**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

| | ***Подп. и дата*** |  | | --- | --- | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **Анимированный переводчик для обмена сообщениями**  **Итоговый документ** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |
| Исполнители:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Малашенко Б.Т, Семенкович Т.А. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.  Заказчик:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Салех Х.М. /  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Компания /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. | |
|  | | |
|  | |  |

**2022**

**СОДЕРЖАНИЕ И СТРУКТУРА ДОКУМЕНТА**

**1.Введение**

1.1.Наименование программы

1.2.Краткая характеристика области применения

**2. Команда**

2.1 Командные роли и распределение ответственности

2.2 Матрица компетенций

**3.Управление проектом**

3.1. Метрики проекта

3.2. Диаграмма Ганта (базовый план и итоговый состав работ)

3.3. Использование инструментов (Canvas) или систем управления проектом (Trello, Jira)

**4. Предметная область и описание продукта**

3.1 Постановка проблемы

3.2 Анализ предметной области, рамки и границы проекта

3.2 Назначение приложения

**5. Описание потенциальных пользователей**

4.1 Цель заказчика и ключевые стейкхолдеры (матрица Управления стейкхолдерами)

4.2 Матрица ответственности

4.3 План управления коммуникациями

**6. Анализ конкурентов**

**7. Функциональные требования**

6.1. Подключение к системе

6.3.Требования к составу выполняемых функций

6.4.Требования к реализации

6.5.Требования к интерфейсу, пр..

**8. Другие требования к продукту (нефункциональные)**

7.1.Требования к надежности

7.2.Требования к хранению

7.3.Требования к безопасности (качеству, стабильности, удобству пользования, проч.

**9. Описание архитектуры приложения и используемых технологий**

**10. Первоначальный прототип**

10.1.Прототипы ранней стадии планирования

10.2.Первые прототипы

**11. Используемые технологии** (Front End / Back End / GitHub / прочие)

**12. Предложения по монетизации/коммерциализации (либо внутр. эффект от автоматизации)**

**13. Будущее развитие проекта**

**14. Отчеты о взаимодействии с заказчиком** (Шаблон 1-6 с подписью)

**15.Отзыв заказчика**

**16. Руководство пользователя**

15.1.Вход в систему

15.2.Выход из системы

**17. Список используемой литературы**

**ВВЕДЕНИЕ**

Наименование программы – Анимированный переводчик для обмена сообщениями.

Краткая характеристика области применения – общение в личных и групповых чатах с возможностью перевода сообщений и синтеза речи.

**КОМАНДА**

| Малашенко Борис Тарасович | Back-end, программист |
| --- | --- |
| Семенкович Тимофей Алексеевич | Разработчик БД, программист |

*Таблица 1. Командные роли*

| RACI | **Борис** | **Тимофей** | **Хади** |
| --- | --- | --- | --- |
| Написание документации | AR | R | I |
| Создание клиент-серверного приложения | C | AR | I |
| Разработка БД | C | AR | I |
| Общение с заказчиком | AR | C | I |
| Работа с ML | AR | C | I |
| Разработка GUI | C | AR | I |
| Тестирование | AR | R | I |

*Таблица 2. Распределение ответственности*

|  | **Борис** | **Тимофей** |
| --- | --- | --- |
| **Postgresql** | **1** | **5** |
| **GUI** | **3** | **4** |
| **Python** | **4** | **5** |
| **PyTorch** | **4** | **2** |
| **NeMo** | **4** | **1** |
| **JavaScript** | **2** | **3** |

