УДК 004.912

EDN PSVGZC

Проектирование бизнес-логики и интерфейса пользователя для программной системы автоматической транскрибации медиа данных

М.А. Ковито[1*], Д.И. Ковалев[2,3]

[1]Сибирский федеральный университет, Красноярск, Россия
[2]Красноярский краевой Дом науки и техники РосСНИО, Красноярск, Россия
[3]Красноярский государственный аграрный университет, Красноярск, Россия

*E-mail: krovik76@gmail.com

Аннотация. В статье рассматриваются современные проблемы проектирования бизнес-логики и интерфейса пользователя для программной системы автоматической транскрибации медиа данных. Отмечается актуальность данного направления исследований и его практическая значимость для процесса расшифровки речи из аудио- или видеозаписи в текст. Сформированы требования к программной системе автоматической транскрибации медиа данных. Также в работе определены ограничения системы при проектировании архитектуры приложения и создан дизайн-макет интерфейса пользователя. Реализация системы включает два блока первый из которых представляет собой интерфейс пользователя в виде графической оболочки для взаимодействия с пользователем, второй реализует бизнес-логику, которая обеспечивает непосредственную обработку данных.

Ключевые слова: медиа данные, транскрибация, бизнес-логика, интерфейс, программная система.

Designing business logic and user interface for a software system for automatic transcription of media data

M.A. Kovito[1*], D.I. Kovalev[2,3]

[1]Siberian Federal University, Krasnoyarsk, Russia
[2]Krasnoyarsk Science and Technology City Hall, Krasnoyarsk, Russia
[3]Krasnoyarsk State Agrarian University, Krasnoyarsk, Russia

*E-mail: krovik76@gmail.com

Abstract. The article deals with modern problems of designing business logic and user interface for a software system for automatic transcription of media data. The relevance of this area of research and its practical significance for the process of transcribing speech from audio or video recording into text is noted. The requirements for the software system for automatic transcription of media data have been formed. Also, the work defines the limitations of the system when designing the application architecture and created a design layout for the user interface. The system implementation includes two blocks, the first of which is a user interface in the form of a graphical shell for interacting with the user, the second implements business logic that provides direct data processing.

Keywords: media data, transcription, business logic, interface, software system.

1. Введение

С ростом цифровизации и всё более широким применением голосовых сообщений и дистанционных встреч происходит накапливание медиа данных, которые могут содержать полезную и важную для человека информацию [1-3]. Однако, процесс её поиска и обработки в данных такого формата достаточно долгий и не очень удобный в отличие от данных, хранящихся в текстовом формате, где, например, доступны средства автоматического поиска и фрагментарного копирования. Например, в работе [1] приведены результаты экспериментов, проведенных с целью сравнительного анализа качества работы существующих сервисов по обработке текстов на русском языке. В работе [2] исследуется возможность применения Google Cloud Speech-to-text API для автоматической транскрибации веб-конференций в реальном времени. Также решению данной задачи посвящены многие интернет-ресурсы [4-7], что говорит об актуальности данного направления исследований и его практической значимости для процесса расшифровки речи из аудио- или видеозаписи в текст.

2. Цель работы

Целью представленной работы является проектирование архитектуры и реализация программной системы автоматической транскрибации медиа данных.

Для достижения поставленной цели необходимо спроектировать программную систему для перевода аудио данных в текстовый формат. Для этого необходимо:

- выделить требования к входным аудио данным, выходным текстовым данным, функциональные требования и требования к используемым технологиям;
- спроектировать архитектуру и бизнес логику программной системы;
- спроектировать интерфейс пользователя и его функциональные блоки.

3. Результаты и их обсуждение

Сформулируем основные требования и ограничения, которым должна отвечать система.

Итак, программная система должна реализовывать следующие функции:

- распознавание аудиозаписи и перевод её в текстовый формат;
- осуществление контроля орфографии;
- распознание записи переговоров из файла мультимедиа;

- добавление специфических отраслевых терминов в словарь распознавания;
- дообучение нейронной сети, выполняющей распознавание речи;
- экспорт полученных текстовых данных в файл.

