

Технический проект: Программное обеспечение для анализа и обработки речевых сигналов

Настоящий документ описывает и обосновывает архитектурные, технологические и проектные решения для создания гибридного настольного программного обеспечения (ПО) для анализа и обработки речевых сигналов.

Система предназначена для исследователей в области лингвистики, фонетики и смежных дисциплин. Она состоит из двух ключевых частей:

1. Клиентская часть: десктоп приложение на C++, предоставляющее пользовательский интерфейс (UI) для управления, визуализации и взаимодействия с данными.
2. Вычислительное ядро (Бэкенд): Локальный сервер на Python, использующий возможности библиотеки SpeechBrain и предварительно обученной нейронной сети, развернутой через Hugging Face, для выполнения наукоемких задач анализа.

Функциональная структура продукта

Функциональная структура системы отражает её модульное устройство и логику взаимодействия компонентов. Продукт включает основные блоки:

- Управление проектами и файлами,
- Предобработка аудио (фильтрация, шумоподавление),
- Анализ и извлечение признаков,
- Визуализация результатов,
- Экспорт и сохранение данных.

Такое разделение обеспечивает ясную логику, независимость компонентов и возможность масштабирования. Для описания структуры применяются диаграммы IDEF0 и DFD, демонстрирующие потоки данных между подсистемами.

