

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## 1. Общая информация о проекте

Название: Разработка приложения для просмотра и управления видеоконтентом

Цель: Разработать и внедрить платформу, которая объединяет возможности видеохостинга и совместного просмотра, предоставляя пользователям единое и интуитивно понятное решение для загрузки, просмотра и совместного взаимодействия с видео в реальном времени, с фокусом на улучшение общего пользовательского опыта и удобства использования.

## 2. Функциональные требования к системе

### 2.1 Авторизация и управление профилем

- Пользователь должен иметь возможность регистрироваться и авторизовываться в приложении через email/пароль или используя сторонние сервисы (Google, Facebook).
- Возможность восстановления пароля через email.

### 2.2 Управление видеоконтентом

- Возможность загрузки видеофайлов в приложение.
- Возможность просмотра списка загруженных видео.
- Возможность одиночного просмотра видеоконтента.

Функции управления видео:

- Удаление видео.
- Создание комнаты для совместного просмотра данного видео.

### 2.3 Обработка видео (автоматическая после загрузки):

- Генерация превью.
- Распознавание речи из видео и суммаризация текста.
- Разделение на главы с таймкодами.
- Проверка на ненормативную лексику.

### 2.4 Совместный просмотр

- Создание комнаты для совместного просмотра видео с уникальной ссылкой/кодом приглашения.
- Удаление комнаты и выход из комнаты пользователем.
- Присоединение к комнате по ссылке/коду приглашения.
- Встроенный чат для текстового общения пользователей во время просмотра.

### **3. Нефункциональные требования**

#### **3.1 Платформенная поддержка**

- Приложение должно быть доступно на Android и iOS, а также в веб версии.

#### **3.2 UI/UX**

- Интуитивно понятный интерфейс, ориентированный на пользователей. Минимум шагов для выполнения основных действий.

#### **3.3 Производительность**

- Быстрая загрузка контента и отзывчивость интерфейса.

### **4. Технические Требования**

#### **4.1 Бэкенд**

- Использование облачных технологий для хранения видео и обработки запросов.

#### **4.2 Фронтенд**

- Разработка на Flutter для обеспечения кроссплатформенной совместимости.

#### **4.3 Базы данных**

- Хранение пользовательских данных и информации о видео.

### **5. Требования к MVP версии приложения**

#### **5.1 Авторизация и управление профилем**

- Пользователь должен иметь возможность регистрироваться и авторизовываться в приложении через email/пароль.
- Возможность восстановления пароля через email.

## 5.2 Управление видеоконтентом

- Возможность просмотра списка загруженных видео.
- Возможность одиночного просмотра видеоконтента.

## 5.3 Обработка видео (автоматическая после загрузки):

- Генерация превью.
- Распознавание речи из видео и суммаризация текста.
- Разделение на главы с таймкодами.
- Проверка на ненормативную лексику.

## 5.4 Совместный просмотр

- Создание комнаты для совместного просмотра видео с уникальной ссылкой/кодом приглашения.
- Присоединение к комнате по ссылке/коду приглашения.

## 5.5 Платформенная поддержка

- Приложение должно быть доступно на Android и iOS, а также в веб версии.

## 5.6 UI/UX

- Интуитивно понятный интерфейс, ориентированный на пользователей. Минимум шагов для выполнения основных действий.

## 5.7 Производительность

- Быстрая загрузка контента и отзывчивость интерфейса.

## Сценарий взаимодействия

Для описания сценария взаимодействия пользователя с системой была составлена таблица (Таблица 1). В таблице расписаны шаги по возможным действиям пользователя и ожидаемому результату. Построена по принципу ПМИ (Программа и методика испытаний).

