

РЕФЕРАТ

с. 126, рис. 20, табл. 41, прил. 12

программный комплекс, видеоконференция, медицинская организация, телемедицина, совещания, недостатки аналогов, открытый исходный код, модернизация.

Разработан программный комплекс для организации и проведения видеоконференций. Комплекс внедрен в медицинскую организацию для телемедицины, мастер-классов и совещаний.

Проведен анализ рынка аналогов и установлены недостатки существующих решений.

Определены технические требования к оборудованию и роли в системе. Определены меры информационной безопасности по защите системы.

Открыт исходный код программного продукта под открытой лицензией «BSD 3-Clause License». Разработана инструкция по установке и сопровождению системы.

Намечен план дальнейшего развития и модернизации системы.

ANNOTATION

p. 126, pic. 20, tab. 41, appx. 12

software system, video conference, medical organization, telemedicine, meetings, disadvantages of analogs, open source, modernization.

A software system has been developed for organizing video conferences. The system is deployed in a medical organization for telemedicine, lessons and meetings.

The analysis of analogs was carried out and the disadvantages of existing solutions were identified.

Technical requirements for hardware and roles in the system are defined. Information security measures to protect the system are defined.

The source code of the software product is open under the open license «BSD 3-Clause License». Instructions for installation and maintenance of the system have been developed.

A plan for further development and modernization of the system has been outlined.

СОДЕРЖАНИЕ

Введение.....	9
1 Технический проект.....	10
1.1 Анализ предметной области.....	10
1.1.1 Программный комплекс для организации и проведения видеоконференций.....	10
1.1.2 Диаграмма деятельности по организации и проведению видеоконференций.....	10
1.1.3 Технология «WebRTC».....	11
1.1.4 Работа с технологией «WebRTC».....	12
1.1.5 Основные технологии (архитектуры) для организации видеоконференций.....	14
1.1.6 Серверная часть и технология «WebSocket».....	17
1.1.7 Медиасервер на основе библиотеки Mediasoup.....	18
1.1.8 Режим администратора.....	18
1.1.9 Протокол «возобновляемой» загрузки файлов TUS.....	18
1.2 Анализ существующей автоматизированной системы на предприятии.....	19
1.2.1 Описание существующей системы.....	19
1.2.2 Требования организации к автоматизируемому процессу.....	19
1.3 Аналоги системы.....	20
1.3.1 Система ВКС – «TrueConf Server».....	20
1.3.2 Система ВКС – «Конференции Zoom».....	21
1.3.3 Система ВКС – «Jitsi Meet».....	21
1.3.4 Сравнительная таблица аналогов.....	22
1.3.5 Достоинства и недостатки аналогов.....	23
1.4 Обоснование выбора инструментов.....	24
1.5 Технология обработки информации.....	26
1.5.1 Диаграмма классов.....	26
1.5.2 Форматы данных.....	30
1.5.3 Алгоритм получения видео- и аудиопотоков пользователя.....	31
1.5.4 Алгоритм получения очередного маршрутизатора потоков на сервере.....	31
1.5.5 Алгоритм создания транспортного канала для исходящих медиапотоков.....	31
1.5.6 Алгоритм создания транспортного канала для входящих медиапотоков.....	32
1.5.7 Алгоритм передачи медиадорожки на сервер.....	32
1.5.8 Алгоритм получения удаленной медиадорожки с сервера.....	32
1.5.9 Алгоритм авторизации в комнате для клиента.....	33
1.5.10 Алгоритм аутентификации и авторизации в комнате для сервера.....	33
1.5.11 Алгоритм организации конференции.....	33

1.5.12 Алгоритм загрузки части файла для возобновляемой загрузки файлов.....	34
1.5.13 Алгоритм создания файла на сервере для возобновляемой загрузки файлов.....	34
1.6 Основные сценарии работы программного продукта.....	34
1.6.1 Диаграмма последовательности процесса захвата медиапотока.....	35
1.6.2 Диаграмма состояний медиапотока.....	35
1.6.3 Диаграмма активностей медиапотока.....	36
1.7 Входные и выходные данные.....	37
1.8 Требования к техническому и программному обеспечению.....	37
1.9 Безопасность информационной системы.....	38
2 Рабочий проект.....	39
2.1 Общие сведения о работе системы.....	39
2.2 Функциональное назначение системы.....	39
2.3 Описание физической архитектуры системы.....	41
2.4 Установка и выполнение программного продукта.....	41
2.5 Описание программы.....	42
2.6 Разработанные меню и интерфейсы.....	77
2.7 Сообщения системы.....	85
3 Программа и методика испытаний.....	86
3.1 Проверка работоспособности системы в роли участника.....	86
3.2 Проверка работоспособности системы в роли организатора.....	87
Заключение.....	88
Список используемых терминов, условных обозначений и сокращений.....	89
Список использованных источников.....	91
Приложение 1 Техническое задание.....	93
Приложение 2 Функциональная диаграмма.....	98
Приложение 3 Диаграмма вариантов использования.....	101
Приложение 4 Пользовательские истории.....	102
Приложение 5 Диаграммы классов.....	110
Приложение 6 Диаграмма последовательности захвата медиапотока.....	118
Приложение 7 Диаграммы состояний медиапотока.....	119
Приложение 8 Диаграммы активностей медиапотока.....	120
Приложение 9 Диаграмма развертывания.....	122
Приложение 10 Содержимое файла Package.json.....	123
Приложение 11 Акт о внедрении.....	125
Приложение 12 Программный продукт на оптическом носителе.....	126

ВВЕДЕНИЕ

В эпоху цифровизации, а также недавней короновирусной пандемии и последовавшей за ней самоизоляции, видеоконференцсвязь стала востребованным решением на современном рынке технологий. Государственные учреждения и организации стали вынуждены поддерживать рабочий процесс в удаленном режиме для сохранения самоизоляции сотрудников.

На рынке информационных технологий существует множество программных комплексов для организации и проведения видеоконференций, однако большая часть из них имеет иностранное происхождение, что не подходит для государственных российских организаций. Также недостатком является и то, что эти программы являются «облачными», и как следствие, данные передаются с некоторой задержкой, и организация, использующая программный комплекс, становится зависима от работоспособности удаленного облачного сервера.

Одной из таких организаций является **ГБУЗ АО АМОКБ**, которая является заинтересованным лицом в получении собственного программного комплекса, и которая планирует организовывать телемедицинские консультации, совещания и видеоконференции, а также мастер-классы в режиме реального времени. В данный момент организация использует коммерческое отечественное решение «**TrueConf Server**», с ограничением на количество пользователей онлайн в системе (не более 30 пользователей). По этим причинам организация начала искать альтернативные варианты.

Одним из главных требований заинтересованного лица касательно разрабатываемого программного комплекса для организации и проведения видеоконференций является поддержка технологии **WebRTC**, которая позволяет осуществлять видеосвязь посредством браузера без установки дополнительных клиентских программ.

Целями разработки программного продукта являются:

- ☛ обеспечение возможности осуществления видеосвязи через веб-браузер;
- ☛ увеличение вместимости конференции (по количеству пользователей);
- ☛ улучшение качества и стабильности видеосвязи;
- ☛ снижение финансовых затрат на приобретение текущей ВКС-системы.

Назначение программного продукта – программа предназначена для организации и проведения видеоконференций.

Инв. № подл.	Подп. и дата		Взам. инв. №	Инв. № дубл	Подп. и дата
<p>видеоконференции, а также мастер-классы в режиме реального времени. В данный момент организация использует коммерческое отечественное решение «TrueConf Server», с ограничением на количество пользователей онлайн в системе (не более 30 пользователей). По этим причинам организация начала искать альтернативные варианты.</p> <p>Одним из главных требований заинтересованного лица касательно разрабатываемого программного комплекса для организации и проведения видеоконференций является поддержка технологии WebRTC, которая позволяет осуществлять видеосвязь посредством браузера без установки дополнительных клиентских программ.</p> <p>Целями разработки программного продукта являются:</p> <ul style="list-style-type: none">☞ обеспечение возможности осуществления видеосвязи через веб-браузер;☞ увеличение вместимости конференции (по количеству пользователей);☞ улучшение качества и стабильности видеосвязи;☞ снижение финансовых затрат на приобретение текущей ВКС-системы. <p>Назначение программного продукта – программа предназначена для организации и проведения видеоконференций.</p>					
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022
					Лист 9

1 ТЕХНИЧЕСКИЙ ПРОЕКТ

1.1 Анализ предметной области

1.1.1 Программный комплекс для организации и проведения видеоконференций

ГБУЗ АО АМОКБ, далее Заинтересованное лицо, является региональным координационным центром телемедицины. Оно организует связь между сотнями структурными подразделениями медицинских организаций. Для организации данной связи заинтересованному лицу необходим программный комплекс для организации и проведения видеоконференций.

Организации требуется разработать собственный программный комплекс для организации и проведения видеоконференций с поддержкой технологии WebRTC[10][11]. Данный комплекс должен предоставить пользователям возможность одновременного захвата нескольких медиаустройств, возможность продемонстрировать экран окна приложения или рабочего стола, а также возможность обмена текстовыми сообщениями и файлами.

Программный комплекс должен быть автономным, то есть он должен при соответствующей необходимости, работать в условиях закрытой сети без доступа к Интернету.

1.1.2 Диаграмма деятельности по организации и проведению видеоконференций

В приложении на рисунке П2.1 представлена функциональная диаграмма IDEF0, описывающая деятельность по организации и проведению видеоконференций.

На данной диаграмме можно выделить 6 основных процессов, которые сопровождаются потоками входных и выходных данных:

1. Создание конференции.
2. Конфигурация конференции.
3. Авторизация участников.
4. Участие в конференции.
5. Администрирование конференции.
6. Завершение конференции.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						Лист	
					ВКРБ 09.03.04.059.2022					10	
Изм	Лист	№ документа	Подпись	Дата							

Процессы «Участие в конференции» и «Администрирование конференции» можно детализировать на подпроцессы.

Таким образом процесс «Участие в конференции» для роли «Участник», который отображен в приложении на рисунке П2.2, разбивается на подпроцессы:

1. Получение списка и видеоокон других участников.
2. Захват медиапотока.
3. Отправление сообщения в чат.
4. Получение сообщения в чате.
5. Отправление файла в чат.
6. Скачивание файла из чата.

Соответственно процесс «Администрирование конференции» для роли «Организатор», который отображен в приложении на рисунке П2.3, разбивается на подпроцессы:

1. Исключение участника из конференции.
2. Изменение имени участника.
3. Отключение медиапотока, исходящего от участника.
4. Блокировка и разблокировка участника по IP-адресу.
5. Изменение названия или пароля от комнаты.
6. Очистка истории чата.

1.1.3 Технология «WebRTC»

WebRTC (англ. **real-time communications** – **коммуникации в реальном времени**) – проект с открытым исходным кодом, предназначенный для организации передачи потоковых данных между браузерами или другими поддерживающими его приложениями по технологии точка-точка[12].

Технология **WebRTC** делает возможной видеосвязь через окно браузера, так что для присоединения к звонку необходимо только перейти по ссылке на соответствующую веб-страницу.

В основном в **WebRTC** используются два аудиокодека, **G.711** и **Opus**, а также видеокодеки **H.264**, **VP8** и **VP9**. С полным списком поддерживаемых медиакодеков можно ознакомиться на официальном сайте проекта WebRTC[10][11].

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						Лист	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					11	

4. Шифрование и дешифрование у пользователя по IP-адресу.

5. Изменение названия или пароля от комнаты.

6. Очистка истории чата.

1.1.3 Технология «WebRTC»

WebRTC (англ. real-time communications – коммуникации в реальном времени) – проект с открытым исходным кодом, предназначенный для организации передачи потоковых данных между браузерами или другими поддерживающими его приложениями по технологии точка-точка[12].

Технология **WebRTC** делает возможной видеосвязь через окно браузера, так что для присоединения к звонку необходимо только перейти по ссылке на соответствующую веб-страницу.

В основном в **WebRTC** используются два аудиокодека, **G.711** и **Opus**, а также видеокодеки **H.264**, **VP8** и **VP9**. С полным списком поддерживаемых медиакодеков можно ознакомиться на официальном сайте проекта WebRTC[10][11].

Основные возможности и особенности технологии:

- ✎ захват изображения с веб-камеры;
- ✎ захват звука с микрофона;
- ✎ захват изображения с экрана компьютера (весь экран или окно приложения);
- ✎ видео- и аудиосвязь;
- ✎ передача данных (текст и файлы);
- ✎ DTLS-SRTP шифрование[14].

Преимущества технологии:

- ✎ проведение конференции в браузере значительно упрощает процесс проведения конференции – пользователю не нужно устанавливать для этого отдельные приложения;
- ✎ используемые кодеки обеспечивают хорошее качество связи;
- ✎ возможность реализации любых элементов интерфейса средствами HTML5 и JavaScript.

Недостатки технологии:

- ✎ технология определяет только общий стандарт передачи данных (видео и звука), но реализации стандарта в разных браузерах не всегда совместимы между собой. Поэтому даже звонки между парой различных браузеров могут представлять отдельную сложность;
- ✎ обеспечение групповых конференций требует дополнительных разработок поверх WebRTC.

1.1.4 Работа с технологией «WebRTC»

WebRTC – это **JavaScript API** (интерфейс прикладного программирования), встроенный в веб-браузер. Он состоит из трех главных компонентов:

1. **Media Streams** – позволяет получить доступ к камере и микрофону через браузер.
2. **RTCPeerConnection** – позволяет установить *peer-to-peer* соединение между браузерами.
3. **RTCDataChannel** – позволяет осуществлять двунаправленную передачу произвольных данных (файлов).

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						Лист	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					12	

звуча), но реализации стандарта в разных браузерах не всегда совместимы между собой. Поэтому даже звонки между парой различных браузеров могут представлять отдельную сложность;	
обеспечение групповых конференций требует дополнительных разработок поверх WebRTC.	
1.1.4 Работа с технологией «WebRTC»	
WebRTC – это JavaScript API (интерфейс прикладного программирования), встроенный в веб-браузер. Он состоит из трех главных компонентов:	
1. Media Streams – позволяет получить доступ к камере и микрофону через браузер.	
2. RTCPeerConnection – позволяет установить peer-to-peer соединение между браузерами.	
3. RTCDataChannel – позволяет осуществлять двунаправленную передачу произвольных данных (файлов).	

Соединение в **WebRTC** осуществляется следующим образом: клиенты оперируют некими "приглашением" и "ответом", в которых содержится важная информация о будущем соединении[13].

Клиент 1 (приглашающий) создает приглашение и передает его Клиенту 2 (отвечающий). Клиент 2, принимая приглашение, генерирует ответ и отправляет его Клиенту 1.

После этого устанавливается *peer-to-peer* соединение, то есть одноранговая сеть без посредников и серверов. В стандарте технологии способ передачи приглашения и ответа никак не регламентируется и должен организовываться силами разработчика приложения, использующего **WebRTC**. Эта передача называется **сигнализацией**. Часто для автоматизации используется сигнальный сервер, который пересылает приглашение и ответ между клиентами. Но сигнализацию можно провести и вручную, например по электронной почте.

На рисунке 1.1 представлена схема такого соединения.

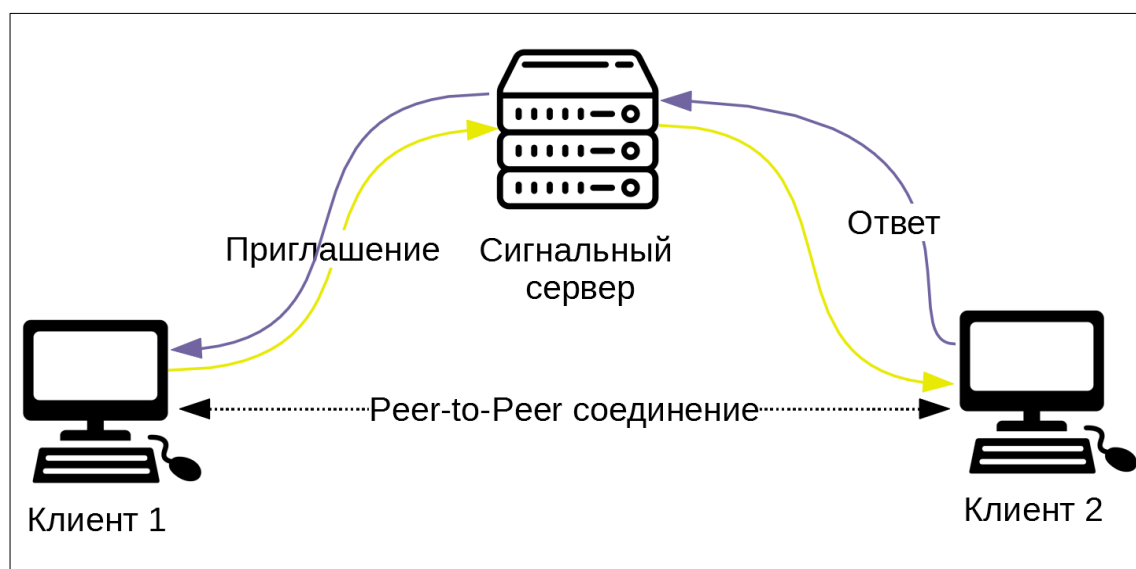


Рисунок 1.1 – Схема peer-to-peer соединения

SDP-объект (Session Description Protocol) – это «приглашение» и «ответ» клиентов. Он представляет из себя текстовые данные в формате ключ-значение. В нем содержится информация:

- ☞ доступные способы кодирования видео и аудио данных;
- ☞ поддерживаемый транспортный протокол;
- ☞ ICE-кандидаты и прочие возможности.

Инв. № подл.	Подп. и дата				Лист 13
	Инв. № дубл.				
Взам. инв. №	Подп. и дата				Лист 13
	Инв. № дубл.				
ВКРБ 09.03.04.059.2022					
Изм	Лист	№ документа	Подпись	Дата	

Рисунок 1.1 – Схема peer-to-peer соединения

SDP-объект (Session Description Protocol) – это «приглашение» и «ответ» клиентов. Он представляет из себя текстовые данные в формате ключ-значение. В нем содержится информация:

- ✎ доступные способы кодирования видео и аудио данных;
- ✎ поддерживаемый транспортный протокол;
- ✎ ICE-кандидаты и прочие возможности.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

Существует три основные архитектуры для организации видеоконференций: **Mesh** (ячеистая топология), **MCU** (Multipoint Control Unit) и **SFU** (Selective Forwarding Unit)[12][17].

Такая архитектура позволяет реализовать с помощью peer-to-peer (P2P) технологии, такой как WebRTC, конференцию, состоящую из более двух человек. Для реализации такой архитектуры не нужен медиасервер, управляющего медиапотокami, достаточно сигнального сервера, который позволит автоматизировать процесс передачи данных о клиентах между ними. Однако технически, при такой архитектуре никакой сервер не обязателен.

The diagram illustrates a secure communication channel using a physical analogy of a locked box. It shows three participants: Alice, Bob, and Carl. Alice starts with a message (represented by a blue square) and a lock (represented by a red padlock). She locks the message with the padlock and sends the locked message to Bob. Bob removes the padlock and retrieves the message. The diagram shows the message being locked and then unlocked, with the padlock being removed after the message is received.

Плюсом данной архитектуры является то, что она — самая простая в реализации, поскольку не требует никакой дополнительной работы с медиапотокami.

Главный минусом архитектуры является то, что при такой топологии, нагрузка на клиентов становится очень большой, когда количество участников

превышает 3-5 человек, поскольку каждый клиент должен осуществлять захват, кодирование и шифрование медиапотока для каждого участника конференции, что повышает требования к клиентской мощности компьютера и сетевому каналу.

В основе архитектуры **MCU**, схема которой отображена на рисунке 1.3, лежат два понятия: **микширование** или сведение, и **транскодирование**.

Микширование – стадия создания из отдельных видео- и аудиопотоков одного конечного потока.

Транскодирование – это преобразование файла из одного метода кодирования (то есть формата файла) в другой.

MCU – передача данных между сервером и клиентами (терминалами) происходит следующим образом (см. рис. 1.3):

1. Получение сервером входящих видеопотоков от всех терминалов.
2. Отправка каждому терминалу сервером копии одного сжимированного видеопотока, состоящего из видеопотоков всех терминалов.
3. Получение терминалами входящего сжимированного видео- и аудиопотоков.

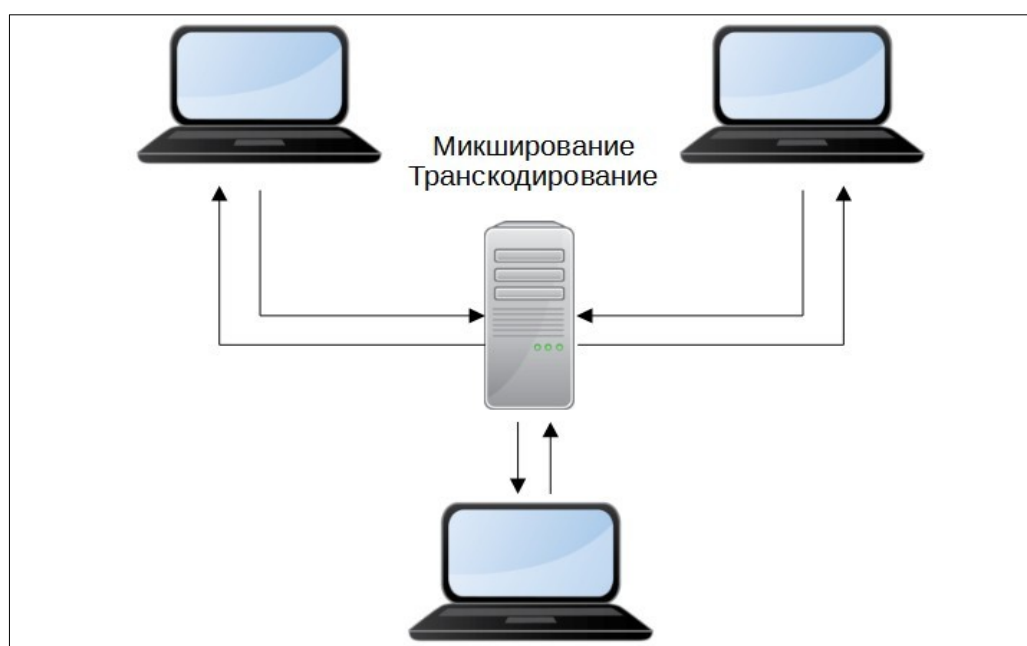


Рисунок 1.3 – Схема MCU

Таким образом, в схеме MCU собеседники не связаны между собой напрямую и передают весь трафик через сервер. Как следствие, сервер MCU должен обладать достаточной пропускной способностью сети и достаточной мощностью центрального процессора для микширования и транскодирования медиапотокa.

Также архитектура MCU не позволяет управлять раскладками видеоэлементов собеседников на терминале, управлением раскладками занимается организатор конференции, у которого есть непосредственный доступ к серверу.

В архитектуре SFU, схема которой отображена на рисунке 1.4 – передача данных между сервером и клиентами (терминалами) происходит следующим образом:

1. Получение сервером входящих видеопотоков от всех терминалов.
2. Отправка каждому терминалу сервером нескольких видеопотоков остальных участников без сжатия.
3. Получение терминалами входящих видеопотоков других терминалов.

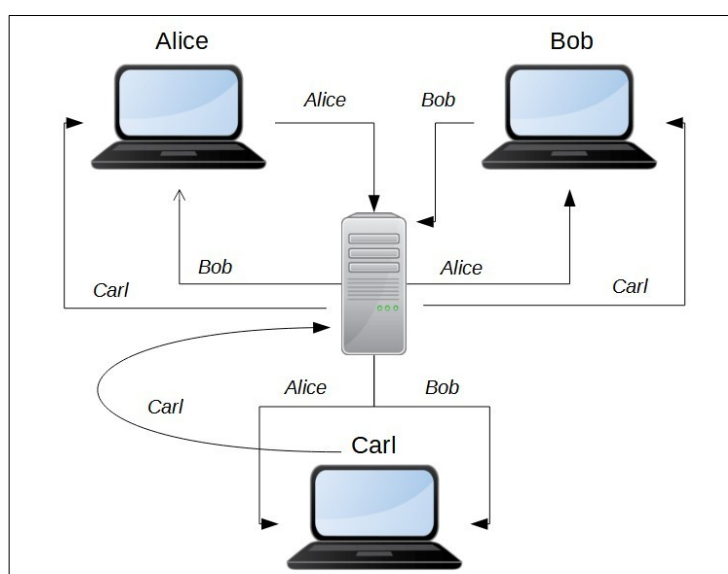


Рисунок 1.4 – Схема SFU

Архитектура SFU имеет несколько преимуществ в сравнении с другими архитектурами:

1. Управление раскладкой видеопотоков на стороне терминала.
2. Относительная нетребовательность к ресурсам сервера по сравнению с другими архитектурами видеоконференций.

Однако, архитектура SFU также имеет и недостатки:

1. Передача избыточного изображения в раскладку: каждый участник получает видеопоток больший, чем ему необходимо.
2. Клиент будет декодировать большое количество входящих медиапотоков, соответственно требуется более мощный процессор и широкий входящий канал (в сравнении с MCU).

Инв. № подл.	Подп. и дата				Лист 16	
	Инв. № дубл					
	Взам. инв №					
	Подп. и дата					
ВКРБ 09.03.04.059.2022						Лист 16
Изм	Лист	№ документа	Подпись	Дата		

The diagram illustrates the SFU (Selective Forwarding Unit) architecture. It shows three participants: Alice, Bob, and Carl. Alice and Bob are at the top, connected to a central box labeled 'Carl'. This 'Carl' box is then connected to a laptop at the bottom, which is also labeled 'Carl'. Arrows indicate the flow of data from Alice and Bob to the central 'Carl' and then to the laptop 'Carl'.

Рисунок 1.4 – Схема SFU

Архитектура SFU имеет несколько преимуществ в сравнении с другими архитектурами:

1. Управление раскладкой видеопотоков на стороне терминала.
2. Относительная нетребовательность к ресурсам сервера по сравнению с другими архитектурами видеоконференций.

Однако, архитектура SFU также имеет и недостатки:

1. Передача избыточного изображения в раскладку: каждый участник получает видеопоток больший, чем ему необходимо.
2. Клиент будет декодировать большое количество входящих медиапотоков, соответственно требуется более мощный процессор и широкий входящий канал (в сравнении с MCU).

1.1.6 Серверная часть и технология «WebSocket»

Процесс сигнализации SDP-объектов «WebRTC», а также иная передача данных между клиентом и сервером может быть реализована различными способами, но одной из самых эффективных технологий для этого является протокол **WebSocket**[16].

WebSocket – протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени. Он делает возможным более гибкое взаимодействие между браузером и веб-сайтом, способствуя передаче интерактивного содержимого и созданию приложений реального времени.

Помимо сигнализации SDP-объектов «WebRTC» и предоставления статического контента (HTML-страницы, файлы со стилями) веб-сервер должен реализовывать серверный механизм авторизации и аутентификации пользователей на основе токенов.

Под токеном подразумевается специальная текстовая строка (электронный ключ доступа), который выдается пользователю после успешной авторизации и используется для идентификации его владельца. То есть токен необходим для аутентификации пользователей в комнатах, защищенных паролем, а также для аутентификации пользователя в панели администратора. Данный механизм работает следующим образом:

1. Для первого запроса пользователя на сервер, генерируются уникальный идентификатор пользователя и токен. Токен представляет из себя текстовую строку, в которую с помощью специального алгоритма закодирован идентификатор пользователя.
2. Данные клиента (например, имя пользователя или идентификаторы комнат, в которых авторизован пользователь) сохраняются на сервере, и сохраненные данные связываются с ранее сгенерированным идентификатором пользователя.
3. Далее создается сессионная куки (небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя) в которой записан только токен. Данная куки добавляется в ответ на запрос и сохраняется в браузере клиента.

Инв. № подл.	Подп. и дата		Взам. инв. №		Инв. № дубл.		Подп. и дата			
<p>аутентификации пользователей в комнатах, защищенных паролем, а также для аутентификации пользователя в панели администратора. Данный механизм работает следующим образом:</p> <ol style="list-style-type: none">1. Для первого запроса пользователя на сервер, генерируются уникальный идентификатор пользователя и токен. Токен представляет из себя текстовую строку, в которую с помощью специального алгоритма закодирован идентификатор пользователя.2. Данные клиента (например, имя пользователя или идентификаторы комнат, в которых авторизован пользователь) сохраняются на сервере, и сохраненные данные связываются с ранее сгенерированным идентификатором пользователя.3. Далее создается сессионная куки (небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя) в которой записан только токен. Данная куки добавляется в ответ на запрос и сохраняется в браузере клиента.										
					ВКРБ 09.03.04.059.2022				Лист	
									17	
Изм	Лист	№ документа	Подпись	Дата						

4. При каждом следующем запросе клиент передает на сервер персональный токен, хранящийся в куки. Сервер сопоставляет идентификатор пользователя, закодированный в токене и распознает клиента.
5. До тех пор пока клиент передает свой персональный токен, сервер распознает клиента. Однако токен может стать невалидным, например, по истечении какого-то установленного времени (таймаут) на сервере.

1.1.7 Медиасервер на основе библиотеки Mediasoup

Система должна реализовать архитектуру видеоконференций SFU. Для этого сервер должен устанавливать прямое **peer-to-peer** соединение с клиентом (транспортный канал), принимать медиапотoki от клиента, делать копии этого потока и рассылать его остальным участникам конференции.

Mediasoup являясь C++ библиотекой, предоставляет низкоуровневое **API** (программный интерфейс), то есть методы и функции для создания транспортного канала **WebRTC**, и методы для работы с медиапотокami в среде **Node.js**[17].

Поскольку библиотека является низкоуровневой, всю сигнализацию и обмен всеми необходимыми данными (видеокодеки, информация о медиапотокax, информация об устройствах клиентов, и так далее) между клиентом и сервером должен осуществлять программист, использующий библиотеку.

1.1.8 Режим администратора

Система должна иметь механизм поддержки комнат конференции, защищенных паролем, которые позволяют изолировать участников конференции одной комнаты от участников другой комнаты.

Для этого в системе должна быть предусмотрена возможность создавать комнаты с заданным именем и паролем, а также возможность удаления ранее созданных комнат.

Также в системе должны быть реализованы следующие базовые функции администратора: исключение участника из комнаты, блокировка участника, отключение передачи видео-/аудиопотоков исходящих от участника, изменение имени участника.