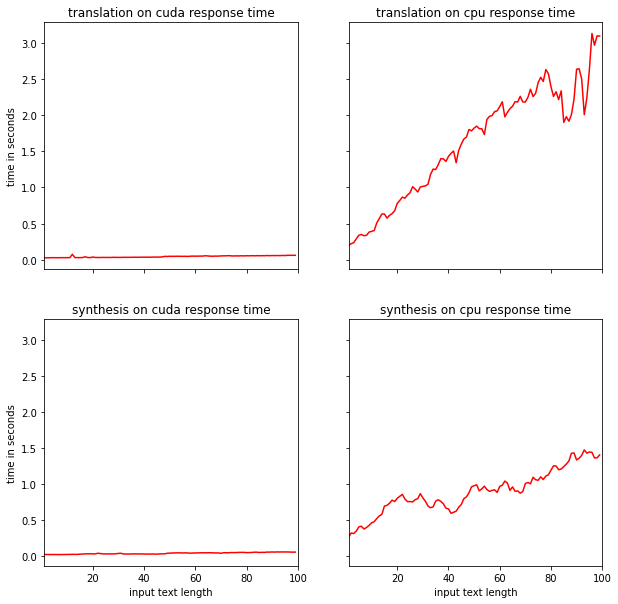
*Таблица 3. Матрица компетенций*

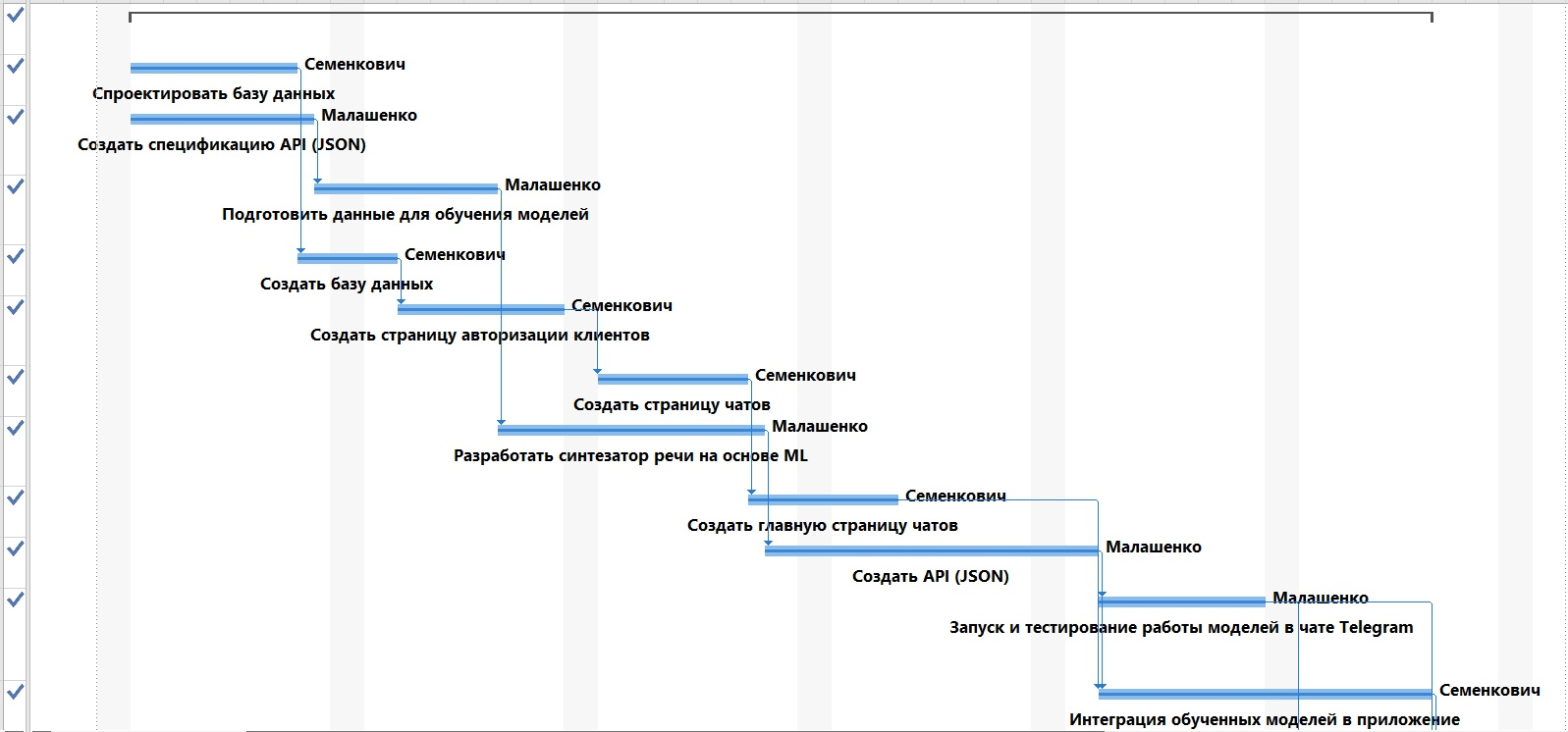
**УПРАВЛЕНИЕ ПРОЕКТОМ**

Метрики проекта:

1. Итоговый вес моделей. Он может быть важен, так как место на сервере ограничено и не бесплатно. При большом числе пользователей необходимо будет запускать кластеры моделей.

| **Модель** | **Вес (MB)** |
| --- | --- |
| nmt\_transformer6x6 | 861.05 |
| fastpitch | 162.07 |
| hifigan | 300.78 |

1. Время ответа API. Сравнение для CUDA и CPU устройств.
2. Пропускная способность системы. На основе предыдущих расчетов, мы можем рассчитать пропускную способность модели в минуту. С учётом работы базы данных и web-приложения, число запросов в минуту при длине среднего запроса в 30 символов составляет 400 и 600 для синтеза речи и перевода на CUDA и 60 и 90 на CPU.
3. Простота поддержки кодовой базы обеспечивается микросервисной архитектурой приложения.
4. Стабильность работы системы на различных устройствах сохраняется при стандартных настройках веб-браузера.
5. Качество работы моделей. Для перевода текста используется метрика WMT20 [1]. Переводчики с английского на русский и с русского на английский имеют качество 35.8 и 22.3 соответственно. Сравнение с другими моделями можно посмотреть здесь [2]. Для оценки качества моделей синтеза речи используется метрика MOS, но так как ее оценка требует большое число участников, мы воспользовались для оценки метрикой Frechet distance, коррелирующей с MOS. Оценка для русского синтезатора составляет 3 балла, 4 балла для английского.

****

*Рисунок 1. Диаграмма Ганта*

**ПРЕДМЕТНАЯ ОБЛАСТЬ И ОПИСАНИЕ ПРОДУКТА**

* Постановка проблемы

Приложение “Анимированный переводчик” - это мессенджер

предназначенный для общения людей, говорящих на разных языках.

Основные функции:

1. Автоматический перевод сообщений
2. Возможность озвучить сообщения с помощью синтезатора голоса

* Анализ предметной области, рамки и границы проекта

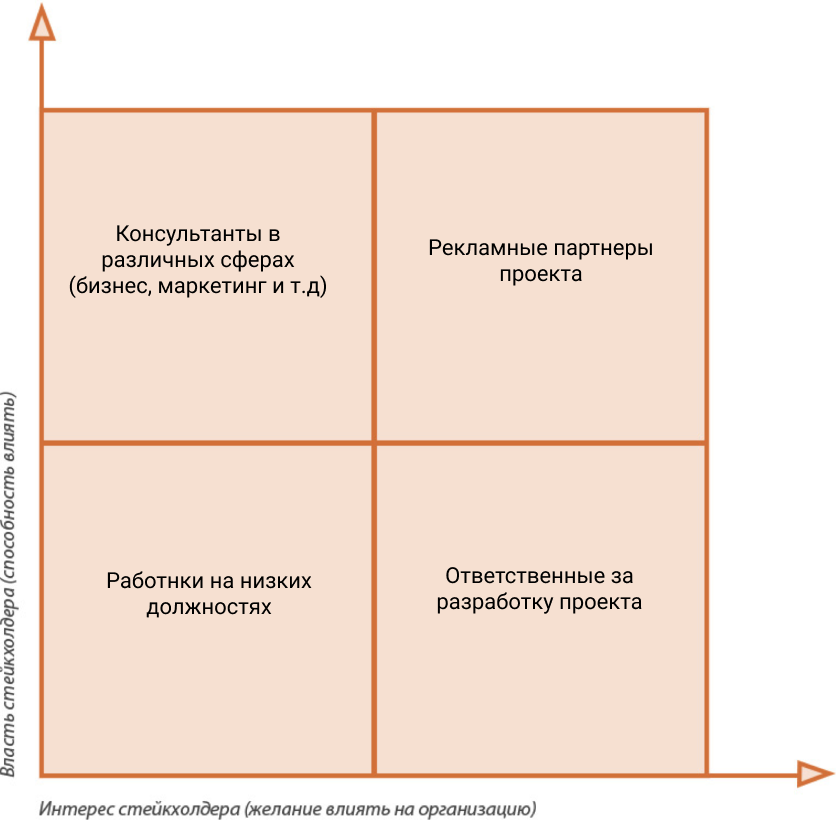
В рамки и границы проекта входит:

1. Выбор подходящей архитектуры приложения.
2. Поиск и утверждение технологий реализации и сервисов для хранения данных.
3. Подбор подходящих моделей.
4. Реализовать удобный интерфейс.
5. Создать наглядное представление качества работы приложения.

* Назначение приложения: клиент должен с помощью мобильного приложения общаться с носителями других языков “на равных”, понимая не только основную информацию, написанную собеседником, но и его эмоции, переданные в синтезированной речи.

**ОПИСАНИЕ ПОТЕНЦИАЛЬНЫХ ПОЛЬЗОВАТЕЛЕЙ**

Цель заказчика – веб-приложение, в котором клиент с помощью мобильного приложения общается с носителями других языков “на равных”, понимая не только основную информацию, написанную собеседником, но и его эмоции, переданные в синтезированной речи.

****

*Рисунок 2. Матрица управления стейкхолдерами*

| **СТЕЙКХОЛДЕРЫ** | **ПРИЧАСТНОСТЬ К ПРОБЛЕМЕ** | **КАК ОНИ МОГУТ СОДЕЙСТВОВАТЬ УСПЕХУ ПРОЕКТА** | **КАКИМ ОБРАЗОМ ИХ СЛЕДУЕТ ВОВЛЕКАТЬ** |
| --- | --- | --- | --- |
| Консультанты (бизнес, маркетинг и т.д.) | Стейкхолдер получает заработную плату, но не относится к проекту напрямую, т.е. не заинтересован в развитии организации. | Помощь в решении ключевых вопросов по проекту. | Уровень вовлеченности низкий. Способ привлечения – размещение объявлений. |
| Рекламные партнеры | Имеют высокое влияние на проект, являются основным источником финансирования. | Высказывают свои идеи по поводу развития проекта, участвуют в разработке бизнесс-плана. | Поиск инвестиций на форумах и площадках. |
| Работники на низких должностях | Не имеют влияния на решения организации. | Выполняют свою работу. | Оплата работы, премии. |
| Ответственные за разработку проекта, ведущие должности. | Заинтересованны в успехе проекта, но ограничены во влиянии. | Верное руководство. | Оплата работы, премии. |