Таблица 1 – Сценарий взаимодействия

Функциональность	Действия пользователя	Ожидаемый результат
<b>Как пользователь, я хочу иметь возможность авторизоваться в приложении, чтобы получить доступ к своему профилю и функционалу приложения</b>		
Авторизация	1. Открыть приложение	1. Открывается экран для авторизации с возможностью авторизоваться (если у пользователя есть аккаунт) или же возможностью зарегистрироваться. Для авторизации пользователю доступны поля с логином и паролем Для регистрации доступны поля с вводом имени пользователя, логина и пароля с его подтверждением. Для подтверждения почты или телефона необходимо пройти верификацию
	2. Пройти авторизацию/регистрацию	2. Авторизация проходит успешно, открывается профиль пользователя На странице профиля пользователю доступны его аватар, его имя и кнопка для выхода из аккаунта (а также разные опции типа настройки, не буду прописывать так как не будем пока реализовывать)
	3. Выйти со страницы профиля	3. Открывается главная страница приложения с загруженными видео (от всех пользователей или только свои) Около каждого видео есть кнопка с опциями. Пользователю доступны кнопки главной страницы, кнопка загрузки видео и кнопка для создания/присоединения к комнате
<b>Как пользователь, я хочу иметь возможность добавить видео в сервис по ссылке, чтобы делиться видеоконтентом</b>		
Добавление видео в сервис	1. В нижнем меню нажать на иконку для добавления видео	1. Открывается модальное окно с полем для вставки ссылки С кнопками Отменить и Готово
	2. Вставить ссылку и	2. Видео появляется на главной странице в списке

	нажать Готово	загруженных видео
<b>Как пользователь, я хочу иметь возможность управлять загруженными видео. Просматривать их, удалять или создавать с ними комнату для совместного просмотра</b>		
Функциональность связанная просмотром видео	1. На главной странице выбрать видео для просмотра. (Открыть для просмотра)	1. Открывается страница для просмотра видео. С проигрывателем видео и кнопкой Создать комнату для совместного просмотра. Также есть кнопка Назад
	2. Вернуться назад на главную страницу. Нажать на кнопку с опциями около выбранного видео	2. У кнопки появляется список с опциями: удалить видео и Создать комнату для просмотра
	3. Нажать Удалить	3. Видео исчезло со страницы
Создание комнаты для совместного просмотра видео	4. Выбрать новое видео, повторить шаг 2, Нажать Создать комнату	4. Пользователя перебрасывает в созданную комнату. Интерфейс выглядит также как в п1 ожидаемых результатов. Посередине экрана модальное окно с кодом для приглашения участников и возможностью скопировать. А также кнопкой для закрытия этого окна.
	5. Скопировать код, закрыть окно	5. Код скопирован в буфер обмена устройства В правом верхнем углу комнаты находится кнопка для повторного вызова модального окна с кодом-приглашением
<b>Как пользователь, я хочу иметь возможность просматривать распознанный текст из видео разделенный по главам</b>		
Просмотр суммаризации содержания видео по тайм кодам	1. Открыть любое видео для просмотра	1. Открывается страница для просмотра видео. С проигрывателем видео и кнопкой для просмотра распознанного текста из видео
	2. Открыть инструмент для просмотра распознанного текста из видео	2. В данном инструменте содержится суммаризация текста из видео разделенная по главам
<b>Как пользователь, я хочу иметь возможность получать предупреждение о наличии нецензурной лексики в видео</b>		
Получение предупреждения о наличии нецензурной лексики в видео	1. Открыть любое видео для просмотра и начать просмотр	1. Открывается страница для просмотра видео. С проигрывателем видео. При приближении видео к фрагменту с нецензурным выражением вылезает предупреждение
<b>Как пользователь, я хочу иметь возможность присоединиться к комнате друга для совместного просмотра выбранного видео</b>		
Присоединение к комнате	1. Перейти в нижнем меню в раздел комнаты	1. В данном разделе отображаются созданные ранее комнаты пользователя с возможностью перехода в

	для совместного просмотра	них, а также иконкой в правой их части с картинкой Удалить. Внизу экрана находится кнопка для присоединения к комнате друга
	2. Нажать на кнопку для присоединения к существующей комнате	2. Открылось модальное окно с полем для ввода кода и кнопками Отменить и готово
	3. Ввести код и нажать Готово	3. Произошел переход в комнату другого пользователя. Интерфейс такой же как у хозяина комнаты, но нет кнопка для приглашения (или оставить)

**Как пользователь, я хочу иметь возможность удалять или выходить из созданной комнаты, чтобы управлять своими комнатами для совместного просмотра**

Управление созданной комнатой	1. В созданной комнате с присоединенными участниками запустить проигрыватель видео	1. Видео запустилось у всех участников комнаты
	2. Нажать на паузу. Приостановить проигрыватель видео	2. Видео остановилось у всех участников

## Диаграмма взаимодействия

Для описания взаимодействия компонентов системы была составлена диаграмма (Рисунок 1).