1.1.9 Протокол «возобновляемой» загрузки файлов TUS

TUS – это открытый протокол для «возобновляемой» загрузки, построенный поверх протокола HTTP. Протокол описывает, каким образом должны загружаться файлы на сервер и как возобновить прерванную загрузку файла[18].

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	информация об устройствах клиентов, и так далее) между клиентом и сервером	
					должен осуществлять программист, использующий библиотеку.	
					1.1.8 Режим администратора	
					Система должна иметь механизм поддержки комнат конференции, защищенных паролем, которые позволяют изолировать участников конференции одной комнаты от участников другой комнаты.	
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	Для этого в системе должна быть предусмотрена возможность создавать комнаты с заданным именем и паролем, а также возможность удаления ранее созданных комнат.	
					Также в системе должны быть реализованы следующие базовые функции администратора: исключение участника из комнаты, блокировка участника, отключение передачи видео-/аудиопотоков исходящих от участника, изменение имени участника.	
					1.1.9 Протокол «возобновляемой» загрузки файлов TUS	
					TUS – это открытый протокол для «возобновляемой» загрузки, построенный поверх протокола HTTP. Протокол описывает, каким образом должны загружаться файлы на сервер и как возобновить прерванную загрузку файла[18].	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист
						18

1.2 Анализ существующей автоматизированной системы на предприятии

1.2.1 Описание существующей системы

Объектом автоматизации является кабинет телемедицины Заинтересованного лица, который является региональным координационным центром телемедицины в Астраханской области.

В данный момент организацией используется коммерческий отечественный продукт «TrueConf Server» от организации ООО «Труконф». Система ВКС используется организацией следующими способами:

- ☞ осуществление дистанционных врачебных и консультационных услуг (телемедицина);
- ☞ мастер-классы и видеотрансляции операций;
- ☞ дистанционные совещания и видеоконференции.

Программный продукт в своей полной комплектации позволяет выполнять все необходимые функции, однако, в соответствии с приобретенной лицензией на продукт (выбор которой определяется исходя из финансовых возможностей организации), система поставляется с ограничением на количество активных пользователей (не более 30 пользователей онлайн в системе) и без поддержки технологии WebRTC.

Не желая тратить финансовые средства на продление лицензии на использование «TrueConf Server», а также стремясь снять ограничение на количество активных пользователей с целью организации соответствующей коммуникационной инфраструктуры между сотнями структурными подразделениями медицинских организаций, заинтересованное лицо начало поиск альтернативных вариантов.

1.2.2 Требования организации к автоматизируемому процессу

Организации требуется разработать собственный программный комплекс для организации и проведения видеоконференций, которая должна отвечать следующим требованиям:

- ☞ веб-клиент и поддержка технологии WebRTC;
- ☞ автономность, то есть возможность развертывания системы на собственном сервере и соответственно работа системы в закрытой сети без подключения к интернету;
- ☞ возможность одновременного захвата нескольких видеоустройств;

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						Лист	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					19	

пользователей (не более 30 пользователей онлайн в системе) и без поддержки технологии **WebRTC**.

Не желая тратить финансовые средства на продление лицензии на использование «**TrueConf Server**», а также стремясь снять ограничение на количество активных пользователей с целью организации соответствующей коммуникационной инфраструктуры между сотнями структурными подразделениями медицинских организаций, заинтересованное лицо начало поиск альтернативных вариантов.

1.2.2 Требования организации к автоматизируемому процессу

Организации требуется разработать собственный программный комплекс для организации и проведения видеоконференций, которая должна отвечать следующим требованиям:

- ☞ веб-клиент и поддержка технологии WebRTC;
- ☞ автономность, то есть возможность развертывания системы на собственном сервере и соответственно работа системы в закрытой сети без подключения к интернету;
- ☞ возможность одновременного захвата нескольких видеоустройств;

- ☞ текстовый чат и надежная передача файлов между участниками конференции, которая возобновляется в случае обрыва соединения;
- ☞ кроссплатформенность, то есть поддержка операционных систем Windows и Linux;
- ☞ открытый исходный код;
- ☞ отсутствие искусственных (то есть не связанных с возможностями аппаратной и программной платформ) ограничений на количество пользователей в одной конференции и в системе в целом;
- ☞ возможность управления качеством видеосвязи, то есть управление битрейтом передаваемого видеоизображения и выбор видеокодеков.

На данный момент ни одна из систем-аналогов не соответствует в полной мере данным требованиям.

1.3 Аналоги системы

1.3.1 Система ВКС – «TrueConf Server»

Разработчиком данного решения является отечественная коммерческая организация ООО «Труконф». Данный сервер видеоконференций включает в себе большое количество возможностей, многие из которых являются дополнительными платными опциями к основному тарифному плану.

Бесплатный тарифный план позволяет организовывать и управлять видеоконференциями с ограничением 10 онлайн пользователей в системе. Есть поддержка текстового чата и передачи файлов, однако отсутствует поддержка веб-клиентов (WebRTC). Участниками могут выступать клиентские приложения для операционных систем Windows, macOS, Linux, Android и iOS. Помимо прочего поддерживается одно соединение по протоколу SIP или H.323.

Стоимость платного тарифного плана определяется количеством онлайн пользователей в системе и дополнительными опциями, и соответственно описать все возможные комбинации не представляется возможным. Но например, стандартный тип лицензии на 30 онлайн пользователей, которая, в сравнении с бесплатным тарифом, позволяет использовать систему автономно (без подключения к интернету) и дополняется базовой технической поддержкой, имеет стоимость в размере 255 000 рублей в год. Дополнительная опция для поддержки публичных веб-конференций (WebRTC) на 50 гостей стоит 100 000 рублей. Подробнее можно ознакомиться на официальном сайте программного продукта[19].

Инв. № подл.	Подп. и дата				Инв. № дубл	Подп. и дата				Взам. инв. №	Подп. и дата																
<p>большое количество возможностей, многие из которых являются дополнительными платными опциями к основному тарифному плану.</p> <p>Бесплатный тарифный план позволяет организовывать и управлять видеоконференциями с ограничением 10 онлайн пользователей в системе. Есть поддержка текстового чата и передачи файлов, однако отсутствует поддержка веб-клиентов (WebRTC). Участниками могут выступать клиентские приложения для операционных систем Windows, macOS, Linux, Android и iOS. Помимо прочего поддерживается одно соединение по протоколу SIP или H.323.</p> <p>Стоимость платного тарифного плана определяется количеством онлайн пользователей в системе и дополнительными опциями, и соответственно описать все возможные комбинации не представляется возможным. Но например, стандартный тип лицензии на 30 онлайн пользователей, которая, в сравнении с бесплатным тарифом, позволяет использовать систему автономно (без подключения к интернету) и дополняется базовой технической поддержкой, имеет стоимость в размере 255 000 рублей в год. Дополнительная опция для поддержки публичных веб-конференций (WebRTC) на 50 гостей стоит 100 000 рублей. Подробнее можно ознакомиться на официальном сайте программного продукта[19].</p>																											
													ВКРБ 09.03.04.059.2022													Лист	
																										20	
Изм Лист № документа Подпись Дата																											

1.3.2 Система ВКС – «Конференции Zoom»

Разработчиком данного решения является американская коммерческая организация «**Zoom Video Communications, Inc. (Zoom)**». Данная система ВКС является облачным ВКС-решением, то есть у нее отсутствует возможность развертывания на собственном сервере и она не может работать автономно без подключения к американским серверам, где хранятся данные о конференциях и их участниках. Поскольку продукт является иностранным решением, он не подходит для использования в государственных учреждениях.

Бесплатный тариф включает возможность организации конференции до 100 участников с одним организатором, но с ограничением по продолжительности конференции до 40 минут. Есть поддержка чата и передачи файлов. Также поддерживаются функции администратора для управления ходом конференции. Есть поддержка веб-клиентов, а также клиентских приложений для операционных систем Windows, Linux, Android и iOS.

Как и предыдущий рассмотренный аналог, данный продукт имеет сложную лицензионную ценовую политику, и стоимость платных тарифов складывается из количества организаторов, количества участников конференции, различных дополнительных опций, таких как поддержка протокола SIP, а также условий эксплуатации и так далее. Подробнее ознакомиться с тарифными планами и возможностями системы можно на официальном сайте продукта[20].

1.3.3 Система ВКС – «Jitsi Meet»

Данный продукт является свободной и бесплатной демонстрационной версией решения на основе WebRTC и движка с открытым исходным кодом, разработанной на языке Java американской коммерческой компанией 8X8.

Данная система имеет возможность развертывания на собственном сервере, совместима с протоколом SIP, поддерживает до 100 участников (включая гостей). Есть поддержка демонстрации экрана компьютера и текстовый чат, однако возможность передачи файлов отсутствует. Есть поддержка веб-клиентов, а также клиентских приложений для операционных систем Android и iOS. Более подробно ознакомиться с функционалом системы можно на сайте продукта[21].

Однако, являясь демонстрационной версией платного решения под названием «**8x8 Meet**», «**Jitsi Meet**» поставляется с минимальным набором функций для администрирования и управления ходом конференции, крайне нестабильной работой

Инв. № подл.	Подп. и дата		Взам. инв. №	Инв. № дубл.	Подп. и дата	дополнительных опций, таких как поддержка протокола SIP, а также условий эксплуатации и так далее. Подробнее ознакомиться с тарифными планами и возможностями системы можно на официальном сайте продукта[20].	
1.3.3 Система ВКС – «Jitsi Meet»							
Данный продукт является свободной и бесплатной демонстрационной версией решения на основе WebRTC и движка с открытым исходным кодом, разработанной на языке Java американской коммерческой компанией 8X8.							
Данная система имеет возможность развертывания на собственном сервере, совместима с протоколом SIP, поддерживает до 100 участников (включая гостей). Есть поддержка демонстрации экрана компьютера и текстовый чат, однако возможность передачи файлов отсутствует. Есть поддержка веб-клиентов, а также клиентских приложений для операционных систем Android и iOS. Более подробно ознакомиться с функционалом системы можно на сайте продукта[21].							
Однако, являясь демонстрационной версией платного решения под названием «8x8 Meet», «Jitsi Meet» поставляется с минимальным набором функций для администрирования и управления ходом конференции, крайне нестабильной работой							
Инв. № подл.	Подп. и дата		Взам. инв. №	Инв. № дубл.	Подп. и дата	<div>ВКРБ 09.03.04.059.2022</div>	
Изм	Лист	№ документа	Подпись	Дата		<div>Лист 21</div>	

медиа сервера, который отвечает за качество передаваемого видеоизображения и аудиопотоков участников конференции. Однако у данного программного продукта есть платная техническая поддержка, которая потенциально может решить все возникающие перечисленные проблемы.

1.3.4 Сравнительная таблица аналогов

В таблице 1.1 приведено сравнение аналогов по различным критериям.

Таблица 1.1 – Сравнительная таблица аналогов

№	Критерий	TrueConf	Zoom	Jitsi Meet	Разрабатываемая система
1	Поддержка веб-клиентов и технологии WebRTC	платная опция	+	+	+
2	Автономность (работа системы в закрытой сети без подключения к интернету)	+	-	+	+
3	Отечественный продукт	+	-	-	+
4	Одновременный захват нескольких видеоустройств	-	-	-	+
5	Текстовый чат	+	+	+	+
6	Передача файлов	+	+	-	+
7	Функции администрирования	+	+	-	+
8	Поддержка протокола SIP	платная опция	платная опция	+	-
9	Возможность управления качеством видеосвязи (регулировка битрейта, выбор видеокодека)	+	-	-	+
10	Отсутствие искусственных ограничений на количество пользователей в системе	-	-	?	+
11	Открытый исходный код	-	-	+	+

Инов. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Продолжение таблицы 1.1

№	Критерий	TrueConf	Zoom	Jitsi Meet	Разрабатываемая система
12	Техническая поддержка	платная	платная	платная	планируется платная
13	Серверная часть поддерживает ОС Windows	+	?	-	+
14	Серверная часть поддерживает Linux	+	?	+	+
15	Клиентское приложение для Windows	+	+	-	-
16	Клиентское приложение для Linux	+	+	-	-
17	Клиентское приложение для Android	+	+	+	-
18	Клиентское приложение для iOS	+	+	+	-

1.3.5 Достоинства и недостатки аналогов

Одно из самых главных недостатка решения «**Конференции Zoom**» – это его иностранное происхождение, что автоматически не позволяет его использовать в государственных учреждениях в связи с нормативно-правовыми актами РФ, регламентирующие импортозамещение программного обеспечения. Также недостатком является и то, что «**Zoom**» является «облачной», и как следствие, данные передаются с некоторой задержкой, и организация, использующая данный продукт, становится зависима от работоспособности удаленного облачного сервера. В соответствии с данными двумя недостатками, Заинтересованное лицо не может использовать и множество других иностранных программных продуктов, которые не обозревались в данном документе.

Решение от компании ООО «Труконф» обладает рядом несомненных преимуществ и в своем полном тарифном плане полностью подходит и реализует все необходимые требования организации. Однако, в соответствии с этой же тарифной политикой и финансовыми возможностями Заинтересованного лица, продукт поставляется с рядом существенных ограничений, такие как лимит на

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
										23
Изм	Лист	№ документа	Подпись	Дата						

количество онлайн участников в системе и отсутствие поддержки технологии WebRTC и веб-клиентов соответственно. Следовательно, к недостаткам можно причислить, помимо платной лицензии и высокой стоимости, закрытый исходный код, что лишает возможности гибко адаптировать программный продукт под нужды бизнес-процесса.

Программный продукт «**Jitsi Meet**» является довольно ограниченной в функционале системой ВКС, которая по большей части является бесплатной демонстрационной версией платного продукта «8x8 Meet». В соответствии с этим, в продукте отсутствуют необходимые возможности, такие как функции администрирования, а также существует ряд технических проблем с качеством и стабильностью организуемой видеоконференцсвязи. Однако, данная система является продуктом с открытым исходным кодом, и соответственно, эти проблемы потенциально решаемы.

Но поскольку главная часть программы, а именно медиасервер, написана на языке Java, было принято решение разрабатывать самостоятельную систему, а не дорабатывать необходимым функционалом «**Jitsi Meet**», так как языку, на котором был разработан медиасервер, сопутствуют риски в плане производительности и масштабируемости системы в зависимости от количества пользователей в конференции, а также существовал большой риск так и не решить проблемы со стабильностью и качеством соединения, которые имеют место быть в данном программном решении.

1.4 Обоснование выбора инструментов

Серверная часть разрабатываемой системы должна базироваться на платформе **Node.js** с использованием языка **TypeScript**. Платформа **Node.js** была выбрана на основе нескольких критериев[4][5][6]:

- ☞ кроссплатформенность: поддержка ОС Windows и Linux;
- ☞ открытый исходный код;
- ☞ производительность (**Node.js** разработан на C++ и позволяет писать C++ модули к программе);
- ☞ масштабируемость;
- ☞ активное развитие платформы;
- ☞ использование платформы крупными организациями.

Инв. № подл.	Подп. и дата				Лист	
	Взам. инв. №					
	Инв. № дубл.					
	Подп. и дата					
ВКРБ 09.03.04.059.2022						24
Изм	Лист	№ документа	Подпись	Дата		

масштабируемости системы в зависимости от количества пользователей в конференции, а также существовал большой риск так и не решить проблемы со стабильностью и качеством соединения, которые имеют место быть в данном программном решении.
<h3>1.4 Обоснование выбора инструментов</h3>
<p>Серверная часть разрабатываемой системы должна базироваться на платформе Node.js с использованием языка TypeScript. Платформа Node.js была выбрана на основе нескольких критериев[4][5][6]:</p> <ul style="list-style-type: none">✎ кроссплатформенность: поддержка ОС Windows и Linux;✎ открытый исходный код;✎ производительность (Node.js разработан на C++ и позволяет писать C++ модули к программе);✎ масштабируемость;✎ активное развитие платформы;✎ использование платформы крупными организациями.

Также важнейшим фактором, который повлиял на выбор платформы, является язык разработки – **JavaScript / TypeScript**, который является де-факто стандартом веб-разработки и позволяет ускорить разработку всей системы, так как язык программирования является общим как для серверной части, так и для клиентской части (веб-приложение).

Язык **TypeScript** был выбран поскольку является обратно совместимым расширением языка **JavaScript**, который компилируется в последний, и который добавляет в **JavaScript** статическую систему типизации, поддержку использования полноценных классов, как в традиционных ООП языках, что повышает скорость разработки, облегчает читаемость и рефакторинг кода, а также позволяет быстрее осуществлять поиск ошибок на этапе разработки[15].

Для реализации схемы **SFU**, в качестве медиасервера должна использоваться библиотека **Mediasoup**, поскольку она является низкоуровневой и производительной библиотекой с открытым исходным кодом, написанной на языке программирования **C++** и имеющей API для **Node.js**.

В соответствии с требованиями Заинтересованного лица, в качестве базы данных в системе должен быть реализован встроенный механизм сохранения данных в файлы формата **.json**. Такой подход сохранения данных имеет свои преимущества и недостатки.

Преимущества встроенного механизма сохранения данных:

- ☞ простота и гибкость;
- ☞ высокая производительность на малых объемах данных;
- ☞ просмотр и редактирование файлов с данными не требует СУБД, а только текстовый редактор;
- ☞ невосприимчивость к SQL-уязвимостям, таким как SQL-инъекции;
- ☞ отсутствие внешней зависимости в виде реляционной СУБД;
- ☞ более простой процесс резервного копирования данных, чем у реляционных баз данных.

Недостатки такого подхода:

- ☞ низкая производительность на больших объемах данных;
- ☞ невозможность написания сложных запросов и отчетов на языке SQL;
- ☞ отсутствуют транзакции, которые присущи сложным СУБД.

Однако должна быть предусмотрена возможность расширения системы до использования реляционной базы данных (SQL).

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022
					Лист 25

1.5 Технология обработки информации

Результатом анализа предметной области стала диаграмма вариантов использования, которая показана в *Приложении 3*.

Исходя из диаграммы можно сделать вывод, что система содержит две роли: участник и организатор.

Вариант использования «Выбрать комнату и авторизоваться в ней» включает две обязательные функции «Получить список и видеоокна других участников» и «Получить сообщение в чате». Также он расширен функциями: «Захватить медиапоток», «Отправить сообщение в чате», «Отправить файл в чат», «Скачать файл из чата».

Вариант использования «Создать конференцию» включает обязательную функцию «Сконфигурировать конференцию».

В *Приложении 4* приведено описание вариантов использования системы в формате пользовательских историй.

1.5.1 Диаграмма классов

В ходе анализа предметной области были разработаны диаграммы классов клиентской и серверной части программы. Они представлены в *Приложении 5* в нотации UML[7].

В приложении на рисунке П5.1 изображена диаграмма классов основной части клиента.

Класс **Room** является центральной сущностью клиентской части системы и олицетворяет собой комнату (конференцию) и включает в себя все соответствующие поля и методы по работе с комнатой.

Класс **Room** задействует класс **UI**, который предоставляет методы для работы с пользовательским интерфейсом. Также класс **Room** включает в себя (агрегация) классы **Mediasoup** и **FileService**.

Класс **Mediasoup** предназначен для работы с медиапотокami: производителями (**Producer**) и потребителями (**Consumer**), а также для подключения к медиасерверу системы. А класс **FileService** инкапсулирует все, что связано со скачиванием файлов с сервера и загрузкой файлов на сервер.

Помимо этого, в клиентской части приложения существует класс **UserMedia**, который занимается непосредственным захватом медиапотокa пользователя

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						
										Лист
					ВКРБ 09.03.04.059.2022					26
Изм	Лист	№ документа	Подпись	Дата						

(веб-камера, микрофон или экран компьютера) и передаёт их классу **Room**, который в свою очередь передаёт их классу **Mediasoup**.

В приложении на рисунках П5.2 и П5.3 отображена диаграмма классов файлового сервиса клиентской части, где ключевой сущностью является класс **FileService**.

Классы **TusHeadRequest**, **TusPatchRequest**, **TusOptionsRequest**, **TusPostCreationRequest** реализуют интерфейс **FileServiceRequest** и представляют собой HTTP-запросы от клиента в соответствии с протоколом возобновляемой передачи файлов TUS.

На рисунке П5.4 изображена диаграмма классов веб-сервиса серверной части.

Класс **WebService** реализует веб-сервер приложения, то есть отвечает на HTTP и HTTPS запросы и формирует соответствующие ответы. Он взаимодействует с репозиториями **RoomRepository** (комнаты), **AuthRoomUserRepository** (записи об авторизованных пользователях в комнатах), **UserBanRepository** (заблокированные пользователи) и **UserAccountRepository** (аккаунты пользователей) в тех случаях, когда это необходимо, например, когда необходимо проверить, имеет ли право участник (идентификатор аккаунта которой был передан в HTTP-запросе в токене) участвовать в выбранной конференции.

Класс **TokenService** инкапсулирует в себе работу по созданию и верификации JWT токенов для пользователей, а класс **FileService** реализует возможность загрузки и скачивания файлов на сервере.

На рисунке П5.5 изображена диаграмма классов сервиса комнат для серверной части.

Класс **Room** представляет собой комнату (конференцию) на сервере и включает в себя и конфигурацию комнаты, и список активных пользователей (объекты класса **ActiveUser**), а также обработку всех событий связанных с конференцией.

Класс **Room** включает в себя (композиция) список объектов класса **ActiveUser** и (агрегация) экземпляр класса **MediasoupService**.

Класс **ActiveUser** предназначен для представления активного участника конференции и хранения ссылок на соответствующие ему медиапотребителей (**Consumer**) и медиапотребителей-производителей (**Producer**).

Класс **MediasoupService** инкапсулирует работу с медиапотоками пользователей, а также управление самим медиасервером приложения.

Инв. № подл.	Подп. и дата				Лист
	Инв. № дубл				
	Взам. инв. №				
	Подп. и дата				
Изм.	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022
					27

участвовать в выбранной конференции.

Класс **TokenService** инкапсулирует в себе работу по созданию и верификации JWT токенов для пользователей, а класс **FileService** реализует возможность загрузки и скачивания файлов на сервере.

На рисунке П5.5 изображена диаграмма классов сервиса комнат для серверной части.

Класс **Room** представляет собой комнату (конференцию) на сервере и включает в себя и конфигурацию комнаты, и список активных пользователей (объекты класса **ActiveUser**), а также обработку всех событий связанных с конференцией.

Класс **Room** включает в себя (композиция) список объектов класса **ActiveUser** и (агрегация) экземпляр класса **MediasoupService**.

Класс **ActiveUser** предназначен для представления активного участника конференции и хранения ссылок на соответствующие ему медиапотoki-потребителей (**Consumer**) и медиапотoki-производителей (**Producer**).

Класс **MediasoupService** инкапсулирует работу с медиапотокami пользователей, а также управление самим медиасервером приложения.

На рисунках П5.6 и П5.7 изображены диаграммы классов файлового сервиса для серверной части.

Классы *TusHeadResponse*, *TusPatchResponse*, *TusOptionsResponse*, *TusPostCreationResponse*, *GetResponse* реализуют интерфейс *FileServiceResponse* и представляют собой HTTP-ответы от сервера в соответствии с протоколом возобновляемой передачи файлов TUS.

Некоторые из этих классов, как видно на диаграмме, создают или получают класс *FileInfo*, в котором содержится вся необходимая метаданная о файле.

Работа с объектами класса *FileInfo* (метаданные файлов на сервере) осуществляется с помощью класса-репозитория *FileRepository*.

Класс *FileService* использует классы *TusHeadResponse*, *TusPatchResponse*, *TusOptionsResponse*, *TusPostCreationResponse*, *GetResponse* в качестве шаблонов для HTTP-ответа, в которых подготавливаются все необходимые HTTP-заголовки и коды ответов.

В приложении на рисунках П5.8, П5.9 и П5.10 изображены диаграммы классов-репозитория серверной части, которые позволяют работать с серверными данными: создавать новые объекты и сохранять их в файлы, загружать из файлов сохраненные данные и так далее.

Класс-репозиторий *AuthRoomUserRepository* позволяет работать со списком авторизованных в комнатах пользователей, и включает в себя (композиция) множество объектов класса *AuthorizationRecords*.

Класс-репозиторий *RoomChatRepository* позволяет работать с историей чатов в конференциях, а класс *UserBanRepository* предназначен для работы со списком заблокированных пользователей (по IP-адресу), который включает в себя (композиция) множество объектов класса *UserBanInfo*.

Работа с аккаунтами пользователей осуществляется посредством класса *UserAccountRepository*, который включает в себя (композиция) множество объектов класса *UserAccount*.

В приложении на рисунках П5.11, П5.12 и П5.13 отображены диаграммы классов сервисов-сокетов, которые обрабатывают запросы и ответы по протоколу WebSocket.

Класс *SocketManager* реализует WebSocket-сервер, то есть осуществляет двустороннюю связь с низкой задержкой с клиентом посредством протокола

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	сохраненные данные и так далее.	
					Класс-репозиторий <i>AuthRoomUserRepository</i> позволяет работать со списком авторизованных в комнатах пользователей, и включает в себя (композиция) множество объектов класса <i>AuthorizationRecords</i> .	
					Класс-репозиторий <i>RoomChatRepository</i> позволяет работать с историей чатов в конференциях, а класс <i>UserBanRepository</i> предназначен для работы со списком заблокированных пользователей (по IP-адресу), который включает в себя (композиция) множество объектов класса <i>UserBanInfo</i> .	
					Работа с аккаунтами пользователей осуществляется посредством класса <i>UserAccountRepository</i> , который включает в себя (композиция) множество объектов класса <i>UserAccount</i> .	
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	В приложении на рисунках П5.11, П5.12 и П5.13 отображены диаграммы классов сервисов-сокетов, которые обрабатывают запросы и ответы по протоколу WebSocket.	
					Класс <i>SocketManager</i> реализует WebSocket-сервер, то есть осуществляет двустороннюю связь с низкой задержкой с клиентом посредством протокола	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист
						28

WebSocket. Данный класс включает в себя (композиция) ряд классов-сервисов: **GeneralSocketService**, **AdminSocketService** и **RoomSocketService**.

Класс **GeneralSocketService** обрабатывает события общего плана, например, связанных с главной страницей приложения.

Класс **AdminSocketService** обрабатывает события, связанных с процессом администрирования конференцией, а класс **RoomSocketService** обрабатывает события, связанных с комнатой (конференцией). При необходимости оба класса-сервиса используют классы-репозитории.

В приложении на рисунке П5.14 представлена диаграмма классов для общих типов, которые используются при передачи информации между клиентом и сервером.

Классы **ChatFileInfo** (информация о файле в чате) и **ChatMessage** (информация о сообщении в чате) относятся к классам, необходимым для работы чата.

Классы **NewWebRtcTransportInfo** (информация о новом транспортном канале связи для медиапоток), **ConnectWebRtcTransportInfo** (событие подключения к транспортному каналу связи для медиапоток), **NewProducerInfo** (информация о новом потоке-изготовителе) и **NewConsumerInfo** (информация о новом потоке-потребителе) предназначены для передачи информации между сервером и клиентом, необходимой для функционирования медиасервера.

Классы **UserInfo** и **UserReadyInfo** содержит информацию о новом участнике конференции.

Класс **PublicRoomInfo** содержит публичную информацию о комнате (для ссылок на комнату), а класс **RoomInfo** расширяет класс **PublicRoomInfo** и используется классом-репозиторием **RoomRepository**.

Классы **NewRoomInfo** (информация о новой комнате), **UpdateRoomInfo** (информация об изменении конфигурации комнаты), **VideoCodec** (видеокодеки для комнаты), **ActionOnUserInfo** (информация об участнике в комнате), **UserBanInfo** (информация о заблокированном пользователе) и **ChangeUserNameInfo** (информация о пользователе, который изменил имя) представляют собой информацию, передаваемую при администрировании конференции.

Классы-интерфейсы **HttpHeaders** (базовые HTTP-заголовки), **OutgoingHttpHeaders** (исходящие HTTP-заголовки), **IncomingHttpHeaders** (входящие HTTP-заголовки), **FileServiceResponse** (HTTP-ответ согласно протоколу

Инв. № подл.	Подп. и дата				Лист	
	Инв. № дубл.					
	Взам. инв. №					
	Подп. и дата					
<p>новом потоке-изготовителе) и <i>NewConsumerInfo</i> (информация о новом потоке-потребителе) предназначены для передачи информации между сервером и клиентом, необходимой для функционирования медиасервера.</p> <p>Классы <i>UserInfo</i> и <i>UserReadyInfo</i> содержит информацию о новом участнике конференции.</p> <p>Класс <i>PublicRoomInfo</i> содержит публичную информацию о комнате (для ссылок на комнату), а класс <i>RoomInfo</i> расширяет класс <i>PublicRoomInfo</i> и используется классом-репозиторием <i>RoomRepository</i>.</p> <p>Классы <i>NewRoomInfo</i> (информация о новой комнате), <i>UpdateRoomInfo</i> (информация об изменении конфигурации комнаты), <i>VideoCodec</i> (видеокодеки для комнаты), <i>ActionOnUserInfo</i> (информация об участнике в комнате), <i>UserBanInfo</i> (информация о заблокированном пользователе) и <i>ChangeUserNameInfo</i> (информация о пользователе, который изменил имя) представляют собой информацию, передаваемую при администрировании конференции.</p> <p>Классы-интерфейсы <i>HttpHeaders</i> (базовые HTTP-заголовки), <i>OutgoingHttpHeaders</i> (исходящие HTTP-заголовки), <i>IncomingHttpHeaders</i> (входящие HTTP-заголовки), <i>FileServiceResponse</i> (HTTP-ответ согласно протоколу</p>						
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	29

TUS), *FileServiceRequest* (HTTP-запрос согласно протоколу TUS) предназначены для реализации протокола возобновляемой загрузки файлов TUS.

1.5.2 Форматы данных

Список активных пользователей в комнате на сервере представляет собой ассоциативный контейнер типа ключ-значения. Ключом является идентификатор аккаунта пользователя, значением контейнера – экземпляр класса *ActiveUser*, описывающего активного пользователя.

Сохранение серверных данных (список комнат, файлов, чатов, аккаунтов пользователей, авторизованных пользователей в комнатах, список заблокированных IP-адресов) должно осуществляться в файлы формата .json в папку «data».

Механизм аутентификации и авторизации должен быть реализован с использованием JWT-токенов.

