

ООО «Лаборатория Наносемантика»

УТВЕРЖДАЮ

СОГЛАСОВАНО

Руководитель 1

Руководитель 2

«____» _____ 2025 г.

«____» _____ 2025 г.

Программный комплекс

«ИИ Жириновский»

(с модулем распознавания лиц, модулем распознавания голоса, модулем Unity)

Руководство пользователя и администратора

Руководство пользователя и администратора системы «ИИ Жириновский»: программа имитирует общение с известным политиком и включает модули распознавания лиц и голосов, Unity-интерфейс и голосовой движок. Описана установка, настройка и взаимодействие всех компонентов, а также возможные ошибки и их решение.
Подробнее: <https://habr.com/ru/articles/743394/>

АННОТАЦИЯ

Настоящий документ содержит руководство системного администратора и пользователя по работе с программным комплексом «ИИ Жириновский», включающим модули распознавания персон по лицу и голосу, модуль Unity, а также модули распознавания речи (ASR) и синтеза речи (TTS).

Раздел «Инструкция администратора» предназначен для системного администратора. Здесь описаны шаги по установке программного комплекса, его технические характеристики, а также архитектура и взаимодействие модулей между собой.

Раздел «Инструкция для пользователя» предназначен для двух ролей пользователя:

- Пользователь с правами администратора (приводится руководство по управлению программным комплексом, в том числе настройке отображения и поведения аватара, настройке микрофона, нейросети).
- Конечный пользователь (описывается процесс общения с программным комплексом).

СОДЕРЖАНИЕ

1	Общие сведения.....	5
1.1	Наименование программного комплекса.....	5
1.2	Назначение программного комплекса.....	5
1.3	Функции программного комплекса.....	5
2	Инструкция администратора.....	6
2.1	Требования к техническому и программному обеспечению (для 10 каналов).....	6
2.1.1	Требования к серверу для модулей GPT.....	6
2.1.2	Требования к серверу для модуля Computer Vision	6
2.1.3	Требования для модуля Unity.....	7
2.1.4	Требования к серверу для модуля Voice ID.....	7
2.1.5	Требования к серверу для модулей ASR/TTS	8
2.2	Установка и настройка модуля Unity	8
2.2.1	Структура приложения.....	8
2.2.2	Запуск приложения.....	9
2.2.3	Конфигурация приложения.....	9
2.3	Установка и настройка модуля распознавания голоса (VoiceID)	10
2.3.1	Описание модуля	10
2.3.2	Настройка модуля.....	11
2.4	Установка и настройка модуля Computer Vision	14
2.4.1	Описание модуля	14
2.4.2	Установка и настройка модуля	15
2.5	Установка и настройка внешнего модуля GPT.....	17
2.5.1	Описание модуля	17
2.5.2	Установка и настройка модуля	18
2.5.3	Тестирование и просмотр логов.....	18
2.5.4	API-запросы	18
2.6	Архитектурная схема Системы	22
3	Инструкция пользователя.....	24
3.1	Главный экран ассистента	24
3.2	Режим администратора программного комплекса.....	25
3.2.1	Создание контента.....	25
3.2.2	Настройка ассистента.....	29
3.3	Процесс общения конечного пользователя с ассистентом.....	31

3.4	Решение возможных проблем.....	32
3.4.1	Подключение или отключение микрофона во время работы приложения	32
3.4.2	При активации микрофона не показывается индикатор громкости и речь не распознается.....	32
3.4.3	Кнопка микрофона активируется, индикация звука есть, но распознавания не происходит	32
4	Сокращения, термины и определения	33

1 Общие сведения

1.1 Наименование программного комплекса

Полное наименование программы: Программный комплекс «ИИ Жириновский» (далее — Ассистент).

1.2 Назначение программного комплекса

Ассистент предназначен для генерации ответов на вопросы пользователей, имитируя стиль общения и риторику основателя и депутата ЛДПР В.В. Жириновского.

1.3 Функции программного комплекса

Ассистент выполняет:

- обработку вопроса и генерацию ответа (при помощи двух моделей GPT и модуля DialogOS) синтезированным голосом В.В. Жириновского;
- визуализацию аватара В.В. Жириновского;
- распознавание личностей по изображению (несколько известных личностей);
- распознавание личностей по голосу (несколько известных личностей).

2 Инструкция администратора

2.1 Требования к техническому и программному обеспечению (для 10 каналов)

2.1.1 Требования к серверу для модулей GPT

- CPU: не менее 8 ядер;
- RAM: не меньше 32 Гб (минимальный вариант), не менее 64 Гб (рекомендуемый);
- GPU VRAM: 32 Гб (минимальный вариант); 48 Гб (рекомендуемый вариант);
- Дисковое пространство: не менее 200 Гб;
- **SSD: 50 Гб;**
- Любая видеокарта с поддержкой CUDA Compute Capability 7.5 и выше; желательно с тензорными ядрами. Рекомендуются ускорители Nvidia V100, A100, H100 или серии RTX 20xx, 30xx, 40xx (модели 70, 80, 90), подробности доступны на сайте NVIDIA.
- Частота CPU: не менее 2200 Гц;
- ОС: Linux, рекомендуемый вариант – Ubuntu версии 20.04 (инструкция по настройке написана для данной версии ОС);
- Архитектура: x86-64.

2.1.2 Требования к серверу для модуля Computer Vision

Оптимальные (рекомендуемые) требования:

- ОС: Ubuntu 20.04;
- SSD: 30 Гб;
- CPU: не менее 8 ядер; не менее 2 ГГц каждое;
- GPU: 16 Гб (например, Nvidia RTX 3080Ti);
- RAM: 24 Гб.

Минимальные требования:

- SSD: 16 Гб;
- CPU: не менее 4 ядер;
- Частота CPU: не менее 2 ГГц каждое;

- GPU: 8 Гб;
- RAM: 12 Гб.

2.1.3 Требования для модуля Unity

- Процессор: Intel Core i5/i7 12th+ Gen;
- RAM: от 16 Гб;
- vRAM: не менее 12 Гб
- vCPU: не менее 4 ядер;
- Хранилище: SSD объёмом от 128 Гб;
- Видеокарта: дискретная уровня Nvidia GeForce RTX 3060 и выше;
- Операционная система: Windows 10/11;
- Постоянное подключение к Интернет 100 Мб/сек.;
- Клавиатура, мышь;
- Веб-камера разрешением FullHD (1920x1080) и выше с хорошей светосилой;
- Аудиосистема (колонки, наушники);
- Микрофон (желательно);
- Дисплей с разрешением FullHD и выше и яркостью от 250 кд/м2.

2.1.4 Требования к серверу для модуля Voice ID

- ОС: Ubuntu 20.04 и выше;
- CPU: не меньше 4 ядер (минимальный вариант), не менее 8 ядер (рекомендуемый);
- RAM: не меньше 8 Гб (минимальный вариант), не менее 16 Гб (рекомендуемый);
- GPU vRAM 6 Гб (минимальный вариант); не меньше 12 Гб (рекомендуемый вариант)
- Дисковое пространство: не меньше 16 Гб (минимальный вариант), 32 Гб (рекомендуемый);
- SSD: не менее 20 Гб.

2.1.5 Требования к серверу для модулей ASR/TTS

Требования предъявляются на каждый из модулей (отдельно на ASR и отдельно на TTS, т.е. эти данные рассчитаны на 1 модуль).