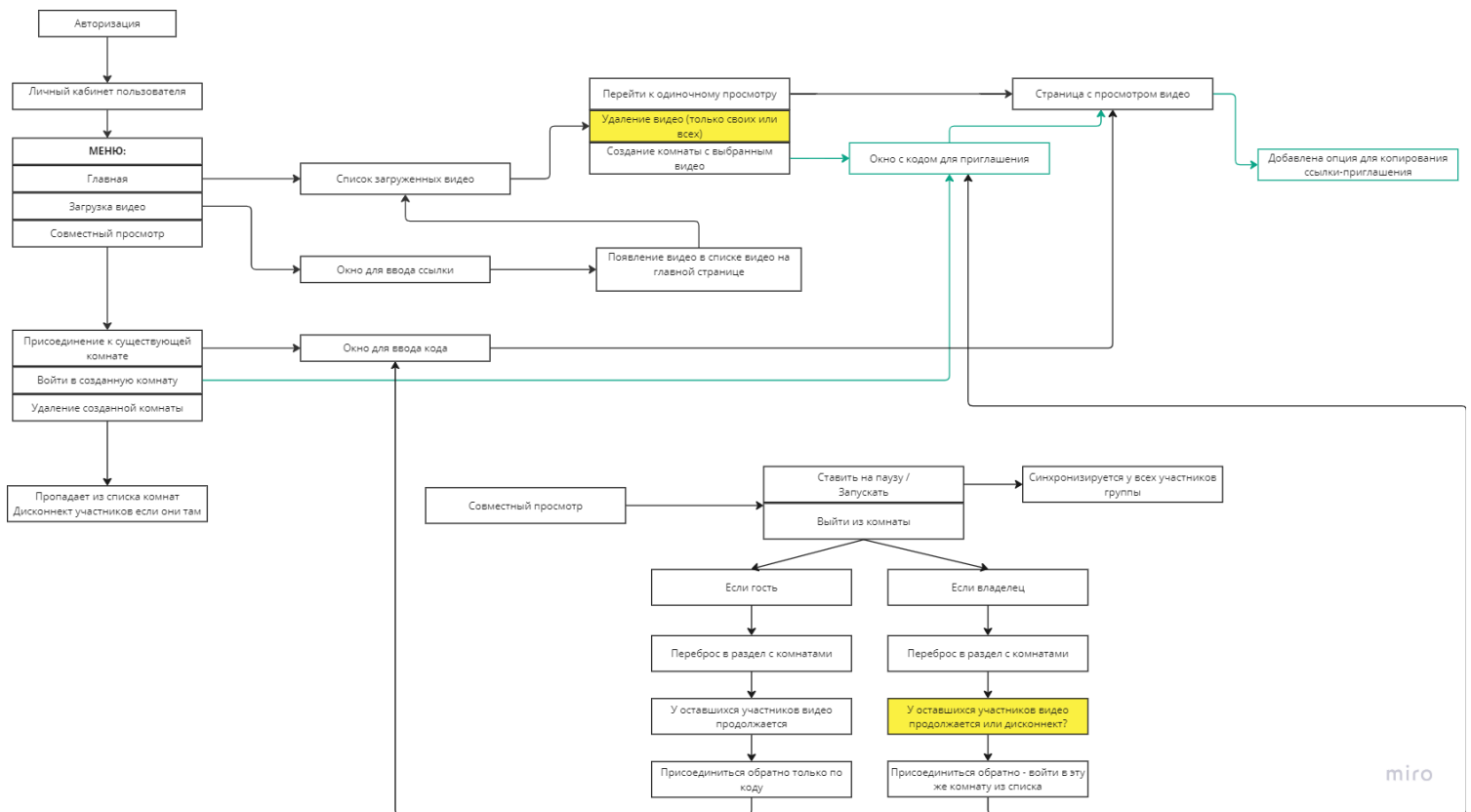


Рисунок 1 – Диаграмма взаимодействия

## Модуль распознавания речи из видео и обработка его на серверной части

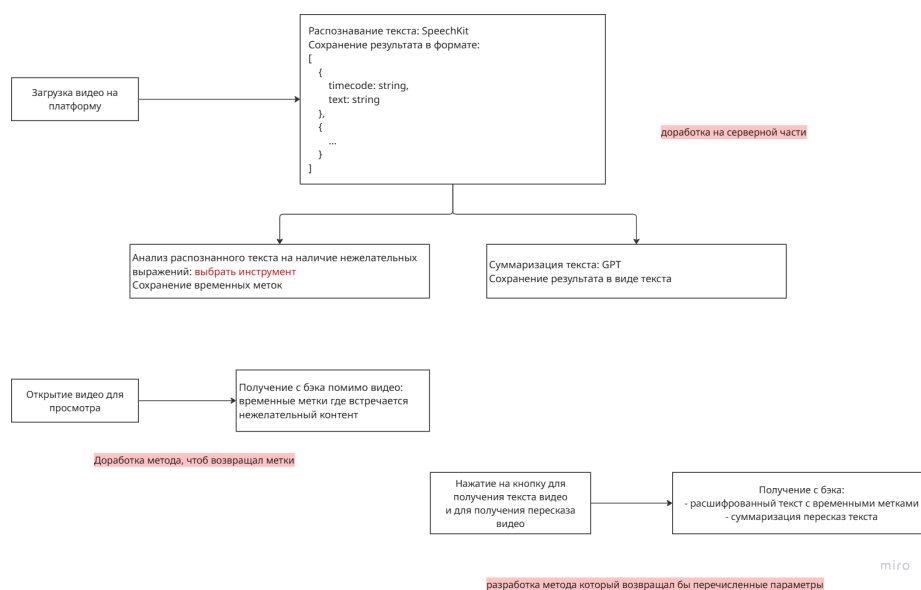


Рисунок 2 – AI модуль приложения

## 1. Загрузка видео

Пользователь загружает видеофайл через интерфейс приложения. После авторизации система сохраняет видео в облачное хранилище и запускает процесс обработки.

## 2. Обработка видео

После загрузки видео помещается в очередь на фоновую обработку. Основные задачи:

- Генерация превью.
- Распознавание речи и разметка глав.
- Проверка на ненормативную лексику.

Технические компоненты:

- Redis + BullMQ (для управления очередью).
- Worker (Node.js): асинхронно обрабатывает задачи из очереди.
- Используется FFmpeg для конвертации видео в WAV-аудио.
- Распознавание речи: Аудио отправляется в Yandex SpeechKit.

Результат: массив фраз с таймкодами (JSON).

- Для суммаризации текст из SpeechKit передается в ChatGPT-4o с промптом

Пример запроса в LLM

```
Make video summary from text in it. Divide video in chapters. Each chapter has title, description, start and finish timecode in ms. Output just JSON (nothing else). Output JSON should have the following format
```

```
{
  "chapters": [
    {
      "title": "chapter title",
      "description": "chapter description",
      "startTimeMs": int // chapter start time in ms
      "endTimeMs": int // chapter end time in ms
    }
  ]
}
```

Video text:

```
{{{text}}} // json, который получился после распознавания речи - фразы и их время
```

Пример ответа LLM



```
[
  {
    "title": "Introduction to Python",
    "description": "Introduction of Python as an interpreted high-level programming language known for its zen-like code.",
    "start_time": 120,
    "end_time": 5029