*Таблица 4. Матрица ответственности стейкхолдеров*

| Предмет коммуникации | Цель | Частота/Дата | Средство коммуникации | Ответственный | Участники |
| --- | --- | --- | --- | --- | --- |
| Разработка требований | Получить полные и однозначные требования | 16.11.2021-23.11.2021 | Zoom | Малашенко Б. | Малашенко Б.  Семенкович Т. |
| Проектирование архитектуры проекта | Получить готовую архитектуру | 02.12.2021-09.12.2021 | Zoom | Малашенко Б. | Малашенко Б.  Семенкович Т. |
| Обучение моделей | Подготовка готовых моделей | 21.01.2022-28.01.2022 | Zoom | Малашенко Б. | Малашенко Б.  Семенкович Т. |
| Демонстрация MVP | Демонстрация работы приложения | 28.01.2022 | Zoom | Малашенко Б. | Малашенко Б. |
| Работа над веб-интерфейсом | Разработка сайта | 28.01.2022-04.02.2022 | Zoom | Малашенко Б. | Малашенко Б.  Семенкович Т. |

*Таблица 5.* План управления коммуникациями

**АНАЛИЗ КОНКУРЕНТОВ**

|  | Google Gboard | iMe Messenger | Google Translate | Line | WeChat |
| --- | --- | --- | --- | --- | --- |
| Доступные платформы | iOS/  Android | iOS/  Android | iOS/  Android | iOS/  Android | iOS |
| Стоимость | Бесплатно | Бесплатно | Бесплатно | Бесплатно | Бесплатно |
| Возможность авторизации клиента | Нет | Есть | Нет | Есть | Есть |
| Возможность регистрации клиента | Нет | Есть | Нет | Есть | Есть |
| Возможность настройки профиля пользователя | Нет | Есть | Нет | Есть | Есть |
| Возможность создать чат для нескольких пользователей | Нет | Нет | Нет | Есть | Есть |
| Возможность перевода сообщений | Есть | Есть | Есть | Есть | Есть |
| Возможность ввода сообщений голосом | Есть | Нет | Нет | Нет | Нет |
| Возможность воспроизвести сообщение голосом | Нет | Нет | Есть | Нет | Нет |
| Анимированный персонаж передающий эмоции сообщения | Нет | Нет | Нет | Нет | Нет |

*Таблица 6. Анализ конкурентов*

**ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ**

1. Возможность авторизации клиента;
2. Возможность создания чата для:

* одного собеседника,
* для нескольких собеседников;

1. Возможность приглашения пользователей в существующие чаты;
2. Возможность перевода сообщения на язык, указанный в профиле;
3. Возможность ввода сообщения с помощью:

* клавиатуры
* голоса

1. Возможность прослушивания текстового сообщения;
2. Возможность изменения информации в своем личном кабинете:

* имя
* пароль
* e-mail
* язык

1. Возможность добавления пользователей в друзья;
2. Возможность удаления пользователей из друзей;
3. Возможность просматривать профиль другого пользователя;
4. Возможность поиска пользователя по его имени;
5. Возможность отображать анимированного персонажа передающего эмоции сообщения;
6. Возможность поиска сообщения по его части;
7. Возможность поиска чата по названию;
8. Возможность удаления чата;
9. Возможность удаления аккаунта и личной информации;
10. Возможность просматривать краткую справку о функционале приложения.

**ДРУГИЕ ТРЕБОВАНИЯ К ПРОДУКТУ**

Программное обеспечение должно соответствовать следующим нефункциональным требованиям:

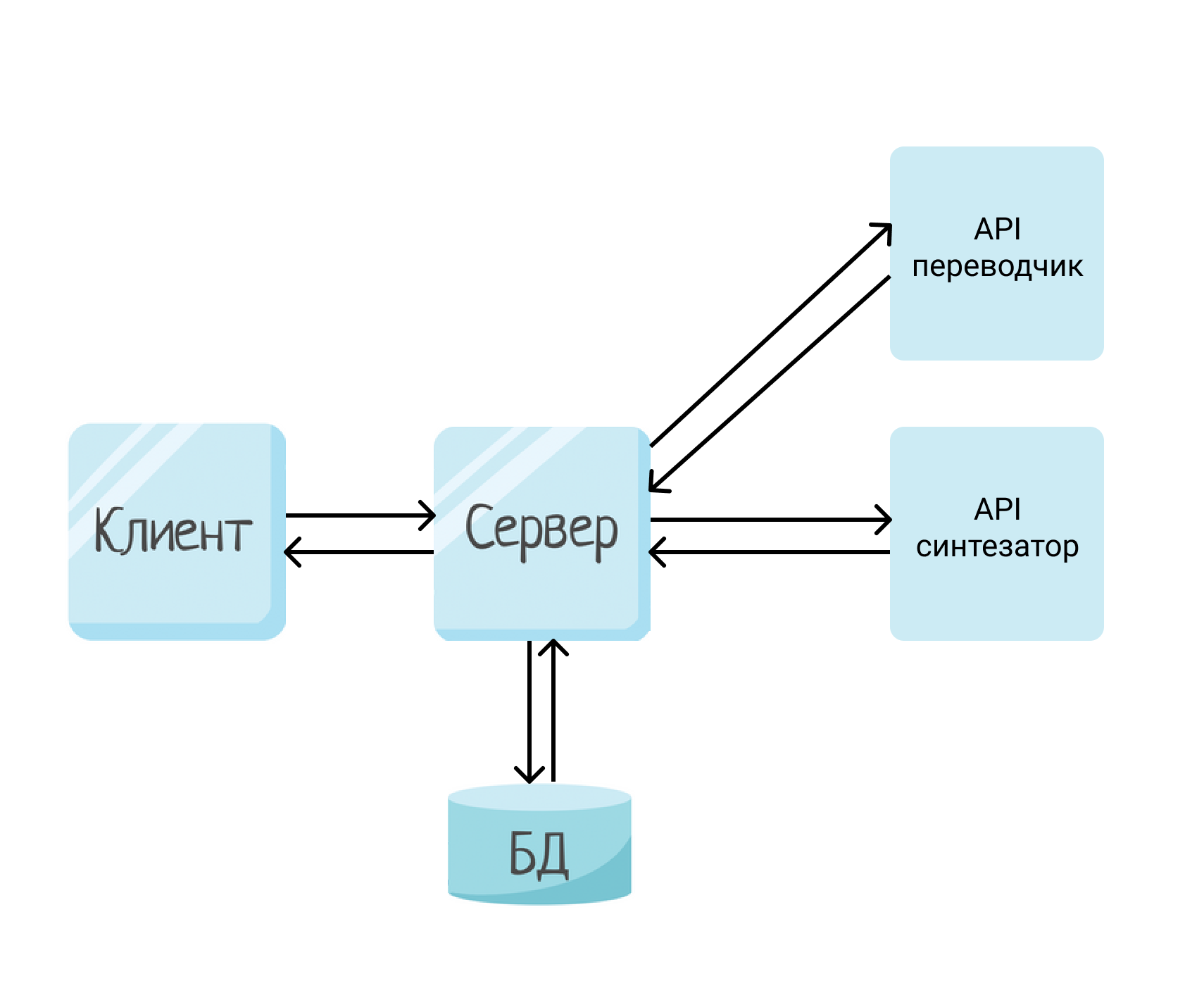
* Удобство в использовании
* Надежность
* Удобство сопровождения

Организационные требования. Серверная часть приложения должна быть реализована на языке Python. Для разработки интерфейса приложения необходимо использовать JavaScript. Данные приложения должны храниться в СУБД PostgreSql [3]. В процессе разработки приложения необходимо использовать систему контроля версий Git. Код должен быть документирован и соответствовать современным конвенциям. Код должен быть покрыт функциональными тестами. Необходимо проводить регулярные встречи с заказчиком для проверки продукта на соответствие ожиданиям и требованиям.