Касаясь требований к входным данным, отметим, что программная система должна корректно обрабатывать файлы следующих форматов: WAV; MP3. Длительность медиа файлов не должна превышать 60 минут.

Программная система должна возвращать корректные текстовые данные, без орфографических ошибок. Также данные могут быть экспортированы в текстовый файл формата «.txt».

На программную систему накладываются ограничения:

отсутствие дополнительных вложений по использованию кодеков;

использование бесплатных библиотек и сервисов.

3.1. Бизнес-логика

Исходя из представленных выше требований, принято решение разделить программную систему на два блока:

- интерфейс пользователя – графическая оболочка для взаимодействия с пользователем;
- бизнес-логика – блок, выполняющий непосредственную обработку данных.

Структурная схема бизнес-логики отображена на рисунке 1. Остановимся более подробно на основных структурных элементах бизнес-логики.

Класс контроллер (Controller) выполняет управление внутренними функциональными модулями распознавания речи и работы с файлами. Он содержит следующие свойства и методы:

- публичное свойство inputFilePath (путь к файлу для транскрибации);
- публичное свойство exportFilePath (путь к файлу для экспорта распознанного текста);
- публичное свойство glossaryPath (путь к файлу словаря);
- публичное свойство glossary (список слов в словаре терминов);
- публичное свойство recognizedText (распознанный текст);

- публичный метод recognize(), который выполняет вызов метода распознавания текста класса модели (SpeechToTextModel);
- публичный метод importMediaFile(), который выполняет вызов метода импорта файлового контроллера (FileController) и сохраняет результат в приватном поле mediaFile;
- публичный метод importGlossaryFile(), который выполняет вызов метода импорта файлового контроллера (FileController) и сохраняет результат в приватном поле glossaryFile;
- публичный метод exportRecognizedText(), который выполняет вызов метода экспорт файлового контроллера (FileController);
- приватное поле mediaFile (загруженный медиа файл для распознавания);
- приватное поле glossaryFile (файл-словарь терминов).

Класс модели распознавания человеческой речи SpeechToTextModel содержит следующие свойства и методы:

- публичный метод recognize(file: file), который на вход принимает медиа файл, а возвращает строку с распознанным текстом;
- публичный метод tune(glossaryFile: file), который дообучает модель на основе словаря терминов.

Класс выполняющий импорт и экспорт файлов **FileController** содержит следующие свойства и методы:

- публичный метод import(filePath: string), который на вход принимает путь к файлу для импорта, а возвращает импортированный файл;
- публичный метод export(text: string, filePath: string), который на вход принимает текст для экспорта и путь, куда экспортировать файл.