Конфигурационный файл **server.conf** является текстовым файлом, со следующими возможными значениями:

- ✦ HTTP_PORT = число (порт для HTTP сервера).
- ✦ HTTPS_PORT = число (порт для HTTPS сервера).
- ✦ SSL_PRIVATE_KEY = строка (название файла – приватного ключа SSL).
- ✦ SSL_PUBLIC_CERT = строка (название файла – публичного сертификата SSL).
- ✦ ADMIN_PASS = строка (пароль администратора).
- ✦ TOKEN_SECRET_KEY = строка (секретный ключ для сессии).
- ✦ MEDIASOUP_LOCAL_IP = строка (IP-адрес сервера в локальной сети).
- ✦ MEDIASOUP_ANNOUNCED_IP = строка (внешний IP-адрес сервера).
- ✦ MEDIASOUP_RTC_MIN_PORT = число (нижняя граница диапазона портов медиасервера).
- ✦ MEDIASOUP_RTC_MAX_PORT = число (верхняя граница диапазона портов медиасервера).
- ✦ NETWORK_INCOMING_CAPABILITY = число (входящая скорость сервера в мегабитах).
- ✦ NETWORK_OUTCOMING_CAPABILITY = число (исходящая скорость сервера в мегабитах).
- ✦ FILE_MAX_SIZE = число (максимальный размер загружаемого файла в байтах).

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						Лист	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					30	

✎	SSL_PRIVATE_KEY = строка (название файла – приватного ключа SSL).
✎	SSL_PUBLIC_CERT = строка (название файла – публичного сертификата SSL).
✎	ADMIN_PASS = строка (пароль администратора).
✎	TOKEN_SECRET_KEY = строка (секретный ключ для сессии).
✎	MEDIASOUP_LOCAL_IP = строка (IP-адрес сервера в локальной сети).
✎	MEDIASOUP_ANNOUNCED_IP = строка (внешний IP-адрес сервера).
✎	MEDIASOUP_RTC_MIN_PORT = число (нижняя граница диапазона портов медиасервера).
✎	MEDIASOUP_RTC_MAX_PORT = число (верхняя граница диапазона портов медиасервера).
✎	NETWORK_INCOMING_CAPABILITY = число (входящая скорость сервера в мегабитах).
✎	NETWORK_OUTCOMING_CAPABILITY = число (исходящая скорость сервера в мегабитах).
✎	FILE_MAX_SIZE = число (максимальный размер загружаемого файла в байтах).

1.5.3 Алгоритм получения видео- и аудиопотоков пользователя

Дано: *stream* – переменная, хранящая прямой медиапоток; *streamConstraints* – переменная, хранящая параметры прямого потока; *localVideo* – html-объект видео.

1. Установить *streamConstraints* = { "audio": true, "video": true }.
2. Вызвать метод *navigator.mediaDevices.getUserMedia(streamConstraints)* и поместить результат в переменную *stream*.
3. Поместить *stream* в *localVideo*.
4. Если произошла ошибка при вызове получения медиапотока, то вывести сообщение об этом в консоль.

1.5.4 Алгоритм получения очередного маршрутизатора потоков на сервере

Дано: *consuming* – булева переменная: маршрутизатор предназначен для потребляющих потоков или производящих.

Если !*consuming* или количество ядер сервера == 1, то

| вернуть *mediasoup.router*, привязанный к первому ядру.

Иначе

| Увеличить индекс последнего маршрутизатора.

| Если индекс равен количеству ядер сервера то присвоить значение
| индексу последнего маршрутизатора = 1.

| Вернуть *mediasoup.router* привязанный к ядру, индекс которого равен
| индексу последнего маршрутизатора.

Конец ветвления

1.5.5 Алгоритм создания транспортного канала для исходящих медиапотоков

Дано: *device* – объект класса *mediasoup.Device* на клиенте.

1. Получить *router* объект класса *mediasoup.Router* – маршрутизатор потоков, привязанный за одним из ядер сервера, см. алгоритм 1.6.4.
2. Создать транспортный канал на сервере, вызвав метод *router.createWebRtcTransport()*.
3. Передать клиенту информацию о созданном транспортном канале.
4. Создать транспортный канал на клиенте, используя полученную информацию при вызове метода *device.createProducerTransport()*.
5. Обработать события «connect» и «produce» созданного транспортного канала.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист 31	
Изм	Лист	№ документа	Подпись	Дата							

1.5.6 Алгоритм создания транспортного канала для входящих медиапотоков

Дано: *device* – объект класса **mediasoup.Device** на клиенте.

1. Получить *router* объект класса **mediasoup.Router** – маршрутизатор потоков, привязанный за одним из ядер сервера, см. алгоритм **1.6.4**.
2. Создать транспортный канал на сервере, вызвав метод **router.createWebRtcTransport()**.
3. Передать клиенту информацию о созданном транспортном канале.
4. Создать транспортный канал на клиенте, используя полученную информацию при вызове метода **device.createConsumerTransport()**.
5. Обработать события «connect» созданного транспортного канала.

1.5.7 Алгоритм передачи медиадорожки на сервер

Дано: *transport* – транспортный канал, объект класса **mediasoup.Transport** на клиенте

1. Получить медиадорожку, см. *алгоритм 1.6.3*.
2. Вызвать метод **transport.produce()** и указать в качестве параметра медиадорожку.
3. Если это первый вызов **transport.produce()**, то сработает событие транспортного канала «connect».
4. Передать на сервер информацию из события, после того, как сработает событие транспорта «produce».
5. Создать на сервере объект класса **mediasoup.Producer** на первом ядре сервера, используя полученную информацию.

1.5.8 Алгоритм получения удаленной медиадорожки с сервера

Дано: *transport* – транспортный канал, *newConsumerInfo* – переменная, хранящая информацию о удаленной медиадорожке (номер, видео или аудио, медиакодеки); **mediaStream** – медиапоток собеседника, привязанный к HTML-видеоэлементу собеседника.

1. Создать на сервере объект класса **mediasoup.Consumer**, вызвав метод **transport.consume()** и указав в качестве параметра номер соответствующего объекта класса **mediasoup.Producer**.
2. Отправить клиенту информацию о созданном **consumer**.
3. Создать объект класса **mediasoup.Consumer** на клиенте, используя полученную информацию и вызвав метод **transport.consume()** на клиенте.

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист 32	
Изм	Лист	№ документа	Подпись	Дата							

4. Если это первый вызов **transport.consume()**, то сработает событие транспортного канала «connect».
5. Поместить медиадорожку из **consumer** в **mediaStream**.

1.5.9 Алгоритм авторизации в комнате для клиента

1. Отправить на сервер запрос с номером комнаты и паролем.
2. Получить ответ на запрос от сервера.
3. Если авторизация успешна, то отправить запрос перехода в комнату на сервер.
4. Иначе сообщить пользователю о некорректном пароле.

1.5.10 Алгоритм аутентификации и авторизации в комнате для сервера

1. Получить идентификатор запрашиваемой комнаты из запроса пользователя.
 2. Прочитать токен из запроса пользователя, проверить его и определить идентификатор пользователя.
 3. Если запрашиваемая комната существует в списке комнат на сервере, то
 - | Если пользователь присутствует в списке авторизованных пользователей в комнате, то отправить страницу с комнатой.
 - | Иначе прочитать пароль от комнаты из запроса пользователя и если пароль верный, то сохранить идентификатор пользователя в список авторизованных пользователей в комнате.
 - | Иначе отправить ответ со статусом ошибки «403».
- Иначе отправить ответ со статусом ошибки «404».

1.5.11 Алгоритм организации конференции

1. Сообщить пользователю название комнаты.
2. Сообщить пользователю перечень поддерживаемых сервером медиакодеков.
3. Дождаться инициализации создания клиентом транспортного канала для входящих медиапоток (см. алгоритм 1.6.6).
4. Сообщить всем пользователям в комнате о том, что подключился новый пользователь.
5. Создать объекты класса **mediasoup.Consumer** (входящие медиапотoki) для всех объектов класса **mediasoup.Producer** (исходящие медиапотoki) других пользователей.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
					<p>пользователей в комнате, то отправить страницу с комнатой.</p> <p>Иначе прочитать пароль от комнаты из запроса пользователя</p> <p>и если пароль верный, то сохранить идентификатор пользователя в</p> <p>список авторизованных пользователей в комнате.</p> <p>Иначе отправить ответ со статусом ошибки «403».</p> <p>Иначе отправить ответ со статусом ошибки «404».</p>					
					<p>1.5.11 Алгоритм организации конференции</p> <p>1. Сообщить пользователю название комнаты.</p> <p>2. Сообщить пользователю перечень поддерживаемых сервером медиакодеков.</p> <p>3. Дождаться инициализации создания клиентом транспортного канала для входящих медиапотокa (см. алгоритм 1.6.6).</p> <p>4. Сообщить всем пользователям в комнате о том, что подключился новый пользователь.</p> <p>5. Создать объекты класса mediasoup.Consumer (входящие медиапотокa) для всех объектов класса mediasoup.Producer (исходящие медиапотокa) других пользователей.</p>					
										Лист
					ВКРБ 09.03.04.059.2022					
Изм	Лист	№ документа	Подпись	Дата						33

6. Передать пользователю информацию о созданных объектах, чтобы он мог создать их локально и начать принимать медиадорожки (*см. алгоритм 1.6.8*).
7. Если пользователь передал медиадорожку на сервер (*см. алгоритм 1.6.8*), то сообщить всем пользователям о новом объекте класса **mediasoup.Producer**, чтобы они могли начать принимать медиадорожки (*см. алгоритм 1.6.8*).
8. Если пользователь отключился, то сообщить всем другим пользователям в комнате, что данный пользователь отключился от сервера.

1.5.12 Алгоритм загрузки части файла для возобновляемой загрузки файлов

1. Отправить **HEAD**-запрос с указанием идентификатора файла для выяснения смещения в байтах, с которых необходимо начать загрузку файла (если файл загружается впервые, то смещение равно нулю).
2. Вернуть значения смещения для загрузки файла клиенту.
3. Отправить **PATCH**-запрос на сервер с указанием идентификатора файла и прикрепив в тело запроса файл, начав загрузку с того места файла, как того требует полученное смещение
4. Принимать файловый поток от клиента, перенаправляя его в файл и сохраняя текущую длину загруженных байт файла на сервере, чтобы в случае повторной загрузки файла клиентом вернуть это значение в качестве смещения.

1.5.13 Алгоритм создания файла на сервере для возобновляемой загрузки файлов

1. Отправить **POST**-запрос с пустым телом запроса на сервер, прикрепив заголовки «Upload-Length» (длина загружаемого файла в байтах) и «Upload-Metadata» (название загружаемого файла и его типа).
2. Создать пустой файл на сервере и вернуть идентификатор файла клиенту.

1.6 Основные сценарии работы программного продукта

Для отображения основных сценариев работы с системой используются диаграмма состояний, диаграмма последовательности и диаграмма активностей[1][2][8][9].

Инв. № подл.	Подп. и дата				Взам. инв. №	Инв. № дубл.				Подп. и дата														
Изм												Лист												Лист
№ документа												Подпись												
Дата												ВКРБ 09.03.04.059.2022												34

того требует полученное смещение

4. Принимать файловый поток от клиента, перенаправляя его в файл и сохраняя текущую длину загруженных байт файла на сервере, чтобы в случае повторной загрузки файла клиентом вернуть это значение в качестве смещения.

1.5.13 Алгоритм создания файла на сервере для возобновляемой загрузки файлов

1. Отправить **POST**-запрос с пустым телом запроса на сервер, прикрепив заголовки «Upload-Length» (длина загружаемого файла в байтах) и «Upload-Metadata» (название загружаемого файла и его типа).

2. Создать пустой файл на сервере и вернуть идентификатор файла клиенту.

1.6 Основные сценарии работы программного продукта

Для отображения основных сценариев работы с системой используются диаграмма состояний, диаграмма последовательности и диаграмма активностей[1][2][8][9].

1.6.1 Диаграмма последовательности процесса захвата медиапотока

Одним из основных сценариев работы рассматриваемой системы является процесс захвата медиапотока участником конференции и передача этого медиапотока другим участникам конференции. Диаграмма последовательности, иллюстрирующая этот процесс, представлена в *Приложении 6*.

При захвате медиапотока у участника есть выбор источника этого захватываемого медиапотока, это может быть как микрофон, веб-камера так и экран компьютера. В соответствии с выбором, браузер возвращает медиапоток (*MediaStream*), который объект класса *UserMedia* передаёт объекту класса *Room*, а тот в свою очередь оборачивает этот медиапоток в поток-производитель (*Producer*) с помощью метода *createProducer*. Объект класса *Mediasoup* создаёт и возвращает поток-производителя, информацию о котором объект класса *Room* отправляет на сервер.

Как только сервер получает в распоряжение информацию о потоке-производителе, он (объект класс *Room*) создаёт серверный поток-производитель с помощью метода *createProducer* серверного класса *Mediasoup*. Как только поток-производитель был создан, объект класса *Room* для каждого участника конференции создаёт поток-потребитель с помощью метода *createConsumer* класса *Mediasoup* и отправляет эту информацию о каждом созданном потоке-потребителе соответствующего потоку участнику.

После того, как клиент получил информацию о потоке-потребителе, он (объект класса *Room*) создает клиентский поток-потребитель с помощью метода *createConsumer* класса *Mediasoup* и вызывает метод *addTrackToRemoteVideo* класса *UI*, чтобы добавить полученный медиапоток в соответствующий видеовиджет собеседника.

1.6.2 Диаграмма состояний медиапотока

В рассматриваемой системе существует два вида медиапотоков: поток-потребитель (*Consumer*) и поток-производитель (*Producer*). В *Приложении 7* представлены диаграммы состояний для двух видов медиапотоков на стороне сервера.

Поток-производитель изначально при создании с помощью метода *createProducer* пребывает в активном состоянии, из которого он может завершиться, если был вызван метод *closeProducer*, или быть приостановленным на паузу методом

Инв. № подл.	Подп. и дата		Инв. № дубл		Подп. и дата	
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		Лист
Подп. и дата		Инв. № дубл		Подп. и дата		Лист
Инв. № подл.		Подп. и дата		Инв. № дубл		Лист
Взам. инв. №		Инв. № дубл		Подп. и дата		

pauseProducer. В состоянии паузы он может быть снят с неё и вернуться в состояние «Активен» с помощью метода *resumeProducer*, а также может завершиться методом *closeProducer*.

Похожие состояния и переходы между ними и для потока-потребителя, однако важно учесть, что поток-потребитель создается изначально в состоянии «Приостановлен».

1.6.3 Диаграмма активностей медиапотока

Медиапотoki переходят из одного состояния в другое в следствие выполнения какой-либо деятельности. В *Приложении 8* представлены диаграммы активностей для двух видов медиапотоков на стороне сервера, которые описывают деятельность при создании, приостановке и снятия медиапотока с паузы.

В приложении на рисунке П8.1 представлена деятельность метода *createProducer*, то есть создание медиапотока-производителя на сервере. Объект класса **Room** создаёт поток-производитель с помощью объекта класса *MediasoupService*. Тот в свою очередь не только создаёт поток-производитель, но и увеличивает счётчик потоков-производителей на сервере, для того чтобы можно было рассчитать максимальный допустимый битрейт медиапотока в следствие ограниченной пропускной способности сервера. После того, как был создан поток, объект класса **Room** вычисляет новый максимально допустимый битрейт, а также привязывается к событию закрытия транспортного канала (по которому этот поток передаётся). После этого, объект класса **Room** создаёт поток-потребитель с помощью метода *createConsumer* для каждого участника конференции. В самом конце, как только все потоки-потребители были созданы, информация о созданном потоке-производителе передаётся клиенту.

На рисунке П8.2 представлена диаграмма активностей при приостановке медиапотока-производителя на сервере. Объект класса **Room** осуществляет проверку, не стоит ли уже на паузе медиапоток, и в случае, если поток в состоянии «Активен», ставит на паузу поток, уменьшает счётчик потоков-производителей и вычисляет новый максимально допустимый битрейт.

На рисунке П8.3 представлена диаграмма активностей при создании медиапотока-потребителя на сервере. Объект класса **Room** с помощью класса *MediasoupService* создаёт поток-потребитель, привязывается к событиям, связанным с этим потоком, и создав объект с информации о новом потоке, отправляет её

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					Лист
										36

было рассчитать максимальный допустимый битрейт медиапотока в следствие ограниченной пропускной способности сервера. После того, как был создан поток, объект класса **Room** вычисляет новый максимально допустимый битрейт, а также привязывается к событию закрытия транспортного канала (по которому этот поток передаётся). После этого, объект класса **Room** создаёт поток-потребитель с помощью метода **createConsumer** для каждого участника конференции. В самом конце, как только все потоки-потребители были созданы, информация о созданном потоке-производителе передаётся клиенту.

На рисунке П8.2 представлена диаграмма активностей при приостановке медиапотока-производителя на сервере. Объект класса **Room** осуществляет проверку, не стоит ли уже на паузе медиапоток, и в случае, если поток в состоянии «Активен», ставит на паузу поток, уменьшает счётчик потоков-производителей и вычисляет новый максимально допустимый битрейт.

На рисунке П8.3 представлена диаграмма активностей при создании медиапотока-потребителя на сервере. Объект класса **Room** с помощью класса **MediasoupService** создаёт поток-потребитель, привязывается к событиям, связанным с этим потоком, и создав объект с информации о новом потоке, отправляет е

клиенту. Необходимо отметить, что здесь не увеличивается счётчик с активными потоками-потребителями, поскольку изначально поток-потребитель создаётся в состоянии «Приостановлен».

На рисунке П8.4 представлена диаграмма активностей при снятии с паузы медиапотока-потребителя на сервере. Объект класса **Room** осуществляет множественную проверку, чтобы поток был на паузе, а соответствующий поток-производитель не был на паузе, и клиент был готов к снятию паузы у этого потока-потребителя. Если проверка успешно выполняется, то поток снимается с паузы и производится перерасчёт максимального битрейта для видеопотока. Сервер оповещает клиента о попытке снять поток с паузы, поскольку попытка снятия с паузы могла быть инициирована не только непосредственно клиентом, но и другими событиями.

1.7 Входные и выходные данные

Входные данные:

- ✎ локальный медиапоток (видео- и/или аудиопоток);
- ✎ текстовые данные или файлы;
- ✎ выбор пунктов меню.

Выходные данные:

- ✎ список комнат;
- ✎ удаленные медиапотоки собеседников;
- ✎ текстовые данные или файлы, полученные от собеседников.

Некоторая часть входных данных является опциональной для участника конференции. То есть допустимы случаи, когда клиент может участвовать в конференции, не отправляя собственных медиапотоков или текстовых (и файловых) данных, но при этом получая медиапотоки от других участников.

1.8 Требования к техническому и программному обеспечению

Рекомендуемая конфигурация для сервера:

- ✎ Процессор: не менее 2.4 ГГц, 4 ядра.
- ✎ Оперативная память: не менее 4 Гб.
- ✎ Свободное место на жестком диске: не менее 1 Гб.
- ✎ Минимальная скорость входящего подключения: 1 Мбит/с.
- ✎ Минимальная скорость исходящего подключения: 1 Мбит/с.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	<div>✎ выбор пунктов меню.</div> <div>Выходные данные:</div> <div>✎ список комнат;</div> <div>✎ удаленные медиапотoki собеседников;</div> <div>✎ текстовые данные или файлы, полученные от собеседников.</div> <div>Некоторая часть входных данных является опциональной для участника конференции. То есть допустимы случаи, когда клиент может участвовать в конференции, не отправляя собственных медиапотокoв или текстовых (и файловых) данных, но при этом получая медиапотoki от других участников.</div> <div>1.8 Требования к техническому и программному обеспечению</div> <div>Рекомендуемая конфигурация для сервера:</div> <div>✎ Процессор: не менее 2.4 ГГц, 4 ядра.</div> <div>✎ Оперативная память: не менее 4 Гб.</div> <div>✎ Свободное место на жестком диске: не менее 1 Гб.</div> <div>✎ Минимальная скорость входящего подключения: 1 Мбит/с.</div> <div>✎ Минимальная скорость исходящего подключения: 1 Мбит/с.</div>	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист 37

- ✦ Максимальное значение сетевой задержки (пинг) - не более 200 мс.
- ✦ Процент потерянных пакетов – не более 1 %.
- ✦ Публичный IP (если система будет работать не в локальной сети).

Операционная система сервера: Linux Debian 11 Bullseye или Windows 10.

Среда выполнения: Node.js LTS версии v12.22.x.

Рекомендуемая конфигурация для клиента:

- ☞ Процессор Intel Pentium 4 или старше с поддержкой SSE3.
- ☞ Оперативная память: не менее 2 Гб.
- ☞ Минимальная скорость входящего подключения: 512 Кбит/с.
- ☞ Минимальная скорость исходящего подключения: 512 Кбит/с.

Веб-браузер: Firefox последней мажорной версии (на данный момент 100.x.x),
Google Chrome последней мажорной версии (на данный момент 100.x.x.x).

Операционная система: дистрибутивы Linux, в которых есть возможность установить браузер необходимой версии (см. выше), а также Windows 7 или более поздние.

1.9 Безопасность информационной системы

Система ВКС должна соответствовать следующим требованиям безопасности:

- ☞ защита системы должна обеспечиваться комплексом программно-технических средств и поддерживающих их организационных мер;
- ☞ разграничение прав доступа пользователей и администраторов системы должно строиться по принципу «что не разрешено, то запрещено»;
- ☞ подключение клиентской части системы к серверной части должно осуществляться с использованием защищенных версий протоколов HTTPS, WSS и DTLS-SRTP для шифрования данных (текстовые данные, метаданные, медиапотoki);
- ☞ доступ к серверным файлам в локальной сети (файлы программного обеспечения и конфигурационные файлы) должен быть ограничен паролем администратора, а вне локальной сети должен быть запрещен;
- ☞ на сервере должны быть запрещены все порты TCP/IP кроме тех, которые необходимые для функционирования системы.

Общий уровень защищенности ИСПДн зависит от организации, использующей данный программный комплекс.

2 РАБОЧИЙ ПРОЕКТ

2.1 Общие сведения о работе системы

Клиентская часть программного продукта разработана в текстовом редакторе Visual Studio Code (версия 1.64.2) на языках HTML5 и TypeScript (версия 4.5.5) с использованием веб-браузеров Mozilla Firefox (версии 100.0.2) и Google Chrome (версия 101.0.4951.67). В данной части системы используется инструмент Webpack (версия 5.67.0) для сборки проекта в набор статических файлов, а также библиотеки howler (версия 2.2.3) для воспроизведения звуков, plyr (версия 3.6.12) для реализации собственного видеоплеера, а также socket.io-client (версия 4.4.1) для поддержки веб-сокетов на клиенте и mediasoup-client (версия 3.6.50) для работы с медиапотокami на клиенте.

Серверная часть программного продукта разработана с применением аналогичных инструментов на языке TypeScript (версия 4.5.5) на платформе Node.JS (версия 12.22.11) и фреймворке Express (версия 4.17.2) для реализации веб-сервера, а также с использованием библиотек Socket.IO (версия 4.4.1) для управления веб-сокетами, jose (версия 4.8.0) для создания и верификации JWT токенов и mediasoup (версия 3.9.6) для управления медиапотокami пользователей и реализации архитектуры SFU.

Работоспособность серверной части программы была проверена под управлением операционных систем Windows 10 v21H2 и Linux Debian 11 Bullseye.

2.2 Функциональное назначение системы

Разработанный программный продукт предназначен для организации и проведения видеоконференций. Программа имеет следующие функциональные возможности:

- ✧ организация видеоконференций согласно архитектуре SFU;
- ✧ браузеры пользователей автоматически устанавливают WebRTC соединение с медиасервером, без личного участия пользователя;
- ✧ программа предоставляет возможность пользователям изменять видео- и аудиопотоки (например, переключение с веб-камеры на демонстрацию экрана) во время уже установленного соединения;
- ✧ программа предоставляет возможность одновременного захвата нескольких видеоустройств с одного клиентского устройства;

Инв. № подл.	Подп. и дата				Взам. инв. №	Инв. № дубл	Подп. и дата							

2.3 Описание физической архитектуры системы

В соответствии с проблематикой предметной области были выделены основные уровни распределения информации, которые отображены на диаграмме развертывания, представленной в *Приложении 9*.

Участник взаимодействует с сервером посредством веб-браузера. Изначально, веб-браузер отправляет запрос к веб-сервису сервера по протоколу «HTTPS» для загрузки файлов, необходимых для работы веб-клиента: «HTML» страницы и скриптовые файлы формата «.js».

После того, как веб-клиент был загружен с сервера, участник может войти в конференцию. Для этого веб-клиент участника отправляет запросы веб-сервису сервера и проходит процесс авторизации и аутентификации в комнате. Также через веб-сервис осуществляется загрузка и скачивание файлов на сервере.

Для участия в конференции и взаимодействия с другими участниками в режиме реального времени, а также для администрирования конференцией, веб-клиент инициирует подключение к сервису веб-сокеты сервера по протоколу «WebSocket».

На сервере, веб-сервис и сервис веб-сокеты взаимодействуют с компонентами репозитория данных в тех случаях, когда необходимо загрузить или сохранить данные, связанных с серверным состоянием, например в случае, когда участнику необходимо отправить список конференций, или когда организатор создал, удалил или отредактировал конференцию.

2.4 Инсталляция и выполнение программного продукта

Для работоспособности разработанного программного продукта необходимо:

1. Скачать и установить LTS версию (v12.22.11) платформы **Node.js** с официального сайта Node.js.
2. Установить менеджер пакетов **NPM v8** для Node.js.
3. Скопировать на жесткий диск компьютера папку с исходными кодами и бинарными файлами «**webrtc-server**», содержащую подпапки «**config**», «**dist**», «**src**» и «**node_modules**», а также файл **package.json**, описывающий программу как **NPM**-пакет (с содержимым этого файла можно ознакомиться в Приложении 10).

- 3.1. Папка «**config**» включает в себя файл с конфигурацией **server.conf** и пустую подпапку «**ssl**».

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
Изм	Лист	№ документа	Подпись	Дата						41

3.2. Папка «**src**» содержит исходные TypeScript-файлы серверной части: **index.ts**, **WebService.ts**, **TokenService.ts**, **Logger.ts**, **Utils.ts** и **MediasoupService.ts**, а также подпапки «**SocketService**» (включает в себя файлы **AdminSocketService.ts**, **RoomSocketService.ts**, **SocketManager.ts** и **GeneralSocketService.ts**), «**FileService**» (включает в себя файлы **FileService.ts**, **FileRepository.ts** и **FileServiceTusProtocol.ts**), «**Room**» (включает в себя файлы **Room.ts**, **RoomChatRepository.ts** и **RoomRepository.ts**) и «**User**» (включает в себя файлы **AuthRoomUserRepository.ts**, **UserAccountRepository.ts** и **UserBanRepository.ts**).

3.3. Папка «**dist**» включает в себя скомпилированные файлы JavaScript из исходных файлов TypeScript из папки «**src**».

3.4. Папка «**node_modules**» все необходимые для работы системы NPM-зависимости (включая веб-клиент программного комплекса).

4. Скачать программу **OpenSSL** и сгенерировать самоподписный SSL-сертификат и закрытый ключ, запустив **OpenSSL** в командной строке со следующими аргументами:

```
openssl req -newkey rsa:2048 -nodes -keyout private.key -new -x509 -days 365 -out public.crt.
```

Полученные в результате файлы **public.crt** и **private.key** поместить в папку «**ssl**».

5. Открыть папку «**webrtc-server**» в терминале **Node.js** (утилита Node.js command prompt) с помощью консольной команды «**cd @**», вместо символа «**@**» указав путь к папке «**webrtc-server**».
6. Запустить проект с помощью консольной команды «**npm start**» в терминале **Node.js** (утилита Node.js command prompt).
7. Перейти на сайт по адресу localhost или 127.0.0.1 с помощью браузера, соответствующему системным требованиям.

В случае возникновения ошибок во время процесса инсталляции ПО, следует обратиться к разработчику.

2.5 Описание программы

Клиентская часть программы состоит из трех модулей: «**public/scripts/index.ts**» – модуль для главной страницы и авторизации в комнате;

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	следующими аргументами: openssl req -newkey rsa:2048 -nodes -keyout private.key -new -x509 -days 365 -out public.crt. Полученные в результате файлы public.crt и private.key поместить в папку « ssl ».
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	5. Открыть папку « webrtc-server » в терминале Node.js (утилита Node.js command prompt) с помощью консольной команды «cd @», вместо символа ”@” указав путь к папке « webrtc-server ».
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	6. Запустить проект с помощью консольной команды «npm start» в терминале Node.js (утилита Node.js command prompt).
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	7. Перейти на сайт по адресу localhost или 127.0.0.1 с помощью браузера, соответствующему системным требованиям.
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	В случае возникновения ошибок во время процесса инсталляции ПО, следует обратиться к разработчику.
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	2.5 Описание программы
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	Клиентская часть программы состоит из трех модулей: «public/scripts/index.ts» – модуль для главной страницы и авторизации в комнате;
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	ВКРБ 09.03.04.059.2022
Изм	Лист	№ документа	Подпись	Дата	Лист
					42

«rooms/scripts/index.ts» – модуль поддержки комнат; «admin/scripts/index.ts» – модуль, реализующий режим администратора.

Модуль «public/scripts/index.ts» состоит из 2 классов IndexSocketService и AuthService.

В таблице 2.1 приведено описание класса IndexSocketService, организующего работу с веб-сокетами на главной странице (файл IndexSocketService.ts).

Таблица 2.1 – Описание класса IndexSocketService

Поле	Тип	Назначение
socket	Socket	Веб-сокет соединение
roomList	HTMLDivElement	Список комнат
Метод		Описание
constructor()		Конструктор для класса
createRoomList(rooms: PublicRoomInfo[])		Добавляет на главную страницу ссылки на комнаты
addRoomToList(room: PublicRoomInfo)		Добавляет на главную страницу ссылку на комнату
removeRoomFromList(id: string)		Удалить ссылку на комнату на главной странице
updateRoomInList(id: string)		Обновить ссылку на комнату на главной странице
onDisconnect(): void		Функция-обработчик события отсоединения от сервера

В таблице 2.2 приведено описание класса AuthService, организующего работу с запросами на странице авторизации в комнату (файл AuthService.ts).

Таблица 2.2 – Описание класса AuthService

Поле	Тип	Назначение
passInput	HTMLInputElement	Поле для ввода пароля
btn_join	HTMLButtonElement	Кнопка «войти»
Метод		Описание
constructor()		Конструктор для класса
btn_join_clicked()		Функция, отправляющая запрос авторизации на сервер с заданным паролем

Модуль «rooms/scripts/index.ts» состоит из 4 классов – для работы с интерфейсом, для работы с библиотекой mediasoup-client, для захвата медиапотоков пользователя и для реализации комнаты на клиенте, а также еще из 6 классов – для загрузки файлов на сервер.

