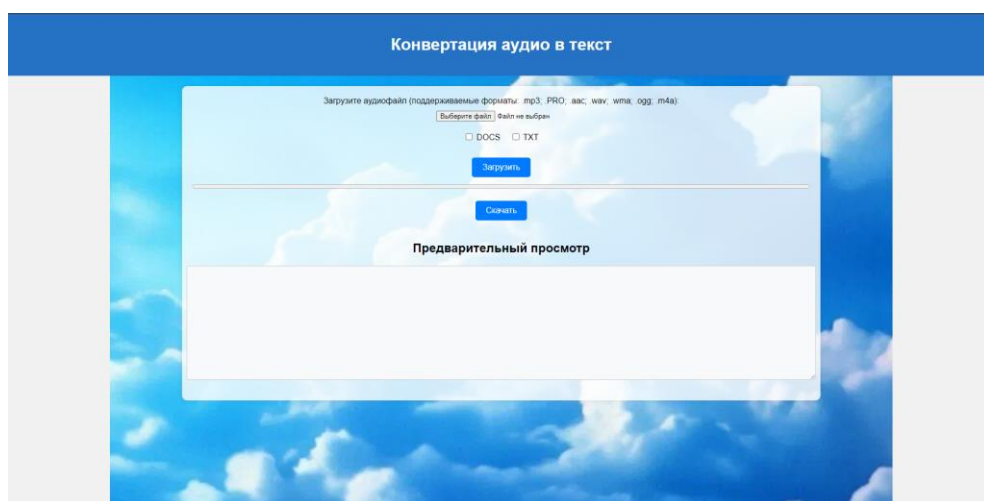


Рисунок 5 – Прототип интерфейса

Для проверки концепции был создан прототип интерфейса, где ключевые элементы расположены в логическом порядке, что помогает пользователям быстро освоиться с системой (рис. 5).

4.2 Наглядный процесс работы с системой

На текущем этапе проектирования разработан функциональный прототип фронтенда, включающий все основные элементы интерфейса:

- Окно загрузки файла: пользователь может загрузить аудиофайл заседания в поддерживаемом формате (рис. 6);

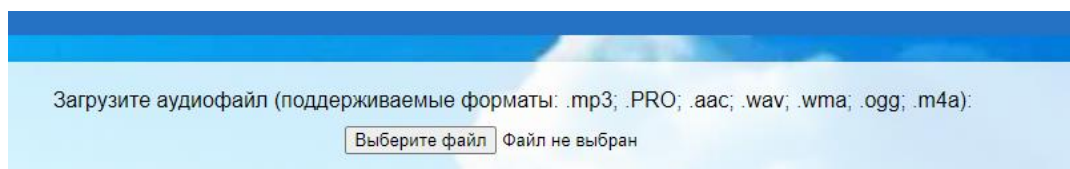


Рисунок 6 – Окно загрузки аудиофайла

- Выбор формата: пользователь выбирает, в каком формате (.docx или .txt) он хочет получить текстовый результат (рис. 7);

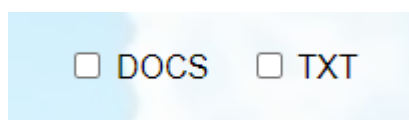


Рисунок 7 – Выбор формата

- Кнопки управления (рис. 8):
- Загрузить: для запуска процесса обработки аудиозаписи;
- Скачать: для получения готового документа.

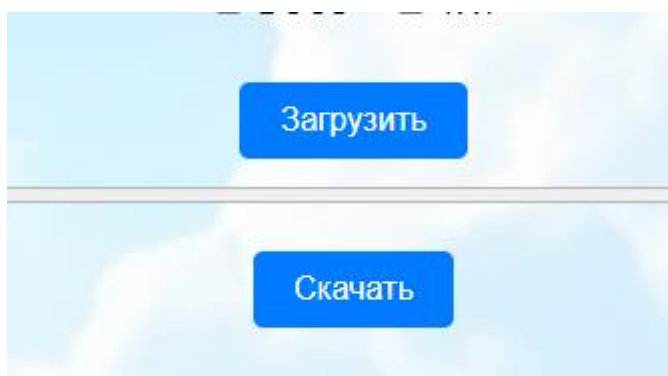


Рисунок 8 – Кнопки запуска обработки файла и скачивания готового результата

- Окно предпросмотра текста: здесь пользователь может увидеть предварительный результат преобразования речи в текст перед его окончательной загрузкой (рис. 9).