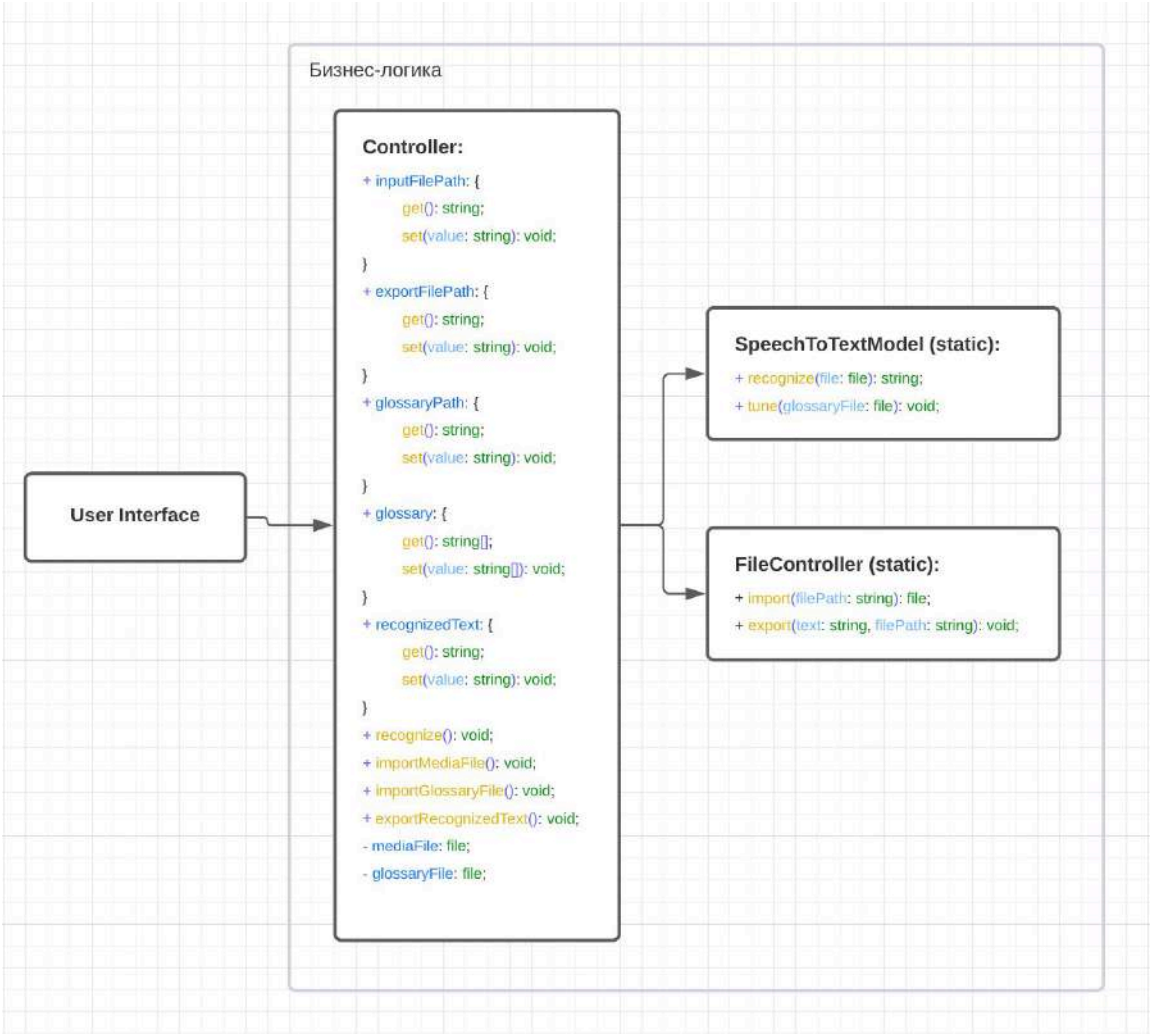

Рисунок 1. Структура бизнес-логики.

3.2. Интерфейс пользователя

Интерфейс пользователя будет обладать минималистичным дизайном, который отображен на рисунках 2-4. Для его разработки был использован сервис Figma.

В главном окне можно загрузить файл для распознавания, нажав на кнопку «Выбрать файл», либо перетащив его в выделенную область. Эти действия произведут вызов метода importMediaFile() контроллера.

После загрузки появится кнопка «Обработать», при нажатии на которую вызывается метод recognize() и запускается процесс распознавания речи. По завершении обработки должен выполниться переход к окну с распознанным текстом (см. рисунок 3), где можно внести изменения в результат распознавания, скопировать или экспортировать текст.