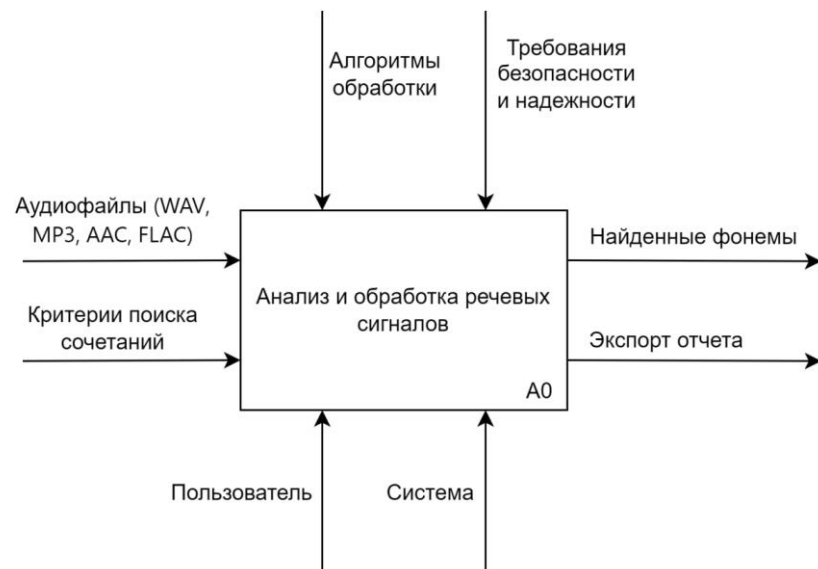


Рисунок 1 – IDEF0

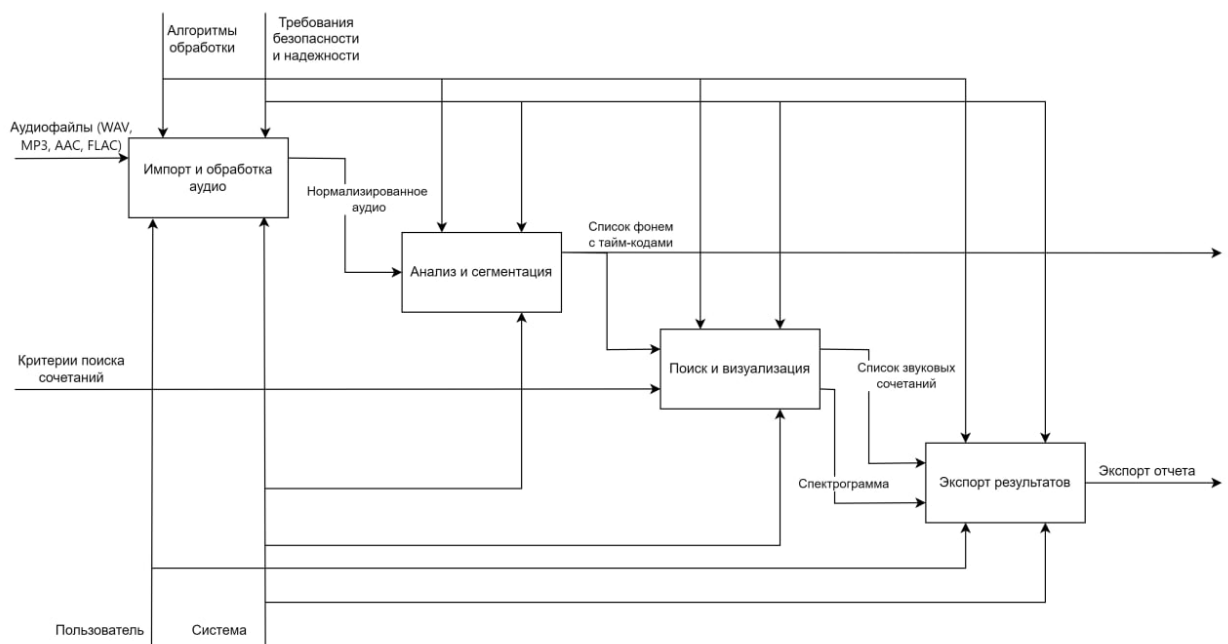


Рисунок 2 – IDEF0 декомпозиция

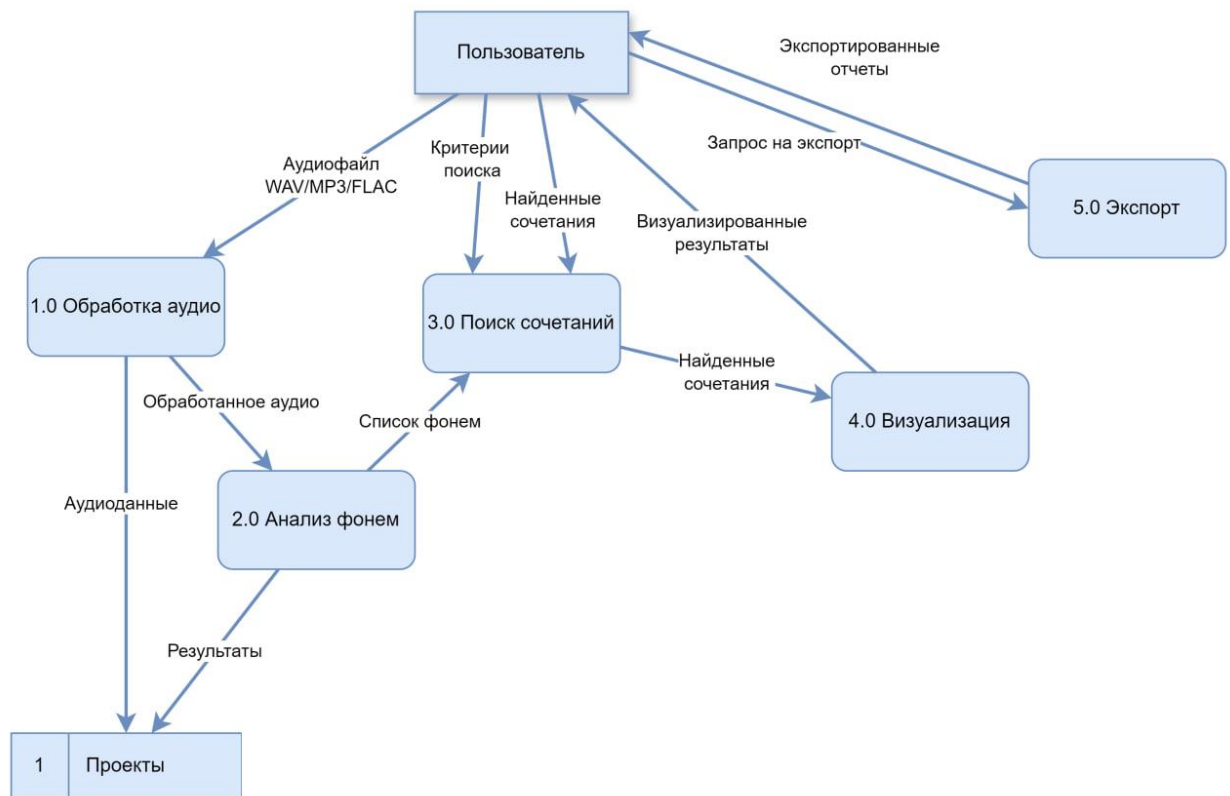


Рисунок 3 – Диаграмма потоков данных

Спецификация процессов

Поведение системы описывается через основные сценарии использования (диаграммы прецедентов) и взаимодействие компонентов (диаграммы последовательностей).