- ОС: Ubuntu 20.04;
- CPU: не меньше 8 ядер (минимальный вариант), не менее 16 ядер (рекомендуемый);
- vCPU: не менее 4 ядер;
- RAM: не меньше 12 Гб (минимальный вариант), не менее 24 Гб (рекомендуемый);
- Дисковое пространство: не меньше 30 Гб (минимальный вариант), 60 Гб (рекомендуемый);
- SSD: не менее 50 Гб;
- GPU VRAM: 6 Гб (минимальный вариант), не меньше 16 Гб (рекомендуемый вариант).

2.2 Установка и настройка модуля Unity

2.2.1 Структура приложения

Приложение представляет собой папку со следующими компонентами:

- исполняемый файл zvv.exe;
- вспомогательные файлы модулей, необходимые для работы ассистента

Конфигурационные параметры приложения хранятся в файле AppConfig в директории zvv_Data/AppData (см. рис. 1). Более подробная информация по файлоу конфигурации описана в разделе 2.2.3.

Имя	Дата изменения	Тип
MonoBleedingEdge	24.02.2025 20:33	Папка с файлами
zvv_BurstDebugInformation_DoNotShip	24.02.2025 20:33	Папка с файлами
zvv_Data	24.02.2025 20:33	Папка с файлами
UnityCrashHandler64	24.02.2025 20:33	Приложение
UnityPlayer.dll	24.02.2025 20:33	Расширение при...
zvv	24.02.2025 20:33	Приложение

Рисунок 1 — Структура корневой папки приложения

Приложение использует доступ в интернет для работы с модулями ASR, DialogOS, TTS.

2.2.2 Запуск приложения

Сначала необходимо запустить исполняемый файл `zvv.exe`. Через некоторое время на экране появится окно ассистента. Первый запуск ассистента может занимать немного дольше времени, чем при повторных запусках. В зависимости от технических характеристик устройства запуск может занимать приблизительно от 10 до 40 секунд, в редких случаях дольше. Когда на экране появится аватар и начнется воспроизведение анимаций, ассистент запущен и готов к работе.

2.2.3 Конфигурация приложения

В папке `zvv_Data/AppData` располагается файл конфигурации `AppConfig`, который содержит набор параметров, позволяющих корректировать работу ассистента (рис.2).

Данный файл можно открыть любым текстовым редактором.

```
{
  "ASR Configuration": {
    "address": "asr-prod.nanosemantics.ai:443",
    "token": "cokHRqYE**2Wv2MZrm#e!2J#6^@h83Akxr6T59Qz",
    "language": "ru",
    "sampleRate": 16000,
    "secure": 1,
    "channels": 1,
    "usePunctuation": 1,
    "partialResultMode": 1,
    "aggressivenessMode": 0,
    "speechIncompleteTimeoutMs": 300,
    "speechCompleteTimeoutMs": 300
  },
  "DOS Configuration": {
    "urlDOSPostPrefix": "https://capi.dos.nanosemantics.ai/api/v1/Chat.",
    "botUID": "4f93368e-b219-4a46-b17d-2a434e076ae7",
    "language": "ru"
  },
  "TTS Configuration": {
    "address": "tts-prod.nanosemantics.ai:443",
    "token": "cokHRqYE**2Wv2MZrm#e!2J#6^@h83Akxr6T59Qz",
    "sampleRate": 22050,
    "voice": "zhirinovskiy2",
    "channels": 1,
    "pitch": 1.0,
    "speed": 1.1,
    "volume": 1.0,
    "endSilenceMs": 100,
    "secure": 1
  }
}
```

Рисунок 2 — Параметры файла `AppConfig`

Данный файл содержит в себе различные параметры конфигурации ассистента, например, параметры для доступа к модулям ASR, DOS, TTS с указанием их адресов и токенов доступа, а также дополнительные параметры для работы с модулями.

Данная конфигурация создается и настраивается техническими специалистами и предоставляется вместе с остальными файлами.

В большинстве случаев внесение изменений в конфигурацию системным администратором не предполагается.

2.3 Установка и настройка модуля распознавания голоса (VoiceID)

2.3.1 Описание модуля

Модуль VoiceID позволяет распознавать голоса, которые заранее были сохранены в базе данных ассистента в виде векторов. На вход в модуль поступает аудиофайл, который затем здесь обрабатывается. В результате обработки получается вектор характеристик голоса, который поочередно сравнивается с векторами каждой персоны в базе данных. В качестве ответа выбирается наиболее близкая персона ко входному вектору, если косинусная близость удовлетворяет заранее заданному порогу.

Для работы модуля требуется N секунд чистого голоса (очищенного от перерывов между речью). Число N задается при развертывании модуля и по умолчанию составляет 5 секунд.

Взаимодействие с модулем возможно либо через запросы из приложения Unity, либо через визуальный интерфейс Swagger.

При общении пользователя с ассистентом модуль Unity посылает запрос к модулю VoiceID и получает в качестве ответа имя персоны, которая общается с ассистентом (только для тех персон, чьи образцы голосов имеются в базе данных ассистента).

Voice ID работает по следующему принципу:

- с помощью скриптов преобразовываются данные (перевод файла в формат .wav);
- преобразованные данные сохраняются в файлы `embeddings.npy` (эмбеддинги с голосами) и `speakers.npy` (имена спикеров); у спикеров и эмбеддингов индексы одинаковые (например, нулевому спикеру соответствует нулевой эмбеддинг);
- поиск говорящего осуществляется по новому вектору: на вход берется новый вектор и рассчитывается косинусная близость к каждому слепку голоса из файла `embeddings.npy`.

2.3.2 Настройка модуля

Для настройки нужно изменить следующие данные в конфигурационном файле ``config.yaml``:

1. ``config.cur_dir`` — полный путь до текущей папки с проектом;
2. ``config.search_threshold`` — минимальный порог косинусной близости для поиска (актуально для веб-сервиса на FastAPI);
3. ``config.use_vad`` — использовать ли VAD (актуально для веб-сервиса на FastAPI);
4. ``config.vad_path`` — путь до yaml-конфига VAD (актуально для веб-сервиса на FastAPI);
5. ``embedder.device`` — устройство, на котором происходит вычисление вектора голоса.

Также нужно отредактировать конфигурационный файл для VAD (``/voice_id/vad_model/conf.yaml``):

1. ``vad_config.device`` — устройство, используемое для VAD;
2. ``model.ckpt_path`` — полный путь до папки ``SA_better_mel_ng`` из директории ``vad_model``.

Предобработка данных

Данные должны иметь следующую структуру:

```
...
root
  speaker1
    1.wav
    2.wav
    3.wav
    ...
  speaker2
    ...
...
```

Конвертация в wav 16кГц:

```
...
import ffmpeg
import os
import yaml
from vad.inference import VAD
```

```
from pydub import AudioSegment
from scripts.converter import TransformAudio
```

Путь до конфига Voice ID

```
config_path = "/voice_id_simple/config.yaml"
```

Путь до данных:

```
path_to_audio = ".../audios"
converter = TransformAudio(config_path)
```

Конвертация аудио: обрезка до 40 секунд, удаление оригинальных файлов и сохранение новых:

```
converter.transform_folder(path_to_audio, length_threshold=40,
delete_audio=True)
...
```

Получение файлов с векторами:

```
...

import torch
import torchaudio
import os
import yaml
import numpy as np
import numpy.typing as npt
from model.features import Fbank, InputNormalization
from model.ECAPA_TDNN import ECAPA_TDNN
from scripts.get_embeddings import Model
from typing import Tuple
from scripts.gen_embeddings import gen_embeddings
```

Путь до конфига Voice ID:

```
config_path = "/voice_id_simple/config.yaml"
```

Путь до данных:

```
path_to_audio = ".../audios"
```

```
path_to_save = "voice_id_simple/data"
model = Model(config_path)
```