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист 43
-----	------	-------------	---------	------	------------------------	------------

В таблице 2.3 приведено описание класса UI, организующего работу с HTML-интерфейсом (файл UI.ts).

Таблица 2.3 – Описание класса UI

Поле	Тип	Назначение
buttons	Map<string, HTMLButtonElement>	Контейнер HTML-кнопок
roomName	HTMLSpanElement	HTML-элемент с названием комнаты
allVideos	Map<string, HTMLVideoElement>	Контейнер всех HTML-элементов видео на экране
chat	HTMLDivElement	Текстовый чат
fileInput	HTMLInputElement	Поле для выбора файла для отправки
filesProgress	HTMLDivElement	Контейнер с прогресс-барами отправки файлов
messageText	HTMLTextAreaElement	Текстовое поле для ввода сообщения, отправляемого в чат
captureSettingsDisplay	HTMLSelectElement	Виджет для выбора разрешения захвата экрана клиента
captureSettingsCam	HTMLSelectElement	Виджет для выбора разрешения веб-камеры
micDevices	HTMLSelectElement	Список устройств-микрофонов
camDevices	HTMLSelectElement	Список видеоустройств (веб-камер)
usernameInput	HTMLInputElement	Поле для ввода имени пользователя
checkboxNotifications	HTMLInputElement	Чекбокс для включения/выключения звуковых оповещений
videoRows	number	Количество строк в раскладке видеоокон
videoColumns	number	Количество столбцов в раскладке видеоокон
mutePolicy	boolean	Текущая политика Mute для видео (свойство muted)
uiSounds	Map<UiSound, Howl>	Звуковые оповещения при событиях

Изм.	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист	44

Продолжение таблицы 2.3

Поле	Тип	Назначение
usernames	Map<string, string>	Таблица-связка userId – имя пользователя
Метод		Описание
handleButtons()		Подключить обработчики к кнопкам
toggleSoundsButtons()		Переключить видимость кнопок включения/выключения звуков собеседника
toggleMicPauseButtons()		Переключить видимость кнопок включения/выключения (пауза) микрофона
toggleMicButtons()		Переключить видимость кнопок захвата/прекращения захвата микрофона
toggleCamButtons()		Переключить видимость кнопок захвата/прекращения захвата вебки
toggleDisplayButtons()		Переключить видимость кнопок захвата/прекращения захвата экрана компьютера
hideMicPauseButtons()		Скрыть видимость кнопок включения/выключения (пауза) микрофона
addOptionToSelect(select: HTMLSelectElement, label: string, value: string)		Добавить новый выбор в виджет Select
prepareButtons(): Map<string, HTMLButtonElement>		Подготовить контейнер map с кнопками
prepareMessageText()		Обработка событий для виджета, куда вводится сообщение пользователя, отправляемое в чат
setNewUsernameFromInput(): string		Установить новое имя для пользователя
displayUserName()		Показать имя пользователя
enableSounds()		Включить звук для всех видео
disableSounds(disable = true)		Выключить звук для всех видео

Инов. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Продолжение таблицы 2.3

Метод	Описание
createVideoItem(userId: string, streamId: string, name: string): HTMLDivElement	Создать видеоэлемент
addVideo(userId: string, videoId: string, name: string)	Добавить новый видеоэлемент пользователя userId
addSecondaryVideo(userId: string, streamId: string, name: string)	Добавить новый неосновной-видеоэлемент пользователя userId
updateVideoLabels(userId: string, newName: string)	Обновления текстовой метки видеоэлемента
removeVideoItem(videoItem: HTMLDivElement)	Удалить видео собеседника (и обновить раскладку)
removeVideos(userId: string)	Удалить все видео от собеседника (и обновить раскладку)
calculateLayout()	Подсчитать количество столбцов и строк в раскладке в зависимости от количества собеседников
resizeVideos()	Перестроить раскладку
prepareVideoPlayer(video: HTMLVideoElement, isLocal: boolean)	Подготовить плеер для видеоэлемента
hideControls(player: Plyr, hide = true)	Скрыть элементы управления у плеера
hideVolumeControl(player: Plyr, hide = true)	Скрыть регулировку звука у плеера
showControls(player: Plyr, hasAudio: boolean)	Показать элементы управления у плеера
prepareVideoLabel(): HTMLSpanElement	Подготовить новую текстовую метку для локального видеоэлемента
setTextIntoVideoLabel(label: HTMLSpanElement, name: string, streamId: string, userId: string)	Подготовить текстовую метку для видеоэлемента
getVideoItem(userId: string, streamId: string): HTMLDivElement null	Получить контейнер с видеоэлементом
getVideoItems(userId: string): NodeListOf<HTMLDivElement>	Получить контейнеры с видеоэлементами пользователя userId

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	

Изм	Лист	№ документа	Подпись	Дата
-----	------	-------------	---------	------

ВКРБ 09.03.04.059.2022

Лист
46

Продолжение таблицы 2.3

Метод	Описание
getCenterVideoLabel(userId: string, streamId: string): HTMLSpanElement null	Получить центральную метку пользователя userId для потока streamId
getCenterVideoLabels(userId: string): NodeListOf<HTMLSpanElement>	Получить все центральные метки пользователя userId
getVideoLabel(userId: string, streamId: string): HTMLSpanElement null	Получить боковую метку пользователя userId для потока streamId
getVideoLabels(userId: string): NodeListOf<HTMLSpanElement>	Получить все боковые метки пользователя userId
toggleVideoLabels(first: HTMLSpanElement, second: HTMLSpanElement)	Переключить видимость для меток на видео
changeCheckboxNotificationsState()	Установить новое состояние для чекбокса notifications
playSound(sound: UiSound)	Воспроизвести звук
getTimestamp(datetime: number): string	Получить время в формате 00:00:00 (24 часа)
displayChatMessage(message: ChatMessage)	Отобразить сообщение в чате
displayChatLink(message: ChatMessage)	Отобразить ссылку на файл в чате
updateNicknameInChat(userId: string)	Обновить имя участника в чате

В таблице 2.4 приведено описание класса Mediasoup, предоставляющего доступ к методам библиотеки mediasoup-client (файл Mediasoup.ts).

Таблица 2.4 – Описание класса Mediasoup

Поле	Тип	Назначение
device	mediasoup.Device	Описание устройства клиента
producerTransport	mediasoup.Transport	Транспортный канал для отправки потоков
consumerTransport	mediasoup.Transport	Транспортный канал для приема потоков
consumers	Map<string, Consumer>	Обработчики входящих медиадорожек (потребители)
producers	Map<string, Producer>	Обработчики исходящих медиадорожек (производители)

Продолжение таблицы 2.4

Поле	Тип	Назначение
linkMapTrackConsumer	Map<string, string>	Контейнер-связка Id потребителя и Id медиадорожки
Метод		Описание
constructor()		Конструктор для класса
async loadDevice(routerRtpCapabilities: mediasoup.RtpCapabilities)		Функция, инициализирующая описание устройства клиента, путем передачи перечня поддерживаемых кодеков сервером
createConsumerTransport(transport: NewWebRtcTransportInfo)		Создать транспортный канал для входящих медиапотоков
createProducerTransport(transport: NewWebRtcTransportInfo)		Создать транспортный канал для исходящих медиапотоков
async createConsumer(newConsumerInfo: NewConsumerInfo): Promise<Consumer>		Создать обработчика входящего потока (потребителя)
async createProducer(newConsumerInfo: NewConsumerInfo): Promise<Producer>		Создать обработчика исходящего потока (производителя)
getConsumer(consumerId: string): Consumer undefined		Получить consumer по Id
getConsumerByTrackId(trackId: string): string undefined		Получить consumer по Id его медиадорожки
deleteConsumer(consumer: Consumer): boolean		Удалить consumer
getProducer(producerId: string): Producer undefined		Получить producer по Id
getProducers(): IterableIterator<Producer>		Получить всех producers (итератор)
deleteProducer(producer: Producer): boolean		Удалить producer

В таблице 2.5 приведено описание класса UserMedia, реализующего захват медиапоточков пользователя (файл UserMedia.ts).

Таблица 2.5 – Описание класса UserMedia

Поле	Тип	Назначение
ui	UI	Объект для работы с интерфейсом
room	Room	Объект - комната
streams	Map<string, MediaStream>	Медиапотoki
capturedVideoDevices	Set<string>	Список захваченных видеоустройств

Продолжение таблицы 2.5

Поле	Тип	Назначение
mainStreamDeviceId	string	Идентификатор видеоустройства, захваченного в главном медиапоток (main)
streamConstraintsMic	MediaStreamConstraints	Настройки медиапотока при захвате микрофона
captureConstraintsDisplay	Map<string, MediaStreamConstraints>	Настройки медиапотока при захвате видеоизображения экрана
captureConstraintsCam	Map<string, MediaStreamConstraints>	Настройки медиапотока при захвате изображения веб-камеры
Метод		Описание
constructor(ui: UI, room: Room)		Конструктор класса
handleDevicesList()		Подготовить список устройств и подключение обработчика событий изменения устройств
handleChoosingCamDevices()		Подключение обработчика изменения выбора видеоустройства
handleButtons()		Подключить обработчики к кнопкам
async handleGetMic()		Обработка нажатия на кнопку "Захватить микрофон"
handleStopMic()		Обработка нажатия на кнопку "Прекратить захват микрофона"
handleGetCam()		Обработка нажатия на кнопку "Захватить веб-камеру"
handleStopCam(deviceId: string)		Обработка нажатия на кнопку "Прекратить захват веб-камеры"
async handleGetDisplay()		Обработка нажатия на кнопку "Демонстрация экрана"
handleStopDisplay()		Обработка нажатия на кнопку "Прекратить захват экрана"
async getUserMedia(streamConstraints: MediaStreamConstraints, deviceId: string): Promise<string>		Получение потока видео (веб-камера) или аудио (микрофон).

<table><tr><td>Инв. № подл.</td><td>Подп. и дата</td><td>Взам. инв. №</td><td>Инв. № дубл</td><td>Подп. и дата</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата							изменения выбора видеоустройства
	Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата							
	handleButtons()	Подключить обработчики к кнопкам										
	async handleGetMic()	Обработка нажатия на кнопку "Захватить микрофон"										
	handleStopMic()	Обработка нажатия на кнопку "Прекратить захват микрофона"										
	handleGetCam()	Обработка нажатия на кнопку "Захватить веб-камеру"										
	handleStopCam(deviceId: string)	Обработка нажатия на кнопку "Прекратить захват веб-камеры"										
	async handleGetDisplay()	Обработка нажатия на кнопку "Демонстрация экрана"										
handleStopDisplay()	Обработка нажатия на кнопку "Прекратить захват экрана"											
async getUserMedia(streamConstraints: MediaStreamConstraints, deviceId: string): Promise<string>	Получение потока видео (веб-камера) или аудио (микрофон).											

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Лист
49

Продолжение таблицы 2.5

Метод	Описание
async updateDevicesAfterGettingPermissions(mediaStream: MediaStream, deviceId: string, isAudioDevice: boolean): Promise<string> isNeededSecondaryVideoStream(): boolean	Обновление списка устройств после получения прав в соответствии с браузером (Firefox или Chrome) Нужен ли дополнительный видеоэлемент под веб-камеру
async getDisplayMedia()	Захват видео с экрана пользователя
async handleMediaStream(streamId: string, mediaStream: MediaStream, videoDeviceId?: string)	Обработка медиапотока с вебкой или микрофоном.
async handleMediaDisplayStream(stream: MediaStream)	Обработка медиапотока (захват экрана)
handleEndedTrack(streamId: string, track: MediaStreamTrack, videoDeviceId?: string)	Обработка закончившейся (ended) дорожки
removeEndedTrack(streamId: string, track: MediaStreamTrack)	Удалить закончившуюся (ended) дорожку
removeEndedDisplayStream()	Удалить закончившийся поток display - экран компьютера: видео и аудио (если есть)
removeTrackFromMainStream(streamId: string, track: MediaStreamTrack)	Удалить медиадорожку из локального основного стрима
removeTrackFromSecondaryStream(streamId: string)	Удалить медиадорожку из локального неосновного стрима
pauseMic()	Выключить микрофон (поставить на паузу)
unpauseMic()	Включить микрофон (снять с паузы)
prepareCaptureDisplayConstraints(): Map<string, MediaStreamConstraints>	Подготовить опции с разрешениями захватываемого видеоизображения
prepareCaptureCamConstraints(): Map<string, MediaStreamConstraints>	Подготовить опции с разрешениями захватываемого изображения с веб- камеры
async prepareDevices(isMicDevices: boolean)	Подготовить опции с устройствами- микрофонами
async refreshDevicesLabels(isMicDevices: boolean)	Обновить названия с устройствами

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Лист
50

В таблице 2.6 приведено описание класса Room, реализующего комнату на клиенте (файл Room.ts).

Таблица 2.6 – Описание класса Room

Поле	Тип	Назначение
ui	UI	Экземпляр класса интерфейса
socket	Socket	Веб-сокет
mediasoup	Mediasoup	Экземпляр класса для работы с библиотекой mediasoup-client
fileService	FileService	Экземпляр класса для загрузки файлов на сервер
soundDelayAfterJoin	boolean	Задержка после входа на воспроизведение звуковых оповещений
roomId	string	Идентификатор комнаты
Метод		Описание
constructor(_ui: UI, _mediasoup: Mediasoup, _fileService: FileService)		Конструктор класса
handleFileUpload()		Обработка загрузки файлов
handleChat()		Обработка событий интерфейса связанных с чатом
handleSocketEvents()		Обработка событий Socket
pauseAndPlayEventsPlayerHandler(userId: string, streamId: string)		Обработка паузы и снятие паузы на плеере
handleButtons()		Обработка нажатий на кнопки
async routerRtpCapabilities(routerRtpCapabilities: MediasoupTypes.RtpCapabilities)		Получение rtpCapabilities (кодеки) сервера и инициализация ими mediasoup device
handleCommonTransportEvents(localTransport: MediasoupTypes.Transport)		Обработка общих событий для входящего и исходящего транспортных каналов
createConsumerTransport(transport: NewWebRtcTransportInfo)		Создать транспортный канал для приема потоков
createProducerTransport(transport: NewWebRtcTransportInfo)		Создать транспортный канал для отдачи потоков
handleProducerTransportEvents(localTransport: MediasoupTypes.Transport)		Обработка событий исходящего транспортного канала
async newConsumer(newConsumerInfo: NewConsumerInfo)		Обработка события - новый входящий медиапоток

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата
-----	------	-------------	---------	------

ВКРБ 09.03.04.059.2022

Продолжение таблицы 2.6

Метод	Описание
closeConsumer(consumerId: string)	Обработка события - входящий медиапоток закрывается
removeRemoteSecondaryVideo(producerU serId: string, streamId: string)	Удалить неосновной видеозлемент другого пользователя
async addMediaStreamTrack(streamId: string, track: MediaStreamTrack)	Добавить медиапоток (одну дорожку) в подключение
removeMediaStreamTrack(trackId: string)	Удалить медиапоток (дорожку) из подключения
pauseMediaStreamTrack(trackId: string)	Поставить медиапоток (дорожку) на паузу
resumeMediaStreamTrack(trackId: string)	Снять медиапоток (дорожку) с паузы

В таблице 2.7 приведено описание класса FileService, реализующего загрузку файлов на сервер (файл FileService.ts).

Таблица 2.7 – Описание класса FileService

Поле	Тип	Назначение
maxSize	number	Максимальный размер файла
Метод		Описание
constructor()		Конструктор класса
fileServiceOptions()		Узнать настройки протокола TUS на сервере
async createFileOnServer(roomId: string, file: File): Promise<string>		Создать пустой файл на сервере и получить его идентификатор
async getUploadOffset(fileId: string): Promise<string>		Узнать смещение для загрузки файла, имеющего идентификатор fileId
async fileUpload(roomId: string, file: File, progressComponent: HTMLDivElement)		Загрузить файл на сервер

В таблице 2.8 приведено описание класса `TusPatchRequest`, являющимся шаблоном для PATCH-запроса, необходимого для реализации загрузки файлов на сервер классом `FileService` (файл `FileServiceTusProtocol.ts`).

Таблица 2.8 – Описание класса TusPatchRequest

Поле	Тип	Назначение
method	string	Тип запроса (PATCH)
path	string	Путь запроса

Продолжение таблицы 2.8

Поле	Тип	Назначение
headers	IncomingHttpHeaders	Заголовки запроса
successfulStatusCode	number	Код ответа – успех
Метод		Описание
constructor(fileId: string, uploadOffset: string)		Конструктор класса

В таблице 2.9 приведено описание класса TusOptionsRequest, являющимся шаблоном для OPTIONS-запроса, необходимого для реализации загрузки файлов на сервер классом FileService (файл FileServiceTusProtocol.ts).

Таблица 2.9 – Описание класса TusOptionsRequest

Поле	Тип	Назначение
method	string	Тип запроса (OPTIONS)
path	string	Путь запроса
successfulStatusCode	number	Код ответа – успех

В таблице 2.10 приведено описание класса TusPostCreationRequest, являющимся шаблоном для POST-запроса, необходимого для реализации загрузки файлов на сервер классом FileService (файл FileServiceTusProtocol.ts).

Таблица 2.10 – Описание класса TusPostCreationRequest

Поле	Тип	Назначение
method	string	Тип запроса (POST)
path	string	Путь запроса
headers	IncomingHttpHeaders	Заголовки запроса
successfulStatusCode	number	Код ответа – успех
Метод		Описание
constuctor(filename: string, filetype: string, uploadLength: number, roomId: string)		Конструктор класса

В таблице 2.11 приведено описание класса TusHeadRequest, являющимся шаблоном для HEAD-запроса, необходимого для реализации загрузки файлов на сервер классом FileService (файл FileServiceTusProtocol.ts).

Таблица 2.11 – Описание класса TusHeadRequest

Поле	Тип	Назначение
method	string	Тип запроса (POST)
path	string	Путь запроса
headers	IncomingHttpHeaders	Заголовки запроса
successfulStatusCode	number	Код ответа – успех
Метод		Описание
constuctor(fileId: string)		Конструктор класса

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
Изм.	Лист	№ документа	Подпись	Дата						53

Модуль «admin/scripts/index.ts» состоит из 1 класса AdminSocketService.

В таблице 2.12 приведено описание класса AdminSocketService, организующего работу с веб-сокетами на странице панели администратора (файл AdminSocketService.ts).

Таблица 2.12 – Описание класса AdminSocketService

Поле	Тип	Назначение
socket	Socket	Веб-сокет соединение для администратора
generalSocket	Socket	Общее веб-сокет соединение
videoCodecSelect	HTMLSelectElement	Поле с выбором видеокодека при создании комнаты
latestSubscribedRoomId	string	Идентификатор комнаты, на получение изменений списка пользователей которой подписался администратор
roomInfoList	Map<string, PublicRoomInfo>	Список комнат
Метод		Описание
constructor()		Конструктор класса
getSelectedRoom(): string		Отдать Id выбранной комнаты
getSelectedUser(): string		Отдать Id выбранного пользователя
getSelectedActionOnUserInfo(): ActionOnUserInfo		Отдать выбранные Id комнаты и пользователя
handleRoomButtons()		Обработка кнопок связанных с комнатами
handleUserButtons()		Обработка кнопок, связанных с пользователем
handleEvents()		Обработка сокет событий
prepareVideoCodecSelect()		Подготовить опции выбора кодека
createRoom()		Создать комнату
roomCreated(info: PublicRoomInfo)		Обрабатываем событие создания комнаты
roomDeleted(roomId: string)		Обрабатываем событие удаления комнаты
roomNameChanged(info: NewRoomNameInfo)		Обрабатываем событие изменения названия комнаты
deleteRoom(roomId: string)		Удалить комнату
clearRoomChat(roomId: string)		Очистить историю чата
deleteRoomFiles(roomId: string)		Удалить все файлы комнаты

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
										54
					Изм	Лист	№ документа	Подпись	Дата	

Продолжение таблицы 2.12

Метод	Описание
setInitialRoomList(roomList: PublicRoomInfo[])	Обрабатываем событие получения начального списка комнат
subscribeUserList(roomId: string)	Сообщаем серверу, что хотим получать список пользователей этой комнаты
unsubscribeUserList(roomId: string)	Сообщаем серверу, что больше не хотим получать список юзеров этой комнаты
setUserList(userList: UserInfo[])	Присваиваем список пользователей
addRoomListItem(info: PublicRoomInfo)	Добавить комнату в список комнат
removeRoomListItem(roomId: string)	Удалить комнату из списка комнат
updateRoomListItem(info: PublicRoomInfo)	Обновить информацию о комнате в списке комнат
kickUser(info: ActionOnUserInfo)	Исключить выбранного пользователя из комнаты
stopUserDisplay(info: ActionOnUserInfo)	Прекратить захват экрана у пользователя
stopUserCam(info: ActionOnUserInfo)	Прекратить захват видеоустройств у пользователя
stopUserAudio(info: ActionOnUserInfo)	Прекратить захват аудио у пользователя
changeUsername(info: ChangeUserNameInfo)	Изменить имя пользователя
banUser(info: ActionOnUserInfo)	Заблокировать пользователя комнаты userId на всём сервере
banUserByIp(userIp: string)	Заблокировать пользователя по IP
unbanUserByIp(userIp: string)	Разблокировать пользователя по IP
changeRoomName(info: NewRoomNameInfo)	Изменить название комнаты
changeRoomPass(info: NewRoomPassInfo)	Изменить пароль комнаты
checkIsSelectOptionCorrect(id: string): boolean	Проверка правильности выбранной опции
setSaveRoomChatPolicy(info: NewRoomSaveChatPolicyInfo)	Установить параметр сохранения истории чата на сервере

Серверная часть программы состоит из одного модуля «index.ts» и 17 классов: WebService – для работы с фреймворком Express, TokenService – для работы с JWT-токенами, UserBanRepository – для хранения списка блокировок пользователей, UserAccountRepository – для хранения аккаунтов пользователей, классы SocketManager, AdminSocketService, GeneralSocketService и RoomSocketService – для

Инов. № подл.	Подп. и дата	Взам. инв. №	Инов. № дубл.	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Лист
55

работы с серверными веб-сокетами, MediasoupService – для работы с библиотекой mediasoup, Room, RoomRepository, AuthRoomUserRepository и RoomChatRepository – для реализации комнаты на сервере, ActiveUser – для описания пользователя в комнате и классы FileService, FileRepository, TusHeadResponse, TusPatchResponse, TusOptionsResponse, TusPostCreationResponse, GetResponse и FileInfo для реализации загрузки и скачивания файлов на сервере.

В таблице 2.13 приведены основные переменные программы «index.ts».

Таблица 2.13 – Основные переменные модуля «index.ts»

Поле	Тип	Назначение
mediasoupService	MediasoupService	Сервис для работы с медиапотоками
userAccountRepository	UserAccountRepository	Репозиторий аккаунтов пользователей
fileRepository	FileRepository	Репозиторий файлов
roomChatRepository	RoomChatRepository	Репозиторий для истории чатов комнат
roomRepository	RoomRepository	Репозиторий комнат
authRoomUserRepository	AuthRoomUserRepository	Репозиторий для записей авторизации пользователей в комнатах
fileService	FileService	Сервис для загрузки и скачивания файлов
userBanRepository	UserBanRepository	Репозиторий блокировок пользователей
tokenService	TokenService	Сервис для работы с токенами
express	WebService	Express веб-сервис
httpServer	http.Server	Объект http веб-сервера
httpPort	number	Порт веб-сервера http
httpsOptions	https.ServerOptions	Конфигурация https сервера, а именно SSL ключ и сертификат
httpsServer	https.Server	Объект https веб-сервера из модуля Node.JS «https»
port	number	Порт веб-сервера https
socketManager	SocketManager	Для работы с серверными веб-сокетами

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

					ВКРБ 09.03.04.059.2022	Лист
						56
Изм	Лист	№ документа	Подпись	Дата		

В таблице 2.14 приведено описание класса `WebService`, организующего работу с фреймворком Express и реализующий веб-сервер (файл `WebService.ts`).

Таблица 2.14 – Описание класса WebService

Поле	Тип	Назначение
app	express.Express	Приложение Express
fileService	FileService	Обработчик файлов
tokenService	TokenService	Сервис для работы с токенами
roomRepository	RoomRepository	Комнаты
userAccountRepository	UserAccountRepository	Аккаунты пользователей
userBanRepository	UserBanRepository	Блокировки пользователей
authRoomUserRepository	AuthRoomUserRepository	Авторизованные пользователи в комнатах
Метод		Описание
constructor(fileService: FileService, tokenService: TokenService, roomRepository: RoomRepository, userAccountRepository: UserAccountRepository, userBanRepository: UserBanRepository, authRoomUserRepository: AuthRoomUserRepository)		Конструктор класса
checkBanMiddleware(req: express.Request, res: express.Response, next: express.NextFunction)		Проверить на наличие блокировки по ip-адресу пользователя
wwwMiddleware(req: express.Request, res: express.Response, next: express.NextFunction)		Убрать www из адреса
httpsMiddleware(req: express.Request, res: express.Response, next: express.NextFunction)		Перенаправить на https
requestHasNotBody(req: express.Request): boolean		Есть ли тело у запроса?
sendCodeAndDestroySocket(req: express.Request, res: express.Response, httpCode: number)		Отправить код ошибки и разорвать соединение с клиентом
rejectRequestWithBodyMiddleware(req: express.Request, res: express.Response, next: express.NextFunction)		Отвергнуть запросы GET, HEAD и OPTIONS с телом в запросе
preventFloodMiddleware(req: express.Request, res: express.Response, next: express.NextFunction)		Защитить от флуд-атаки через body в реквесте

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

Продолжение таблицы 2.14

Метод	Описание
handleRoutes()	Обработать маршруты
async createAuthToken(res: express.Response, role: string): Promise<string>	Создать для клиента JWT-токен и прикрепить его к ответу в виде httpOnly cookie.
async roomRoute(req: express.Request, res: express.Response, next: express.NextFunction)	Маршруты для комнаты
async adminRoute(req: express.Request, res: express.Response, next: express.NextFunction)	Маршруты для администратора
handleFilesRoutes()	Обрабатываем маршруты, связанные с файлами
handleStatic()	Открыть доступ к статическим файлам
endPoint()	Самый последний обработчик запросов

В таблице 2.15 приведено описание класса TokenService, организующего работу с JWT-токенами (файл TokenService.ts).

Таблица 2.15 – Описание класса TokenService

Поле	Тип	Назначение
secret	Crypto.KeyObject	Секретная фраза для JWT-токенов
userAccountRepository	UserAccountRepository	Аккаунты пользователей
Метод	Описание	
constructor(userAccountRepository: UserAccountRepository)	Конструктор класса	
async tokenExpressMiddleware(req: express.Request, res: express.Response, next: express.NextFunction)	Middleware для Express	
async tokenSocketMiddleware(socket: Socket, next: SocketNextFunction)	Middleware для Socket.IO	
async create(data: TokenData, exp: number): Promise<string>	Создать токен	
async verify(jwt: string): Promise<string undefined>	Проверить токен	

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм.	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					58

В таблице 2.16 приведено описание класса SocketManager, создающий экземпляры классов, организующих работу с веб-сокетами (файл SocketManager.ts).

Таблица 2.16 – Описание класса SocketManager

Поле	Тип	Назначение
io	SocketIO.Server	Объект сервера веб-сокеты
generalSocketService	GeneralSocketService	Обработчик общих веб-сокеты событий
adminSocketService	AdminSocketService	Обработчик админских веб-сокеты событий
roomSocketService	RoomSocketService	Обработчик веб-сокеты событий, связанных с комнатой
userBanRepository	UserBanRepository	Список заблокированных пользователей
Метод		Описание
constructor(server: https.Server, tokenMiddleware: TokenSocketMiddleware, fileRepository: FileRepository, mediasoupService: MediasoupService, roomRepository: RoomRepository, userAccountRepository: UserAccountRepository, userBanRepository: UserBanRepository, authRoomUserRepository: AuthRoomUserRepository, roomChatRepository: RoomChatRepository)		Конструктор класса
createSocketServer(server: https.Server): SocketIO.Server		Создать Socket.IO сервер
applyCheckBanMiddleware(socket, next)		Применить Middleware для проверки заблокированных пользователей

В таблице 2.17 приведено описание класса GeneralSocketService, обрабатывающего общие веб-сокеты события (файл GeneralSocketService.ts).

Таблица 2.17 – Описание класса GeneralSocketService

Поле	Тип	Назначение
generalIo	SocketIO.Namespace	Объект пространства имен веб-сокеты
roomRepository	RoomRepository	Менеджер комнат

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист
											59

Продолжение таблицы 2.17

Метод	Описание
constructor(generalIo: SocketIO.Namespace, roomRepository: RoomRepository)	Конструктор класса
clientConnected()	Обработка события – «клиент подключился»
notifyAboutCreatedRoom(info: RoomLinkInfo)	Оповестить всех о созданной комнате
notifyAboutDeletedRoom(id: string)	Оповестить всех об удаленной комнате
sendUserListToAllSubscribers(roomId: string)	Сообщить новый список пользователей подписчикам этой комнаты
sendUserListToSubscriber(subscriberId: string, roomId: string)	Сообщить новый список пользователей комнаты определенному подписчику этого события

В таблице 2.18 приведено описание класса RoomSocketService, обрабатывающего веб-сокеты события, связанных с комнатой (файл RoomSocketService.ts).

