

Лицей

Индивидуальная выпускная работа

**Приложение для сохранения кратких записей**

**После онлайн встреч BriefTalk AI**

*Выполнил Кром Илья Васильевич*

Москва 2025

# Краткое описание проекта

Проект представляет из себя приложение, которое делится на две части: backend и десктопное приложение. Backend представляет из себя API для регистрации пользователей и взаимодействия с моделями искусственного интеллекта. Десктопное приложение в свою очередь является удобным интерфейсом для взаимодействия с backend'ом. Также десктопное приложение позволяет вести аудио запись онлайн встреч на данном устройстве.

Основная идея, собственно — записать онлайн встречу, отправить её на backend, там она преобразуется в текст, после чего из него будут выделены основные тезисы. Таким образом после онлайн встречи у пользователя останется краткая сводка о том, что обсуждалось на встрече.

До этого проекта я «баловался» написанием небольших программ, но полноценные большие проекты (хотя бы сравнимые по объёму с BriefTalk AI v 1.1) я не писал. Также я был совсем не знаком с такими вещами как Docker, CI\CD, деплой приложений, ORM и т.д.

## Проблемное поле

Современные деловые встречи и звонки часто занимают 2-3 часа, и при их большом количестве участникам становится трудно запомнить и структурировать полученную информацию.

На данный момент на рынке практически отсутствуют решения, которые бы эффективно записывали аудио, автоматически преобразовывали его в текст, суммировали основные темы обсуждения и сохраняли результат в удобных форматах. А то, что есть, в очень часто не поддерживает работу с русским языком, или ограничено одним конкретным приложением для встреч (например, Zoom)

Заявляемый продукт решает эту проблему, предоставляя пользователям возможность получения как полного текста разговоров, так и краткого пересказа

ключевых моментов каждой встречи на русском языке, а также возможность вести аудио запись встречи вне зависимости от «места» её проведения (будь то Google Meet или Яндекс Телемост или еще что-нибудь)

В качестве доказательства проблемного поля прикрепляю результаты проведённого мною опроса:

Otter.ai просит 8.33 \$/месяц за:

- возможность преобразовать 10 аудио/видео в текст в месяц
- максимальную длину одного разговора 1.5 часа

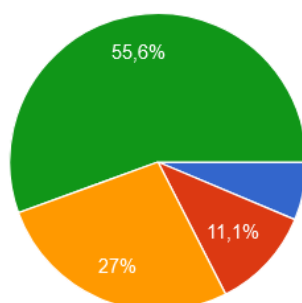
При этом поддержка существует только для английского, французского и испанского языков

за 20 \$/месяц вы получите:

- максимальную длину разговора 4 часа
- Безлимитное кол-во преобразований в текст

Как вы считаете, насколько такие цены справедливы? Выберите наиболее подходящий для вас вариант

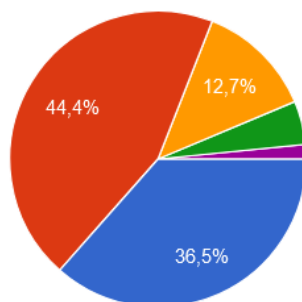
63 ответа



- Отличная цена, при необходимости я бы легко заплатил(а) столько за подобный функционал
- Хорошая цена, ну может чуть больше чем хотелось бы
- Если будет острая необходимость - я могу заплатить такую цену, но считаю её достаточно высокой
- Слишком высокая цена, я не готов(а) столько платить за подобный функц...