Пример: пользователь загружает аудиофайл → система выполняет предобработку → анализирует → визуализирует результат → сохраняет проект.

Для алгоритмических частей (нормализация, шумоподавление, сегментация) применяются схемы BPMN или блок-схемы, показывающие последовательность шагов. Для объектов с жизненным циклом (проект, фрагмент) создаются диаграммы состояний, отражающие переходы от загрузки до экспорта.

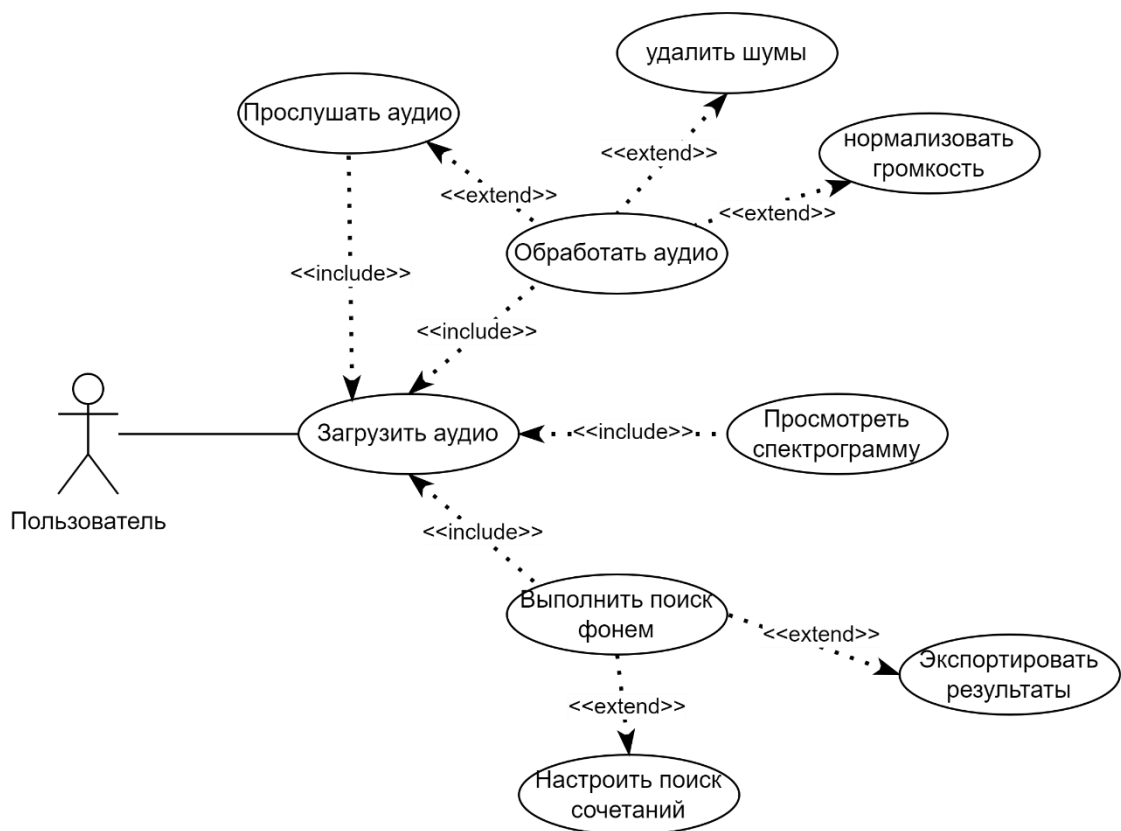


Рисунок 4 – Диаграмма прецедентов

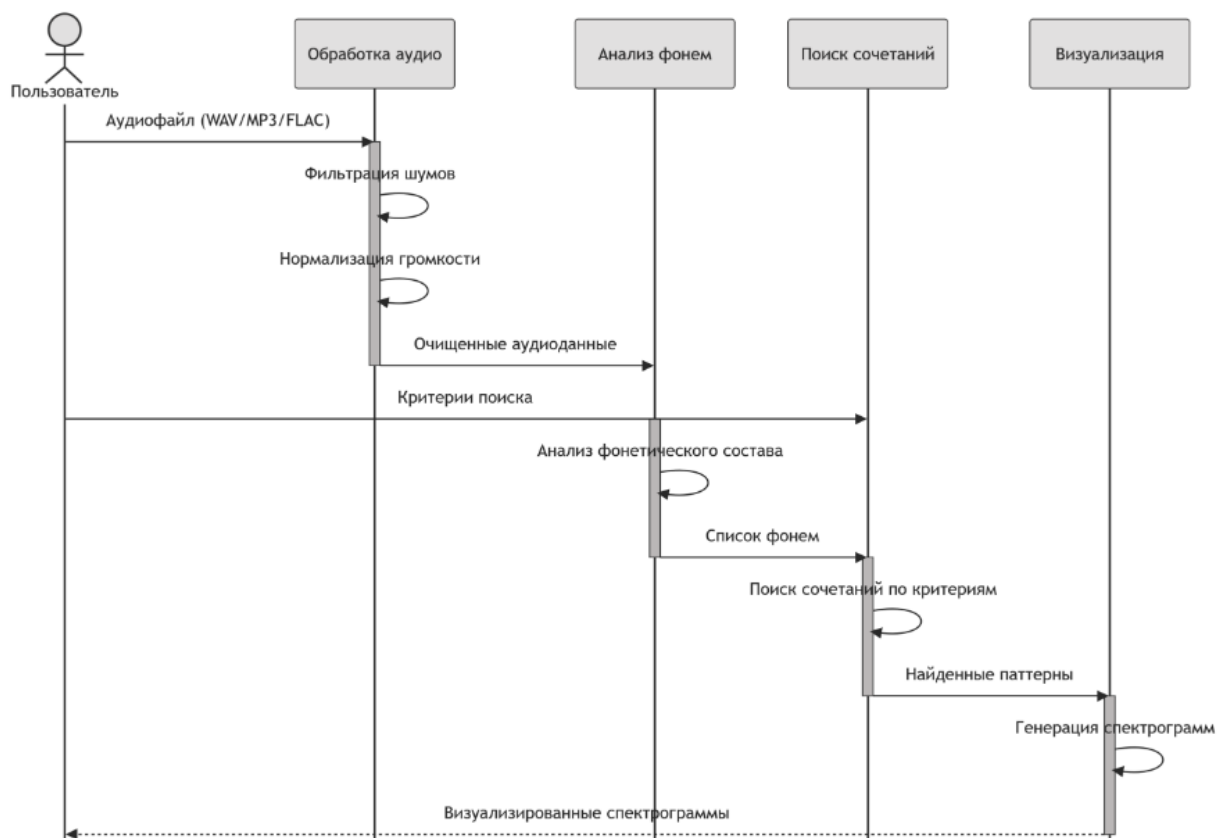