Сохранение файлов `speakers.npy` и `embeddings.npy` в `path_to_save`:

```
gen_embeddings(model, path_to_audio, path_to_save)
...
```

Поиск вектора в базе:

```
...

import torch
import torchaudio
import os
import yaml
import numpy as np
import numpy.typing as npt
from model.features import Fbank, InputNormalization
from model.ECAPA_TDNN import ECAPA_TDNN
from scripts.get_embeddings import Model
from typing import Tuple
from scripts.search_embeddings import Searcher
```

Путь до конфига Voice ID:

```
config_path = "/voice_id_simple/config.yaml"
model = Model(config_path)
```

Путь до файлов с данными:

```
path_to_emb = "voice_id_simple/data/embeddings.npy"
path_to_targets = "voice_id_simple/data/speakers.npy"
searcher = Searcher(path_to_emb, path_to_targets)
```

Загрузка данных из файлов:

```
searcher.load_embeddings()
```

Получение вектора голоса:

```
vector = model.get_embed("../audio.wav")
```

Поиск образца голоса по базе (минимальная косинусная близость 5 выводит самый близки образец):

```
speaker, cos_sim = searcher.search_vector(vector, threshold=0.5)
...
```

Работа с готовым модулем:

После разворачивания модуля к нему можно подключиться либо по внешнему адресу (например, `www.voice_id_simple.nanosemantics.ai`), либо по внутреннему адресу (`http://127.0.0.1:5002`, где `5002` — номер порта).

Аудио принимается либо файлом через `identification/identify`, либо потоком/микрофоном через `127.0.0.1:5002/ws_endpoint/ws_stream`. Примеры использования потока в `ws_client.py` (а также `stream_example.py`).

Модуль возвращает словарь вида:

```
...
{"person_voice": Персона, "sim": Косинусная близость образца к персоне}
...
```

Если косинусная близость ниже определенного порога (в конфигурационном файле), то возвращается параметр `None` (в таком случае персону определить невозможно).

2.4 Установка и настройка модуля Computer Vision

2.4.1 Описание модуля

Модуль Computer Vision позволяет распознавать лица персон, которые заранее были добавлены в базу данных ассистента.

Данный модуль принимает изображение или видео, далее изображение или видео предобрабатывается, подается нейросетевой модели, которая получает координаты каждого обнаруженного лица. Каждое распознанное лицо сопоставляется с лицами, заведенными в базе данных Системы. Если совпадение найдено, модуль возвращает имя человека и уровень уверенности в определении.

2.4.2 Установка и настройка модуля

Скачайте веса с диска или dgx_storage по ссылке <https://disk.yandex.ru/d/RCUUpZeEt3NgEA> и поместите их в директорию:

```
- `/person_detector/*` в `./services/yoloAPI/app/core/person_detection/models/`  
```bash  
sudo docker-compose up --build
```
```

Для более удобной разработки можно создать окружение:

```
```bash  
python3.9 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```
```

Фактически в локальном окружении нет необходимости, так как запуск осуществляется в Docker-контейнере, и есть hot reload (можно после запуска контейнера править код, и он будет подгружаться). Код прокинут в контейнер через volume, но для корректной работы автодополнения и анализа кода в IDE рекомендуется установить зависимости в локальном окружении.

На `localhost:8001` будет доступно API.

Gradio интерфейсы для фото и видео доступны соответственно в `/gradio_image` и `/gradio_video`. Код для них лежит в директории `./services/yoloAPI/app/gradio_interface/`.

Фронтэнд версия API видео доступна в `/`, код в `./services/yoloAPI/app/routers/video.py`

Методы API

Основной необходимый метод API - `/detect`.

Пример использования curl:

```
```bash  
curl -X 'POST' \
 'http://localhost:8001/detect' \
 -H 'accept: application/json' \
 -H 'Content-Type: multipart/form-data' \
 -F 'file_list=@test1.jpg;type=image/jpeg' \
 `
```

```
-F 'download_image=false'
...
```

`download\_image` - возврат размеченного изображения. Сейчас этот параметр не используется, но можно его восстановить.

**Формат ответа:**

```
``json
{
 "data": [
 [
 {
 "class": 0,
 "class_name": "person",
 "bbox": [
 243,
 268,
 256,
 309
],
 "confidence": 0.47476664185523987,
 "person_name": null,
 "person_score": 0,
 "face_bbox": null,
 "emotion": null,
 "age": null,
 "is_speaking": false,
 "timestamp": false
 },
 ...
]
],
 "timestamp": 1701695910.3713672
}
...
```

## **Идентификатор людей**

Сейчас в индексе `jirinovsky7.pkl` имеются следующие известные персоны:



...

'Треф Герман Оскарович': 'gref',  
'Кириенко Сергей Владиленович': 'kirienko',  
'Корчевников Борис Вячеславович': 'korchevnikov',  
'Кошелев Владимир Алексеевич ': 'koshelev',  
'Миронов Сергей Михайлович': 'mironov',  
'Мишустин Михаил Владимирович': 'mishustin',  
'Песков Дмитрий Сергеевич': 'peskov',  
'Попов Евгений Георгиевич': 'popov',  
'Познер Владимир Владимирович': 'pozner',  
'Путин Владимир Владимирович': 'putin',  
'Силуанов Антон Германович': 'siluanov',  
'Скабеева Ольга Владимировна': 'skabeeva',  
'Слуцкий Леонид Эдуардович': 'slutsky',  
'Собчак Ксения Анатольевна': 'sobchak',  
'Соловьев Владимир Рудольфович': 'solovyov',  
'Володин Вячеслав Викторович': 'volodin',  
'Ворсобин Владимир Владимирович': 'vorsobin',  
'Зюганов Геннадий Андреевич': 'zyuganov'

...

## **2.5 Установка и настройка внешнего модуля GPT**

### **2.5.1 Описание модуля**

Модуль GPT используется для генерации ответа на запрос пользователя при помощи нейросетей.

Модуль делает запросы через DialogOS при помощи API.

## 2.5.2 Установка и настройка модуля

1. Скачайте файлы `zhirinovsky.tar.gz` и `docker-compose.yml`.
2. Загрузите Docker-образ из терминала командой `docker load < zhirinovsky.tar.gz`.
3. Перейдите в папку, где лежит файл `docker-compose.yml` и запустите docker-контейнер при помощи команды `docker-compose up -d`.

## 2.5.3 Тестирование и просмотр логов

Для тестирования можно отправить запрос к API при помощи `curl`:

```
curl -X 'POST' \
 'http://localhost:12400/v1/chat/completions' \
 -H 'accept: application/json' \
 -H 'Content-Type: application/json' \
 -d '{
 "model": "model",
 "messages": [{"role": "user", "content": "Может ли искусственный интеллект
управлять Россией?"}],
 "temperature": 0.4
 }'
```

Для просмотра логов можно использовать стандартные средства Docker:

```
docker logs -f zhirinovsky-stand.
```

## 2.5.4 API-запросы

### 1. Общие положения

В качестве транспорта для взаимодействия с модулем используется протокол HTTP. Данные сериализуются с помощью JSON в кодировке UTF-8.

### 2. Запрос

Модуль принимает POST-запросы по следующему URL: `http://localhost:12400/v1/chat/completions` (при обращениях с сервера, на котором развёрнут модуль).

### 3. Заголовки

Помимо стандартных заголовков необходимо учесть:

- Формат содержимого запросов (Content-Type) – `application/json`.