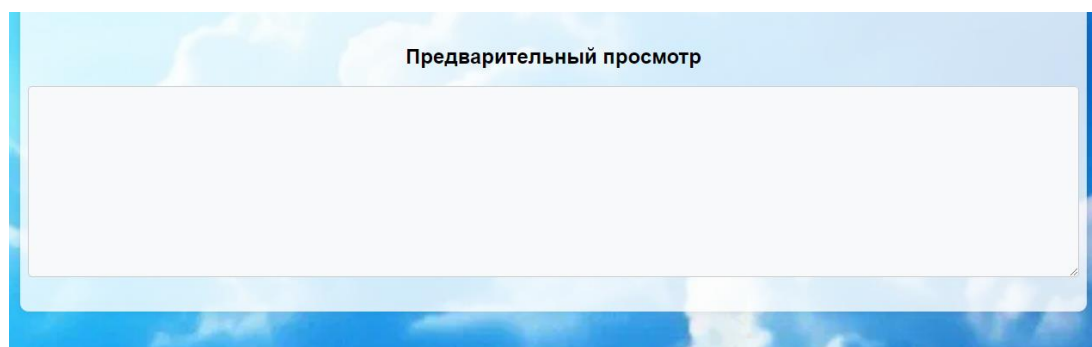


Рисунок 9 – Окно предпросмотра содержимого обработанного файла

Процесс взаимодействия с системой может быть описан следующим образом:

1. Пользователь заходит на веб-страницу и видит окно загрузки файла;
2. Загружает аудиозапись заседания в одном из поддерживаемых форматов;
3. Выбирает желаемый формат выходного документа;
4. Запускает обработку, после чего система отображает статус выполнения задачи;
5. После завершения обработки пользователь видит текстовый результат в окне предпросмотра;
6. Пользователь скачивает обработанный файл.

Разработанный функциональный прототип демонстрирует интуитивно понятный и удобный интерфейс пользователя. Простота использования и наличие окна предпросмотра обеспечивают высокую степень комфорта и контроля над процессом транскрипции. Дальнейшая работа будет направлена на оптимизацию производительности и расширение функциональности системы, включая поддержку дополнительных аудиоформатов и возможностей редактирования полученного текста.

4.3 Оценка удобства использования интерфейса

На этапе проектирования интерфейса были учтены основные требования к удобству использования:

1. Простота и интуитивность: элементы интерфейса расположены логично, что позволяет легко понять процесс работы даже без инструкций;
2. Минимальное количество шагов: взаимодействие с системой организовано таким образом, чтобы пользователь мог быстро достичь цели (получить стенограмму) с минимальным количеством действий;
3. Гибкость: наличие окна предпросмотра позволяет проверить результат до его сохранения, что особенно важно при работе с преобразованием аудиофайлов в текст.

Интерфейс разработан с учётом возможного расширения функционала (рис. 6). Например, в будущем может быть добавлена функция распознавания акторов, чтобы система могла автоматически определять, кто говорит в данный момент, что повысит точность и информативность стенограмм.

Кроме того, использование сгенерированного фонового изображения подчёркивает внимание к деталям и инновационный подход к созданию визуальной составляющей. Интерфейс, несмотря на свою простоту, демонстрирует высокие стандарты качества, что делает его важным элементом разрабатываемой системы.(рис.10)



Рисунок 10 – Сгенерированное фоновое изображение

Таким образом, разработанный интерфейс отвечает принципам простоты, эффективности и гибкости, обеспечивая комфортное взаимодействие пользователя с системой. Встроенные возможности расширения гарантируют адаптацию к будущим потребностям и развитию функциональности.