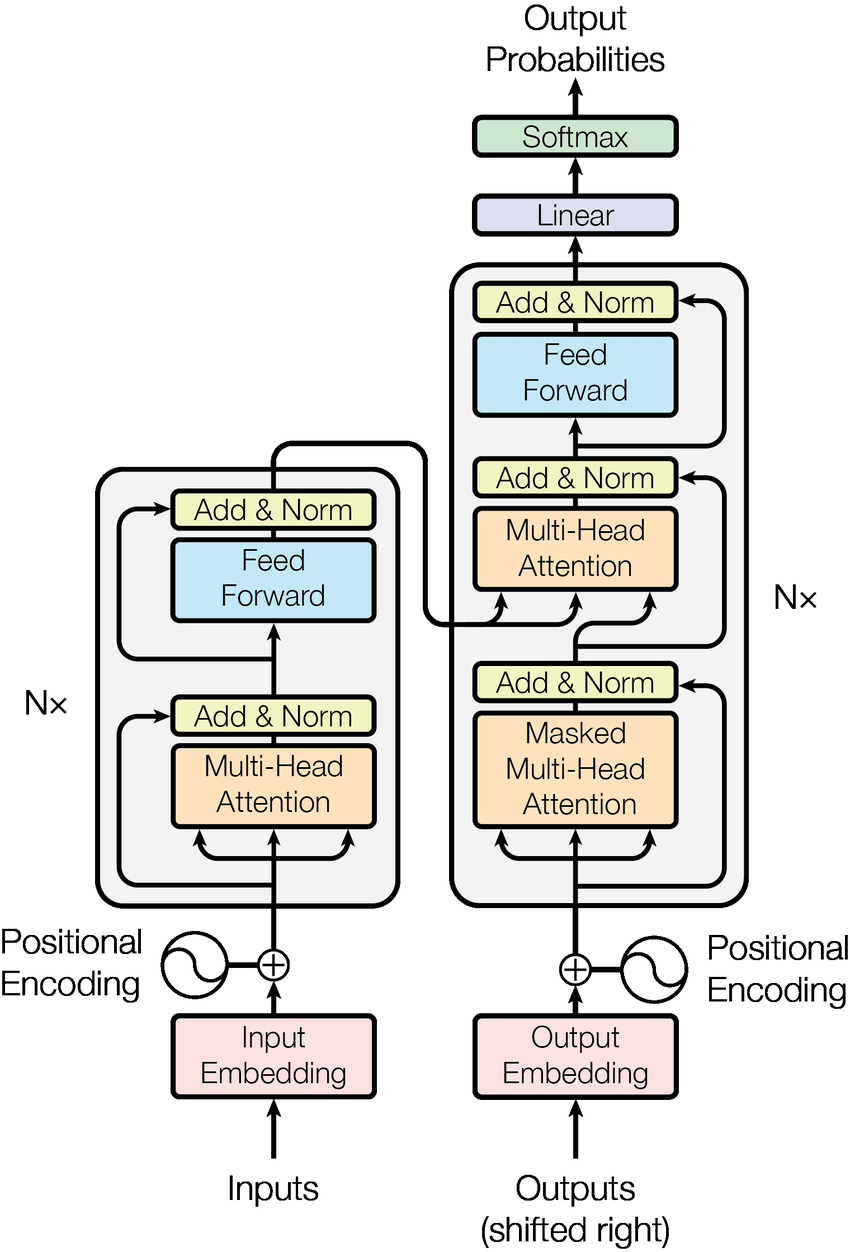
Внешние требования. Необходимо учитывать факторы, внешние по отношению к разрабатываемой системе и процессу её разработки. Они включают требования, определяющие взаимодействие данной системы с другими системами, юридические требования, следование которым гарантирует, что система будет разрабатываться и функционировать в рамках существующего законодательства, а также этические требования. Последние должны гарантировать, что система будет приемлемой для пользователей или заказчика.

**АРХИТЕКТУРА ПРИЛОЖЕНИЯ И ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ**

Для разработки использована микросервисная архитектура. Микросервисы будут обновляться независимо от веб-приложения и будут далее использоваться в других проектах. Кроме того приложение станет устойчивым к сбоям и будет возвращать ответ за корректное время, что соответствует надежности и удобству в использовании.

*****Рисунок 3. Архитектура приложения*

Модели для перевода основаны на архитектуре трансформеров [4]. Модель имеет 6 слоев в энкодере и 6 в декодере. Для получения токенов используется YouTokenizeToMe [5] токенайзер.



*Рисунок 4. Архитектура трансформера*

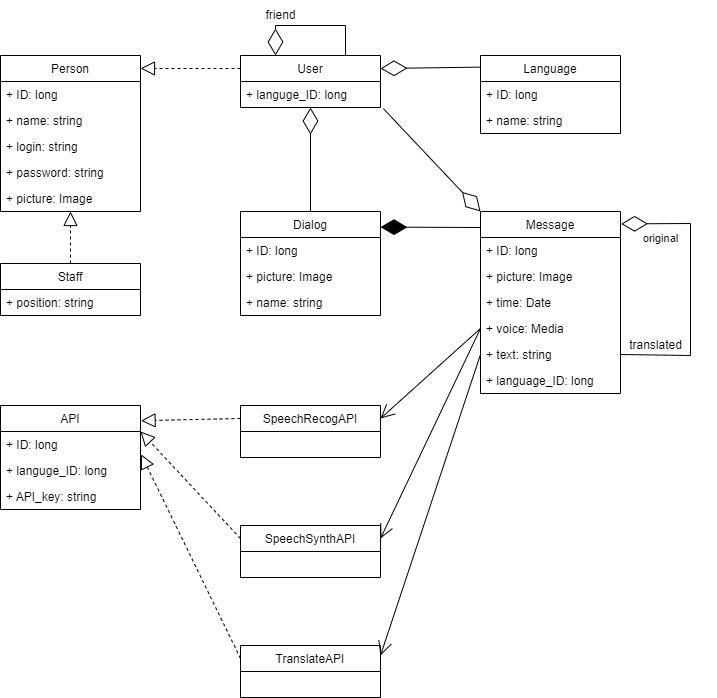
Для перевода текста с английского языка на русский и с русского на английский были взяты предобученные модели [6,15].

Синтез речи состоит из 2 этапов. Генерация спектрограмм по тексту и их преобразование в речь. Для этого используются модели Fastpitch и Hifigan. Для английского языка были взяты предобученные модели [7,8]. Для обучения модели русской речи были произведены следующие шаги:

1. Для снижения времени обучения взята предобученная модель [7].
2. При помощи библиотеки pitch вычислены значения pitch\_fmin, pitch\_fmax, pitch\_avg, pitch\_svd.

|  | pitch\_fmin | pitch\_fmax | pitch\_avg | pitch\_std |
| --- | --- | --- | --- | --- |
| MCV | 65.406 | 2093.004 | 141.892 | 126.442 |
| RUSLAN | 80 | 640 | 211.275 | 52.185 |

1. Созданы файлы манифестов (файл dataset\_preparation).
2. В качестве оптимизатора используем adam с параметрами betas = (0.9, 0.98), weight\_decay = 1.0e-06 и lr =2e-2 или 2e-4 для MCV и RUSLAN, соответственно.
3. Примеры обучения содержатся в файле FastPitch\_Hifigan\_Training.