Рисунок 5 – Диаграмма последовательности UML

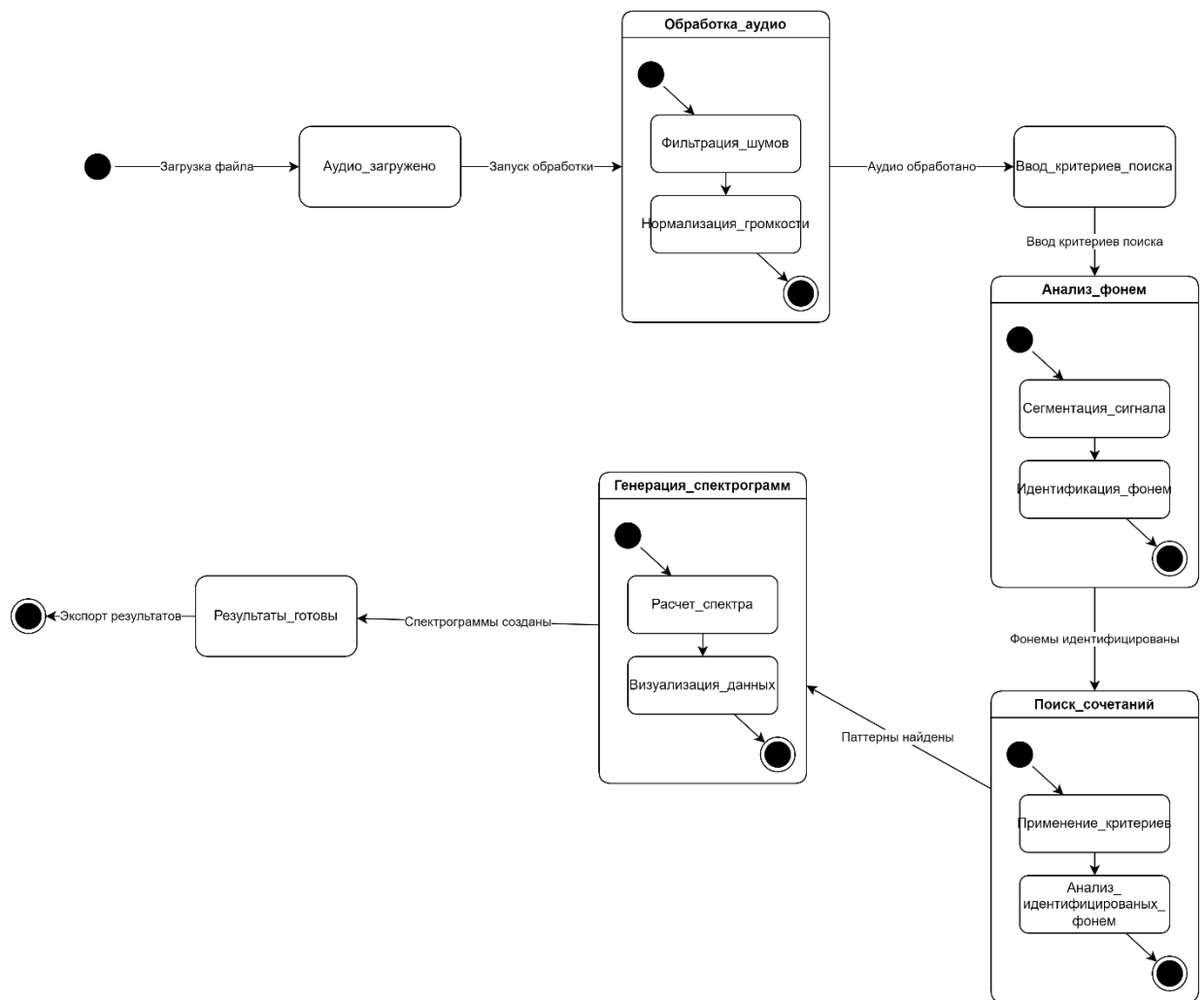


Рисунок 6 – Диаграмма состояний

Структура хранимой информации

Данные организованы вокруг сущностей: *Проект*, *Аудиофайл*, *Фрагмент*, *Результат анализа*, *Звуковые сочетания*.

Связи: один проект содержит несколько аудиофайлов; каждый файл — набор фрагментов; фрагменты связаны с результатами анализа.

Информация хранится в реляционной БД (ER-диаграмма) с метаданными и ссылками на бинарные файлы. Такой подход обеспечивает целостность и гибкость обработки.