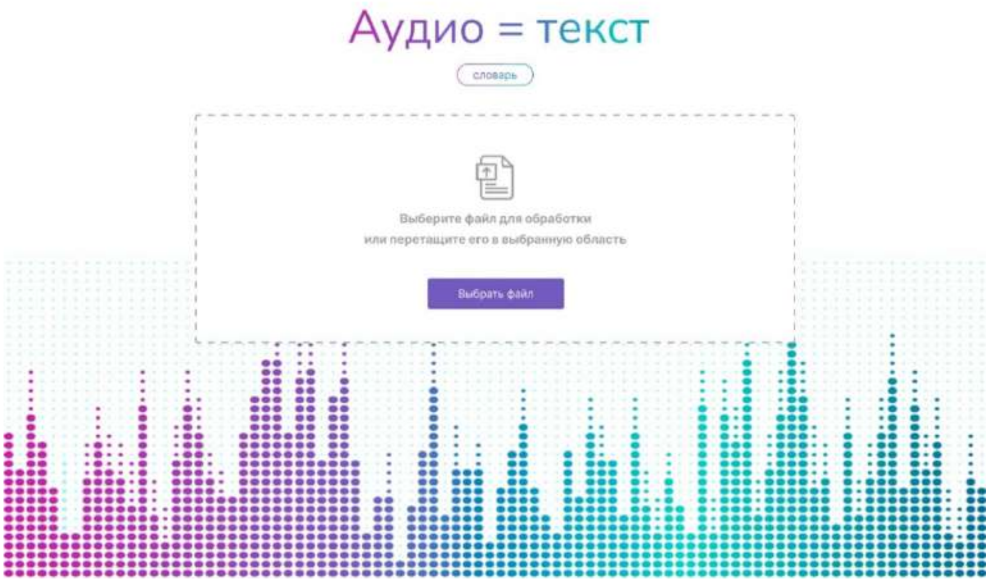

Рисунок 2. Главное окно приложения.

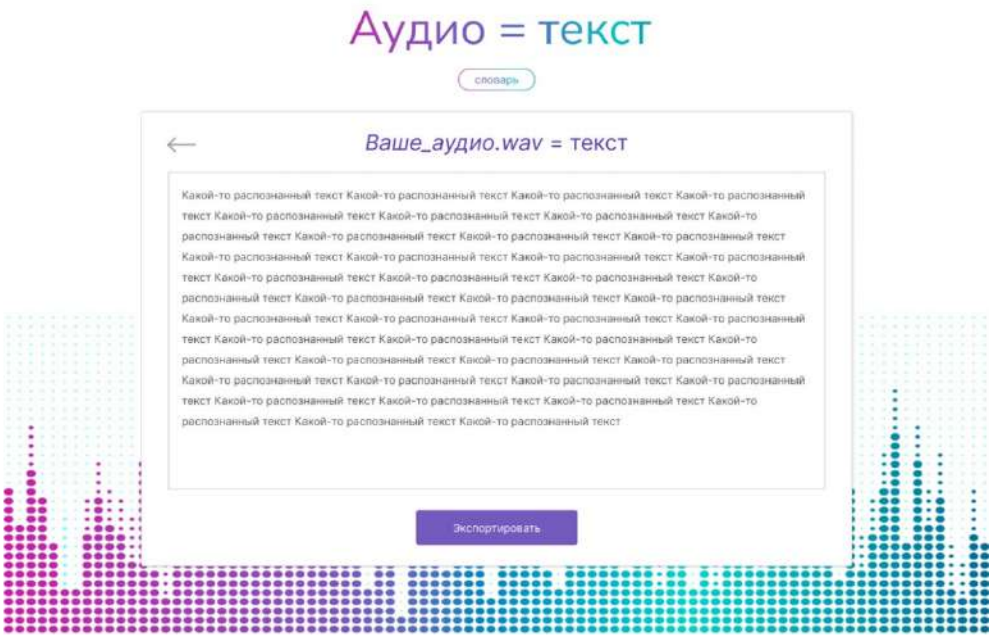

Рисунок 3. Окно с распознанным текстом.