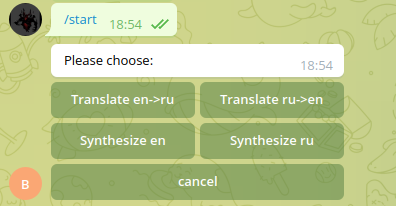
*Рисунок 5. ER-диаграмма БД*

В качестве БД используется Postgresql. В ней хранятся информация для авторизации пользователей, а также информация о чатах и сообщениях.

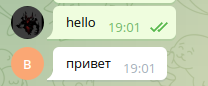
**ПЕРВОНАЧАЛЬНЫЙ ПРОТОТИП**

Первоначальный прототип был выполнен на основе мессенджера telegram.

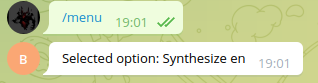
Рассмотрим use-case работы.



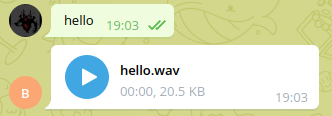
*Рисунок 6. Начало диалога с ботом*



*Рисунок 7. Перевод сообщения*



*Рисунок 8. Выбор функции синтеза речи*



*Рисунок 9. Синтез речи*

Бот выполняет логгирование пользователей и сообщений и использует реализованный API.

**ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ**

* **FrontEnd**

Весь FrontEnd представлен в виде html страниц. На некоторых присутствуют скрипты написанные на языке JavaScript. Все модели и стили взяты с сайта [https://getbootstrap.com/](https://vk.com/away.php?to=https%3A%2F%2Fgetbootstrap.com%2F&cc_key=).

* **BackEnd**

Весь BackEnd написан на языке Python с использованием библиотек Flask[9], SQLAlchemy, SocketIO[10] и другими (полный список можно найти в файле requirements.txt). Весь код находится в файле app.py, он описывает логику сервера, работает с БД и передает клиенту необходимые данные.

* **GitHub**

Весь код сервера и веб-клиента находится в следующем репозитории: <https://github.com/BaronFonMonc/web>

* **Heruko**

Данное веб-приложение мы хостим на сайте <https://wizardchat.herokuapp.com/> [11].

* **Fastapi**

API реализован при помощи фреймворка Fastapi, обеспечивающего быструю и удобную работу. Кроме того фреймворк поддерживает автокомментирование.

* **Uvicorn**

Это реализация веб-сервера ASGI для Python. Является минимальным низкоуровневым интерфейсом приложения для асинхронных фреймворков.

* **Localtunnel**

Так как для веса текущего API не было найдено бесплатных хостингов, мы воспользовались библиотекой Localtunnel, позволяющей предоставить общий доступ к приложению на вашем локальном компьютере для разработки, не вмешиваясь в настройки DNS и брандмауэра.

**ПРЕДЛОЖЕНИЯ ПО МОНЕТИЗАЦИИ**

Интеграция рекламы в приложение. Создание платной подписки, открывающей дополнительный функционал: стикеры, другие языки или голоса.

**БУДУЩЕЕ РАЗВИТИЕ ПРОЕКТА**

Повышения качества моделей, добавление новых языков.

В качестве нового функционала – реализация воспроизведения эмоционального окраса текста, анализ сентиментов.

Повышение скорости и стабильности работы приложения.

Добавление страницы профиля и возможность его редактирования.

Добавление функционала для редактирования, удаления сообщений, чатов и для отправки аудио, видео или фотографий в сообщениях.

**ОТЧЕТЫ О ВЗАИМОДЕЙСТВИИ С ЗАКАЗЧИКОМ**

| **Название** | **Описание** | **Дата** |
| --- | --- | --- |
| Постановка задачи | Описание проекта, конкуренты, основные задачи и пользователи. Основные требования к инструментам и технологиям. | 03.12.2021 |
| Функциональные и нефункциональные требования | Краткое тех. задание и перечисление требований к проекту. Анализ конкурентов и функционала. | 03.12.2021 |
| План задач и распределение ролей | Распределение ролей, ответственности, матрица компетенций, таблица задач по проекту. | 03.12.2021 |
| Архитектура решения | Описание и назначение программы | 16.01.2022 |
| Проектирование: статические и динамические модели | Модель данных, диаграмма классов, диаграмма компонентов, диаграмма ER, use-case диаграмма. | 16.01.2022 |
| Тестирование и приёмка | Баг-репорт, приёмное тестирование | 15.03.2022 |

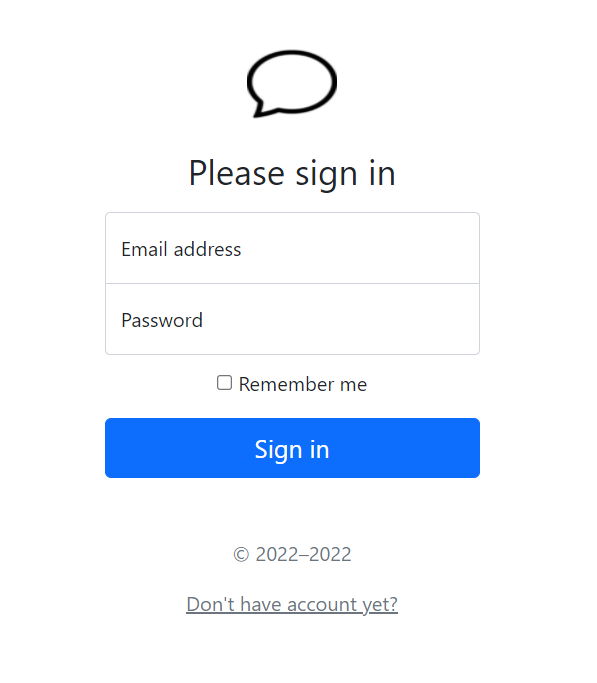
*Таблица 7. Отчёты о взаимодействии с заказчиком*

**ОТЗЫВ ЗАКАЗЧИКА**

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

* Вход в систему
  + Авторизация

Перейдя по ссылке <https://wizardchat.herokuapp.com/login> или нажав на кнопку Login/Logout на главной странице откроется следующая страница.

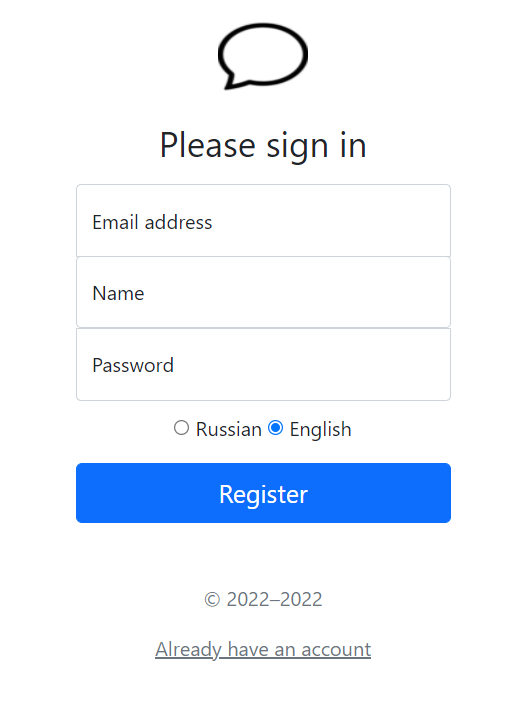


*Рисунок 10. Страница авторизации*

Здесь вы можете ввести свою почту и пароль, после чего вас вернет на главную страницу. Система сообщит вам, если вы ввели не существующую почту или неправильный пароль.