  },
  {
    "title": "Popularity and Applications",
    "description": "Discussion on Python's popularity and its use in serious projects. It's widely known for its statistics and friendly nature.",
    "start_time": 5380,
    "end_time": 15610
  },
  {
    "title": "History of Python",
    "description": "Python was created by Guido van Rossum and released in 1991, named after Monty Python's Flying Circus.",
    "start_time": 15780,
    "end_time": 25880
  },
  {
    "title": "Common Uses of Python",
    "description": "Python is used for server-side applications, data analysis, and machine learning, often leveraging Django.",
    "start_time": 25970,
    "end_time": 35980
  },
  {
    "title": "Python's Readability and Philosophy",
    "description": "Emphasis on Python's code readability and its adherence to the Zen of Python.",
    "start_time": 36079,
    "end_time": 50700
  },
  {
    "title": "Code Structure in Python",
    "description": "How Python organizes code using indentation instead of braces, and its support for both functional and OOP paradigms.",
    "start_time": 50760,
    "end_time": 75299
  },
  {
    "title": "Python's Ecosystem",
    "description": "Overview of Python's vast ecosystem, including third-party libraries and frameworks like TensorFlow.",
    "start_time": 75400,
    "end_time": 132140
  },
  {
    "title": "Conclusion",
    "description": "Final thoughts on Python's popularity and versatility, with a call to action for viewers."
  }
]
```

```
"start_time": 132319,  
"end_time": 145600  
}  
1
```

Таблица 1 – Таблица интеграций

Сервис	Назначение
Yandex SpeechKit	Распознавание речи из аудио.
ChatGPT-4o	Суммаризация, главы, NSFW-анализ.
FFmpeg	Извлечение аудиодорожки.
Redis/BullMQ	Очередь фоновых задач.