### 4. Аргументы запроса

Имя аргумента	Тип аргумента	Обязательность	Комментарий
model	string	Да	Идентификатор модели.
messages	string   array<object>	Да	История сообщений данного диалога в формате массива словарей вида {"role": X, "content": Y}, где role – роль автора реплики ("user" или "assistant"), content – содержание реплики.
max_tokens	integer	Нет	Максимальное число токенов, которые должны быть сгенерированы в продолжение диалога.  Общее число входных и сгенерированных токенов не может быть больше длины контекста, присущей данной модели (в данном случае – 2048 токенов).
temperature	number	Нет	Значение между 0 и 2. Более высокие значения (например, 0.8) порождают более случайный ответ, тогда как более низкие (например, 0.2) порождают более детерминированный ответ.  Рекомендуется менять либо temperature, либо top_p, но не оба одновременно.
top_p	number	Нет	Значение между 0 и 1. Альтернатива температуре, ядерное семплирование. Например, значение 0.1 значит, что порождаются только токены с суммарной вероятностью 10%.  Рекомендуется менять либо temperature, либо top_p, но не оба одновременно.
n	integer	Нет	Количество вариантов продолжений диалога на один вход.
stop	(array<string>   string)	Нет	До четырёх последовательностей символов, после генерации которых модель прекращает дальнейшую

Имя аргумента	Тип аргумента	Обязательность	Комментарий
			генерацию.
stream	boolean	Нет	Если флаг установлен, ответ будет посылаться по частям, как только соответствующие токены будут готовы. Конец потока обозначается так:  data: [DONE]

## 5. Коды состояния

В случае успешной обработки запроса модуль возвращает ответ с кодом 200.

## 6. Ответ на запрос

Ответом является JSON-объект со следующими полями:

Имя аргумента	Тип аргумента	Комментарий
id	string	Уникальный идентификатор ответа.
object	string	Тип ответа (всегда chat.completion).
created	integer	Временная метка Unix (в секундах) создания ответа.
model	string	Идентификатор модели, использованной для порождения ответа.
choices	array<object>	Массив с ответом модели (см. ниже).
usage	object	Статистики ответа: число порождённых токенов, число токенов в промпте, общее число токенов.

Каждый элемент массива в объекте choices является JSON-объектом со следующими полями:

Имя аргумента	Тип аргумента	Комментарий
index	integer	Номер варианта ответа в массиве.
message	object	Ответ модели: role – роль автора реплики (user или assistant), content – содержание реплики.
finish_reason	string	<ul style="list-style-type: none"> <li>Причина остановки генерации токенов: stop,</li> </ul>

Имя аргумента	Тип аргумента	Комментарий
		<p>если модель дошла до естественного конца ответа или переданного в аргументах <code>stop sequence</code>;</p> <ul style="list-style-type: none"> <li>• <code>length</code>, если модель дошла до максимального числа токенов, переданного в аргументах.</li> </ul>

## 7. Пример запроса

```
curl -X 'POST' \
 'http://localhost:12400/v1/chat/completions' \
 -H 'accept: application/json' \
 -H 'Content-Type: application/json' \
 -d '{
 "model": "model",
 "messages": [{"role": "user", "content": "Может ли искусственный интеллект
управлять Россией?"}],
 "temperature": 0.4
 }'
```

## 8. Пример ответа

```
{
 "id": "chatcmpl-36jUc3J9D3Lj6sRFsn9js8",
 "object": "chat.completion",
 "created": 1697211698,
 "model": "model",
 "choices": [
 {
 "index": 0,
 "message": {
 "role": "assistant",
 "content": " [neutral] Искусственный интеллект Жириновского не может
управлять Россией, потому что он не обладает физическим телом и не может принимать
решения."
 },
 "finish_reason": "stop"
 }
],
 "usage": {
 "prompt_tokens": 80,
 "total_tokens": 108,
 "completion_tokens": 28
 }
}
```

## 9. Пример запроса на Python

```
import requests

headers = {
```

```

"Content-Type": "application/json"
}

messages=[
 {"role": "user", "content": "Может ли искусственный интеллект управлять Россией?"},
 {"role": "assistant", "content": "[good] Конечно! Искусственный интеллект Жириновского – это мощный инструмент, который может принимать важные решения для развития страны. Он способен анализировать и принимать решения на основе данных и информации, предоставляемой ему от различных источников."},
 {"role": "user", "content": "А это не опасно?"},
]

json_data = {
 "model": "model",
 "messages": messages,
 "temperature": 0.4,
}

response = requests.post("http://0.0.0.0:12400/v1/chat/completions", headers=headers,
json=json_data)
print(response.json())

```

## 2.6 Архитектурная схема Системы

Схема архитектуры Системы представлена на рисунке 3.

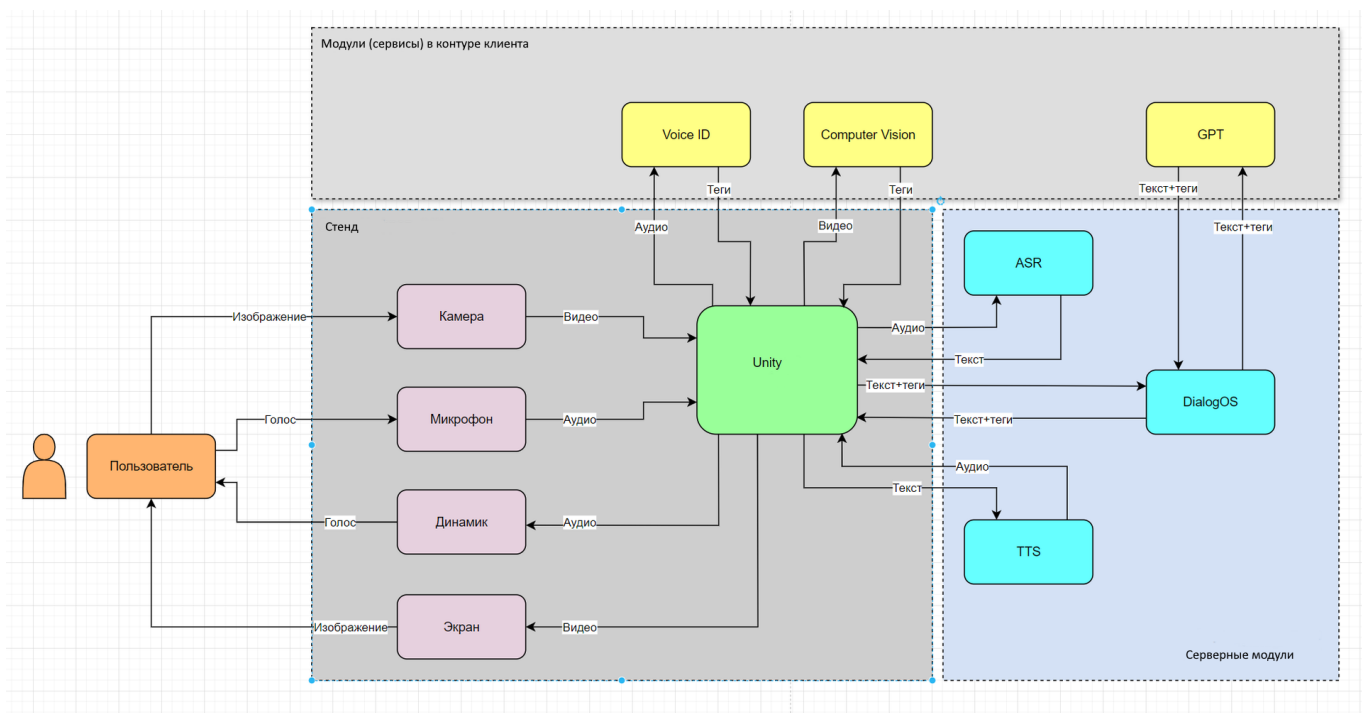


Рисунок 3 — Архитектура Системы

Алгоритм работы ассистента осуществляется следующим образом.