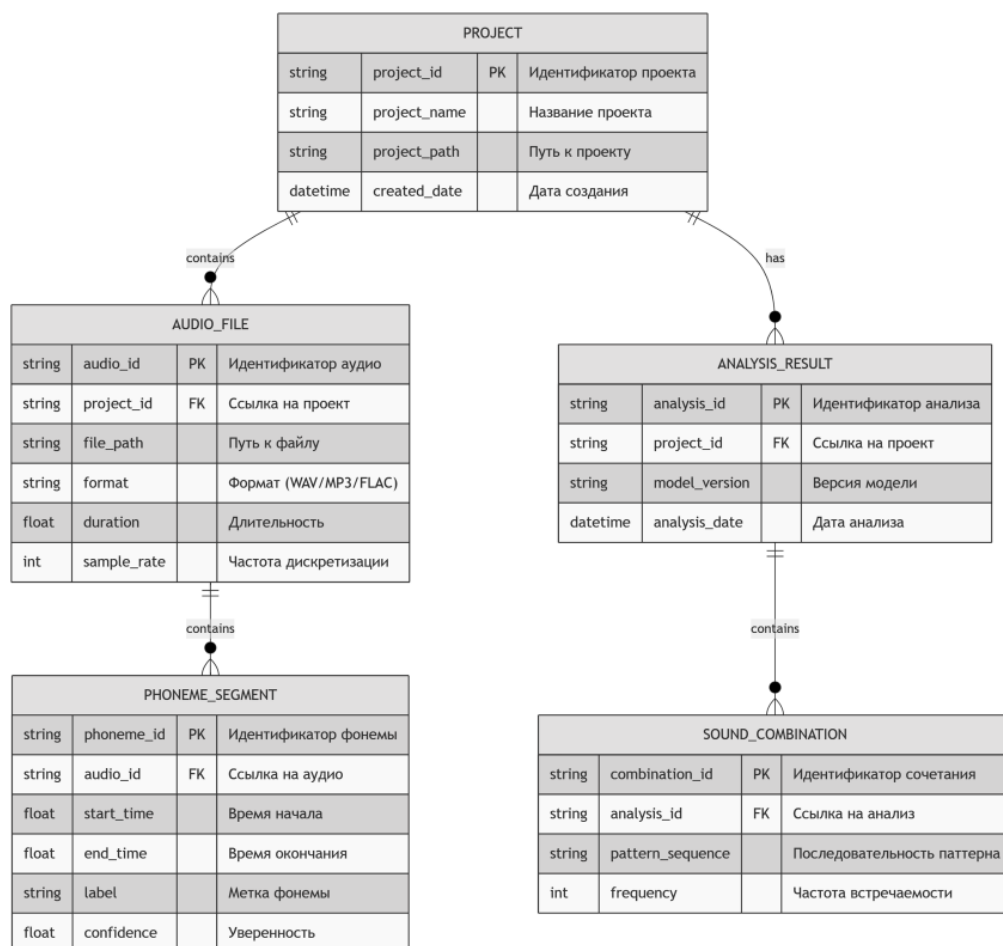


Рисунок 7 – ER диаграмма

Архитектурно-структурное решение

Продукт использует двухуровневую клиент-серверную архитектуру, полностью функционирующую на компьютере пользователя. Она состоит из двух независимых приложений:

1. GUI-клиент на C++ (SpeechAnalyzer.exe), отвечающий за интерфейс и взаимодействие с пользователем.
2. Локальный бэкенд-сервис на Python (analysis_server.exe), который выполняет всю обработку данных и вычисления с помощью ML-моделей.