Таблица 2.18 – Описание класса RoomSocketService

Поле	Тип	Назначение
roomIo	SocketIO.Namespace	Объект пространства имен веб-сокетов
roomRepository	RoomRepository	Комнаты
roomChatRepository	RoomChatRepository	Чаты комнат
userAccountRepository	UserAccountRepository	Аккаунты пользователей
userBanRepository	UserBanRepository	Заблокированные пользователи
authRoomUserRepository	AuthRoomUserRepository	Авторизованные пользователи
generalSocketService	GeneralSocketService	Сервис для обработки общих событий веб-сокетов
fileRepository	FileRepository	Репозиторий файлов
mediasoupService	MediasoupService	Сервис для работы с медиапотоками
Метод		Описание
clientConnected()		Клиент подключился
async clientJoined(room: Room, socket: Socket, userId: string)		Пользователь заходит в комнату

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм.	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					60

Продолжение таблицы 2.18

Метод	Описание
constructor(roomIo: SocketIO.Namespace, generalSocketService: GeneralSocketService, tokenMiddleware: TokenSocketMiddleware, fileRepository: FileRepository, mediasoupService: MediasoupService, roomRepository: RoomRepository, userAccountRepository: UserAccountRepository, userBanRepository: UserBanRepository, authRoomUserRepository: AuthRoomUserRepository, roomChatRepository: RoomChatRepository)	Конструктор класса
async requestCreateWebRtcTransport(room: Room, socket: Socket, user: ActiveUser, consuming: boolean)	Запросить создание транспортного канала по запросу клиента
async userReady(room: Room, socket: Socket, user: ActiveUser, info: UserReadyInfo)	Запросить потоки других пользователей для нового пользователя и оповестить всех о новом пользователе
async requestCreateConsumer(socket: Socket, room: Room, consumerUser: ActiveUser, producerUserId: string, producer: MediasoupTypes.Producer)	Запросить создание потока-потребителя для пользователя consumerUser из потока-производителя пользователя producerUserId
handleConsumerEvents(socket: Socket, room: Room, consumer: MediasoupTypes.Consumer, consumerUser: ActiveUser)	Обработка событий у потока-потребителя
async requestCreateProducer(socket: Socket, room: Room, user: ActiveUser, newProducerInfo: NewProducerInfo)	Запросить создание потока-производителя для пользователя user
handleProducerEvents(socket: Socket, room: Room, user: ActiveUser, producer: MediasoupTypes.Producer)	Обработка событий у потока-производителя

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата	Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист
											61

Продолжение таблицы 2.18

Метод	Описание
getSocketBySocketId(id: string): Socket undefined	Получить веб-сокет соединение по Id
async userChangedName(roomId: string, socket: Socket, userId: string, username: string)	Пользователь изменил ник
async userSentChatMsg(roomId: string, userId: string, msg: string)	Пользователь отправил сообщение в чат
async userSentChatFile(userId: string, roomId: string, fileId: string)	Пользователь отправил ссылку на файл в чат
userDisconnected(room: Room, socket: Socket, userId: string, reason: string)	Пользователь отключился
emitMaxVideoBitrate(newMaxVideoBitrate: number)	Разослать клиентам во всех комнатах новое значение максимального битрейта для видеопотоков
kickUser(info: ActionOnUserInfo)	Исключить пользователя из комнаты
kickAllUsers(roomId: string)	Исключить всех пользователей из комнаты
stopUserDisplay(info: ActionOnUserInfo)	Сообщить клиенту пользователя, о том, что необходимо прекратить захват экрана
stopUserCam(info: ActionOnUserInfo)	Сообщить клиенту пользователя, о том, что необходимо прекратить захват видеоустройств
stopUserAudio(info: ActionOnUserInfo)	Сообщить клиенту пользователя, о том, что необходимо прекратить захват аудиодорожки
changeUsername(info: ChangeUserNameInfo)	Изменить имя пользователя
async banUser(info: ActionOnUserInfo)	Заблокировать пользователя userId, находящегося в комнате, на сервере
async sendChatHistory(socket: Socket, roomId: string)	Отправить пользователю историю чата

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

					ВКРБ 09.03.04.059.2022	Лист
						62
Изм	Лист	№ документа	Подпись	Дата		

В таблице 2.19 приведено описание класса AdminSocketService, обрабатывающего веб-сокеты события, связанные с функциями администратора (файл AdminSocketService.ts).

Таблица 2.19 – Описание класса AdminSocketService

Поле	Тип	Назначение
adminIo	SocketIO.Namespace	Объект пространства имен веб-сокетов
generalSocketService	GeneralSocketService	Обработчик общих веб-сокет событий
roomSocketService	RoomSocketService	Обработчик веб-сокет событий, связанных с комнатой
roomRepository	RoomRepository	Комнаты
authRoomUserRepository	AuthRoomUserRepository	Авторизованные пользователи
userBanRepository	UserBanRepository	Заблокированные пользователи
userAccountRepository	UserAccountRepository	Аккаунты пользователей
roomChatRepository	RoomChatRepository	Чаты комнат
fileRepository	FileRepository	Репозиторий файлов
Метод		Описание
constructor(adminIo: SocketIO.Namespace, tokenMiddleware: TokenSocketMiddleware, generalSocketService: GeneralSocketService, roomSocketService: RoomSocketService, roomRepository: RoomRepository, userBanRepository: UserBanRepository, authRoomUserRepository: AuthRoomUserRepository, userAccountRepository: UserAccountRepository, roomChatRepository: RoomChatRepository, fileRepository: FileRepository)		Конструктор класса
checkIp()		Проверить IP-адрес
clientConnected()		Клиент подключился
async createRoom(info: NewRoomInfo)		Создать комнату
async deleteRoom(roomId: string)		Удалить комнату
async kickUser(info: ActionOnUserInfo)		Исключить пользователя
async changeRoomName(info: NewRoomNameInfo)		Изменить название комнаты
async changeRoomPass(info: NewRoomPassInfo)		Изменить пароль комнаты

Инов. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

В таблице 2.20 приведено описание класса RoomRepository, являющимся менеджером комнат – экземпляров класса Room (файл RoomRepository.ts).

Таблица 2.20 – Описание класса RoomRepository

Поле	Тип	Назначение
rooms	Map<string, Room>	Комнаты
mediasoup	MediasoupService	Сервис для работы с медиапотоками
userAccountRepository	UserAccountRepository	Аккаунты пользователей
Метод		Описание
constructor(mediasoup: MediasoupService, userAccountRepository: UserAccountRepository)		Конструктор класса
async writeToFile()		Полностью обновить содержимое файла с записями о комнатах
async create(info: NewRoomInfo): Promise<string>		Создать комнату
async remove(id: string): Promise<void>		Удалить комнату
async update(info: UpdateRoomInfo): Promise<void>		Изменить информацию о комнате
get(id: string): Room undefined		Получить комнату
has(id: string): boolean		Есть ли такая комната?
getRoomLinkList(): PublicRoomInfo[]		Получить список ссылок на комнаты
getActiveUserList(roomId: string): UserInfo[]		Получить список пользователей в комнате roomId
getActiveUserSocketId(roomId: string, userId: string): string		Получить socketId у активного пользователя userId в комнате roomId
async checkPassword(id: string, pass: string): Promise<boolean>		Проверить правильность пароля от комнаты
isEmptyPassword(id: string): boolean		Проверить, есть ли пароль у комнаты
getSaveChatPolicy(id: string): boolean		Проверить, нужно ли сохранять сообщения в историю чата комнаты
setSaveChatPolicy(id: string): boolean		Изменить параметр сохранения истории чата

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	Promise<void>	
					get(id: string): Room undefined	Получить комнату
					has(id: string): boolean	Есть ли такая комната?
					getRoomLinkList(): PublicRoomInfo[]	Получить список ссылок на комнаты
					getActiveUserList(roomId: string): UserInfo[]	Получить список пользователей в комнате roomId
					getActiveUserSocketId(roomId: string, userId: string): string	Получить socketId у активного пользователя userId в комнате roomId
					async checkPassword(id: string, pass: string): Promise<boolean>	Проверить правильность пароля от комнаты
					isEmptyPassword(id: string): boolean	Проверить, есть ли пароль у комнаты
					getSaveChatPolicy(id: string): boolean	Проверить, нужно ли сохранять сообщения в историю чата комнаты
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	setSaveChatPolicy(id: string): boolean	Изменить параметр сохранения истории чата
ВКРБ 09.03.04.059.2022						Лист
						64

Изм	Лист	№ документа	Подпись	Дата

В таблице 2.21 приведено описание класса Room, представляющего из себя комнату на сервере (файл Room.ts).

Таблица 2.21 – Описание класса Room

Поле	Тип	Назначение
id	string	Идентификатор комнаты
name	string	Название комнаты
password	string	Пароль комнаты
saveChatPolicy	boolean	Сохранять историю чатов?
videoCodec	VideoCodec	Видеокодек комнаты
mediasoup	MediasoupService	Сервис для работы с медиапотоками
mediasoupRouters	MediasoupTypes.Router[]	Массив роутеров (каждый роутер на своём ядре процессора)
latestRouterIdx	number	Индекс последнего задействованного роутера
activeUsers	Map<string, ActiveUser>	Список активных пользователей в комнате
Метод		Описание
static async create(info: RoomInfo, mediasoup: MediasoupService): Promise<Room>		Асинхронно создать комнату
constructor(info: RoomInfo, mediasoup: MediasoupService, mediasoupRouters: MediasoupTypes.Router[])		Конструктор класса
routerRtpCapabilities(): MediasoupTypes.RtpCapabilities		Получить RTP возможности (кодеки) роутера
getRouter(consuming: boolean): MediasoupTypes.Router		Получить очередной роутер для создания транспортного канала
async createWebRtcTransport(user: ActiveUser, consuming: boolean): Promise<MediasoupTypes.WebRtcTransport>		Создать транспортный канал по запросу клиента
transportClosed(user: ActiveUser, consuming: boolean)		Транспортный канал был закрыт, поэтому необходимо обработать это событие
async connectWebRtcTransport(user: ActiveUser, info: ConnectWebRtcTransportInfo)		Подключиться к транспортному каналу по запросу клиента

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	Инв. № подл.

Изм	Лист	№ документа	Подпись	Дата
-----	------	-------------	---------	------

ВКРБ 09.03.04.059.2022

Продолжение таблицы 2.21

Метод	Описание
async createConsumer(consumerUser: ActiveUser, producer: MediasoupTypes.Producer): Promise<MediasoupTypes.Consumer>	Создание потока-потребителя для пользователя consumerUser из потока-производителя пользователя producerUserId.
consumerClosed(consumer: MediasoupTypes.Consumer, consumerUser : ActiveUser)	Поток-потребитель был завершен, поэтому необходимо обработать это событие
async userRequestedPauseConsumer(user: ActiveUser, consumerId: string): Promise<boolean>	Пользователь user запросил поставить на паузу поток-потребитель с идентификатором consumerId
async pauseConsumer(consumer: MediasoupTypes.Consumer): Promise<boolean>	Поставить consumer на паузу
async userRequestedResumeConsumer(user: ActiveUser, consumerId: string): Promise<boolean>	Пользователь user запросил снять с паузы поток-потребитель с идентификатором consumerId
async resumeConsumer(consumer: MediasoupTypes.Consumer): Promise<boolean>	Снять consumer с паузы
async createProducer(user: ActiveUser, newProducerInfo: NewProducerInfo): Promise<MediasoupTypes.Producer>	Создать поток-производитель, который описан с помощью newProducerInfo, для пользователя user
producerClosed(producer: MediasoupTypes.Producer, user: ActiveUser)	Поток-потребитель был завершен, поэтому необходимо обработать это событие
userRequestedCloseProducer(user: ActiveUser, producerId: string)	Пользователь user запросил закрыть поток-производитель с идентификатором producerId
async userRequestedPauseProducer(user: ActiveUser, producerId: string): Promise<boolean>	Пользователь user запросил поставить на паузу поток-производитель с идентификатором producerId
async pauseProducer(producer: MediasoupTypes.Producer): Promise<boolean>	Поставить на паузу поток-производитель producer

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

Продолжение таблицы 2.21

Метод	Описание
async userRequestedResumeProducer(user: ActiveUser, producerId: string): Promise<boolean>	Пользователь user запросил снять с паузы поток-производитель с идентификатором producerId
async resumeProducer(producer: MediasoupTypes.Producer): Promise<boolean>	Снять с паузы поток-производитель producer
userDisconnected(userId: string)	Пользователь отключился из комнаты
close()	Заккрыть комнату

В таблице 2.22 приведено описание класса ActiveUser, описывающего пользователя в комнате (файл Room.ts).

Таблица 2.22 – Описание класса ActiveUser

Поле	Тип	Назначение
userId	string	Идентификатор пользователя в комнате
socketId	string	Идентификатор сокета пользователя
rtpCapabilities?	MediasoupTypes.RtpCapabilities	RTP возможности пользователя (поддерживаемые кодеки)
consumerTransport?	MediasoupTypes.WebRtcTransport	Транспортный канал пользователя для потребляющих медиапотоков
producerTransport?	MediasoupTypes.WebRtcTransport	Транспортный канал пользователя для производящих медиапотоков
producers	Map<string, MediasoupTypes.Producer>	Контейнер с медиапотоками-производителями
consumers	Map<string, MediasoupTypes.Consumer>	Контейнер с медиапотоками-потребителями
Метод		Описание
constructor(roomId: string, userId: string, socketId: string)		Конструктор класса
getTransportById(transportId: string): MediasoupTypes.WebRtcTransport undefined		Вернуть транспортный канал по идентификатору

Интв. № подл.	Подп. и дата	Взам. интв. №	Интв. № дубл	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Лист

67

В таблице 2.23 приведено описание класса MediasoupService, организующего работу с библиотекой mediasoup (файл MediasoupService.ts).

Таблица 2.23 – Описание класса MediasoupService

Поле	Тип	Назначение
mediasoupWorkers	MediasoupTypes.Worker[]	Экземпляры класса mediasoup.Worker – «обертка» над C++ подпроцессами, которые обрабатывают медиапотoki
networkIncomingCapability	number	Входящая скорость сервера (в мегабитах Mbit)
networkOutcomingCapability	number	Исходящая скорость сервера (в мегабитах Mbit)
maxAudioBitrate	number	Максимальный битрейт (Кбит) для аудиопотоков на сервере
maxVideoBitrate	number	Максимальный битрейт (Кбит) для видеопотоков на сервере.
videoConsumersCount	number	Количество видеопотоков-потребителей
audioConsumersCount	number	Количество аудиопотоков-потребителей
videoProducersCount	number	Количество видеопотоков-производителей
audioProducersCount	number	Количество аудиопотоков-производителей
audioCodecConf	MediasoupTypes.RtpCodecCapability	Объект-настройка для аудикодека Opus
videoCodecVp9Conf	MediasoupTypes.RtpCodecCapability	Объект-настройка для видеокодека VP9
videoCodecVp8Conf	MediasoupTypes.RtpCodecCapability	Объект-настройка для видеокодека VP8
videoCodecsH264Conf	MediasoupTypes.RtpCodecCapability[]	Объект-настройка для видеокодека H264
Метод		Описание
constructor(workers: Array<MediasoupTypes.Worker>) Приватный конструктор для класса		constructor(workers: Array<MediasoupTypes.Worker>) Приватный конструктор для класса

Продолжение таблицы 2.23

Метод	Описание
static async create(numWorkers: number): Promise<Mediasoup>	Функция, возвращающая экземпляр класса Mediasoup
async createRouters(codecChoice: VideoCodec): Promise<MediasoupTypes.Router[]>	Функция, создающая маршрутизаторы медиапоток для комнаты
async createWebRtcTransport(user: User, consuming: boolean, router: MediasoupTypes.Router): Promise<MediasoupTypes.WebRtcTransport>	Функция, создающая транспортный канал WebRTC, который необходим для передачи медиапотока от клиента на сервер и наоборот
createProducer(user: User, newProducerInfo: NewProducerInfo, routers: MediasoupTypes.Router[]): Promise<MediasoupTypes.Producer>	Функция, создающая обработчика исходящего от клиента медиапотока (производитель)
createConsumer(user: User, producer: MediasoupTypes.Producer, router: MediasoupTypes.Router): Promise<MediasoupTypes.Consumer>	Функция, создающая экземпляр класса обработчика входящего для клиента медиапотока (потребитель)
calculateNewMaxVideoBitrate()	Рассчитать новое значение максимального битрейта для видеопотоков исходя из количества аудио- и видеопотоков, обрабатываемых сервером

В таблице 2.24 приведено описание класса RoomChatRepository, реализующего работу с историями чатов комнат (файл RoomChatRepository.ts).

Таблица 2.24 – Описание класса RoomChatRepository

Поле	Тип	Назначение
CHATS_FOLDER_PATH	string	Путь до папки с историями чатов комнат
Метод	Описание	
constructor()	Конструктор класса	
async addMessage(roomId: string, msg: ChatMessage)	Добавить сообщение в историю чата для roomId	

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	MediasoupTypes.Router):	клиента медиапотока (потребитель)
					Promise<MediasoupTypes.Consumer> calculateNewMaxVideoBitrate()	Рассчитать новое значение максимального битрейта для видеопотоков исходя из количества аудио- и видеопотоков, обрабатываемых сервером

В таблице 2.24 приведено описание класса RoomChatRepository, реализующего работу с историями чатов комнат (файл RoomChatRepository.ts).

Таблица 2.24 – Описание класса RoomChatRepository

Поле	Тип	Назначение
CHATS_FOLDER_PATH	string	Путь до папки с историями чатов комнат
Метод		Описание
constructor()		Конструктор класса
async addMessage(roomId: string, msg: ChatMessage)		Добавить сообщение в историю чата для roomId

Изм.	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Лист
69

Продолжение таблицы 2.24

Метод	Описание
async removeAll(roomId: string)	Удалить историю чата для roomId
async getAll(roomId: string): Promise<ChatMessage[] undefined>	Получить список сообщений в чате комнаты roomId
has(roomId: string): boolean	Есть ли история чата для комнаты roomId?

В таблице 2.25 приведено описание класса UserAccountRepository, реализующего работу с аккаунтами пользователей (файл UserAccountRepository.ts).

Таблица 2.25 – Описание класса UserAccountRepository

Поле	Тип	Назначение
USERS_FILE_PATH	string	Путь до файла с аккаунтами пользователей
users	Map<string, UserAccount>	Аккаунты пользователей
Метод	Описание	
constructor()	Конструктор класса	
async writeToFile()	Полностью обновить содержимое файла с записями о пользователях	
async create(info: NewUserAccountInfo): Promise<string>	Создать запись об аккаунте пользователя	
async remove(userId: string)	Удалить запись об аккаунте пользователя	
get(userId: string): UserAccount undefined	Получить запись об аккаунте пользователя	
has(userId: string): boolean	Есть ли запись об этом аккаунте?	
async setUsername(userId: string, name: string)	Установить новое имя пользователя	
async setRole(userId: string, role: string)	Установить новую роль для пользователя	
getUsername(userId: string): string undefined	Получить имя пользователя	
isAdmin(userId: string): boolean	Является ли пользователь администратором?	
async getRole(userId: string)	Получить текущую роль пользователя в системе	

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
--------------	--------------	--------------	-------------	--------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

Поле	Тип	Назначение
AUTH_ROOM_USERS_FILE_PATH	string	Путь до файла со списком авторизованных пользователей
authRoomUsers	Map<string, Authorization Records>	Идентификаторы авторизованных пользователей в комнате
Метод		Описание
constructor()		Конструктор класса
async writeToFile()		Полностью обновить содержимое файла с Id пользователей, авторизованных в комнатах
async create(roomId: string, userId: string)		Создать запись об авторизации пользователя userId в комнате roomId
async remove(roomId: string, userId: string)		Удалить запись об авторизации пользователя userId в комнате roomId
async removeAll(roomId: string)		Удалить все записи об авторизациях пользователей в комнате roomId
get(roomId: string): string[] undefined		Получить список авторизованных пользователей в комнате roomId
has(roomId: string, userId: string): boolean		Есть ли запись об авторизованном пользователе userId в комнате roomId?

Таблица 2.27 – Описание класса UserBanRepository

Поле	Тип	Назначение
BANS_FILE_PATH	string	Путь до файла со списком заблокированных пользователей
bans	Map<string, UserBanInfo>	Заблокированные пользователи

Продолжение таблицы 2.27

Метод	Описание
constructor()	Конструктор класса
async writeToFile()	Полностью обновить содержимое файла с записями о блокировках пользователя
async create(info: UserBanInfo): Promise<string>	Создать запись о блокировке пользователя
async remove(ip: string)	Удалить запись о блокировке пользователя
get(ip: string): UserBanInfo undefined	Получить запись о блокировке пользователя по IP-адресу
has(ip: string): boolean	Проверка IP-адреса на блокировку

В таблице 2.28 приведено описание класса FileService – сервиса, реализующего работу с файлами (файл FileService.ts).

Таблица 2.28 – Описание класса FileService

Поле	Тип	Назначение
FILES_PATH	string	Путь до папки с файлами
fileRepository	FileRepository	Репозиторий метаданных файлов
authRoomUserRepository	AuthRoomUserRepository	Авторизованные пользователи
roomRepository	RoomRepository	Комнаты
Метод	Описание	
assignHeaders(fromTus: FileServiceResponse, toExpress: express.Response)	Присвоить HTTP-заголовки ответу Response	
sendStatus(res: express.Response, statusCode: number, statusMsg?: string)	Отправить HTTP код-ответа	
sendStatusWithFloodPrevent(conditionForPrevent: boolean, req: express.Request, res: express.Response, statusCode: number, statusMsg?: string)	Отправить HTTP код-ответа с учетом флудовой атаки	
tusHeadInfo(req: express.Request, res: express.Response)	Обработка запроса Head	
async tusPatchFile(req: express.Request, res: express.Response)	Обработка запроса Patch	

Инов. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Изм.	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Лист

72

Продолжение таблицы 2.28

Метод	Описание
tusOptionsInfo(req: express.Request, res: express.Response)	Обработка запроса Options
downloadFile(req: express.Request, res: express.Response)	Обработка запроса Get (скачивание файла клиентом)
async tusPostCreateFile(req: express.Request, res: express.Response)	Обработка запроса Post

В таблице 2.29 приведено описание класса FileRepository, реализующего работу с метаданными файлов (файл FileRepository.ts).

Таблица 2.29 – Описание класса FileRepository

Поле	Тип	Назначение
INFO FILE PATH	string	Путь до файла с метаданными файлов
FILES PATH	string	Путь до папки с файлами
files	Map<FileId, FileInfo>	Метаданные файлов
Метод		Описание
constructor()		Конструктор класса
async writeToFile()		Обновить файл с метаданными файлов
async create(newFileInfo: NewFileInfo): Promise<string>		Создать запись о файле
async update(info: FileInfo)		Изменить информацию о файле
async remove(id: string)		Удалить запись о файле
async removeByRoom(roomId: string)		Удалить все записи о файлах, относящихся к комнате roomId
async writeFile(inStream: Readable, fileId: string, oldBytesWritten: number): Promise<number>		Запись потока в файл на сервере
async removeFile(fileId: string)		Удалить файл с сервера
get(id: string): FileInfo undefined		Получить запись о файле
has(id: string): boolean		Проверка наличия записи об файле

В таблице 2.30 приведено описание класса FileInfo, описывающего файл на сервере (метаданные) (файл FileRepository.ts).

Таблица 2.30 – Описание класса FileInfo

Поле	Тип	Назначение
id	string	Идентификатор файла
name	string	Оригинальное название файла

Инв. № подл.	Подп. и дата
Взам. инв. №	Инв. № дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ документа	Подпись	Дата
------	------	-------------	---------	------

ВКРБ 09.03.04.059.2022

Продолжение таблицы 2.30

Поле	Тип	Назначение
type	string	Тип файла
size	number	Размер файла в байтах
bytesWritten	number	Сколько байт уже было получено сервером
ownerId	string	Id аккаунта пользователя, загружающего файл
roomId	string	Id комнаты, в которой загружали файл
OriginalMetadata	string undefined	Оригинальные метаданные, которые указал пользователь

В таблице 2.31 приведено описание классов TusHeadResponse и TusPatchResponse, являющимися шаблонами для ответа на HEAD-запрос и PATCH-запрос соответственно для класса FileService (файл FileServiceTusProtocol.ts).

Таблица 2.31 – Описание классов TusHeadResponse и TusPatchResponse

Поле	Тип	Назначение
headers	OutgoingHttpHeaders	Заголовки ответа
statusCode	number	Код ответа
successful	boolean	Успешна ли была обработка запроса пользователя
Метод		Описание
constructor(req: express.Request, fileInfo: FileInfo undefined)		Конструктор для класса

В таблице 2.32 приведено описание класса `GetResponse`, являющимся шаблоном для ответа на GET-запрос для класса `FileService` (файл `FileServiceTusProtocol.ts`).

Таблица 2.32 – Описание класса `GetResponse`

Поле	Тип	Назначение
statusCode	number	Код ответа
statusMsg	string	Сообщение ответа
successful	boolean	Успешна ли была обработка запроса пользователя
Метод		Описание
constructor(req: express.Request, fileInfo: FileInfo undefined, filePath: string)		Конструктор для класса

В таблице 2.33 приведено описание класса TusPostCreationResponse, являющимся шаблоном для ответа на POST-запрос для класса FileService (файл FileServiceTusProtocol.ts).

Таблица 2.33 – Описание класса TusPostCreationResponse

Поле	Тип	Назначение
headers	OutgoingHttpHeaders	Заголовки ответа
statusCode	number	Код ответа
successful	boolean	Успешна ли была обработка запроса
fileInfo	FileInfo undefined	Данные о файле
Метод		Описание
constructor(req: express.Request, fileId: string, ownerId: string, roomId: string)		Конструктор класса
parseMetadata(metadata: string undefined): Map<string, string>		Распарсить метаданные

В таблице 2.34 приведено описание класса TusOptionsResponse, являющимся шаблоном для ответа на OPTIONS-запрос для класса FileService (файл FileServiceTusProtocol.ts).

Таблица 2.34 – Описание класса TusOptionsResponse

Поле	Тип	Назначение
headers	OutgoingHttpHeaders	Заголовки ответа
statusCode	number	Код ответа
successful	boolean	Успешна ли была обработка запроса

Серверная и клиентская часть программы используют общие структуры данных: интерфейсы HttpHeaders, IncomingHttpHeaders, OutgoingHttpHeaders, FileServiceRequest и FileServiceResponse для реализации возобновляемой загрузки файлов на сервер.

В таблице 2.35 приведено описание интерфейса HttpHeaders, являющимся HTTP-заголовками (файл FileServiceTypes.ts).

Таблица 2.35 – Описание класса HttpHeaders

Поле	Тип	Назначение
"Upload-Offset"	string undefined	Смещение в байтах, необходимое для загрузки файла
"Upload-Length"	string undefined	Длина загружаемого файла в байтах
"Tus-Resumable"	string undefined	Версия протокола TUS

Подп. и дата	Поле		Тип		Назначение	
	headers		OutgoingHttpHeaders		Заголовки ответа	
	statusCode		number		Код ответа	
	successful		boolean		Успешна ли была обработка запроса	

Серверная и клиентская часть программы используют общие структуры данных: интерфейсы `HttpHeaders`, `IncomingHttpHeaders`, `OutgoingHttpHeaders`, `FileServiceRequest` и `FileServiceResponse` для реализации возобновляемой загрузки файлов на сервер.

В таблице 2.35 приведено описание интерфейса `HttpHeaders`, являющимся HTTP-заголовками (файл `FileServiceTypes.ts`).

Таблица 2.35 – Описание класса `HttpHeaders`

Подп. и дата	Поле		Тип		Назначение	
	"Upload-Offset"		string undefined		Смещение в байтах, необходимое для загрузки файла	
	"Upload-Length"		string undefined		Длина загружаемого файла в байтах	
	"Tus-Resumable"		string undefined		Версия протокола TUS	

Инв. № подл.

Изм

Лист

№ документа

Подпись

Дата

ВКРБ 09.03.04.059.2022

Лист

75

Продолжение таблицы 2.35

Поле	Тип	Назначение
"Cache-Control"	string undefined	Механизм кеширования в запросах и ответах
"Upload-Metadata"	string undefined	Метаданные, в которых содержится оригинальное название и тип файла

В таблице 2.36 приведено описание интерфейса IncomingHttpHeaders, которое расширяет интерфейс HttpHeaders, поскольку является HTTP-заголовками для запроса (файл FileServiceTypes.ts).

Таблица 2.36 – Описание класса IncomingHttpHeaders

Поле	Тип	Назначение
"Content-Type"	string undefined	Тип передаваемого содержимого в теле запроса
"Room-Id"	string undefined	Номер комнаты, в которой загружается файл

В таблице 2.37 приведено описание интерфейса OutgoingHttpHeaders, которое расширяет интерфейс HttpHeaders, поскольку является HTTP-заголовками для ответа на запрос (файл FileServiceTypes.ts).

Таблица 2.37 – Описание класса OutgoingHttpHeaders

Поле	Тип	Назначение
"Tus-Version"	string undefined	Поддерживаемая сервером версия протокола TUS
"Tus-Extension"	string undefined	Список расширений протокола TUS
"Tus-Max-Size"	string undefined	Максимальный допустимый размер для файла

В таблице 2.38 приведено описание интерфейса FileServiceRequest, которое представляет собой базовый шаблон для запроса, используемого в процессе загрузки файла (файл FileServiceTypes.ts).

Таблица 2.38 – Описание класса FileServiceRequest

Поле	Тип	Назначение
method	string	Тип запроса
path	string	Путь запроса
headers	IncomingHttpHeaders undefined	Заголовки запроса

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022
					Лист 76

В таблице 2.39 приведено описание интерфейса FileServiceResponse, которое представляет собой базовый шаблон для ответа на запрос, используемого в процессе загрузки файла (файл FileServiceTypes.ts).

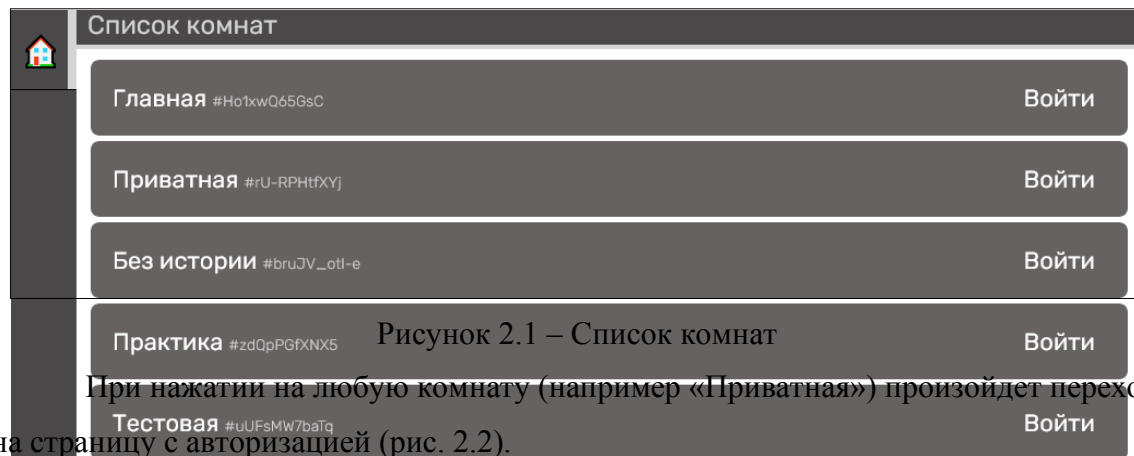
Таблица 2.39 – Описание класса FileServiceResponse

Поле	Тип	Назначение
headers	OutgoingHttpHeaders undefined	Заголовки ответа
statusCode	number	Код ответа

Также стоит отметить, что сохранение серверных данных осуществляется с применением паттерна «Репозиторий», что позволяет в дальнейшем при необходимости расширить данный функционал системы для сохранения информации в базу данных.