5 Структура базы данных

5.1. Обоснование выбора базы данных

В рамках проектирования программной системы формирования стенограмм заседаний Думы Владивостока был произведён выбор системы управления базами данных (СУБД). После анализа доступных решений было принято решение использовать MySQL. Этот выбор обусловлен рядом факторов, которые делают данную СУБД наиболее подходящей для целей и задач проекта. [4]

Во-первых, MySQL широко известна и активно используется в разработке программного обеспечения, благодаря чему она обладает большим количеством документации, примеров и готовых решений. Это делает процесс проектирования и работы с системой более прозрачным и предсказуемым.

Во-вторых, MySQL Workbench предоставляет удобный графический интерфейс для работы с базой данных[5]. С его помощью можно визуализировать структуру данных, проектировать таблицы, управлять связями между ними и проводить анализ производительности. Это особенно полезно на этапе проектирования, когда структура базы данных часто пересматривается и уточняется.

Кроме того, MySQL демонстрирует высокую производительность при работе с большими объемами данных. Учитывая, что система будет обрабатывать множество аудиофайлов и сопоставлять их с текстовыми стенограммами, критически важно, чтобы база данных была способна быстро обрабатывать запросы[6].

Еще одним преимуществом является гибкость MySQL, которая обеспечивает поддержку различных типов данных и позволяет легко интегрировать базу данных с другими компонентами системы, такими как серверная часть и интерфейс пользователя.

Наконец, использование MySQL на этапе проектирования и начального развертывания системы не требует существенных финансовых затрат благодаря модели open-source, что делает её идеальным выбором для текущего этапа проекта.

5.2. Описание структуры БД

На этапе проектирования предполагается минималистичный подход к структуре базы данных, включающий только две основные таблицы:

1. Таблица AudioFiles

- Назначение: хранение информации о загруженных аудиофайлах.

Основные поля:

- file_id (идентификатор файла, первичный ключ);

- file_name (имя файла);
- uploaded_by (идентификатор пользователя, загрузившего файл);
- upload_date (дата и время загрузки файла);
- file_path (путь к физическому расположению файла на сервере).

2. Таблица Transcripts

- Назначение: хранение результатов обработки загруженных файлов.

Основные поля:

- transcript_id (идентификатор записи, первичный ключ);
- file_id (внешний ключ, связывающий запись с таблицей AudioFiles);
- transcript_text (распознанный текст, полученный в результате обработки);
- processing_date (дата и время создания стенограммы);
- format (выбранный формат сохранения документа, например .txt или .docx);
- accuracy (процент точности работы нейронной сети, если применимо).

Эти две таблицы обеспечивают базовую функциональность системы, включая загрузку аудиофайлов и сохранение результатов их обработки[7].

Данный подход выбран для упрощения разработки и ускорения реализации. Две таблицы обеспечивают:

- прямую связь между исходными файлами и результатами их обработки;
- легкость в модификации структуры, если в будущем появится необходимость добавления новых функций или таблиц;
- удобство для интеграции с пользовательским интерфейсом, где необходимо быстро отображать загруженные файлы и результаты их обработки.

По мере развития проекта структура базы данных может быть дополнена для поддержки новых возможностей, таких как управление пользователями, сохранение настроек обработки или архивирование данных.

5.3. Хранение и обработка результатов работы нейронной сети

Результаты работы системы, основанной на нейронной сети, представляют собой текстовые данные, извлеченные из аудиофайлов[8]. Эти данные включают:

1. Текстовые стенограммы: результаты преобразования аудиофайлов в текст, которые являются ключевым продуктом системы;
2. Метаданные стенограмм: информация о времени обработки, точности результатов и параметрах распознавания.

Этапы работы с результатами:

1. Сохранение исходных данных

- После загрузки аудиофайл сохраняется в таблице AudioFiles. На сервере он обрабатывается нейронной сетью.
- 2. Создание текстовых стенограмм
 - Нейронная сеть обрабатывает аудиофайл и создает текст. Этот текст вместе с метаданными сохраняется в таблице Transcripts.
- 3. Интеграция с пользовательским интерфейсом
 - Пользователи могут просматривать предварительные результаты обработки стенограммы через интерфейс до ее сохранения или загрузки.
- 4. Архивирование данных
 - Для оптимизации работы и предотвращения избыточной загрузки серверных ресурсов старые данные могут архивироваться.