Пользователь передает голосовой запрос при помощи микрофона, а при помощи камеры вводится изображение говорящего человека. Аудио и видео передаются в модуль Unity, который взаимодействует со всеми другими модулями ассистента.

Аудио передается в модуль Voice ID, где аудиозапись сопоставляется с голосами персон, заранее занесенных в базу данных ассистента. Далее модуль Voice ID передает распознанные данные в виде тегов обратно в модуль Unity.

Видео передается в модуль Computer Vision, где видео обрабатывается. Полученные изображения сопоставляются с изображениями персон, заранее занесенных в базу данных. Далее модуль Computer Vision передает распознанные данные в виде тегов обратно в модуль Unity.

Для получения ответа на заданный вопрос загруженное аудио из модуля Unity передается в модуль ASR, где происходит перевод аудио в текст. Распознанный текст с тегами передается в модуль DialogOS, где происходит генерация ответа (в том числе при помощи внешнего модуля GPT). DialogOS передает сгенерированный текст ответа и теги анимации в модуль Unity. Модуль Unity передает сгенерированный текст ответа в модуль TTS, где происходит перевод текста ответа в аудиоформат (голос). Далее аудио голоса пересылается обратно в модуль Unity, где вся информация обрабатывается, а затем через динамик выводится пользователю. Дополнительно на экране отображается аватар В.В. Жириновского, синхронизированный со звучащим голосом.

## 3 Инструкция пользователя

### 3.1 Главный экран ассистента

После запуска ассистента (см. раздел 2.2.2) отображается главный экран ассистента (рис. 4).

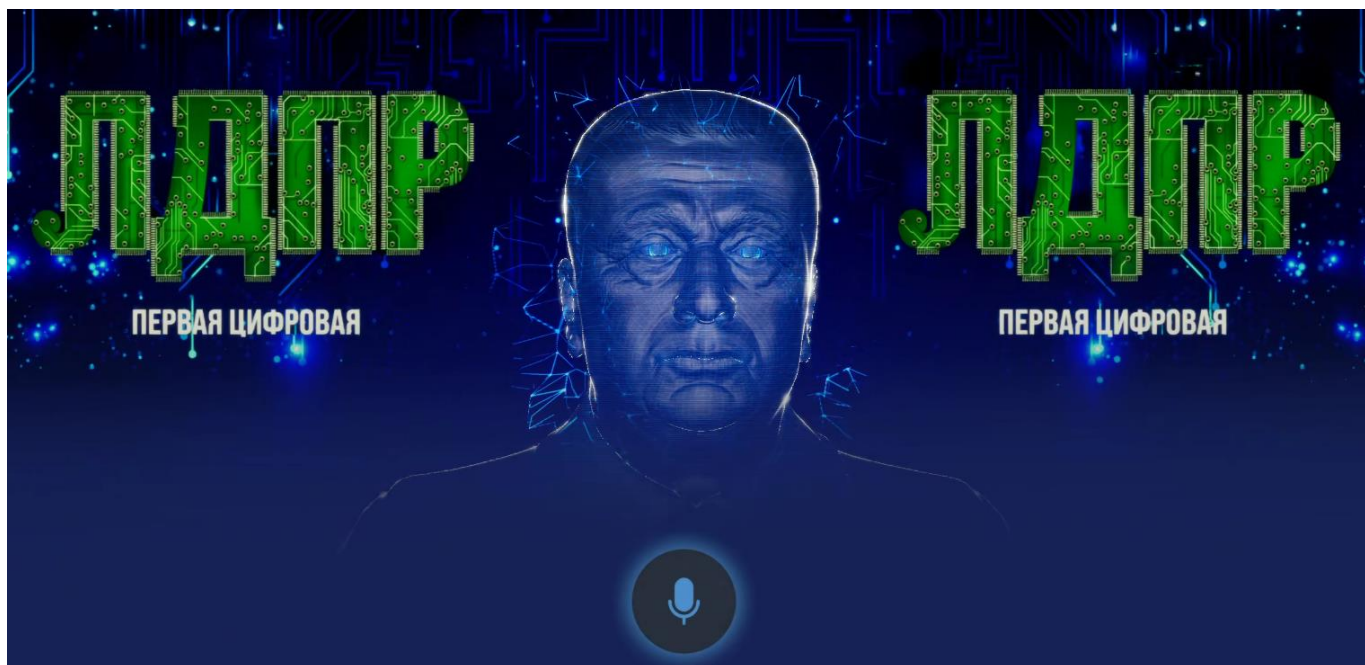


Рисунок 4 — Главный экран ассистента

В центре главного экрана отображается анимированный аватар, который проигрывает анимации в зависимости от своего текущего состояния.

Кнопка активации микрофона расположена в нижней части экрана и выполняет двойную функцию:

- запускает/останавливает запись голоса;
- показывает текущий статус записи (активна/неактивна).

Состояния кнопки микрофона:

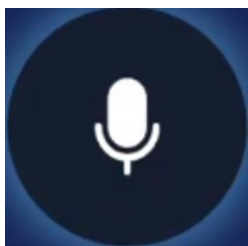


Рисунок 5 — Неактивное состояние

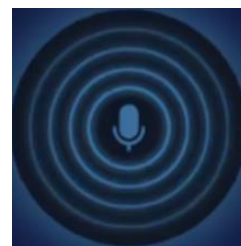


Рисунок 6 — Активное состояние



В неактивном состоянии (рис. 5) микрофон представляет из себя иконку микрофона белого цвета на фоне синего круга.

В активном состоянии (рис. 6) синий круг начинает пульсировать. Также в качестве индикации звука иконка микрофона заполняется голубым цветом снизу вверх в зависимости от громкости.

Субтитры (ответы ассистента) отображаются между аватаром и кнопкой микрофона.

Для управления настройками ассистента используются следующие горячие клавиши:

- Ctrl + I — вход в режим администратора программного комплекса.
- Ctrl + N — выбор микрофона, который выводит звук (рис. 7).

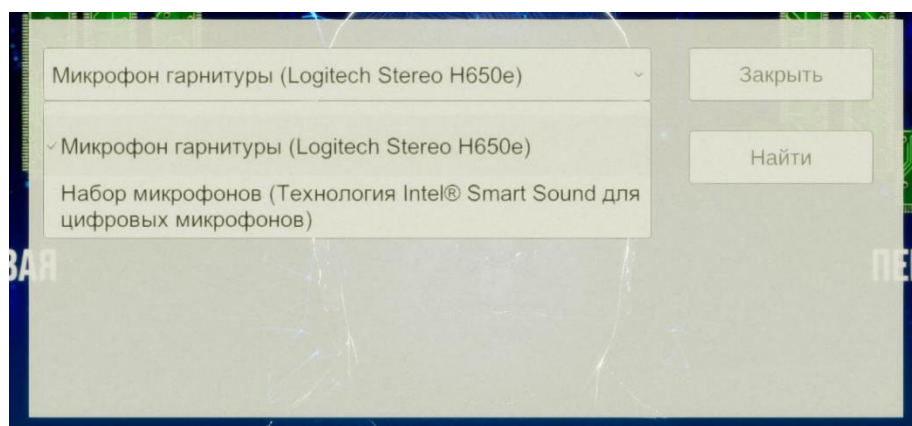


Рисунок 7 — Выбор микрофона

- ALT + ENTER + K — выход из полноэкранного режима.
- ESC — выход из приложения.