2.6 Разработанные меню и интерфейсы

После перехода на главную страницу программы по адресу, например: «https://localhost/», в окне браузера пользователя отобразится страница, содержащая список комнат (рис. 2.1).



При нажатии на любую комнату (например «Приватная») произойдет переход на страницу с авторизацией (рис. 2.2).

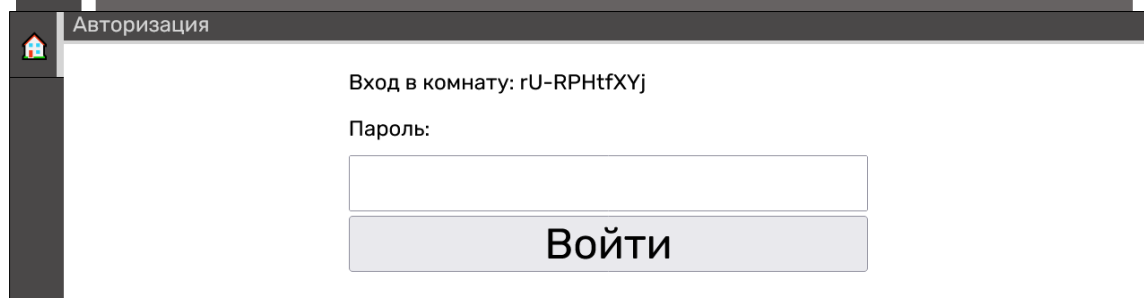


Рисунок 2.2 – Страница авторизации комнаты

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата	<p>Рисунок 2.1 – Список комнат</p> <p>При нажатии на любую комнату (например «Приватная») произойдет переход на страницу с авторизацией (рис. 2.2).</p> <p>Рисунок 2.2 – Страница авторизации комнаты</p>					Лист
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					77

Если пользователь ввел неправильный пароль, веб-приложение сообщит об этом выводом сообщения (рис. 2.3).

Рисунок 2.3 – Неправильный пароль

В случае, если пользователь ввел пароль верно, веб-сервер отправит клиенту страницу с выбранной комнатой (рис. 2.4).

Рисунок 2.4 – Вход в комнату «Приватная»

В верхней части окна находится поле для ввода имени, ряд кнопок: «Захватить микрофон», «Захватить веб-камеру», «Демонстрация экрана компьютера», «Включить звуки собеседников», первые три из них позволяют

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
										78
					Изм	Лист	№ документа	Подпись	Дата	

пользователю захватывать медиапотoki. Около кнопок захвата видеоизображения (веб-камера или экран) располагается виджет с выбором «Выберите разрешение». Также там располагается чекбокс для включения или выключения звуковых оповещений, а также два списка для выбора устройств, которые необходимо захватить.

В случае, если пользователь изменит имя и нажмет кнопку «Сохранить имя», имя пользователя изменится и сохранится на сервере. Также изменится и метка на видео (рис. 2.5).



Рисунок 2.5 – Смена имени пользователя

Назначение кнопок верхнего ряда становится ясно из их названий. Виджет с выбором «Выберите разрешение» позволяет выбрать разрешение изображения захватываемого видеоизображения (на выбор 640x360, 1280x720, 1920x1080 и так далее) (рис. 2.6).

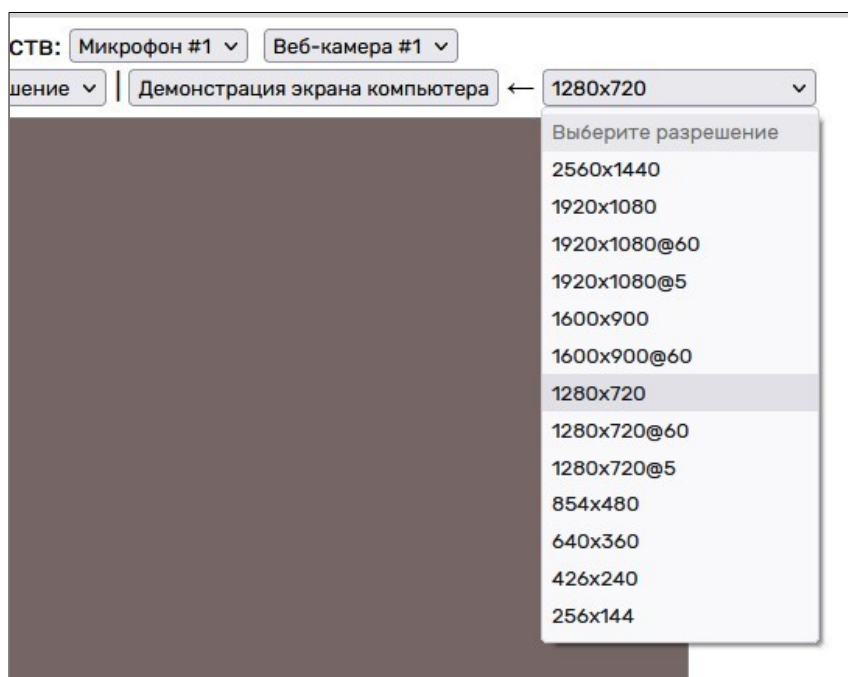


Рисунок 2.6 – Виджет для выбора разрешения захватываемого изображения

выбором «Выберите разрешение» позволяет выбрать разрешение изображения захватываемого видеоизображения (на выбор 640x360, 1280x720, 1920x1080 и так далее) (рис. 2.6).

The image shows a software window with a dark background. At the top, there are two buttons: 'Микрофон #1' and 'Веб-камера #1'. Below them is a label 'Демонстрация экрана компьютера' with a left-pointing arrow. To the right of this label is a dropdown menu currently set to '1280x720'. The dropdown menu is open, showing a list of resolution options: '2560x1440', '1920x1080', '1920x1080@60', '1920x1080@5', '1600x900', '1600x900@60', '1280x720' (which is highlighted), '1280x720@60', '1280x720@5', '854x480', '640x360', '426x240', and '256x144'. The main area of the window is a dark, blurry rectangle.

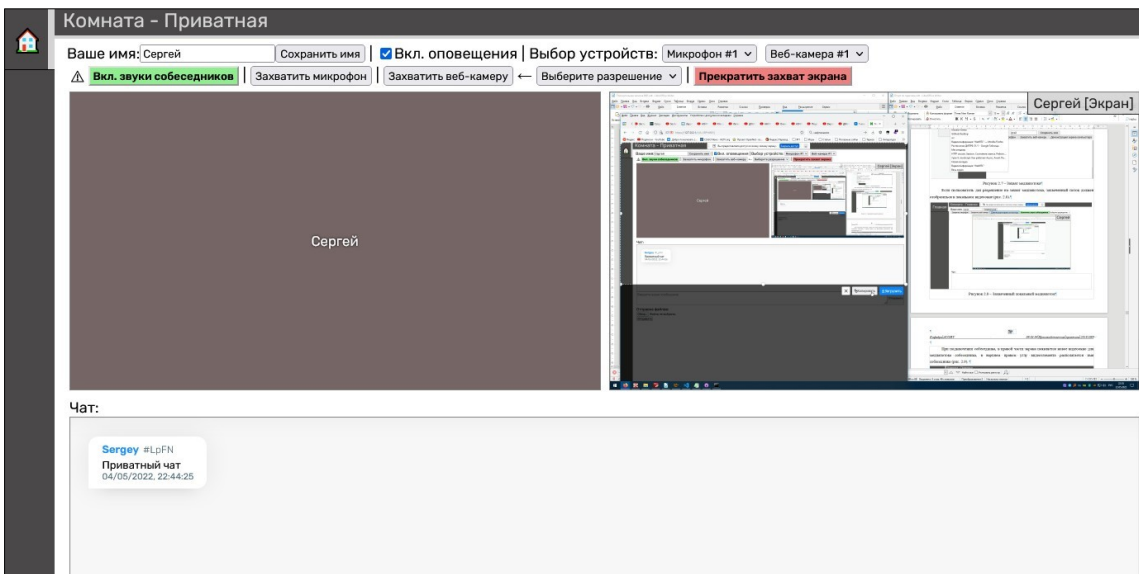
Рисунок 2.6 – Виджет для выбора разрешения захватываемого изображения

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Лист
79

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата



ВКРБ 09.03.04.059.2022

При подключении собеседника, в правой части экрана появляется новое видеоокно для медиапотока собеседника, в верхнем правом углу видеоэлемента располагается имя собеседника (рис. 2.9).

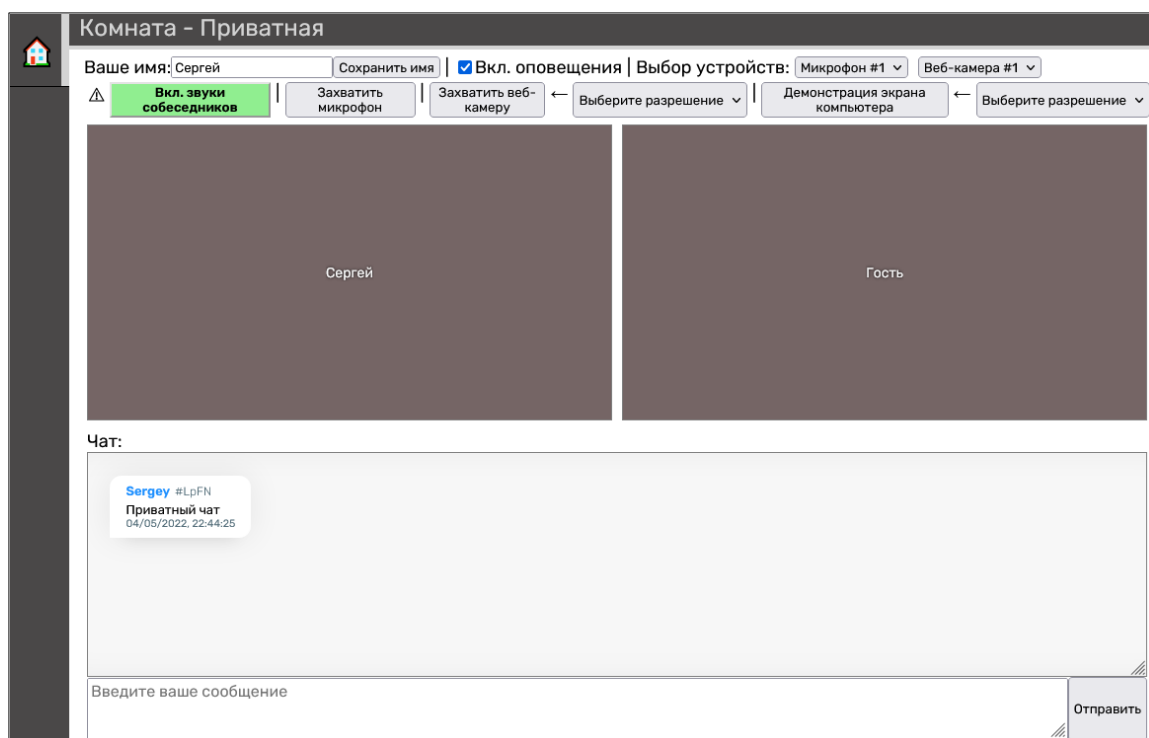


Рисунок 2.9 – Подключение собеседника с именем «Гость»

После того, как собеседник осуществит захват медиапотоков, изображение в видеофрейме изменится (рис. 2.10).

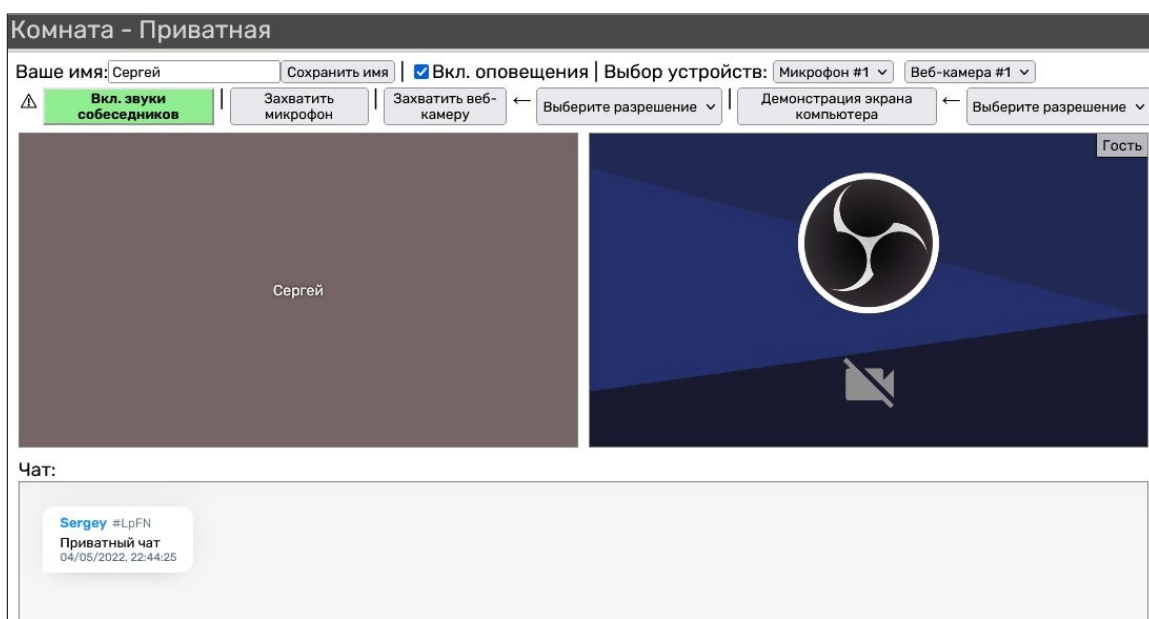


Рисунок 2.10 – Появление изображения в видеоокне собеседника

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист	
					Изм	Лист	№ документа	Подпись	Дата	81	

Чат:

Sergey #LpFN

Приватный чат

04/05/2022. 22:44:25

Гость #ku7w

ПРИВЕТ

22:23:41

Введите ваше сообщение

Отправить

В случае подключения нескольких собеседников, экран пользователя принимает вид (рис. 2.12). Пользователь «Сергей» может наблюдать на своем экране видеорежимы всех участников конференции, а именно пользователей «Гость» и «Bob».



В случае отключения собеседника, его видеоокно удаляется с экрана клиента (рис. 2.13).

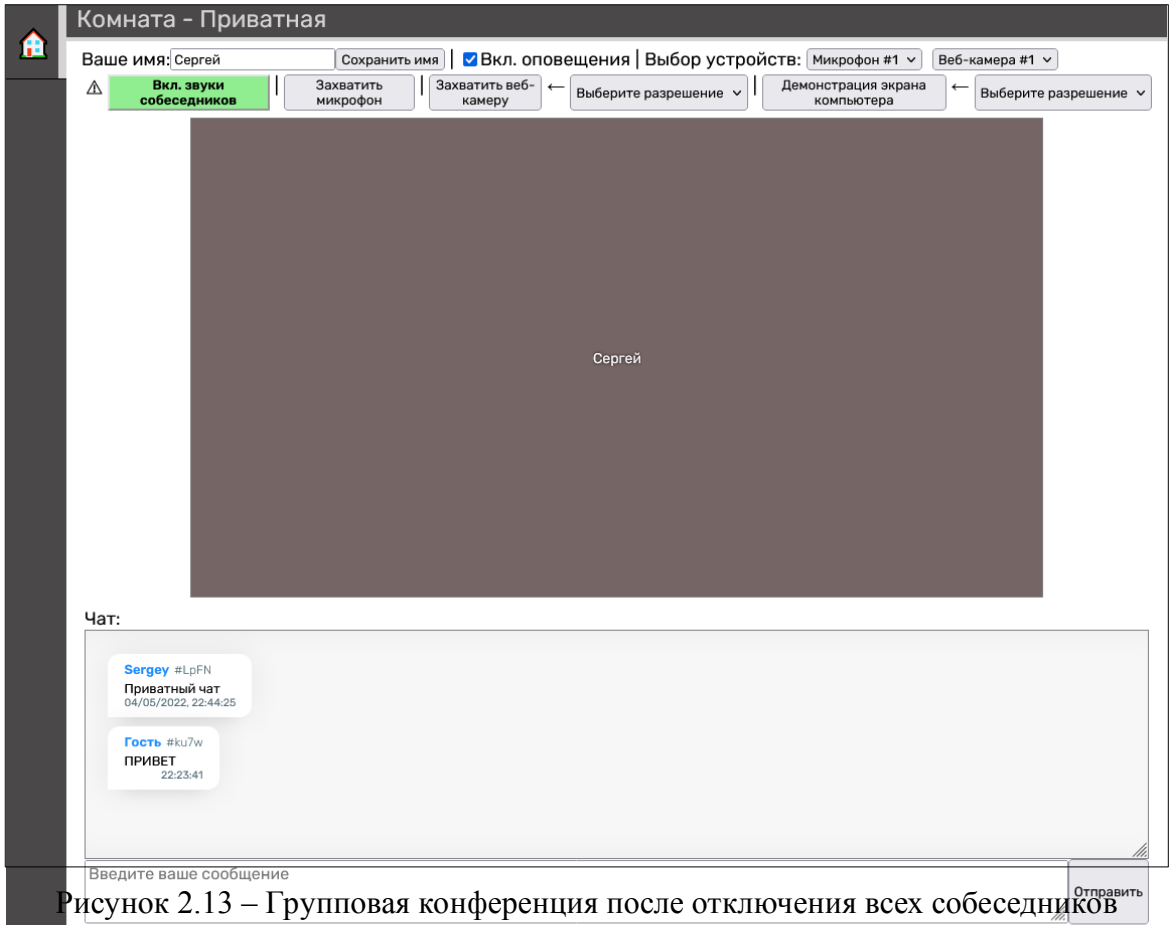


Рисунок 2.13 – Групповая конференция после отключения всех собеседников

Помимо виджета с чатом, в нижней части экрана присутствует поле для выбора загружаемого файла (рис. 2.14).

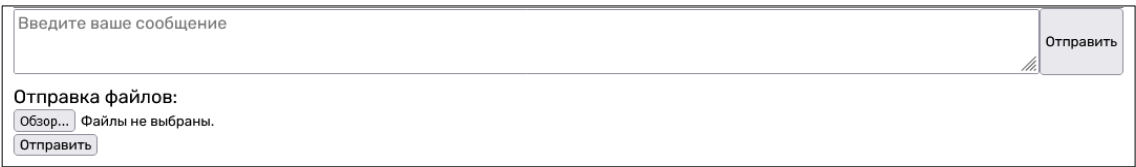


Рисунок 2.14 – Поле для загрузки файла

После выбора файла и нажатия на кнопку «Отправить», система начинает загрузку файла на сервер и отображает прогресс в виде прогресс-бара (рис. 2.15).

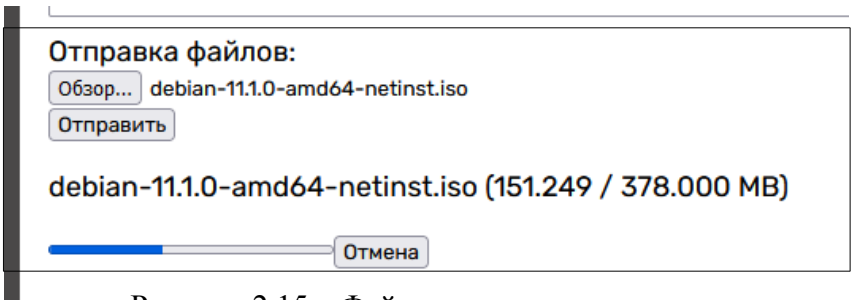


Рисунок 2.15 – Файл загружается на сервер

Инв. № подл.					ВКРБ 09.03.04.059.2022	Лист 83	
	Изм	Лист	№ документа	Подпись			Дата
Подп. и дата	Рисунок 2.15 – Файл загружается на сервер						
Взам. инв №	После выбора файла и нажатия на кнопку «Отправить», система начинает загрузку файла на сервер и отображает прогресс в виде прогресс-бара (рис. 2.15).						
Инв. № дубл	Рисунок 2.14 – Поле для загрузки файла						
Подп. и дата	Рисунок 2.13 – Групповая конференция после отключения всех собеседников						

Введите ваше сообщение

Отправить

Отправка файлов:

Обзор... Файлы не выбраны.

Отправить

Введите ваше сообщение

Отправить

Отправка файлов:

Обзор... Файлы не выбраны.

Отправить

Отправка файлов:

Обзор... debian-11.1.0-amd64-netinst.iso

Отправить

debian-11.1.0-amd64-netinst.iso (151.249 / 378.000 MB)

Отмена

Для перехода в режим администратора, необходимо перейти по адресу «https://localhost/admin» и авторизоваться (авторизация в панель администратора с точки зрения интерфейсов практически не отличается от авторизации в комнаты (см. рис. 2.2).

После авторизации, перед пользователем предстает панель администратора, где он может создавать и удалять комнаты, с помощью соответствующих кнопок «Создать комнату» и «Удалить комнату». Также он может редактировать комнату с помощью кнопок «Изменить название», «Изменить пароль», «Сохранять историю чата», «Не сохранять историю чата», «Очистить историю чата» и «Удалить все файлы комнаты».

Помимо этого, администратор может также исключить пользователя из комнаты, заблокировать пользователя, изменить имя пользователя или отключить исходящее от пользователя видео- или аудиодорожку, нажав соответствующие кнопки «Кикнуть пользователя», «Заблокировать пользователя», «Прекратить демонстрацию экрана пользователя», «Отключить веб-камеры пользователя», «Отключить аудио пользователя», «Изменить ник пользователя» (рис. 2.16).

The screenshot shows the 'Панель администратора' (Administrator Panel) with a title bar 'Менеджер комнат' (Room Manager). It is divided into three main sections:

- Создание комнат (Room Creation):** Includes fields for 'Название комнаты' (Room Name), 'Пароль комнаты' (Room Password), and 'Видеокод для комнаты' (Room Video Code) with a dropdown menu showing 'VP8'. There is a checkbox for 'Сохранять историю чата?' (Save chat history?) which is checked. A 'Создать комнату' (Create room) button is at the bottom.
- Выбор комнаты (Room Selection):** Features a 'Список комнат' (Room List) dropdown menu currently showing 'не выбрано...'. Below it are buttons for 'Новое название комнаты' (New room name), 'Изменить название' (Change name), 'Новое пароль комнаты' (New room password), 'Изменить пароль' (Change password), 'Сохранять историю чата' (Save chat history), 'Не сохранять историю чата' (Don't save chat history), 'Очистить историю чата' (Clear chat history), 'Удалить все файлы комнаты' (Delete all room files), and a red 'Удалить комнату' (Delete room) button.
- Выбор пользователя (User Selection):** Includes a 'Список пользователей' (User List) dropdown menu showing 'не выбрано...'. Buttons include 'Кикнуть пользователя' (Kick user), a red 'Заблокировать пользователя' (Block user) button, 'Прекратить демонстрацию экрана пользователя' (Stop user screen sharing), 'Отключить веб-камеры пользователя' (Turn off user webcam), 'Отключить аудио пользователя' (Turn off user audio), 'Новое имя пользователя' (New user name), 'Изменить ник пользователя' (Change user nickname), 'IP-адрес' (IP address) field, 'Заблокировать по IP' (Block by IP), and 'Разблокировать по IP' (Unblock by IP).

Рисунок 2.16 – Панель администратора

Также администратор может заблокировать или разблокировать пользователя, вручную введя IP-адрес и нажав на кнопки «Заблокировать по IP» или «Разблокировать по IP».

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист 84	
Изм.	Лист	№ документа	Подпись	Дата							

2.7 Сообщения системы

Клиентская часть программы сообщает об ошибках в консоли браузера. В таблице 2.40 приведены сообщения системы.

Таблица 2.40 – Сообщения системы

№ п\п	Сообщение	Причина возникновения, способ устранения
1	> [UserMedia] getUserMedia error:	Не удалось захватить микрофон или веб-камеру пользователя
2	> [UserMedia] getUserMedia success:	Захват микрофона или веб-камеры пользователя прошел успешно
3	> [UserMedia] getDisplayMedia success:	Захват экрана компьютера пользователя прошел успешно
4	> [UserMedia] getDisplayMedia error:	Не удалось захватить экран пользователя
5	Создано веб-сокеты подключение	Было создано подключение веб-сокета
6	Client ID:	Идентификатор созданного веб-сокета
7	[Room] > remoteUser disconnected:	Собеседник отключился от сервера
8	Вы были отсоединены от веб-сервера (websocket disconnect)	Возможно у клиента неполадки в сети или сервер был выключен
9	> [Mediasoup] Browser not supported	Браузер не поддерживает WebRTC
10	> [Mediasoup] createRecvTransport error	Ошибка при создании транспортного канала для исходящих медиапоток
11	> [Mediasoup] createSendTransport error	Ошибка при создании транспортного канала для входящих медиапоток
12	> [Mediasoup] consume error	Ошибка при обработке входящей медиаторожки
13	> [Mediasoup] Transport failed. Check your proxy settings	Не удалось соединиться с медиасервером. Необходимо проверить настройки прокси

В случае появления других сообщений следует обратиться к разработчику.

Инов. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

Лист

85

3 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

3.1 Проверка работоспособности системы в роли участника

1. Запустить веб-сервер. Перейти по адресу <https://127.0.0.1/> или <https://localhost/>, появится список комнат (см. рис. 2.1).
2. Нажать на кнопку перехода в комнату, убедиться, что в окне браузера содержимое сменилось инструкцией страницей авторизации (см. рис. 2.2).
3. Нажать кнопку «Главная» в левой части экрана, убедиться, что произошел возврат к списку комнат (см. рис. 2.1).
4. Выполнить пункт 2. Ввести неправильный пароль, убедиться, что было выведено сообщение об неправильном пароле (см. рис. 2.3).
5. Ввести правильный пароль, убедиться, что был осуществлен переход в выбранную комнату (см. рис. 2.4).
6. Сменить имя пользователя, убедиться, что имя было изменено (см. рис. 2.5).
7. Нажать кнопку «Захватить веб-камеру», убедиться, что появилось окно с запросом разрешения на захват медиапотока (см. рис. 2.7).
8. Разрешить захват медиапотока, убедиться, что в видеоокне появилось изображение с веб-камеры (см. рис. 2.8).
9. Повторить пункты 5, 7, 8, только вместо нажатия кнопки «Захватить веб-камеру» нажать на кнопку «Демонстрация экрана компьютера».
10. Нажать на кнопку «Захватить микрофон», убедиться, что появилось окно с запросом разрешения на захват медиапотока (см. рис. 2.7).
11. Разрешить захват медиапотока, убедиться, что в видеоокне появилась аудиодорожка микрофона (см. рис. 2.8).
12. Выбрать в виджете для выбора разрешения захватываемого изображения (см. рис. 2.6) другое разрешение и повторить пункт 9, убедиться, что выбранное разрешение соответствует изображению в видеоокне.
13. Повторить пункт 12 для всех вариантов разрешений.
14. С другого устройства выполнить пункты 1, 2, 5, 6, убедиться, что соединение было установлено и что на экране первого устройства отображено два видеоокна (см. рис. 2.9).
15. На другом устройстве выполнить пункт 9, убедиться, что на первом устройстве изображения второго видеокна обновилось (см. рис. 2.10).

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	
Изм.	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022
					Лист 86

ЗАКЛЮЧЕНИЕ

Во время работы над ВКР был выполнен анализ актуальности программного комплекса для организации и проведения видеоконференций, а также был проведен анализ отечественных и зарубежных аналогов с выявлением недостатков, с которыми сталкиваются пользователи таких систем.

Была предложена функциональная модель, а также разработаны алгоритмы для реализации групповых видеоконференций с возможностью передачи текстовых и файловых данных.

Был разработан ПП, который соответствует поставленным функциональным требованиям.

Программный комплекс следует развивать по следующим направлениям:

1. Разработка отдельного компьютерного клиента, поскольку собственный клиент позволит повысить удобство работы с системой, а также сделает программный комплекс невосприимчивым к ситуациям, когда обновление веб-браузера может затронуть работоспособность веб-клиента и лишить возможности пользоваться системой ВКС.
2. Разработка отдельного мобильного клиента, который позволит реализовать те функции клиента, которые недоступны в мобильной версии веб-браузера, а также позволит снизить заряд расходуемой батареи мобильного устройства и повысить производительность работы интерфейса системы.
3. Реализация поддержки СУБД SQL для производительности и устойчивости системы при очень больших объемах данных.

Результаты работы были внедрены в организацию «ГБУЗ АО АМОКБ» в кабинет Телемедицины. Соответствующий акт о внедрении представлен в *Приложении 11*.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					Лист
										88

2.	Разработка отдельного мобильного клиента, который позволит реализовать те функции клиента, которые недоступны в мобильной версии веб-браузера, а также позволит снизить заряд расходуемой батареи мобильного устройства и повысить производительность работы интерфейса системы.
3.	Реализация поддержки СУБД SQL для производительности и устойчивости системы при очень больших объемах данных.

Результаты работы были внедрены в организацию «ГБУЗ АО АМОКБ» в кабинет Телемедицины. Соответствующий акт о внедрении представлен в *Приложении 11*.

СПИСОК ИСПОЛЬЗУЕМЫХ ТЕРМИНОВ, УСЛОВНЫХ ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

API (интерфейс прикладного программирования) – специальный набор компонентов, с помощью которых одна компьютерная программа может взаимодействовать с другой программой.

Битрейт медиапотока – количество бит, используемых для передачи медиаданных в единицу времени. Битрейт используется при измерении эффективной скорости передачи потока данных по сетевому каналу, то есть минимального размера канала, который сможет пропустить этот поток без задержек и с оптимальным качеством медиаданных. Требуемый битрейт для оптимального качества передаваемых медиаданных зависит от следующих факторов, на примере видеопотока: *разрешение видеоизображения, видеокодек, количество кадров в секунду*.

БД – база данных.

Видеоконференция – способ обмена видеоизображения, звука и иных данных между двумя или более точками.