В итоге, система обеспечивает надежное хранение и обработку результатов работы нейронной сети, включая текстовые стенограммы и метаданные, обеспечивая целостность данных и эффективное управление хранилищем. Процесс обработки и сохранения данных организован таким образом, чтобы гарантировать удобство использования и производительность системы.

6. Состав программы

6.1. Общая архитектура системы

Для реализации веб-сервиса по преобразованию аудиофайлов в текстовый формат будет использована многослойная архитектура, разделенная на несколько ключевых компонентов: frontend, backend и база данных[9].

1. Frontend (HTML, CSS, JavaScript):

Этот слой будет отвечать за взаимодействие пользователя с системой. На фронтенде будут реализованы все основные элементы интерфейса, такие как форма для загрузки файлов, выбор формата сохранения результата, кнопка загрузки и предпросмотра стенограммы. Фронтенд будет работать в браузере пользователя, обеспечивая интуитивно понятный и удобный интерфейс для работы с системой[10].

- для стилизации интерфейса будет использован CSS, включая использование сгенерированного нейронной сетью фона в синих оттенках для поддержания уникальности дизайна[11];

- JavaScript будет использоваться для динамического обновления страницы и асинхронных операций, таких как загрузка файлов и отображение прогресса их обработки.

2. Backend (Python, JavaScript):

Серверная часть системы будет работать на Python с использованием популярных фреймворков и библиотек для обработки аудиофайлов и работы с нейронной сетью. На серверной стороне будет обрабатываться логика обработки аудиофайлов с помощью модели нейронной сети для распознавания речи и генерации текстовых стенограмм[12]. Также сервер будет взаимодействовать с базой данных, обеспечивая хранение загруженных файлов и обработанных текстов.

- Python будет использоваться для обработки аудиофайлов, интеграции с нейронной сетью и выполнения алгоритмов распознавания речи[13];

- JavaScript будет применяться для асинхронных операций на сервере, например, обработки запросов от клиента и взаимодействия с фронтендом через API[14].

3. База данных (MySQL):

База данных будет использоваться для хранения метаданных аудиофайлов и результатов их обработки. Это позволит систематизировать данные и обеспечить эффективное управление файлами и стенограммами[15].

Эта архитектура обеспечит модульность системы, где каждый компонент выполняет свою задачу, что позволит легко расширять функциональность системы в будущем[16].

6.2. Диаграммы архитектуры

Для наглядности и лучшего понимания структуры системы можно представить диаграмму, которая будет включать следующие компоненты (рис. 7):

1. Frontend — компонент, с которым взаимодействует пользователь. Пользователь загружает аудиофайлы, выбирает формат и скачивает готовую стенограмму;
2. Backend — серверная часть, которая обрабатывает аудиофайлы с помощью нейронной сети, взаимодействует с базой данных и отправляет результат обратно на frontend[17];
3. База данных — хранит информацию о загруженных файлах и обработанных текстах;
4. Модель нейронной сети — выполняет преобразование аудиофайлов в текст с использованием алгоритмов распознавания речи.