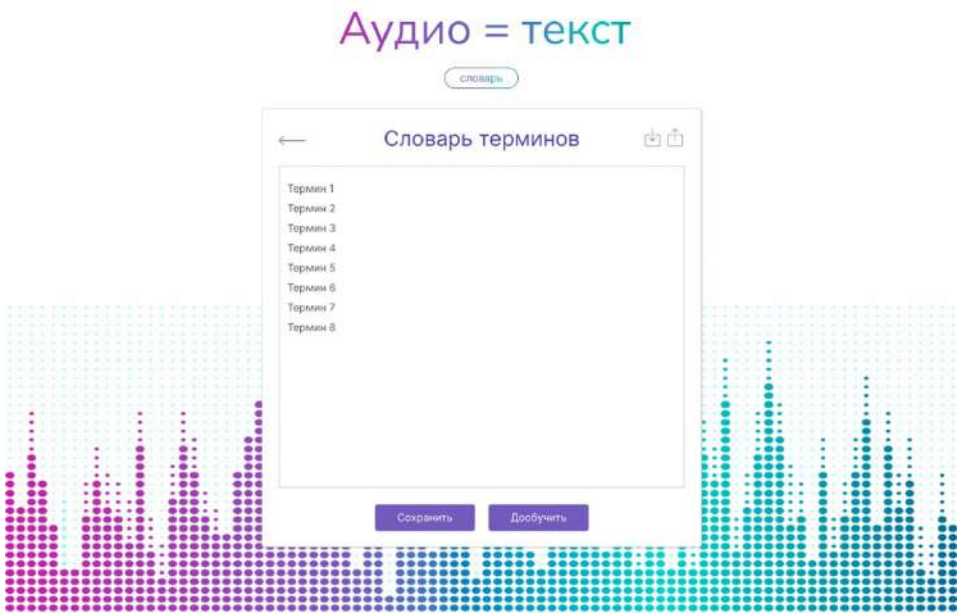

Рисунок 4. Словарь терминов.

Также во всех окнах присутствует кнопка «словарь», при нажатии на которую должно открываться модальное окно со словарем терминов (см. рисунок 4). В этом окне можно внести термины вручную, импортировать их из файла, экспортировать в файл, сохранить результат или дообучить модель распознавания на основе добавленных терминов.

4. Заключение

В результате выполненного исследования были сформированы требования к программной системе автоматической транскрибации медиа данных, определены её ограничения, спроектирована архитектура приложения, а также создан дизайн-макет интерфейса пользователя. Реализация системы представлена двумя блоками, отражающими интерфейс пользователя - графическую оболочку для взаимодействия с пользователем и бизнес-логику – блок, выполняющий непосредственную обработку данных.

Список литературы

1. Мухамедиев, Р.И. Облачные сервисы для обработки текстов на естественном языке / Р.И. Мухамедиев, А. Сымагулов, Я.И. Кучин, С. Абдуллаева, Ф.Н. Абдолдина // Современные информационные технологии и ИТ-образование. – 2018. – № 14(4). – С. 872-880.
2. Каменская, А.С. Адаптация Google Cloud Speech-to-text API для автоматической транскрибации веб-конференций в реальном времени / А.С. Каменская // Автоматика и программная инженерия. – 2019. – № 2(28). – С. 19-23.
3. UML Class Diagram Tutorial // Lucidchart: [сайт]. – URL: https://www.lucidchart.com/pages/uml-class-diagram (дата обращения: 12.11.2022).
4. The Top Free Speech-to-Text APIs, AI Models, and Open Source Engines // AssemblyAI : [сайт]. – URL: https://www.assemblyai.com/blog/the-top-free-speech-to-text-apis-and-open-source-engines/ (дата обращения: 12.11.2022).
5. Распознавание речи // Yandex Cloud: [сайт]. – URL: https://cloud.yandex.ru/docs/speechkit/stt/ (дата обращения: 12.11.2022).
6. Golos // GutHub: [сайт]. – URL: https://github.com/salute-developers/golos (дата обращения: 12.11.2022).
7. Guide to Developer Handoff // Figma: [сайт]. – URL: https://www.figma.com/best-practices/guide-to-developer-handoff/components-styles-and-documentation/ (дата обращения: 12.11.2022).