Видеокодек – конференция может быть настроена с выбором одного из трёх видеокодеков: *VP8, VP9 и H264*. Видеокодек используется для кодирования видеопотоков в конференции.

Видеоокно участника – виджет приложения, который отображает видеоизображение (видеодорожка) и воспроизводит звук (аудиодорожка) исходящие от участника.

Конференция (или комната) – функциональность приложения, предоставляющая возможность участникам, находящимся в комнате, пользоваться чатом и получать видеоокна от других участников.

Медиапоток – поток медиаданных, например *-изображение веб-камеры, звук микрофона или изображение с экрана компьютера*.

Одноранговая или пиринговая (Peer-to-Peer, P2P) сеть – компьютерная сеть, основанная между двумя узлами (peer). Часто в такой сети отсутствуют выделенные серверы, а каждый узел (peer) как является клиентом, так и выполняет функции сервера.

ПО – программное обеспечение.

ПП – программный продукт.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					89

Пользовательская история – это описание функциональной возможности ПО, составленное с точки зрения конечного пользователя. В истории отражены: название прецедента, действующее лицо, цель, предусловия, последовательность действий.

СУБД – система управления базами данных.

Система ВКС – система видеоконференцсвязи.

Сообщение – текстовый набор символов.

Телемедицина – использование современных средств коммуникаций для дистанционного предоставления врачебных и консультационных услуг.

Файл – бинарный файл, например изображения, документы, исполняемые программы.

Чат – виджет или окно приложения, посредством которого участники могут обмениваться сообщениями и файлами.

	Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата
Изм	Лист	№ документа	Подпись	Дата	
ВКРБ 09.03.04.059.2022					Лист 90

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Фаулер М. UML. Основы, 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.
2. Хританков А. С., Полежаев В. А., Андрианов А. И. Проектирование на UML. Сборник задач по проектированию программных систем. 2-е. изд. – Екатеринбург: Издательские решения, 2017. – 240 с.
3. Белов С.В., Лаптев В.В., Морозов А.В., Толасова В.В., Мамлеева А.Р. Требования к оформлению студенческих работ. / АГТУ – Астрахань, 2019. 60 с.
4. Закас Н., ECMAScript 6 для разработчиков. – СПб.: Питер, 2017. – 352 с.: ил. – (Серия «Библиотека программиста»).
5. Дуглас К., Как устроен JavaScript. – СПб.: Питер, 2019. – 304 с. – (Серия «Для профессионалов»).
6. Янг А., Мек Б., Кантелон М., Node.js в действии. 2-е изд. – СПб.: Питер, 2018. – 432 с.: ил. – (Серия «Для профессионалов»).
7. PlantUML – [Электронный ресурс] режим доступа: <http://plantuml.com/ru/> (02.05.2022).
8. Теория и практика UML. Диаграмма последовательности | Статьи – [Электронный ресурс] режим доступа: http://www.it-gost.ru/articles/view_articles/94. (03.05.2022).
9. Теория и практика UML. Диаграмма состояний | Статьи – [Электронный ресурс] режим доступа: http://www.it-gost.ru/articles/view_articles/97 (03.05.2022).
10. WebRTC – [Электронный ресурс] режим доступа: <https://webrtc.org> (05.05.2022).
11. WebRTC - MDN Web Docs Glossary: Definitions of Web-related terms | MDN – [Электронный ресурс] режим доступа: https://developer.mozilla.org/en-US/docs/Web/API/MediaStream_Recording_API (05.05.2022).
12. WebRTC - преимущества, недостатки, секреты, как работает, подключение – [Электронный ресурс] режим доступа: <https://trueconf.ru/webrtc.html> (06.05.2022).
13. WebRTC samples – [Электронный ресурс] режим доступа: <https://webrtc.github.io/samples/> (06.05.2022).

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
										91
Изм	Лист	№ документа	Подпись	Дата						

14. Объяснение протокола SRTP / Блог компании Voximplant / Хабр – [Электронный ресурс] режим доступа: <https://habr.com/ru/company/Voximplant/blog/413165/> (07.05.2022).
15. TypeScript: The starting point for learning TypeScript – [Электронный ресурс] режим доступа: <https://www.typescriptlang.org/docs/handbook/> (07.05.2022).
16. Introduction | Socket.IO – [Электронный ресурс] режим доступа: <https://socket.io/docs/v3> (10.05.2022).
17. mediasoup :: Documentation – [Электронный ресурс] режим доступа: <https://mediasoup.org/documentation/> (12.05.2022).
18. Resumable upload protocol 1.0.x | tus – [Электронный ресурс] режим доступа: <https://tus.io/protocols/resumable-upload.html> (14.05.2022).
19. Купить видеоконференцсвязь – Цены TrueConf – [Электронный ресурс] режим доступа: <https://trueconf.ru/prices.html> (15.05.2022).
20. Видеоконференцсвязь Zoom – Тарифные планы и цены | Zoom – [Электронный ресурс] режим доступа: <https://zoom.us/pricing> (15.05.2022).
21. Jitsi Meet – [Электронный ресурс] режим доступа: <https://meet.jit.si/> (15.05.2022).

Инв. № подл.	Подп. и дата				Взам. инв. №	Инв. № дубл	Подп. и дата
Инв. № подл.	Подп. и дата				Взам. инв. №	Инв. № дубл	Подп. и дата
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022		
					92		

ПРИЛОЖЕНИЕ 1

ТЕХНИЧЕСКОЕ ЗАДАНИЕ на разработку программного продукта «Программный комплекс для организации и проведения видеоконференций»

Направление 090304 – Программная инженерия

Исполнитель: обучающийся гр. ДИПР641 Катунин С.С.

1 Назначение, цели и задачи разработки

Цели разработки:

- ☞ обеспечение возможности осуществления видеосвязи через веб-браузер;
- ☞ увеличение вместимости конференции (по количеству пользователей);
- ☞ улучшение качества и стабильности видеосвязи;
- ☞ снижение финансовых затрат на приобретение текущей ВКС-системы.

Назначение разработки: система ВКС предназначена для организации и проведения видеоконференций.

Основные задачи, решаемые разработчиком в процессе разработки системы:

- ☞ анализ предметной области;
- ☞ разработка программного продукта в соответствии с требованиями;
- ☞ документирование проекта в соответствии с установленными требованиями.

2 Характер разработки: прикладная квалификационная работа.

3 Основания для разработки

- ☞ Учебный план направления 09.03.04 «Программная инженерия» 2018 года набора.
- ☞ Распоряжение по кафедре АСОИУ № 53 от «20» декабря 2022 г.

4 Плановые сроки выполнения

Начало «10» декабря 2021 г.

Окончание «28» июня 2022 г.

5 Требования к проектируемой системе

5.1 Требования к функциональным характеристикам

Проектируемая система должна обеспечивать выполнение следующих основных функций:

- ☞ позволять организовывать видеоконференций согласно архитектуре SFU;
- ☞ позволять организовывать конференции в закрытой локальной сети без доступа к Интернету;

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	<div>☞ документирование проекта в соответствии с установленными требованиями.</div> <div>2 Характер разработки: прикладная квалификационная работа.</div> <div>3 Основания для разработки<div>☞ Учебный план направления 09.03.04 «Программная инженерия» 2018 года набора.</div><div>☞ Распоряжение по кафедре АСОИУ № <u>53</u> от «<u>20</u>» <u>декабря</u> <u>2022</u> г.</div></div> <div>4 Плановые сроки выполнения<div>Начало «<u>10</u>» <u>декабря</u> <u>2021</u> г.</div><div>Окончание «<u>28</u>» <u>июня</u> <u>2022</u> г.</div></div> <div>5 Требования к проектируемой системе<div>5.1 Требования к функциональным характеристикам<div>Проектируемая система должна обеспечивать выполнение следующих основных функций:<div>☞ позволять организовывать видеоконференций согласно архитектуре SFU;</div><div>☞ позволять организовывать конференции в закрытой локальной сети без доступа к Интернету;</div></div></div></div>				
					<div><div><div>Изм</div><div>Лист</div><div>№ документа</div><div>Подпись</div><div>Дата</div></div><div>ВКРБ 09.03.04.059.2022</div><div>Лист 93</div></div>				

- ✧ предоставлять возможность пользователям изменять видео- и аудиопотоки (например, переключение с веб-камеры на демонстрацию экрана) во время уже установленного соединения;
- ✧ предоставлять возможность одновременного захвата нескольких видеоустройств с одного клиентского устройства;
- ✧ предоставлять возможность выбора разрешения и количества кадров видеоизображения, захваченного с видеоустройства или экрана компьютера;
- ✧ поддерживать клиентов, пользующихся интернет-связью с недостижимым NAT (например, некоторые виды мобильного интернета), то есть когда полученные данные от STUN-сервер не соответствуют сетевым данным пользователя, поскольку между ним и локальным компьютером клиента находится более одного устройства с NAT (более одного маршрутизатора), так как в библиотеке mediasoup предусмотрена реализация спецификации ICE-Lite;
- ✧ предоставлять возможность участникам обмениваться текстовыми сообщениями и файлами с помощью чата;
- ✧ предоставлять администратору возможность создавать, редактировать и удалять комнаты, а также устанавливать видеокодеки для заданной комнаты и задавать параметр сохранения истории чата в комнате;
- ✧ предоставлять администратору следующие возможности: исключение участника из комнаты, отключение передачи видео-/аудиопотоков исходящих от пользователя, изменение имени пользователя, блокировка пользователя по IP-адресу.

Программа предусматривает функциональные ограничения:

- ✧ программа не должна обеспечивать обычным пользователям возможность самостоятельно выбирать кодеки для видео- и аудиопотоков;
- ✧ в системе не предусмотрен механизм личных сообщений и личного кабинета пользователя.

Инв. № подл.	Подп. и дата				Взам. инв. №	Инв. № дубл	Подп. и дата	
<p>сообщениями и файлами с помощью чата;</p> <p>☞ предоставлять администратору возможность создавать, редактировать и удалять комнаты, а также устанавливать видеокодеки для заданной комнаты и задавать параметр сохранения истории чата в комнате;</p> <p>☞ предоставлять администратору следующие возможности: исключение участника из комнаты, отключение передачи видео-/аудиопотоков исходящих от пользователя, изменение имени пользователя, блокировка пользователя по IP-адресу.</p> <p>Программа предусматривает функциональные ограничения:</p> <p>☞ программа не должна обеспечивать обычным пользователям возможность самостоятельно выбирать кодеки для видео- и аудиопотоков;</p> <p>☞ в системе не предусмотрен механизм личных сообщений и личного кабинета пользователя.</p>								
					ВКРБ 09.03.04.059.2022			Лист
								94
Изм	Лист	№ документа	Подпись	Дата				

5.2 Требования к эксплуатационным характеристикам

Программа не должна аварийно завершаться при любых действиях пользователя.

Время реакции программы на действия пользователя не должно превышать 15 секунд.

5.3 Требования к программному обеспечению:

Операционная система сервера: Linux Debian 11 Bullseye или Windows 10.

Среда выполнения на сервере: Node.js LTS версии v12.22.x.

Веб-браузер клиента: Firefox последней мажорной версии (на данный момент 100.x.x), Google Chrome последней мажорной версии (на данный момент 100.x.x.x).

Операционная система клиента: дистрибутивы Linux, в которых есть возможность установить браузер необходимой версии (см. выше), а также Windows 7 или более поздние.

5.4 Требования к аппаратному обеспечению:

Рекомендуемая конфигурация для сервера:

- ✦ Процессор: не менее 2.4 ГГц, 4 ядра.
- ✦ Оперативная память: не менее 4 Гб.
- ✦ Свободное место на жестком диске: не менее 1 Гб.
- ✦ Минимальная скорость входящего подключения: 1 Мбит/с.
- ✦ Минимальная скорость исходящего подключения: 1 Мбит/с.
- ✦ Публичный IP (если система будет работать не в локальной сети).

Рекомендуемая конфигурация для клиента:

- ✦ Процессор Intel Pentium 4 или старше с поддержкой SSE3.
- ✦ Оперативная память: не менее 2 Гб.
- ✦ Минимальная скорость входящего подключения: 512 Кбит/с.
- ✦ Минимальная скорость исходящего подключения: 512 Кбит/с.

6 Стадии и этапы разработки

6.1 Эскизный проект (ЭП)

- ✦ Анализ предметной области.
- ✦ Подготовка проектной документации.

6.2 Технический проект (ТП)

- ✦ Разработка структур и форм представления данных.
- ✦ Разработка структуры программного комплекса.
- ✦ Подготовка пояснительной записки.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						Лист	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					95	

✎	Оперативная память: не менее 4 Гб.
✎	Свободное место на жестком диске: не менее 1 Гб.
✎	Минимальная скорость входящего подключения: 1 Мбит/с.
✎	Минимальная скорость исходящего подключения: 1 Мбит/с.
✎	Публичный IP (если система будет работать не в локальной сети).
Рекомендуемая конфигурация для клиента:	
✎	Процессор Intel Pentium 4 или старше с поддержкой SSE3.
✎	Оперативная память: не менее 2 Гб.
✎	Минимальная скорость входящего подключения: 512 Кбит/с.
✎	Минимальная скорость исходящего подключения: 512 Кбит/с.

6 Стадии и этапы разработки

6.1 Эскизный проект (ЭП)

✎ Анализ предметной области.

✎ Подготовка проектной документации.

6.2 Технический проект (ТП)

✎ Разработка структур и форм представления данных.

✎ Разработка структуры программного комплекса.

✎ Подготовка пояснительной записки.

6.3 Рабочий проект (РП)

- ✎ Программная реализация.
- ✎ Тестирование и отладка программы.
- ✎ Подготовка программной и эксплуатационной документации.

6.4 Эксплуатация (Э)

Описание и анализ результатов проведенного исследования.

7 Требования к документированию проекта

К защите выпускной квалификационной работы должны быть представлены следующие документы:

- ✎ Пояснительная записка к ВКР.
- ✎ Презентация доклада.
- ✎ Программа, презентация и пояснительная записка к ВКР на оптическом носителе.

Требования к структуре документов определены соответствующими стандартами ЕСПД.

Требования к оформлению определены соответствующими методическими указаниями.

8 Порядок контроля и приемки

Контроль выполнения выпускной квалификационной работы проводится научным руководителем поэтапно в соответствии с утвержденным графиком выполнения ВКР.

На завершающем этапе руководитель осуществляет нормоконтроль представленной исполнителем документации и принимает решение о допуске (недопуске) ВКР к защите.

В процессе защиты ВКР исполнитель представляет документацию, делает краткое сообщение по теме разработки и демонстрирует ее программную реализацию.

При выставлении оценки учитывается:

- ✎ степень соответствия представленной разработки требованиям технического задания;
- ✎ качество программной реализации, документации и доклада по теме проекта;
- ✎ соблюдение исполнителем графика выполнения ВКР.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	Инв. № подл.					
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022		Лист			
							96			

9 Литература

1. Фаулер М. UML. Основы, 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.
2. Закас Н., ECMAScript 6 для разработчиков. – СПб.: Питер, 2017. – 352 с.: ил. – (Серия «Библиотека программиста»).
3. Дуглас К., Как устроен JavaScript. – СПб.: Питер, 2019. – 304 с. – (Серия «Для профессионалов»).
4. Янг А., Мек Б., Кантелон М., Node.js в действии. 2-е изд. – СПб.: Питер, 2018. – 432 с.: ил. – (Серия «Для профессионалов»).

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата	Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист
											97

ФУНКЦИОНАЛЬНАЯ ДИАГРАММА

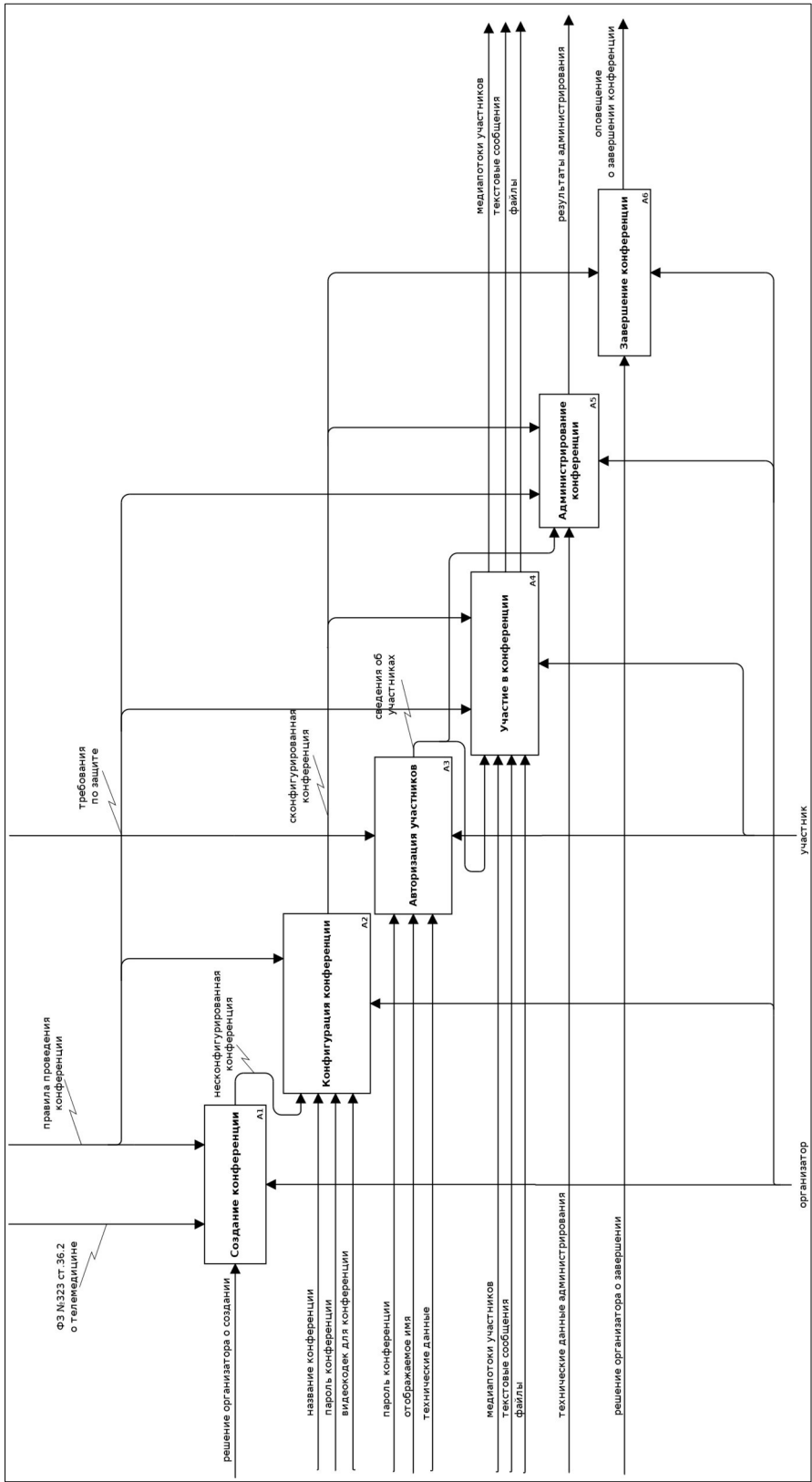


Рисунок П2.1 – Функциональная диаграмма - А0

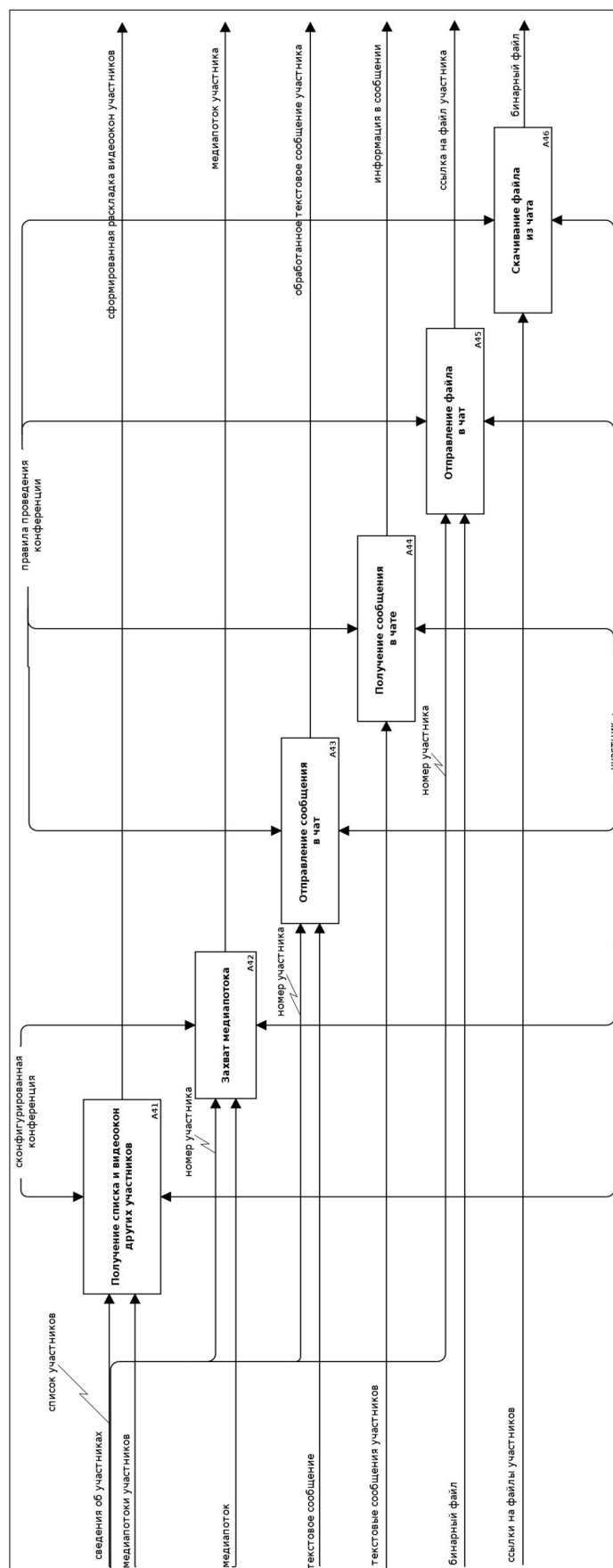


Рисунок П2.2 – Функциональная диаграмма - А4

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата
Изм	Лист	№ документа	Подпись	Дата

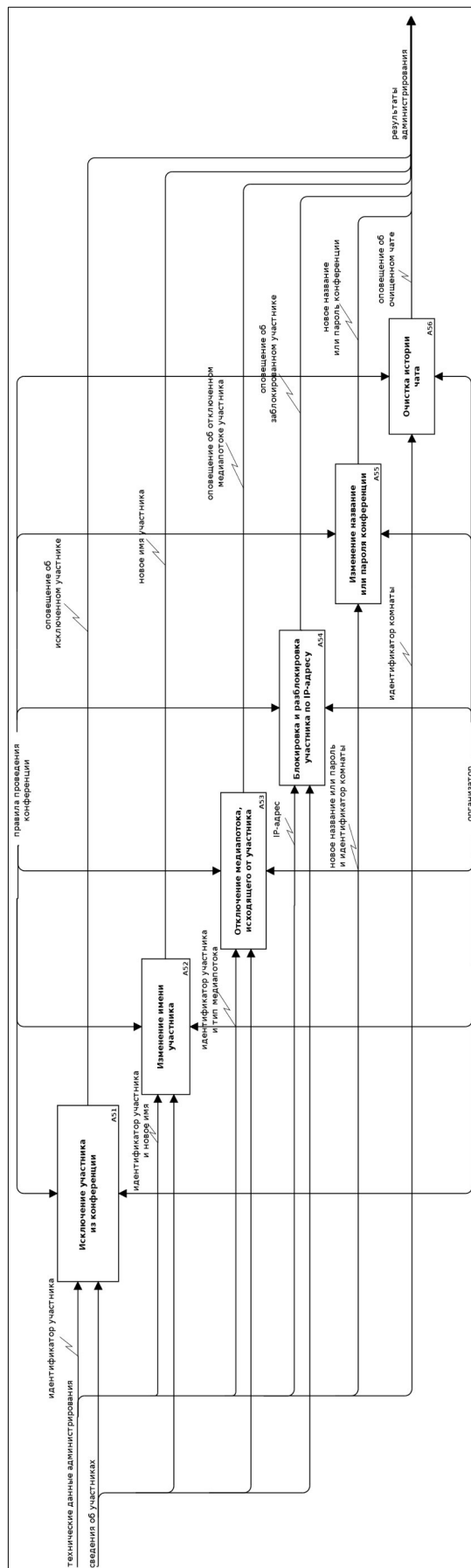


Рисунок П2.3 – Функциональная диаграмма - А5

ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



Рисунок ПЗ – Диаграмма вариантов использования

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Изм.	Лист	№ документа	Подпись	Дата

ПОЛЬЗОВАТЕЛЬСКИЕ ИСТОРИИ

Название прецедента: выбрать комнату и авторизоваться в ней.

Действующее лицо: участник.

Цель: зайти в выбранную конференцию с целью участвовать в ней.

Предусловия: система выдала участнику *список комнат*.

Главная последовательность:

1. Участник выбирает в *списке комнат* определенную комнату.
2. Система показывает участнику *окно авторизации*, содержащее *поля для ввода пароля от комнаты*, а также кнопку «войти в комнату».
3. Участник вводит пароль от комнаты и нажимает кнопку «войти в комнату».
4. Система проверяет пароль комнаты.
5. Система показывает участнику *окно комнаты*, содержащее *виджет с чатом и виджет для видеоокон других участников*.

Альтернативная последовательность (участник ввел неправильный пароль):

1. Участник выбирает в *списке комнат* определенную комнату.
2. Система показывает участнику *окно авторизации*, содержащее *поля для ввода пароля от комнаты*, а также кнопку «войти в комнату».
3. Участник вводит пароль от комнаты и нажимает кнопку «войти в комнату».
4. Система проверяет пароль комнаты.
5. Система сообщает участнику о том, что введенный им пароль неправильный.

Название прецедента: получить список и видеоокна других участников.

Действующее лицо: участник.

Цель: коммуницировать с другими участниками посредством видео- и аудиосвязи.

Предусловия: участник находится в *комнате* с другими участниками.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	пароль):					
					1. Участник выбирает в <i>списке комнат</i> определенную комнату.					
					2. Система показывает участнику <i>окно авторизации</i> , содержащее <i>поля для ввода пароля от комнаты</i> , а также кнопку « <i>войти в комнату</i> ».					
					3. Участник вводит пароль от комнаты и нажимает кнопку « <i>войти в комнату</i> ».					
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	4. Система проверяет пароль комнаты.					
					5. Система сообщает участнику о том, что введенный им пароль неправильный.					
					Название прецедента: получить список и видеоокна других участников.					
					Действующее лицо: участник.					
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата	Цель: коммуницировать с другими участниками посредством видео- и аудиосвязи.					
					Предусловия: участник находится в <i>комнате</i> с другими участниками.					
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					Лист
										102

Главная последовательность:

1. Система отправляет участнику *список* и *видеоокна других участников*.
2. Система отображает видеокна других участников.

Альтернативная последовательность (в комнату зашел новый другой участник):

1. Система добавляет нового участника в *список* и отправляет его *видеоокно*.
2. Система отображает видеокно нового участника.

Альтернативная последовательность (другой участник покинул комнату):

1. Система удаляет другого участника из *списка*.
2. Система перестает отображать на экране видеокно другого участника.

Название прецедента: получить сообщение в чате.

Действующее лицо: участник.

Цель: коммуницировать с другими участниками посредством текстовой связи.

Предусловия: участник находится в *комнате* с другими участниками.

Главная последовательность:

1. Система отправляет участнику сообщение от другого участника.
2. Система отображает сообщение в чате в соответствии с *типом сообщения*.

Бизнес-правила: сообщение может быть гиперссылкой, которое необходимо отображать как ссылку, а не как текстовое сообщение.

Название прецедента: захватить медиапоток.

Действующее лицо: участник.

Цель: отправить медиапоток другим участникам для коммуникации с ними.

Предусловия:

- ☛ Участник находится в *комнате* с другими участниками.
- ☛ Участник имеет устройство, с которого можно захватить медиапоток.

Главная последовательность:

1. Участник нажимает на кнопку «*захватить микрофон*» или «*захватить веб-камеру*».
2. Система захватывает медиапоток с устройства пользователя и отправляет его другим участникам.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата					
<p>Предусловия: участник находится в комнате с другими участниками.</p> <p>Главная последовательность:</p> <ol style="list-style-type: none">1. Система отправляет участнику сообщение от другого участника.2. Система отображает сообщение в чате в соответствии с <i>типом сообщения</i>. <p>Бизнес-правила: сообщение может быть гиперссылкой, которое необходимо отображать как ссылку, а не как текстовое сообщение.</p> <p>Название прецедента: захватить медиапоток.</p> <p>Действующее лицо: участник.</p> <p>Цель: отправить медиапоток другим участникам для коммуникации с ними.</p> <p>Предусловия:</p> <ul style="list-style-type: none">☞ Участник находится в <i>комнате</i> с другими участниками.☞ Участник имеет устройство, с которого можно захватить медиапоток. <p>Главная последовательность:</p> <ol style="list-style-type: none">1. Участник нажимает на кнопку «<i>захватить микрофон</i>» или «<i>захватить веб-камеру</i>».2. Система захватывает медиапоток с устройства пользователя и отправляет его другим участникам.									
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022				Лист
									103

Альтернативная последовательность (участник захватывает изображение с экрана компьютера):

1. Участник выбирает в *выпадающем меню* желаемое разрешение захватываемого видеоизображения.
2. Участник нажимает на кнопку «*захватить изображение с экрана компьютера*».
3. Система захватывает медиапоток в выбранном разрешении и отправляет его другим участникам.

Бизнес-правила:

- ✦ Видеоизображение захватывается в формате 16:9.
- ✦ Минимальное разрешение захватываемого изображения с экрана компьютера 240p (426x240 пикселей).
- ✦ Максимальное разрешение захватываемого изображения с экрана компьютера 1440p (2560x1440 пикселей).

Название прецедента: отправить сообщение в чате.

Действующее лицо: участник.

Цель: отправить сообщение другим участникам для передачи информации.

Предусловия: участник находится в *комнате* с другими участниками.

Главная последовательность:

1. Участник вводит текстовое сообщение в *текстовое поле чата*.
2. Участник нажимает на кнопку «*отправить сообщение*».
3. Система получает и обрабатывает сообщение, удаляя из него лишние знаки пробела и табуляции.
4. Система отправляет сообщение другим участникам.