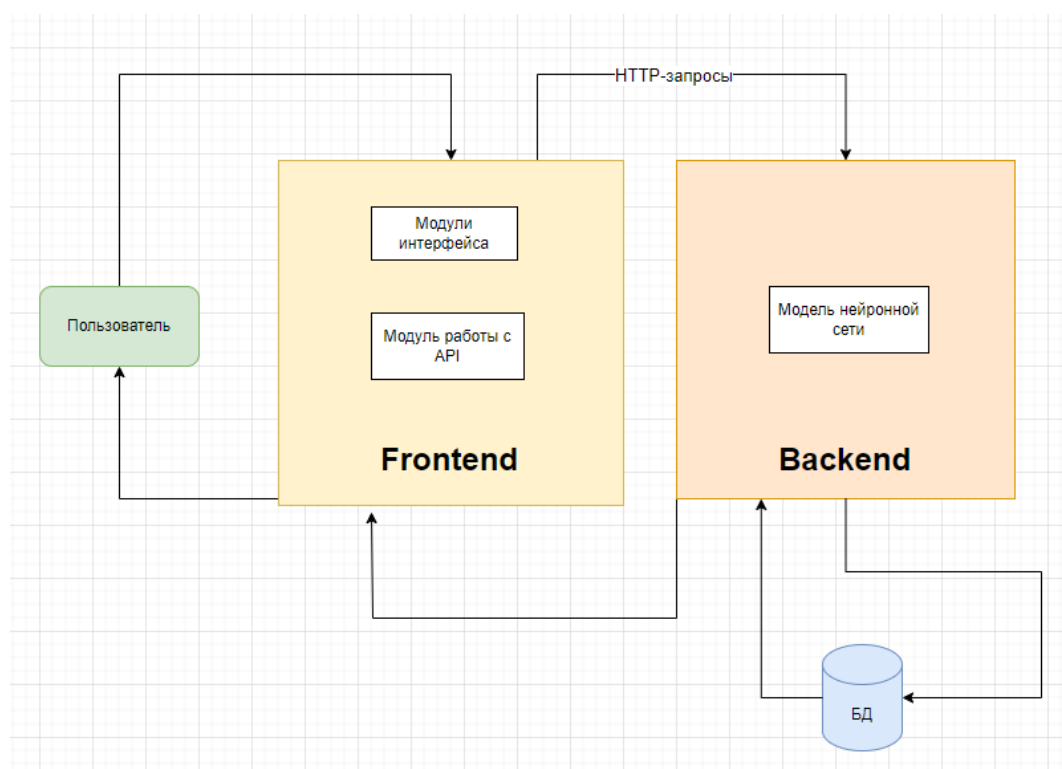


Рисунок 7 – Диаграмма архитектуры

Представленная архитектура системы демонстрирует четкое разделение компонентов, обеспечивая эффективную обработку данных и удобный пользовательский интерфейс.

6.3. Основные алгоритмы обработки данных

Основным алгоритмом обработки данных будет использование нейронной сети для распознавания речи. На этом этапе предполагается использование модели Whisper, которая будет преобразовывать аудиофайлы в текст[18].

Алгоритм обработки данных можно разделить на несколько шагов:

1. Загрузка аудиофайла:

- Пользователь загружает аудиофайл через интерфейс системы;
- Backend получает файл, сохраняет его в соответствующем месте и начинает его обработку.

2. Обработка аудиофайла:

- С помощью библиотеки Python и нейронной сети Whisper происходит преобразование аудиофайла в текст;
- Алгоритм будет включать предобработку аудиофайла (например, удаление шума), если это необходимо для повышения точности распознавания.

3. Преобразование в текст:

- Модель Whisper анализирует аудиофайл, распознает речь и преобразует её в текстовый формат;
- Результат сохраняется в базе данных, а также становится доступным для пользователя в выбранном формате (.txt, .docx и т. д.).

4. Отправка результата пользователю:

- После обработки текста пользователь может скачать стенограмму в выбранном формате;
- Результат сохраняется в базе данных с возможностью его дальнейшего использования или архивирования.

Этот процесс может быть улучшен и дополнен, например, добавлением функционала для определения акторов (кто из участников говорит в данный момент), что будет разрабатываться на более поздних этапах проекта.