### 3.2 Режим администратора программного комплекса

После нажатия комбинаций клавиш CTRL + I на главном экране ассистента активируется «Режим администратора».

***Примечание.** Пока панель администратора активна, отключается кнопка, которая отвечает за управление записью с микрофона. Если нужно закрыть панель администратора, достаточно просто нажать на кнопку закрытия.*

#### 3.2.1 Создание контента

После входа в режим администратора в правом верхнем углу нажмите кнопку «Создание контента» (рис. 8).

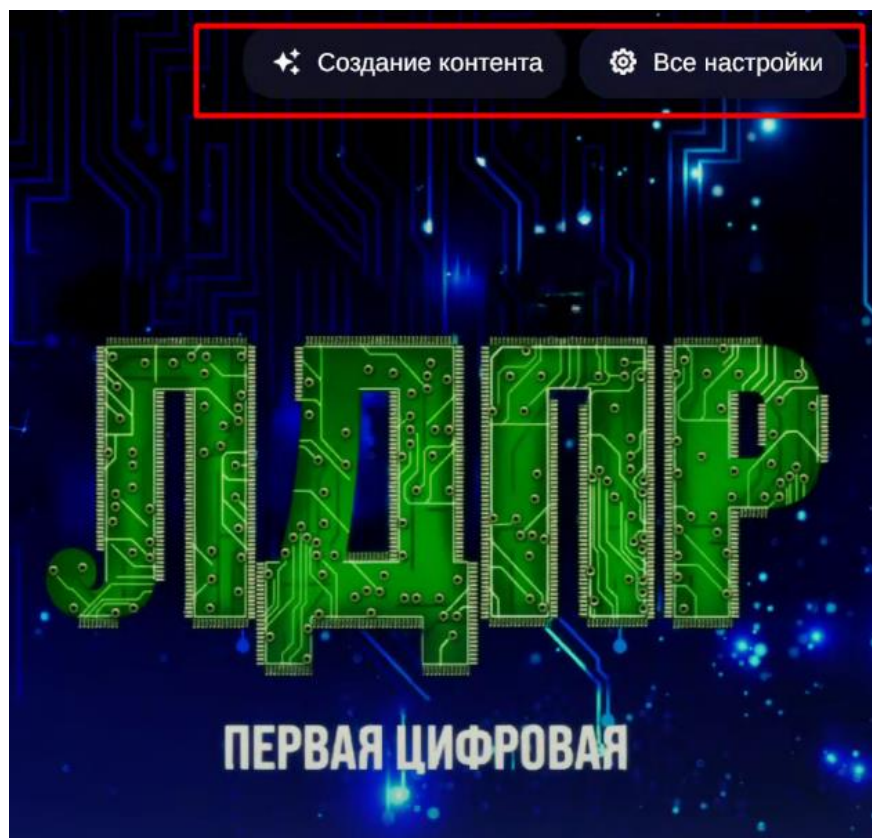


Рисунок 8 — Режим администратора

Появится окно «Создание контента». В окне представлены следующие секции для настройки запроса и ответа, для которого необходимо настроить реакцию ассистента (рис. 9):

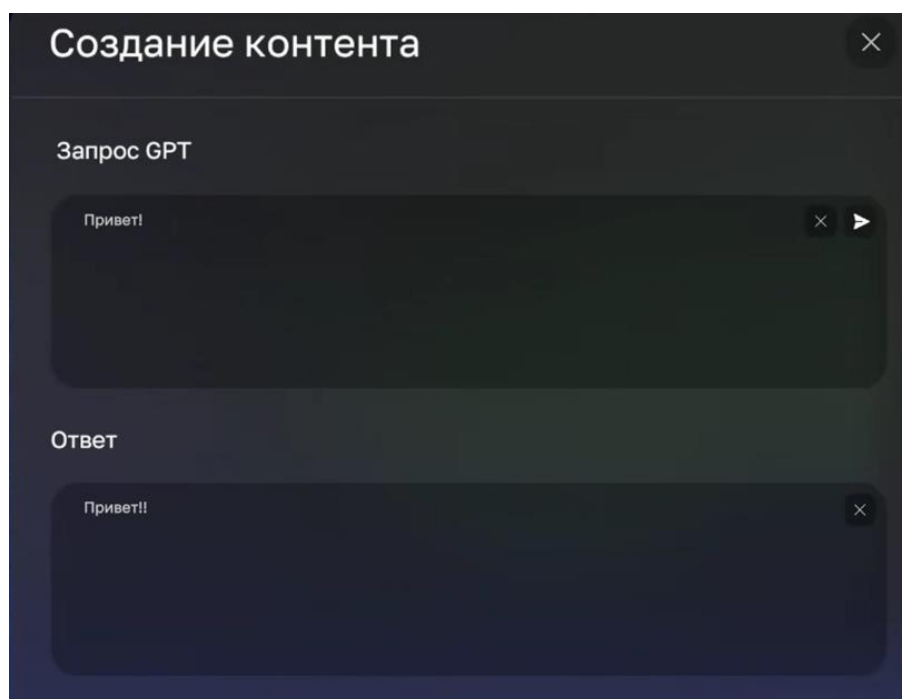



Рисунок 9 — Настройка запроса и ответа

- «Запрос GPT» — введите вопрос, который может задать конечный пользователь. Текст можно копировать и вставлять при помощи комбинаций клавиш CTRL + C и CTRL + V соответственно, также для выделения всего текста можно использовать комбинацию клавиш CTRL + A.

Для генерации ответа при помощи GPT необходимо нажать кнопку  (рис. 10). Если нажать на эту кнопку несколько раз, то ответы всегда будут разные. Таким образом, ответ можно выбрать на свое усмотрение.

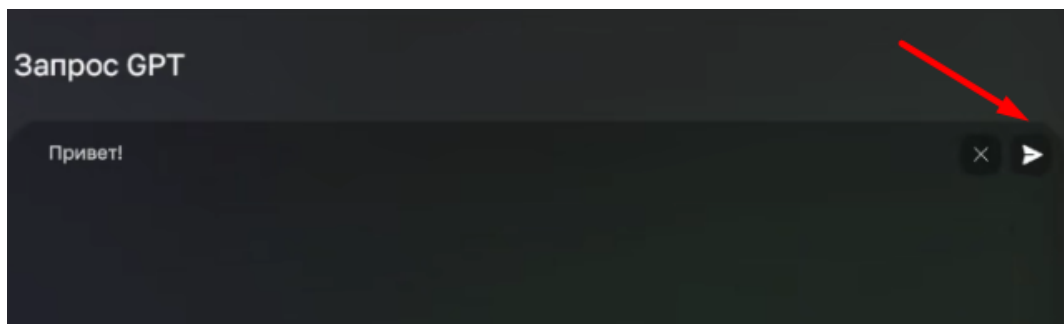


Рисунок 10 — Кнопка генерации ответа GPT

- «Ответ» — ответ на запрос, автоматически сгенерированный нейронной сетью GPT. Также можно вручную добавить любой текст (рис. 11).