* + Регистрация

Перейдя по ссылке <https://wizardchat.herokuapp.com/register> или нажав на надпись “Don’t have account yet?” попадем на следующую страницу регистрации



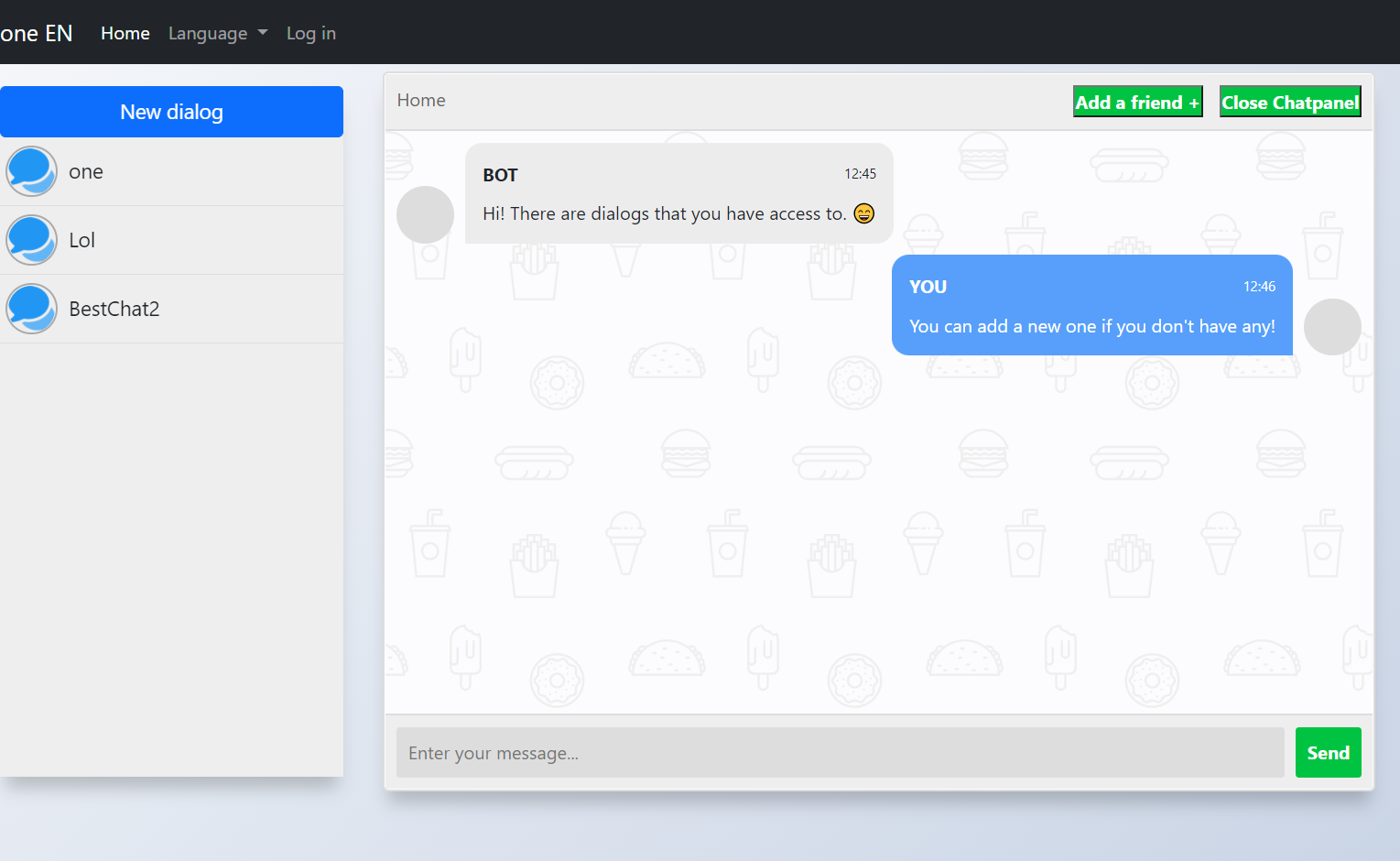
*Рисунок 11. Страница регистрации*

Вы можете ввести сюда свою почту, имя, пароль и выбрать ваш язык. Страница проводит проверку на корректность данных и сообщает, если данная почта уже используется. После успешного ввода данных вас перенесет на страницу авторизации, где нужно будет ввести вашу почту и пароль.

Если вы попали на эту страницу случайно, можете нажать на “Already have an account”, что переведет вас на страницу авторизации.

* Главное окно чатов

Перейдя по ссылке <https://wizardchat.herokuapp.com/home> или <https://wizardchat.herokuapp.com> вы попадете на главную страницу чатов. Сюда можно попасть нажав на кнопку Home на шапке страницы.



*Рисунок 12. Главная страница чатов*

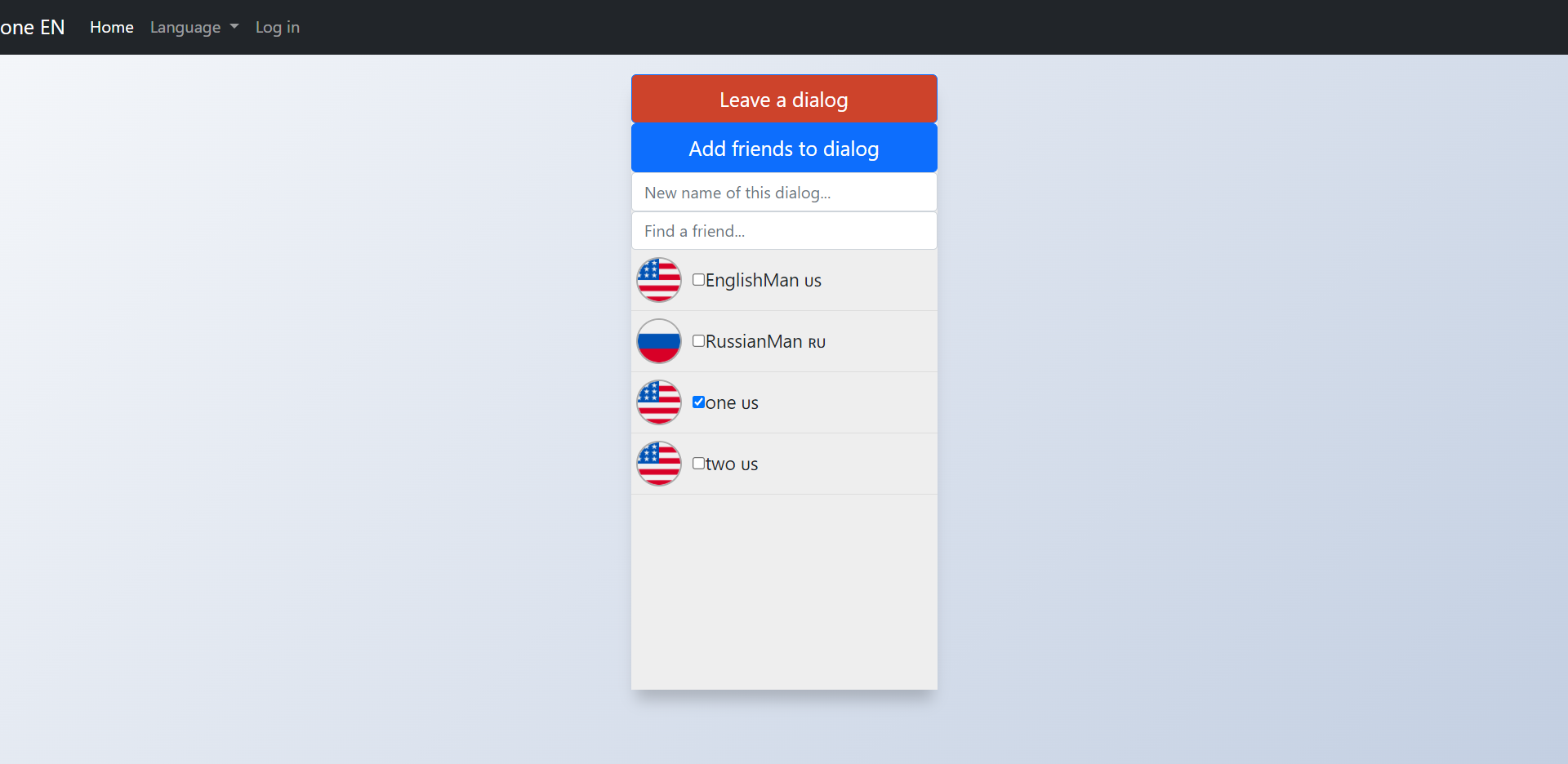
Здесь отображены чаты, в которых вы участвуете (их можно скрыть с помощью кнопки Close Chatpanel), ваше имя в шапке страницы, кнопка Logout/Login, кнопка для создания нового чата, а также небольшая инструкция в виде чата.