Заключение

В ходе выполнения данной курсовой работы был разработан проект программной системы, предназначенной для автоматизации процесса формирования текстовых стенограмм заседаний Думы города Владивостока. В рамках работы было выполнено следующее:

1. Изучены и описаны требования к системе, а также целевые группы пользователей, их роли и обязанности, что позволило определить основные задачи, которые система должна решать;
2. Проектирован функционал системы с учётом потребностей пользователей и возможных ограничений, а также продуманы дополнительные функции, которые могут быть реализованы в будущем для повышения эффективности работы системы;
3. Спроектирован пользовательский интерфейс с учётом принципов удобства и минимализма, чтобы обеспечить интуитивно понятное взаимодействие с системой;
4. Разработана структура базы данных, обеспечивающая хранение загруженных аудиофайлов и текстовых стенограмм, а также обоснован выбор технологии MySQL для работы с данными;
5. Определена общая архитектура системы, которая разделяет frontend и backend для обеспечения гибкости и масштабируемости. Также были описаны основные алгоритмы обработки данных, включающие использование нейронной сети для преобразования речи в текст.

Разработка данной системы имеет значительный потенциал для автоматизации рутинных задач по созданию стенограмм, что может существенно повысить точность, оперативность и удобство работы с протоколами заседаний. Несмотря на проектный характер курсовой работы, реализация предложенных решений способна стать основой для создания реального программного продукта.

Данная работа показала важность междисциплинарного подхода, сочетающего элементы программирования, анализа данных, проектирования интерфейсов и взаимодействия с базами данных. В будущем проект может быть доработан, включая внедрение дополнительных функций, таких как распознавание акторов, повышение точности обработки речи и интеграция с внешними сервисами для расширения функционала.

Список использованных источников

1. Методы и подходы к распознаванию речи в нейронных сетях // Хабр. - URL: <https://habr.com/ru/articles/567849/> (дата обращения: 01.12.2025).
2. Особенности проектирования пользовательских интерфейсов для веб-приложений // UX Planet. - URL: <https://uxplanet.org/designing-web-apps-ui/> (дата обращения: 01.12.2025).
3. Современные инструменты разработки фронтенда // Хабр. - URL: <https://habr.com/ru/articles/565710/> (дата обращения: 01.12.2025).
4. Сравнение популярных технологий для веб-разработки // Tproger. - URL: <https://tproger.ru/articles/web-dev-tools-comparison/> (дата обращения: 01.12.2025).
5. Как выбрать базу данных для вашего проекта // Tproger. - URL: <https://tproger.ru/articles/db-selection-guide/> (дата обращения: 01.12.2025).
6. Преимущества MySQL для работы с большими данными // MySQL.com. - URL: <https://dev.mysql.com/doc/refman/8.0/en/> (дата обращения: 01.12.2025).
7. Whisper: современная нейронная сеть для обработки речи // Towards Data Science. - URL: <https://towardsdatascience.com/whisper-ai-speech-recognition/> (дата обращения: 01.12.2025).
8. Обзор технологий Python для веб-разработки // Python.org. - URL: <https://python.org/web/> (дата обращения: 01.12.2025).
9. Использование JavaScript для интерактивности веб-приложений // MDN Web Docs. - URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата обращения: 01.12.2025).
10. Введение в HTML и CSS для проектирования веб-интерфейсов // W3Schools. - URL: <https://www.w3schools.com/html/> (дата обращения: 01.12.2025).
11. Принципы разработки RESTful API для веб-приложений // Хабр. - URL: <https://habr.com/ru/articles/447202/> (дата обращения: 01.12.2025).
12. Почему важны прототипы в проектировании интерфейсов // UX Collective. - URL: <https://uxdesign.cc/prototyping-in-ui-design/> (дата обращения: 01.12.2025).
13. Выбор СУБД: шпаргалка, чтобы не запутаться // Хабр. - URL: <https://habr.com/ru/articles/775574/> (дата обращения: 01.12.2025).
14. Интеграция нейронных сетей в веб-приложения: лучшие практики // Towards AI. - URL: <https://towardsai.net/deploying-ai-models> (дата обращения: 01.12.2025).
15. Алгоритмы обработки данных в системах искусственного интеллекта // Tproger. - URL: <https://tproger.ru/articles/data-processing-algorithms/> (дата обращения: 01.12.2025).
16. Основы тестирования нейронных сетей // Хабр. - URL: <https://habr.com/ru/articles/636772/> (дата обращения: 01.12.2025).