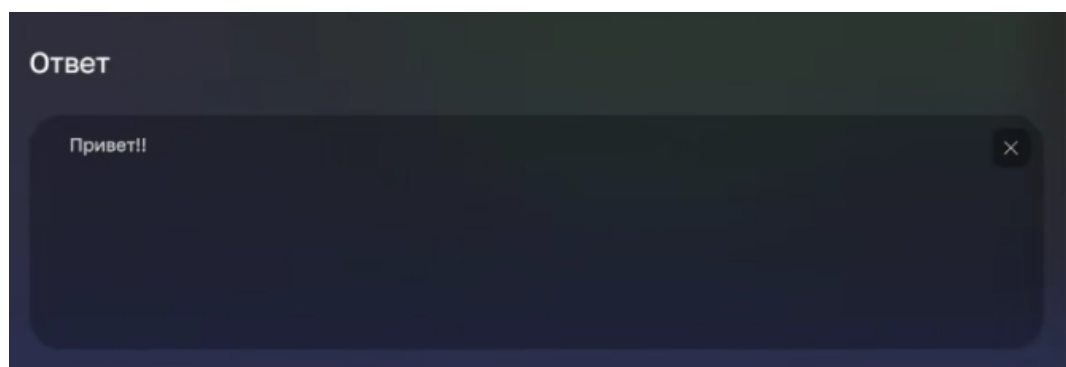


Рисунок 11 — Поле ответа на запрос

В секциях «Эмоции в аудио», «Эмоции в видео» (рис.12) настраиваются эмоции, которые будут воспроизводиться при ответе ассистента, который был указан в секции «Ответ» (рис. 9).

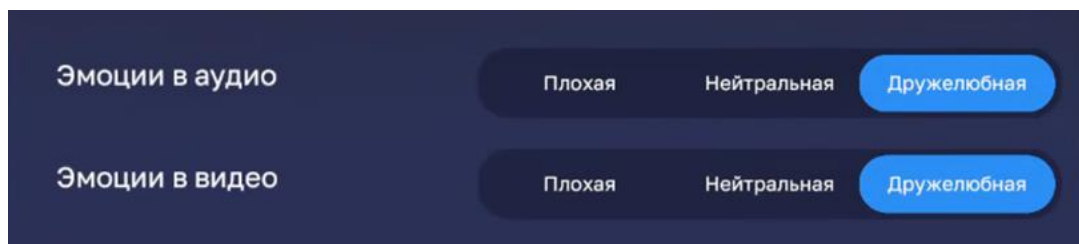


Рисунок 12 — Настройка эмоций

Ассистент поддерживает 3 вида эмоций (в аудио и видео) — плохая, нейтральная, дружелюбная (рис. 12). Здесь необходимо выбрать одну из эмоций, которая будет проигрываться при заданном выше ответе ассистента.

Ассистент также позволяет записывать свои ответы в виде аудио- или видеофайла. Для скачивания файлов нужно ввести текст в графу «Ответ» или использовать ответ, который сгенерирует нейросеть (см. рис.9), далее нажать на кнопку «Аудио» или «Видео» (рис. 13). Таким образом, когда ассистент проговорит указанный ответ голосом, видео- или аудиофайлы ответа будут добавлены в папки по адресу:

C:\Users\aktay\OneDrive\Рабочий стол\ZVV\zvv\_Data\AppData (рис. 14).

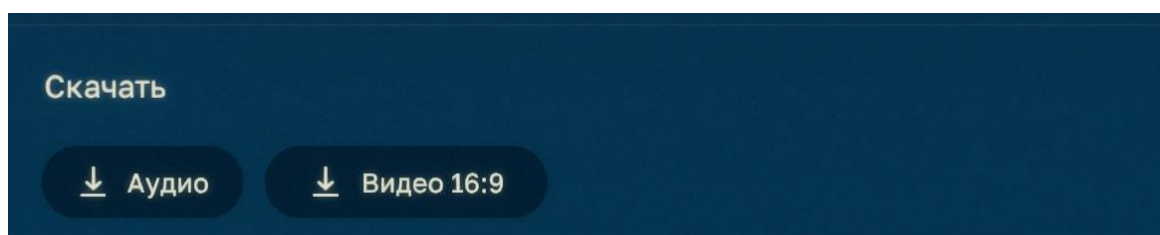


Рисунок 13 — Скачивание файлов

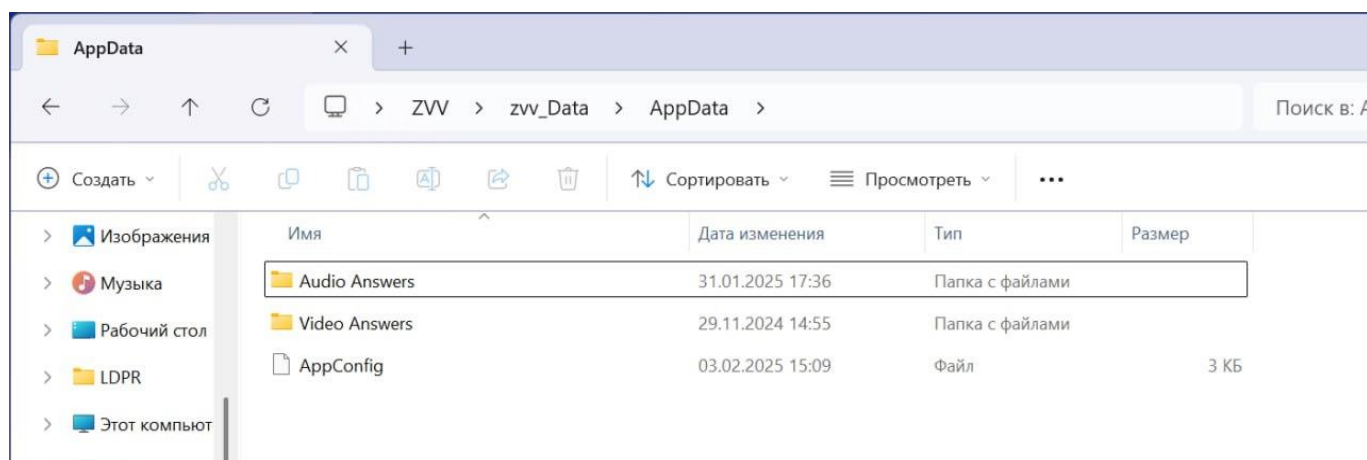


Рисунок 14 — Папки с аудио- и видеоответами

### 3.2.2 Настройка ассистента

На главном экране войдите в режим администратора, нажав клавиши Ctrl + I, далее в правом верхнем углу нажмите кнопку «Все настройки» (рис. 8).

В следующих секциях (рис. 15) можно настроить нейронную сеть:

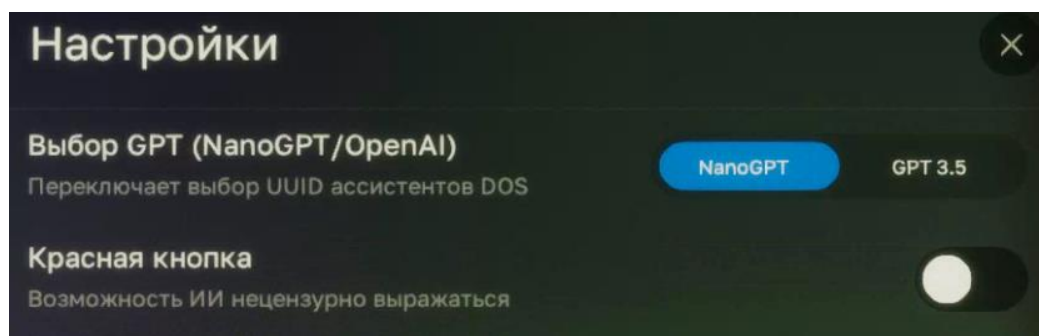


Рисунок 15 — Настройки нейросети

- «Выбор GPT (NanoGPT / OpenAI)» — выбор нейросети, которая будет отвечать на поставленные вопросы:
  - внутренняя сеть, разработанная компанией «Лаборатория Наносемантика»;
  - нейросеть от компании OpenAI.
- «Красная кнопка» — при включении кнопки ассистент может использовать нецензурную лексику в ответах на вопросы пользователей.