* + Отправка сообщений

Вы можете отправлять в данный чат сообщения и его могут увидеть люди, которые тоже в данный момент находятся на главной странице. Данные сообщения не сохраняются, поэтому при обновлении страницы они удаляются. Также, данные сообщения не переводятся и не озвучиваются с помощью соответствующих API.

* + Создание нового чата

Нажав на кнопку “Create a new dialog” или “Add a friend” вы откроете следующую страницу.

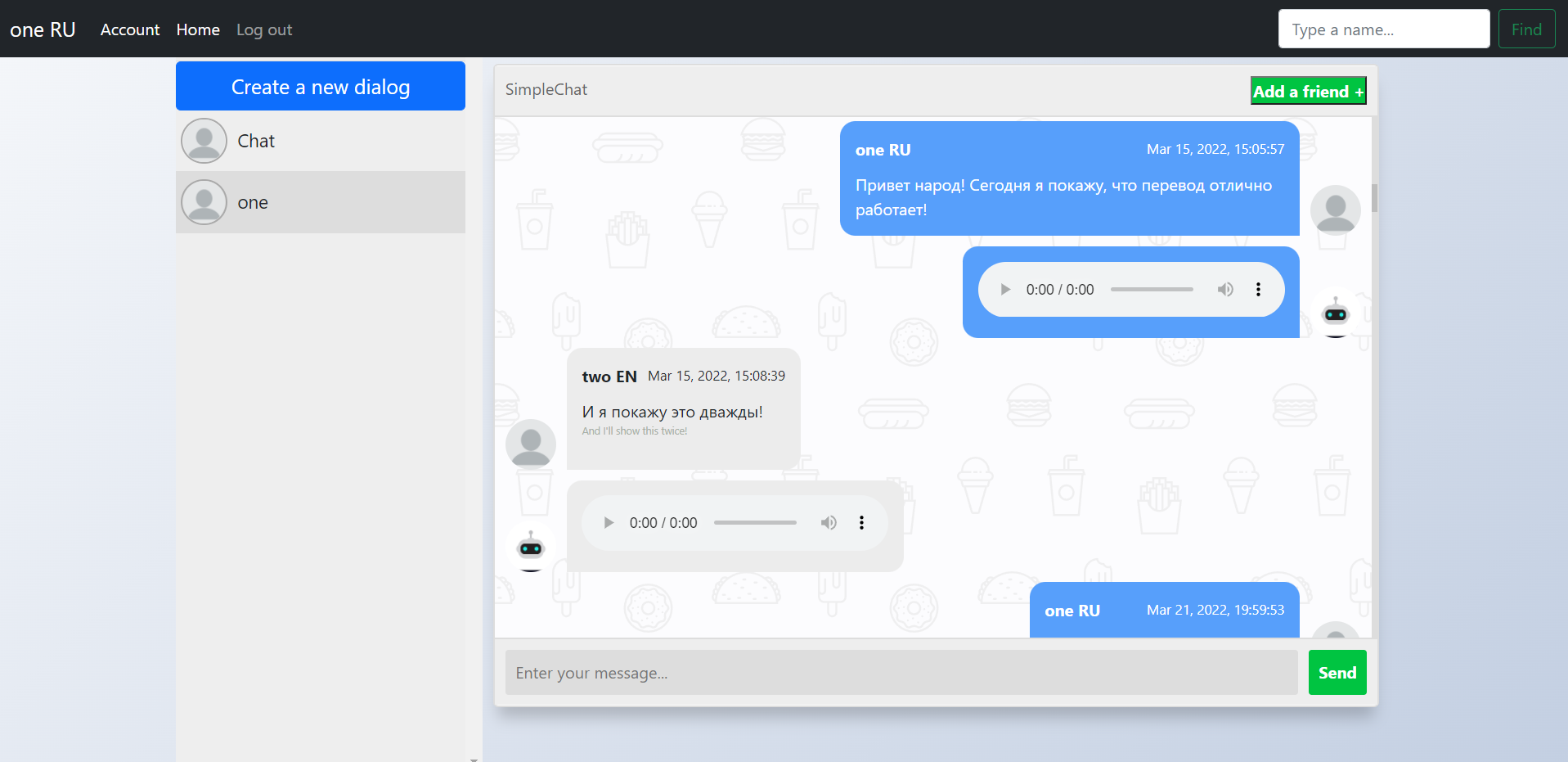


*Рисунок 13. Создание чата/Добавления друзей в чат*

Здесь вы можете, покинуть чат, задать имя нового чата (или новое название чата), выбрать друзей, которых вы хотите добавить с помощью соответствующих чекбоксов (или удалить если вы админ) и воспользоваться поисковой строкой среди всех пользователей мессенджера. После этого вас перенесет на главную страницу где вы сможете выбрать, соответствующий чат.

* Окно чата

Перейдя по ссылке <https://wizardchat.herokuapp.com/chat/><id> или нажав на соответствующий чат в списке вы перейдете на страницу чата. Она может выглядеть, например, следующим образом:



*Рисунок 14. Страница чата*

Если вы попытаетесь зайти на страницу чата, в котором вас нет, то вас перенесет обратно на главную страницу.