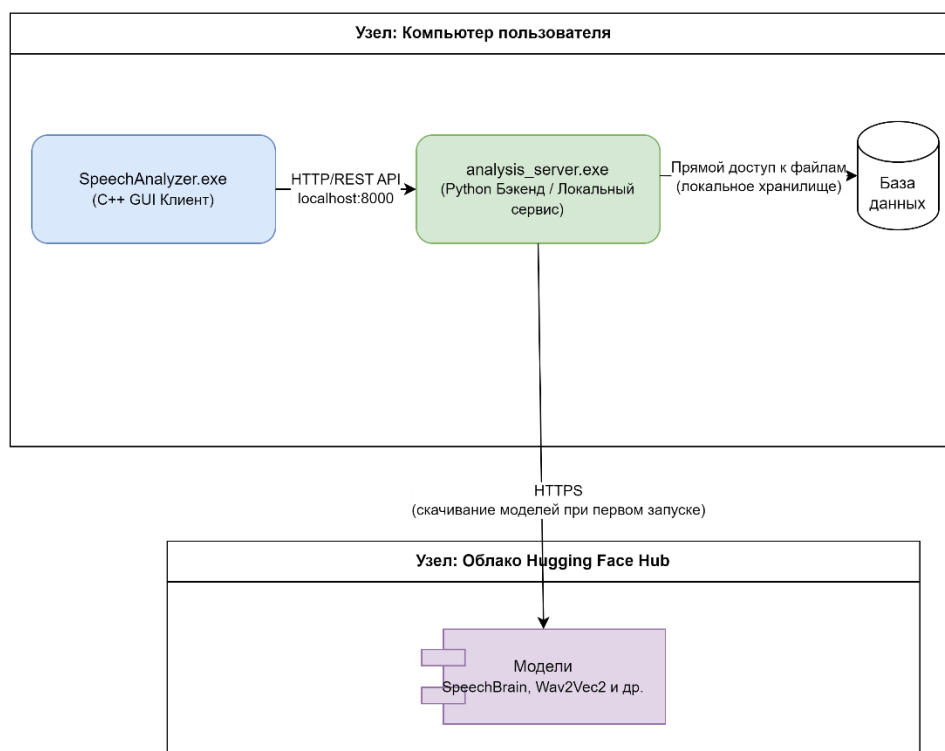


Рисунок 8 – Диаграмма размещения

Интерфейсы

Пользовательский интерфейс (UI) спроектирован с учетом потребностей целевой аудитории — исследователей, лингвистов и аналитиков. Основные принципы, заложенные в основу взаимодействия:

1. Фокус на данных: Интерфейс минимизирует отвлекающие элементы, концентрируя внимание пользователя на самом важном — аудиосигнале, его спектрограмме и результатах анализа.
2. Сравнительный анализ: Ключевой особенностью является возможность одновременного просмотра и прослушивания исходного («до») и обработанного («после») аудио. Это позволяет мгновенно оценивать результат применения фильтров и других инструментов.
3. Прямое манипулирование: Пользователь может взаимодействовать непосредственно с визуализацией данных, например, выделять фрагменты на спектрограмме для дальнейшей работы.

4. Эффективность рабочего процесса: Интерфейс построен так, чтобы поддерживать логичный и последовательный рабочий процесс: загрузка -> обработка -> анализ -> экспорт.

При запуске программы открывается стартовый экран с возможностью создать новый проект, загрузив аудио, или вернуться к старым проектам. Прототип окна стартового экрана представлен на рисунке 9.

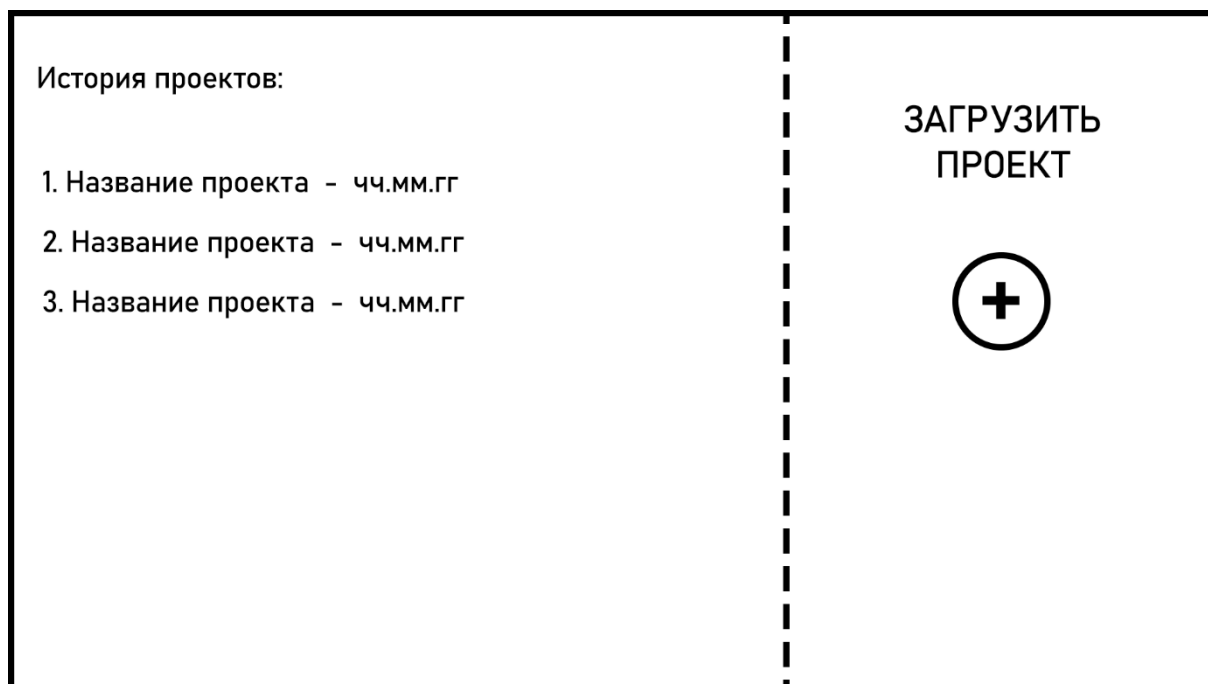


Рисунок 9– Прототип экранной формы стартового окна

На стартовом экране предоставляются возможности:

- просматривать список ранее созданных проектов с указанием названия и даты создания/последнего изменения;
- загружать существующий проект из списка;
- создавать новый проект с помощью кнопки «Загрузить проект» (с иконкой «+»).

Для работы с выбранным проектом и аудиофайлами программа должна предоставлять главный экран, являющийся основным рабочим пространством пользователя. Прототип окна главного экрана представлен на рисунке.

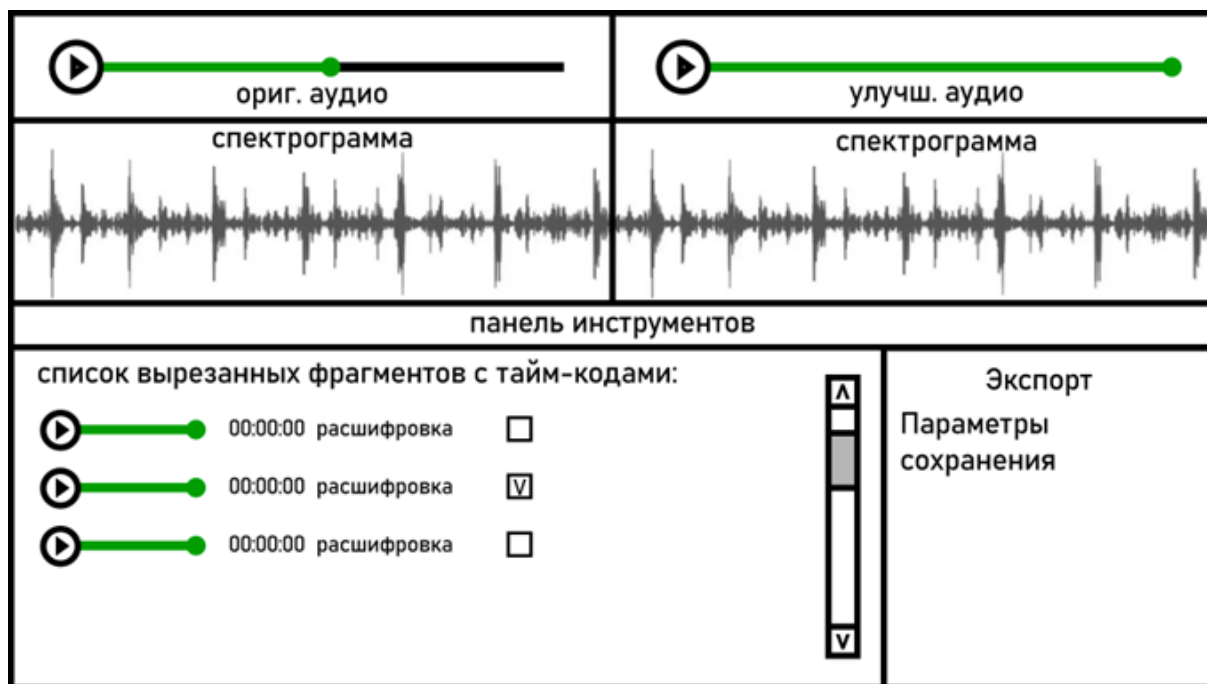


Рисунок10– Прототип экранной формы главного окна

В рабочем пространстве представлены следующие возможности:

- воспроизводить оригинальное аудио и улучшенное аудио через встроенные плееры;
- просматривать спектрограмму для оригинального и обработанного аудио;
- управлять воспроизведением (старт, пауза, перемотка, регулировка громкости);
- вырезать и сохранять отдельные аудиофрагменты с тайм-кодами;
- просматривать список найденных фрагментов аудио;
- прослушать каждый фрагмент отдельно;
- отмечать найденные фрагменты для экспорта;
- экспортировать результаты в выбранном формате.