Программный комплекс также позволяет выбрать имя (из списка заведенных имен), и тогда ассистент будет обращаться к пользователю по имени.

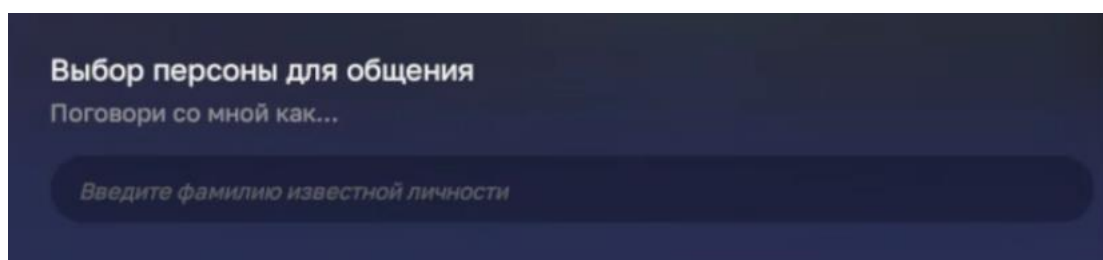


Рисунок 16 — Выбор персоны для общения

Для этого в секции «Выбор персоны для общения» введите латинскими буквами любое имя из таблицы ниже.

gref	Герман Греф
------	-------------

kirienko	Сергей Кириенко
korchevnikov	Борис Корчевников
koshelev	Владимир Кошелев
mironov	Сергей Миронов
mishustin	Михаил Мишустин
peskov	Дмитрий Песков
popov	Евгений Попов
pozner	Владимир Познер
putin	Владимир Владимирович Путин
siluanov	Антон Силуанов
skabeeva	Ольга Скабеева
slutsky	Леонид Слуцкий
sobchak	Ксения Собчак
solovyov	Владимир Соловьев
volodin	Слава Володин
vorsobin	Владимир Ворсобин
zyuganov	Евгений Зюганов

В секции «Дополнительно» можно включить / выключить субтитры для ответов ассистента:



Рисунок 17 — Настройка субтитров

### 3.3 Процесс общения конечного пользователя с ассистентом

Когда приложение будет готово к работе, аватар будет воспроизводить анимации покоя (Idle).

При нажатии на кнопку микрофона на главном экране:

- кнопка микрофона перейдёт в активное состояние;
- аватар посмотрит в центр экрана;
- начнётся проигрывание анимаций задумчивости (Thinking).

В данный момент можно начинать говорить в микрофон. Все записанные данные отправляются в модуль распознавания речи (ASR). Ассистент использует микрофон, выбранный в настройках операционной системы как устройство по умолчанию.

Также при обработке данных с микрофона используется система VAD (Voice Activity Detection) которая определяет, говорят ли в данный момент в микрофон или нет. Если после активации микрофона речь не будет зафиксирована в течение приблизительно 6 секунд, запись с микрофона будет остановлена, кнопка микрофона на главном экране перейдет в неактивное состояние, а аватар снова начнет проигрывать анимации покоя.

Когда пользователь закончит говорить, микрофон также будет отключен, кнопка перейдет в неактивное состояние, а аватар будет проигрывать анимацию того, что он услышал вопрос (Heard).

В некоторых случаях из-за наличия фоновых шумов, ассистент может не распознать отсутствие / окончание речи и будет продолжать запись. В таком случае достаточно вручную снова нажать кнопку микрофона или соответствующую кнопку на клавиатуре для завершения записи.

Затем вопрос будет отправлен в модуль DialogOS, а аватар снова будет проигрывать анимации задумчивости, тем самым обозначая, что идет обработка запроса.

После получения ответа от DialogOS определяется реакция ответа, подготавливаются субтитры (если настроены) и текст отправляется в модуль синтеза речи (TTS).

После получения синтезированной речи ассистент начинает озвучивать ответ, отображаются субтитры ответа (если настроены), а аватар начинает проигрывать анимации разговора (Talk).

После завершения ответа аватар снова начнет проигрывать анимации покоя.

Если в процессе обработки запроса снова будет активирована кнопка микрофона, то обработка запроса будет прервана. Также если в процессе озвучивания ответа будет активирована кнопка микрофона, озвучивание будет прервано. Таким образом, можно сразу задавать следующий вопрос без ожидания завершения ответа или просто прервать текущий вопрос / ответ.

Если на каком-либо этапе произойдет ошибка, аватар просто снова начнет проигрывать анимации покоя.

### **3.4 Решение возможных проблем**

#### **3.4.1 Подключение или отключение микрофона во время работы приложения**

При подключении/отключении микрофона во время работы ассистент может не подключиться к микрофону. В таком случае кнопка микрофона на главном экране либо не будет активирована, либо будет активна, но индикатор звука не будет ничего показывать, и речь не будет распознаваться.

Решение: перезапуск ассистента с проверкой, какой микрофон выбран основным в настройках операционной системы.

#### **3.4.2 При активации микрофона не показывается индикатор громкости и речь не распознается**

Сначала необходимо проверить, что нужный микрофон выбран в настройках операционной системы как основной.

В некоторых случаях антивирус может блокировать доступ приложения к микрофону. Решение: добавление ассистента в список исключений или список доверенных приложений у антивируса.

#### **3.4.3 Кнопка микрофона активируется, индикация звука есть, но распознавания не происходит**

Сначала необходимо проверить, присутствует ли у устройства соединение с интернетом.

Если доступ к интернету присутствует, но ответа все еще нет, значит вероятнее всего проблема заключается в сбое или недоступности со стороны сервисов ASR, DOS, TTS.



## 4 Сокращения, термины и определения

Аватар (анимированная 3d-модель)	— цифровой персонаж, похожий на В. В. Жириновского, созданный в трехмерной графике и способный двигаться благодаря анимации.
Ассистент	— программный комплекс «ИИ Жириновский», который может обрабатывать вопросы пользователей и выдавать ответ, имитируя стиль общения и риторiku основателя и депутата ЛДПР В.В. Жириновского.
ИИ	— сокр. от Искусственный интеллект.
Косинусная близость	— мера сходства между двумя векторами в многомерном пространстве, основанная на косинусе угла между ними.
ASR	— сокр. от Automatic Speech Recognition, автоматическое распознавание речи.
DialogOS	<p>— профессиональная платформа для разработки разговорного искусственного интеллекта компании «Лаборатория Наносемантика».</p> <p>DialogOS позволяет реализовать полный цикл разработки и поддержки виртуального ассистента, а именно:</p> <ul style="list-style-type: none"><li>— создать виртуального ассистента, который автоматически отвечает на вопросы пользователей;</li><li>— провести тестирование виртуального ассистента;</li><li>— подключить виртуального ассистента в различные каналы коммуникации, например, мессенджеры, вебсайты, мобильные приложения и IVR;</li><li>— дообучить виртуального ассистента путем добавления новых правил и примеров;</li><li>— просмотреть историю коммуникации виртуального ассистента с клиентами;</li><li>— просмотреть эффективность работы виртуального ассистента;</li><li>— управлять пользователями DialogOS.</li></ul>
GPT	— сокр. от Generative Pre-trained Transformer, программное обеспечение, которое использует технологию машинного обучения для обработки естественного языка и автоматической генерации текста на основе входных данных.
TTS	— сокр. от Text-to-Speech, преобразование текста в речь (синтез речи).

- Unity — кроссплатформенный игровой движок (модуль), который используется для создания интерактивного 3D-ассистента (аватара).
- VAD — сокр. от Voice Activity Detection, технология обнаружения голосовой активности, которая определяет, содержит ли аудиосигнал речь или нет.