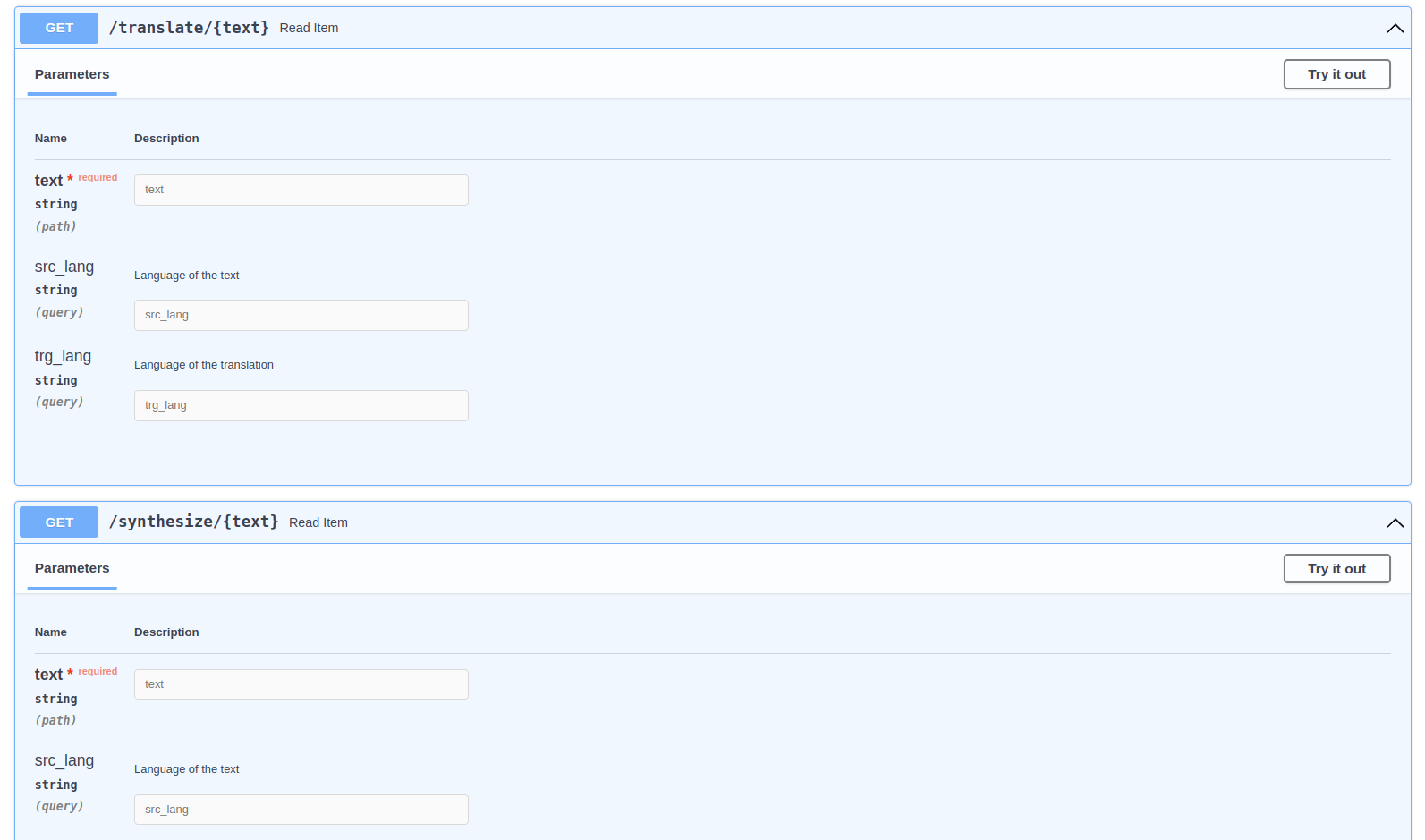
* + Отправка сообщений

Когда вы отправляете сообщение, оно передаются в API для перевода и синтезатора речи. После этого оно сохраняется в базе данных и отображается всем пользователям в чате на соответствующем языке (все русскоговорящие видят сообщения переведенные на русский или их русский оригинал). Если сообщение переведено, то снизу под ним серым шрифтом будет написан оригинал. Также, каждое сообщение озвучивается с помощью API и отправляется от пользователя с иконкой робота.

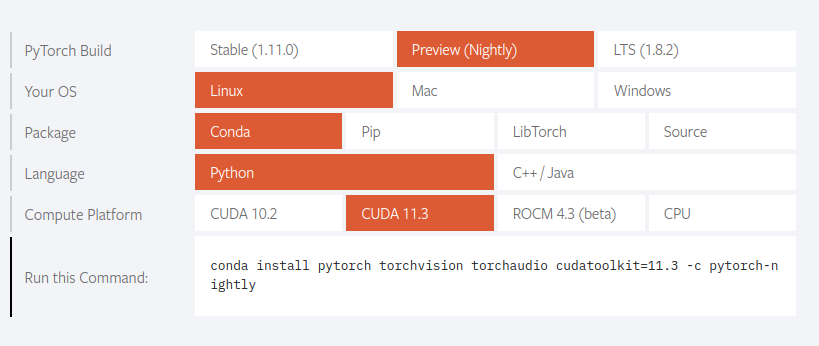
* + Добавление друзей

Находясь в чате, можно нажать на кнопку “Add a friend +” и перейти на соответствующую страницу добавления друзей. Если вы попытаетесь перейти на эту страницу не находясь в чате, вас перенесет обратно на главную страницу.

* Для руководства использования API сгенерирована документация [12].



*Рисунок 15. Часть документации API*

* Для развертывания API необходимо выполнить следующие шаги:
  + Воспользоваться гайдлайном с официального репозитория NeMo [13] и установить nemo и соответствующие оборудованию версии библиотек pytorch и cuda. Для корректной работы лучше использовать чистую среду conda и загрузить nightly версию pytorch [14]. 
  + Установить библиотеки wavfile, soundfile и noisereduce.
  + На данном этапе файл nemo\_util.py должен выполняться корректно. В этом случае производится установка fastapi и uvicorn. Для запуска API на localhost достаточно выполнить команду uvicorn main:app --host 0.0.0.0 --port <p>, где p – предпочтительный порт.
  + Для проверки работы можно воспользоваться эндпоинтами / или /dummy\_audio.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Mikolov T., Le Q. V., Sutskever I. Exploiting similarities among languages for machine translation //arXiv preprint arXiv:1309.4168. – 2013.
2. Mathur N. et al. Results of the WMT20 metrics shared task //Proceedings of the Fifth Conference on Machine Translation. – 2020. – С. 688-725.
3. PostgreSQL B. PostgreSQL //Web resource: http://www. PostgreSQL. org/about. – 1996.
4. Vaswani A. et al. Attention is all you need //Advances in neural information processing systems. – 2017. – Т. 30.
5. YouTokenToMe // Web resource: <https://github.com/VKCOM/YouTokenToMe>. – 2020
6. Nmt\_transformer6x6 en\_ru // Web resource: <https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/nmt_en_ru_transformer6x6>. - 2021
7. Fastpitch // Web resource: <https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/tts_en_fastpitch/>. – 2021
8. Hifigan // Web resource: <https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/tts_hifigan>. – 2021
9. Grinberg M. Flask web development: developing web applications with python. – " O'Reilly Media, Inc.", 2018.
10. Rai R. Socket. IO Real-time Web Application Development. – Packt Publishing Ltd, 2013.
11. Middleton N., Schneeman R. Heroku: up and running: effortless application deployment and scaling. – " O'Reilly Media, Inc.", 2013.
12. API documentation // Web resource: <https://editor.swagger.io/?url=https://raw.githubusercontent.com/BlessedTatonka/Translator_for_message_exchange/main/src/openapi.yaml>. - 2022
13. Nvidia NeMo // Web resource: <https://github.com/NVIDIA/NeMo>. - 2022
14. PyTorch installation // Web resource: <https://pytorch.org/>. - 2022
15. Nmt\_transformer6x6 en\_ru // Web resource: <https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/nmt_ru_en_transformer6x6>. - 2021