Документация до запуску

Запуск APK:

1. Разрешить установку из неизвестных источников
  - В настройках перейти в раздел "Безопасность" или "Приложения" (в зависимости от вашего устройства).
  - Найдите и активируйте опцию "Неизвестные источники" или "Установка приложений из неизвестных источников". Это позволит устанавливать приложения, не загружаемые из Google Play Маркета.
2. Скопировать APK файл на устройство
3. Установить APK
  - Найдите APK файл и нажмите на него, чтобы начать процесс установки.
  - Подтвердите установку, нажав "Установить".
5. Запустить приложение
  - После завершения установки вы можете открыть приложение сразу из окна установки, нажав "Открыть".
  - Альтернативно, найдите установленное приложение в меню приложений вашего устройства и запустите его оттуда

Запуск в браузере:

<https://echo.spb.ru/#/home>

Запуск веб-версии из репозитория:

(Описана в Readme репозитория)

## Установка

---

Перед началом работы убедитесь, что у вас установлен [Node.js](#).

```
cd .\videoService\  
npm i  
cd ..  
cd .\server\  
npm i
```



## Web

---

### Запуск

```
npm run dev
```



### Запуск проверки кода

```
npm run check
```



### Запуск eslint и prettier

```
npm run eslint
```



## Back

---

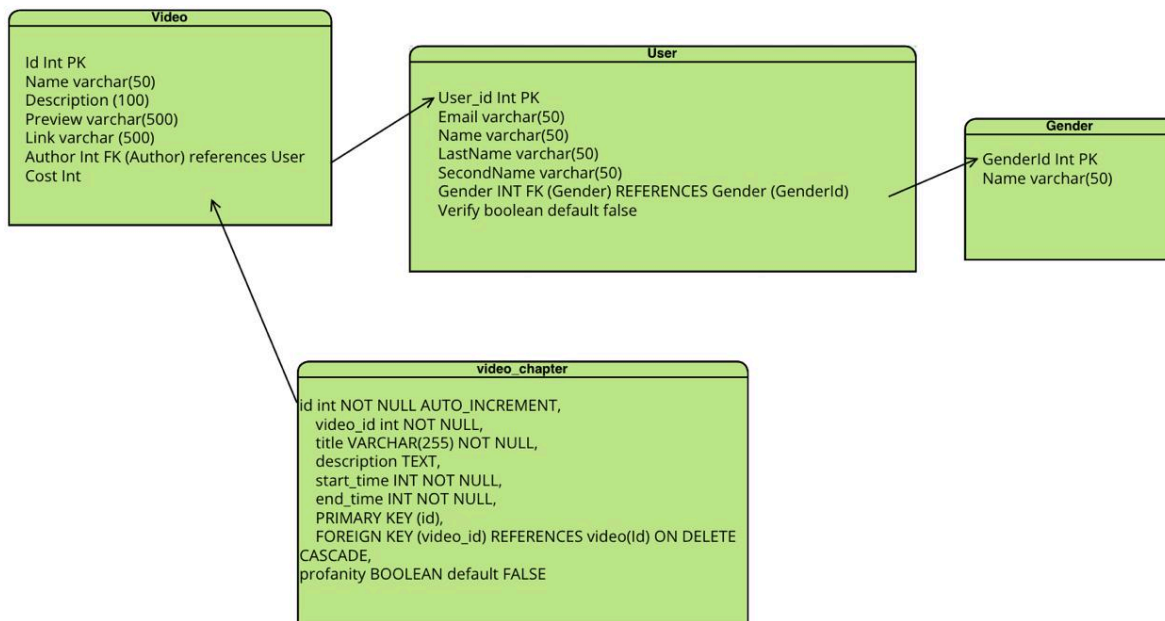
### Запуск

```
npm run server
```



Запускается nodemon, который автоматически пересобирает проект при изменениях в проекте

## ER-диаграмма



## **Ссылки на доп. материалы**

[Доска в jira](#)

[Доска в miro с расписанным User Story и диаграммой взаимодействия](#)

[Гитхаб репозиторий](#)

[Разработанный дизайн в figma](#)

[Swagger описание методов](#)