Считаете ли вы полноценную поддержку **только** английского языка серьёзным минусом для подобного приложения? (если брать глобально, не учитывая что вы носитель русского языка)  
Выберете наиболее подходящий для вас вариант

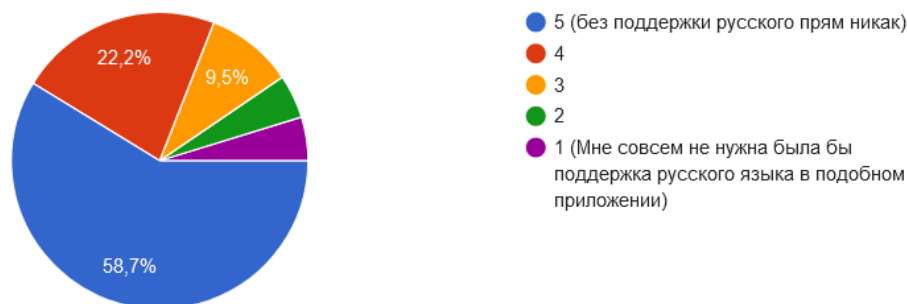
63 ответа



- Да, это очень серьёзный минус, нужно больше языков!
- Да, это существенный минус
- Не сказал(а) бы что это значимый минус, но это определённо минус (=D)
- Для подобного приложения это супер незначительный минус
- Не сказал(а) бы что это вообще минус

Представим что вы пользуетесь \ собираетесь пользоваться подобным функционалом.  
Насколько вам нужна была бы поддержка русского языка (т.е. чтобы можно было вообще работать с аудио на русском языке)?  
Оцените потребность от 1 до 5

63 ответа



## Целевая аудитория

Все люди, которые регулярно сидят на созвонах \ встречах онлайн. Любой возраст, любая профессия, главное, что сидят на онлайн встречах. К выделяемым группам относятся:

- Руководители и менеджеры проектов
- Аналитики и консультанты
- Люди, работающие удалённо

## Функциональные требования

В проект были заложены следующие функциональные требования:

- Автоматическая запись аудио созвонов и встреч. Была выбрана идея записывать вообще весь входящий и исходящий аудиосигнал, таким образом можно записывать звук созвонов вне зависимости от платформы их проведения.
- Преобразование аудиозаписей в текст.
- Суммаризация текстовых данных, выделение ключевых тем.

- Сохранение полного текста и суммарного отчета в популярные форматы (txt, docx и т.д.) с возможностью их дальнейшего редактирования

Это было сделано исходя из проблемного поля, основная идея в котором - отсутствие хороших решений для русскоязычных пользователей. Следовательно, моей целью было сделать продукт, который был бы доступен и мог бы работать с русским языком.

## Конкуренты на рынке

### 1. Otter.ai

Сильные стороны:

- Высокая точность распознавания речи на английском языке.
- Интеграция с Zoom, Microsoft Teams, Google Meet.
- Автоматическое создание сводок встреч с ключевыми моментами.
- Удобный поиск по тексту и возможность делиться записями.

Слабые стороны:

- Отсутствие поддержки русского языка.
- Высокая стоимость подписки (\$8/месяц и выше).

### 2. Fireflies.ai

Сильные стороны:

- Поддержка широкого спектра платформ (Zoom, Slack, Google Meet).
- Возможность аннотирования текста и поиска по записям.
- Встроенная система управления задачами и делегирования.

Слабые стороны:

- Низкая поддержка русского языка.

- Неудобный интерфейс для редактирования текста.

### 3. **Descript**

Сильные стороны:

- Уникальная функция редактирования аудио через текст.
- Высокая точность распознавания речи.
- Поддержка подкастов и видео, а не только деловых встреч.
- Встроенные инструменты для удаления слов-паразитов и редактирования.

Слабые стороны:

- Ориентирован на создание контента, а не на бизнес-встречи.
- Нет поддержки русского языка.
- Высокая стоимость (\$15–30/месяц).
- Сложный интерфейс для новичков.

### 4. **Rev**

Сильные стороны:

- Поддержка профессиональных аудиофайлов.
- Гибкие возможности обработки нестандартных форматов.

Слабые стороны:

- Длительное время обработки (несколько часов для длинных встреч).
- Высокая стоимость (\$1,5/минута при использовании редактора-человека).
- Нет автоматической генерации сводок.
- Не интегрирован с платформами для видеозвонков.

### 5. **Zoom**

Сильные стороны:

- Встроенная функция автоматической расшифровки встреч (Live Transcription).
- Глубокая интеграция с экосистемой видеозвонков.
- Возможность записи и хранения встреч.

Слабые стороны:

- Качество расшифровки ниже, чем у специализированных сервисов.
- Ограниченная поддержка языков, русский работает с ошибками.
- Нет функций автоматического сокращения текста.
- Файл с расшифровкой требует ручной постобработки.

**Вывод:** Существующие решения ориентированы в основном на английский язык, имеют высокую стоимость и ограниченные возможности редактирования.

## Стек технологий

- Python – основной язык разработки
- Qt (PySide6 в частности) – основной фреймворк для создания графического интерфейса и дизайна
- PyTorch – фреймворк для работы с ML моделями
- SQLite – СУБД для хранения настроек приложения и хранения данных на удалённом сервере
- FastAPI – фреймворк для backend'а
- SQLAlchemy – библиотека для работы с базой данных
- Docker и docker compose – для сбора backend части приложения в контейнеры
- Github Actions – CI\CD

# Рефлексия

1) Возникавшие в процессе работы над продуктом проблемы и способы решения: Все проблемы которые у меня возникали делятся на два типа:

А) «Баги» решение которых я нашел в интернете либо спросил у кого-то как их исправить. Таких было очень много и всяких разных, это, я считаю, особенность данной области

Б) «Баги» и проблемы для которых я не смог найти решения ни в интернете, ни у других людей. Пример такого «бага»:

Когда я запускаю моё десктопное приложение, некоторые классы объектов должны обязательно находиться в одном файле кода чтобы всё работало корректно. Это проблема, потому что этот один конкретный файл кода становится слишком большим и неудобным для дальнейшей разработки. Однако объяснения почему так происходит я так и не нашел. Попытки решить проблему путём экспериментирования также ни к чему не привели. Такого сорта проблемы встречаются сильно реже.

2) Дальнейшие перспективы развития

a. Добавление новых «фичей» в десктопное приложение для более удобного и интуитивно понятного взаимодействия с интерфейсом

b. Обезопасить backend приложение от всевозможных информационных атак насколько это возможно

c. Распространить приложение среди пользователей

d. Возможно, добавление платных функций и\или рекламы в приложение дабы оно начало приносить доход.

e. Выступление с данным проектом на различных олимпиадах и конкурсах

3) За время выполнения проекта я научился:

a. Работать с БД через ORM

b. Оборачивать приложения в контейнеры с помощью Docker



- c. Настраивать подключения по SSH, а также настраивать сервера для безопасной работы с ними
- d. Разворачивать свои приложения на удалённых серверах
- e. Запускать процесс CI\CD
- f. Структурировать большое количество файлов в одну большую систему

В дальнейшем все эти навыки и знания пригодятся мне для написания собственных продуктов, как на основе этого проекта, так и с другими задумками.

4) Если говорить о моём приложении не как об MVP, а как о готовом решении, которое можно брать и пользоваться, то:

- Стоило уделить больше внимания выбору ИИ моделей для проекта и подобрать решения которые лучше бы выполняли поставленные задачи (так как в данной версии (1.1) суммаризация текста делается путём вычленения наиболее важных предложений из общей массы, а не созданием краткой сводки. Также при преобразовании из аудио в текст текущая версия приложения не распознаёт разные голоса и довольно часто (хоть и не критично) ошибается при переводе аудио в текст) (также стоит сказать что обучить свою модель – практически нереальная задача, так как этим занимаются большие компании которые тратят огромные средства на вычислительные мощности и еще больше средств они тратят на сбор и структурирование данных для обучения моделей. Речь идет **как минимум** о миллионах рублей)
- Для хранения данных стоило использовать PostgreSQL так как он более масштабируем и имеет больше возможностей для разработчиков по сравнению с SQLite
- Стоило добавить больше оптимизационных решений, вроде Redis
- Также, стоило заложить больше масштабируемости в проект добавив, например, Kubernetes