Альтернативная последовательность (участник отправляет пустое сообщение):

1. Участник вводит пустое текстовое сообщение в *текстовое поле чата*.
2. Участник нажимает на кнопку «*отправить сообщение*».
3. Система получает и обрабатывает сообщение, удаляя из него лишние знаки пробела и табуляции.
4. В результате обработки сообщения система определяет, что оно пустое и поэтому не отправляет сообщение другим участникам.

Инв. № подл.	Подп. и дата		Взам. инв. №		Инв. № дубл		Подп. и дата		
<p>Цель: отправить сообщение другим участникам для передачи информации.</p> <p>Предусловия: участник находится в <i>комнате</i> с другими участниками.</p> <p>Главная последовательность:</p> <ol style="list-style-type: none">1. Участник вводит текстовое сообщение в <i>текстовое поле чата</i>.2. Участник нажимает на кнопку «<i>отправить сообщение</i>».3. Система получает и обрабатывает сообщение, удаляя из него лишние знаки пробела и табуляции.4. Система отправляет сообщение другим участникам. <p>Альтернативная последовательность (участник отправляет пустое сообщение):</p> <ol style="list-style-type: none">1. Участник вводит пустое текстовое сообщение в <i>текстовое поле чата</i>.2. Участник нажимает на кнопку «<i>отправить сообщение</i>».3. Система получает и обрабатывает сообщение, удаляя из него лишние знаки пробела и табуляции.4. В результате обработки сообщения система определяет, что оно пустое и поэтому не отправляет сообщение другим участникам.									
					ВКРБ 09.03.04.059.2022				
Изм	Лист	№ документа	Подпись	Дата					Лист
									104

Название прецедента: отправить файл в чат.

Действующее лицо: участник.

Цель: отправить файл другим участникам для передачи информации и данных.

Предусловия: участник находится в *комнате* с другими участниками.

Главная последовательность:

1. Участник выбирает файл *в поле для выбора файла*.
2. Участник нажимает на кнопку «*отправить файл*».
3. Система проверяет размер файла на допустимость.
4. Система получает файл и формирует ссылку на скачивание этого файла.
5. Система отправляет ссылку на скачивание файла всем участникам чата.

Альтернативная последовательность (участник отправляет файл с недопустимым размером):

1. Участник выбирает файл *в поле для выбора файла*.
2. Участник нажимает на кнопку «*отправить файл*».
3. Система проверяет размер файла на допустимость.
4. Система определяет, что размер файла превышает допустимый размер и прекращает принимать файл от участника.

Название прецедента: скачать файл из чата.

Действующее лицо: участник.

Цель: скачать файл другого участника для получения информации и данных.

Предусловия:

- ✦ Участник находится в *комнате* с другими участниками.
- ✦ В *чате* должна быть ссылка на скачивание файла другого участника.

Главная последовательность:

1. Участник нажимает на ссылку на скачивание файла другого участника.
2. Система отправляет файл участнику.
3. Участник скачивает и сохраняет файл на устройство.

Название прецедента: создать конференцию.

Действующее лицо: организатор.

Цель: организовать конференцию (комнату), для коммуникации участников находящихся в ней.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
										105
					Изм	Лист	№ документа	Подпись	Дата	

Главная последовательность:

1. Организатор открывает *окно* для *создания конференции*.
2. Организатор конфигурирует будущую конференцию начальными данными (см. прецедент *сконфигурировать конференцию*).
3. Система создает конференцию и помещает ссылку на неё в *список комнат* на главной странице.

Название прецедента: сконфигурировать конференцию.

Действующее лицо: организатор.

Цель: инициализировать начальные настройки конференции или скорректировать уже существующие.

Предусловия:

- ☞ Конференция создана.
- ☞ Конференция не завершена.

Главная последовательность:

1. Организатор открывает *окно* для *конфигурации конференции*.
2. Организатор вводит *название* и *пароль конференции* в соответствующие *текстовые поля*.
3. Организатор указывает иные технические данные для конференции.
4. Система применяет новые настройки для конференции незамедлительно.
5. Система обновляет настройки для всех участников конференции.

Бизнес-правила: под техническими данными имеется в виду видеокодек для видеодорожек в конференции: VP9, VP8 или H264 на выбор.

Название прецедента: завершить конференцию.

Действующее лицо: организатор.

Цель: прекратить любую активность со стороны участников в конференции (комнате).

Предусловия:

- ☞ Конференция создана.
- ☞ Конференция не завершена.

Главная последовательность:

1. Организатор открывает *окно* для *конфигурации конференции*.
2. Организатор нажимает на *кнопку завершения конференции*.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
Изм.	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					Лист
										106

2. Организатор вводит название и номер конференции в соответствующие текстовые поля.

3. Организатор указывает иные технические данные для конференции.

4. Система применяет новые настройки для конференции незамедлительно.

5. Система обновляет настройки для всех участников конференции.

Бизнес-правила: под техническими данными имеется в виду видеокодек для видеодорожек в конференции: VP9, VP8 или H264 на выбор.

Название прецедента: завершить конференцию.

Действующее лицо: организатор.

Цель: прекратить любую активность со стороны участников в конференции (комнате).

Предусловия:

- ☞ Конференция создана.
- ☞ Конференция не завершена.

Главная последовательность:

1. Организатор открывает *окно для конфигурации конференции.*

2. Организатор нажимает на *кнопку завершения конференции.*

3. Система завершает конференцию и удаляет её из *списка конференций* на *главной странице*.
4. Система перенаправляет всех участников, находящихся в конференции на *главную страницу*.

Название прецедента: исключить участника из конференции.

Действующее лицо: организатор.

Цель: прекратить активность со стороны участника в конференции (комнате).

Предусловия:

- ☞ Конференция создана.
- ☞ Конференция не завершена.
- ☞ Участник находится в комнате (участвует в конференции).

Главная последовательность:

1. Организатор открывает *окно – панель администрирования*.
2. Организатор выбирает в *выпадающем списке участников* того участника, которого необходимо исключить.
3. Организатор нажимает на *кнопку исключения участника*.
4. Система перенаправляет исключенного участника, находящихся в конференции на *главную страницу*.

Название прецедента: заблокировать или разблокировать участника.

Действующее лицо: организатор.

Цель: запретить (или наоборот, снять запрет) участнику заходить в систему.

Главная последовательность:

1. Организатор открывает *окно – панель администрирования*.
2. Организатор вводит в *текстовое поле* IP-адрес участника.
3. Организатор нажимает на *кнопку блокировки* или *кнопку разблокировки*.
4. Система блокирует (или снимает блокировку) участника.

Название прецедента: изменить имя участника.

Действующее лицо: организатор.

Цель: изменение отображаемого имени для других участников конференции.

Инв. № подл.	Подп. и дата		Взам. инв. №		Инв. № дубл		Подп. и дата	
<p>3. Организатор нажимает на <i>кнопку</i> <i>исключить участника</i>.</p> <p>4. Система перенаправляет исключенного участника, находящихся в конференции на <i>главную страницу</i>.</p> <p>Название прецедента: заблокировать или разблокировать участника.</p> <p>Действующее лицо: организатор.</p> <p>Цель: запретить (или наоборот, снять запрет) участнику заходить в систему.</p> <p>Главная последовательность:</p> <ol style="list-style-type: none">1. Организатор открывает <i>окно – панель администрирования</i>.2. Организатор вводит в <i>текстовое поле</i> IP-адрес участника.3. Организатор нажимает на <i>кнопку блокировки</i> или <i>кнопку разблокировки</i>.4. Система блокирует (или снимает блокировку) участника. <p>Название прецедента: изменить имя участника.</p> <p>Действующее лицо: организатор.</p> <p>Цель: изменение отображаемого имени для других участников конференции.</p>								
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист		
						107		

Предусловия:

- ☞ Конференция создана.
- ☞ Конференция не завершена.
- ☞ Участник находится в комнате (участвует в конференции).

Главная последовательность:

1. Организатор открывает *окно – панель администрирования*.
2. Организатор выбирает в *выпадающем списке участников* того участника, которому необходимо изменить отображаемое имя.
3. Организатор вводит новое имя для участника в *текстовое поле*.
4. Организатор нажимает на *кнопку изменения имени участника*.
5. Система изменяет отображаемое имя участника для всех участников конференции.

Название прецедента: отключить медиапоток, исходящий от участника.

Действующее лицо: организатор.

Цель: прекратить передачу медиапотока со стороны участника в конференции (комнате).

Предусловия:

- ☞ Конференция создана.
- ☞ Конференция не завершена.
- ☞ Участник находится в комнате (участвует в конференции).
- ☞ Участник передает медиапоток в систему.

Главная последовательность:

1. Организатор открывает *окно – панель администрирования*.
2. Организатор выбирает в *выпадающем списке участников* того участника, который передает медиапоток.
3. Организатор нажимает на *кнопку отключения медиапотока*.
4. Система сообщает устройству участника, что необходимо прекратить передачу медиапотока.
5. Устройство участника прекращает передачу медиапотока в систему.

Название прецедента: очистить чат конференции.

Действующее лицо: организатор.

Цель: удаление всех сообщений участников из чата комнаты.

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата						
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					Лист
										108

конференции (комнаты).

Предусловия:

- ☞ Конференция создана.
- ☞ Конференция не завершена.
- ☞ Участник находится в комнате (участвует в конференции).
- ☞ Участник передает медиапоток в систему.

Главная последовательность:

1. Организатор открывает *окно – панель администрирования*.
2. Организатор выбирает в *выпадающем списке участников* того участника, который передает медиапоток.
3. Организатор нажимает на *кнопку отключения медиапотока*.
4. Система сообщает устройству участника, что необходимо прекратить передачу медиапотока.
5. Устройство участника прекращает передачу медиапотока в систему.

Название прецедента: очистить чат конференции.

Действующее лицо: организатор.

Цель: удаление всех сообщений участников из чата комнаты.

Предусловия:

- ☞ Конференция создана.
- ☞ Конференция не завершена.

Главная последовательность:

1. Организатор открывает *окно – панель администрирования*.
2. Организатор выбирает в *выпадающем списке комнат* ту комнату, где необходимо очистить чат.
3. Организатор нажимает на *кнопку очистки чата*.
4. Система очищает чат комнаты.

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата						
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022					Лист
										109

ДИАГРАММЫ КЛАССОВ

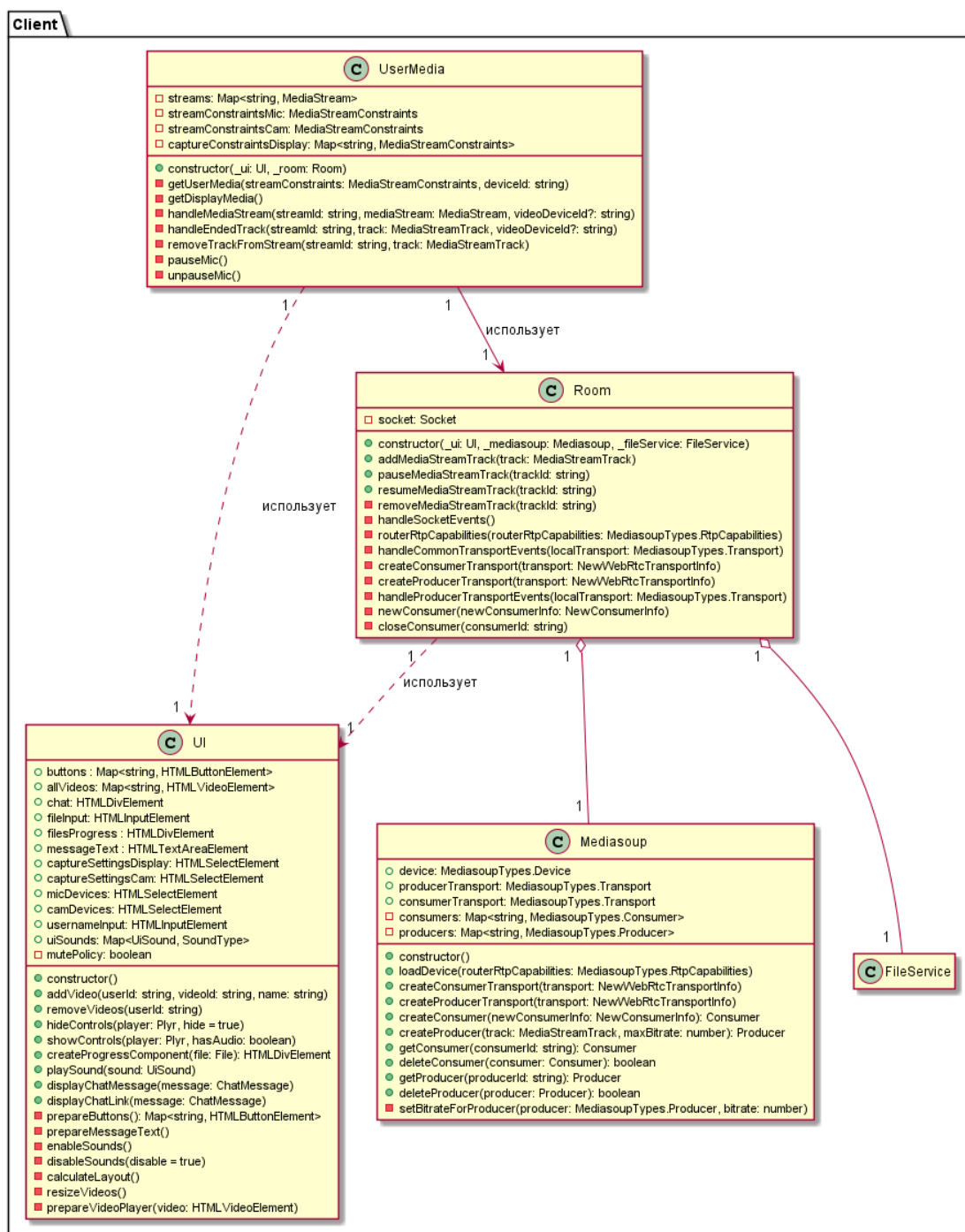


Рисунок П5.1 – Диаграмма классов клиентской части – главная часть

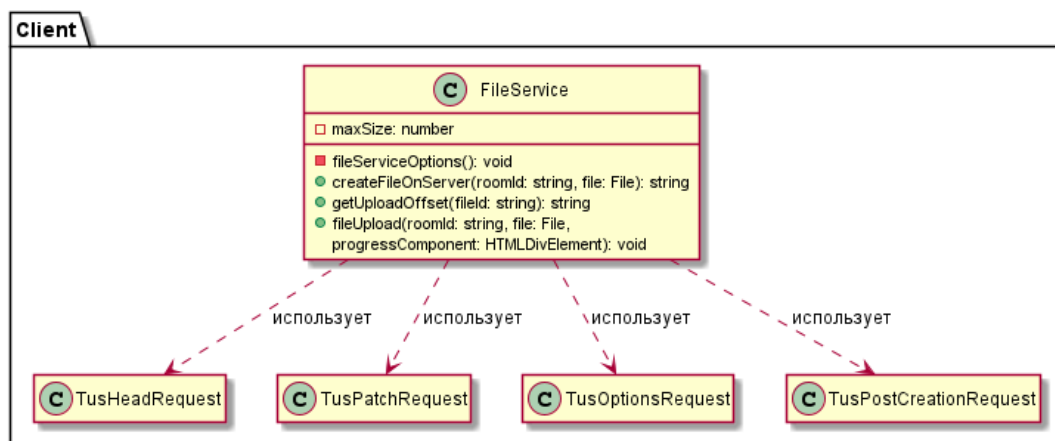


Рисунок П5.2 – Диаграмма классов клиентской части – файловый сервис

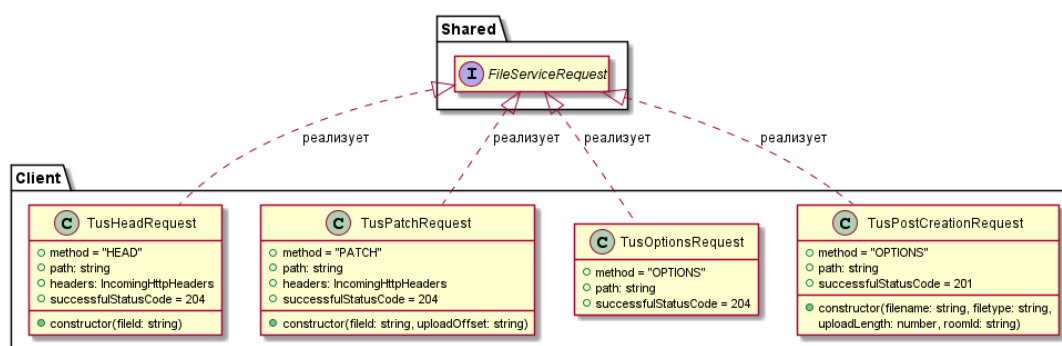


Рисунок П5.3 – Диаграмма классов клиентской части – протокол TUS

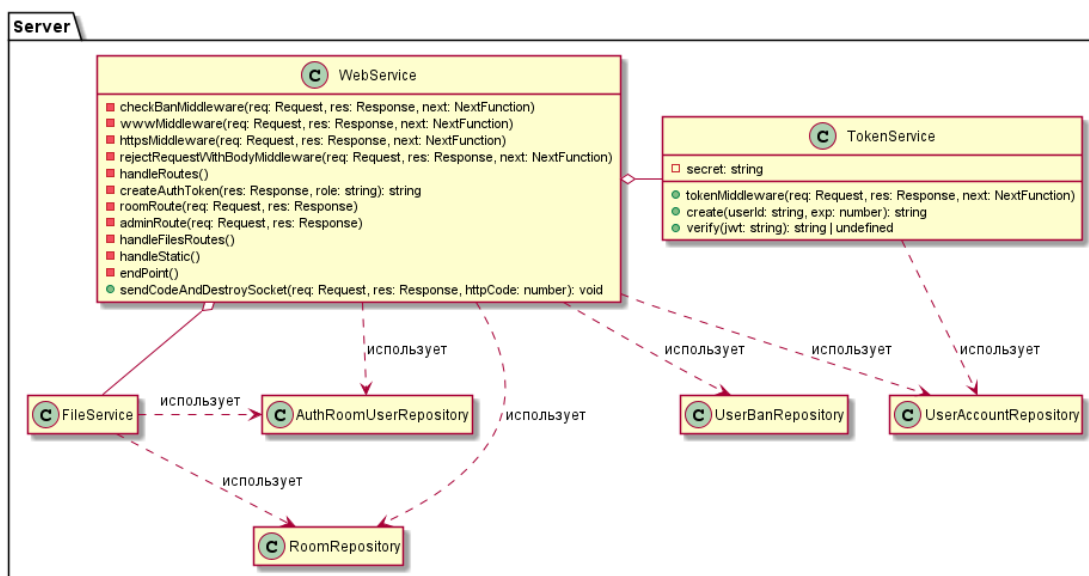


Рисунок П5.4 – Диаграмма классов серверной части – веб-сервис

Инв. № подл.	Взам. инв. №	Инв. № дубл.	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
									111
Изм.	Лист	№ документа	Подпись	Дата					

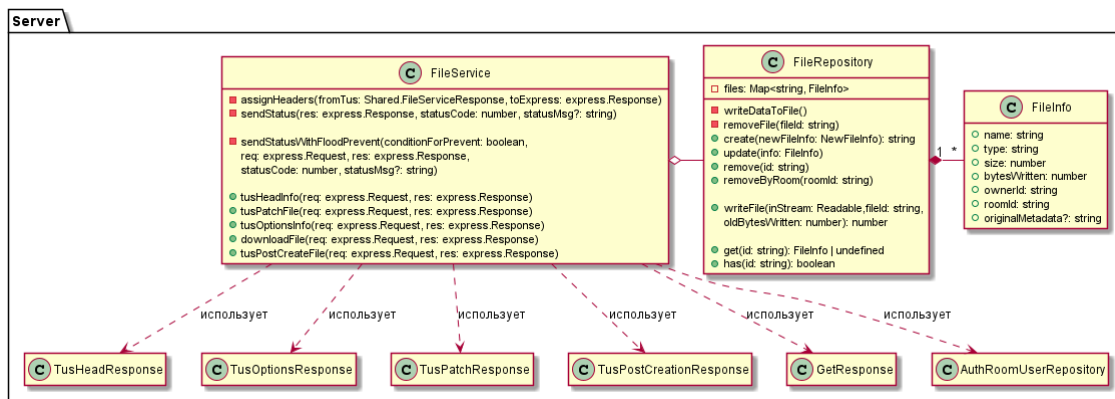


Рисунок П5.6 – Диаграмма классов серверной части – файловый сервис

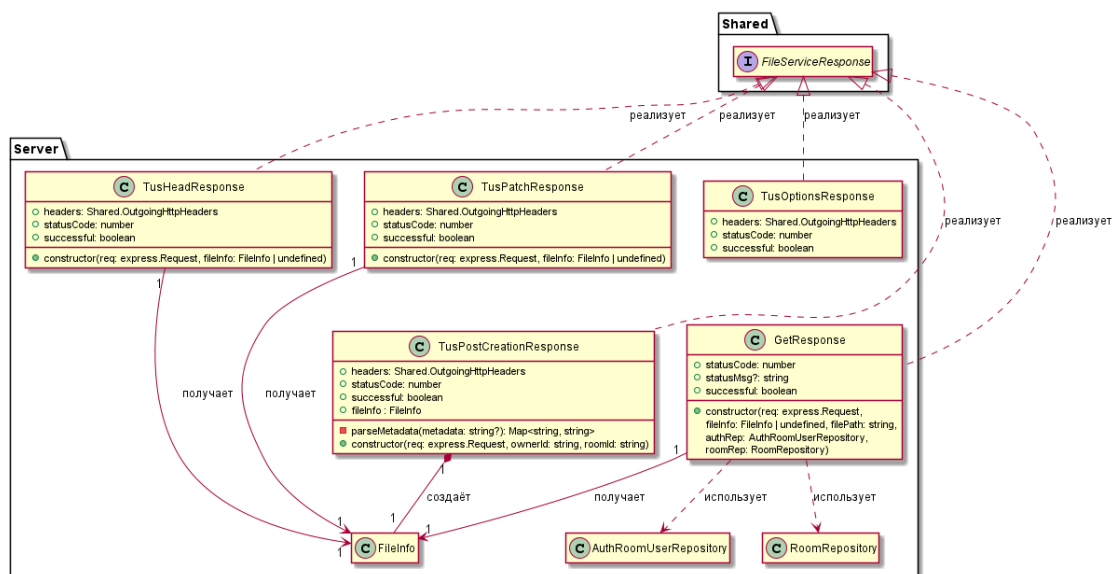


Рисунок П5.7 – Диаграмма классов серверной части – протокол TUS

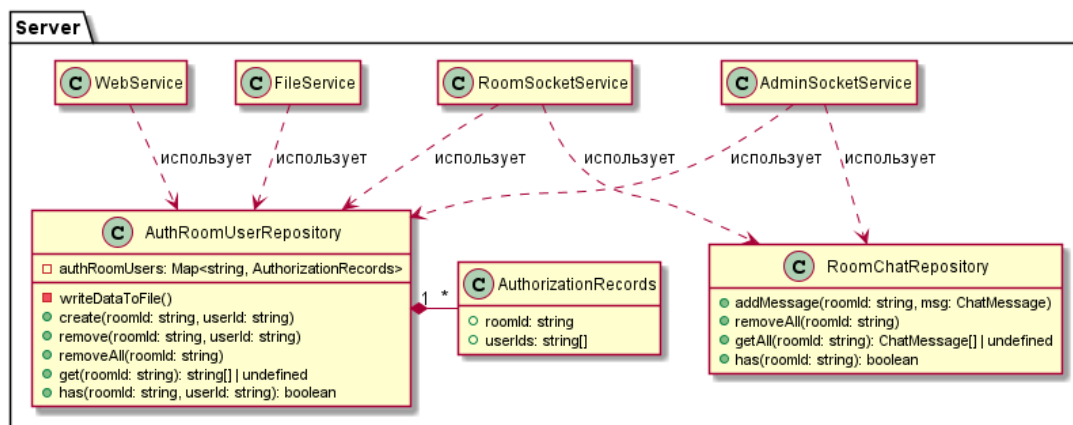


Рисунок П5.8 – Диаграмма классов серверной части – репозитории данных – часть 1

Инв. № подл.	Подп. и дата
Взам. инв. №	Инв. № дубл.
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ документа	Подпись	Дата
------	------	-------------	---------	------

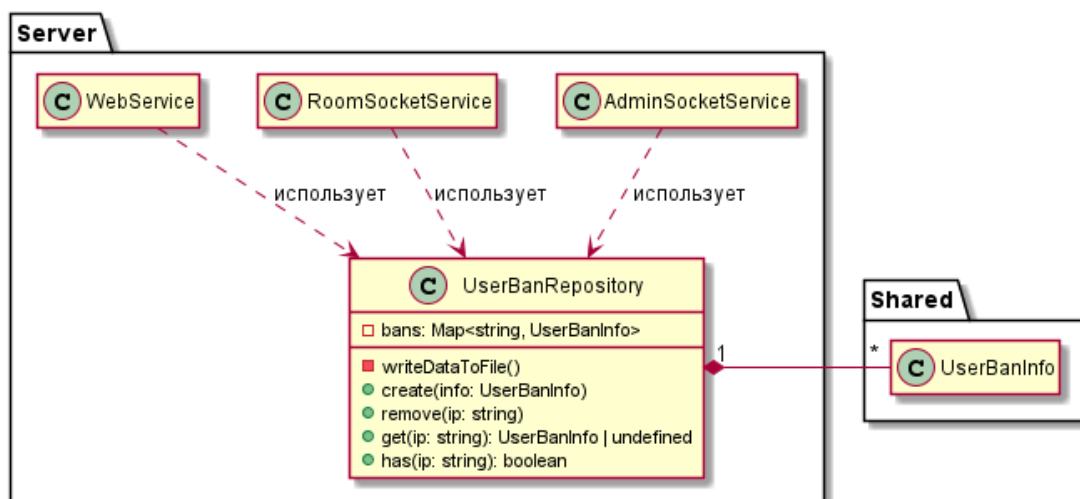


Рисунок П5.9 – Диаграмма классов серверной части – репозитории данных – блокировки пользователей

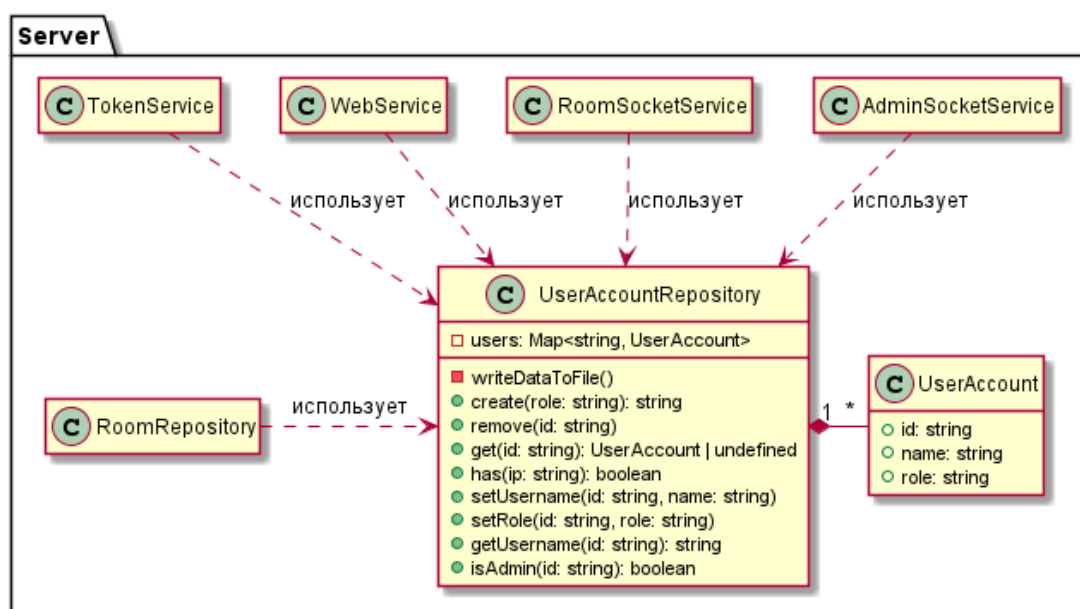
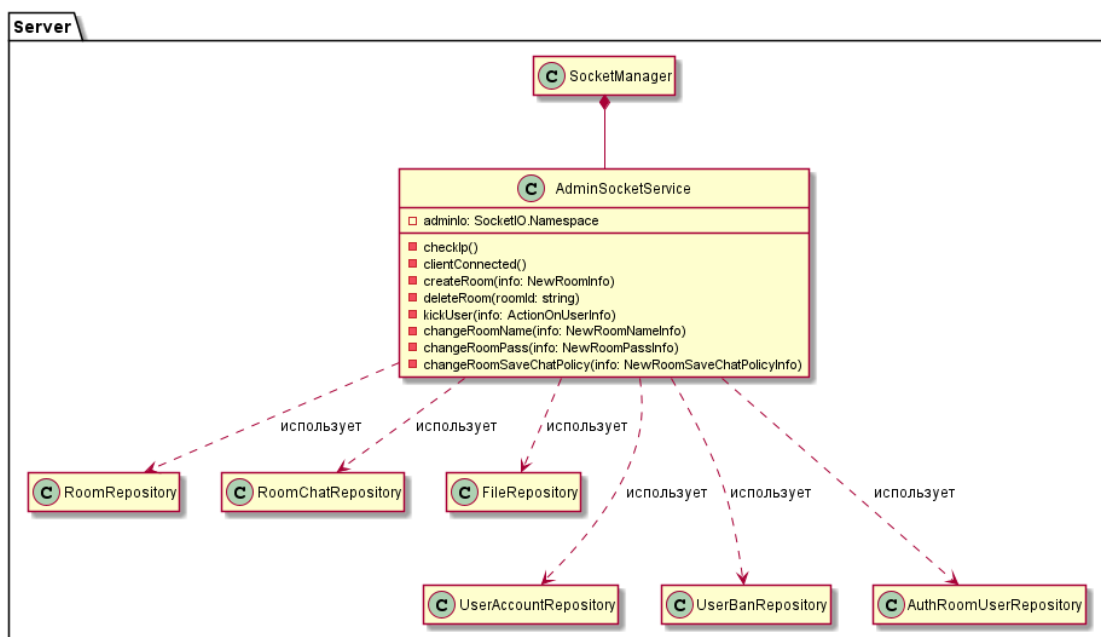
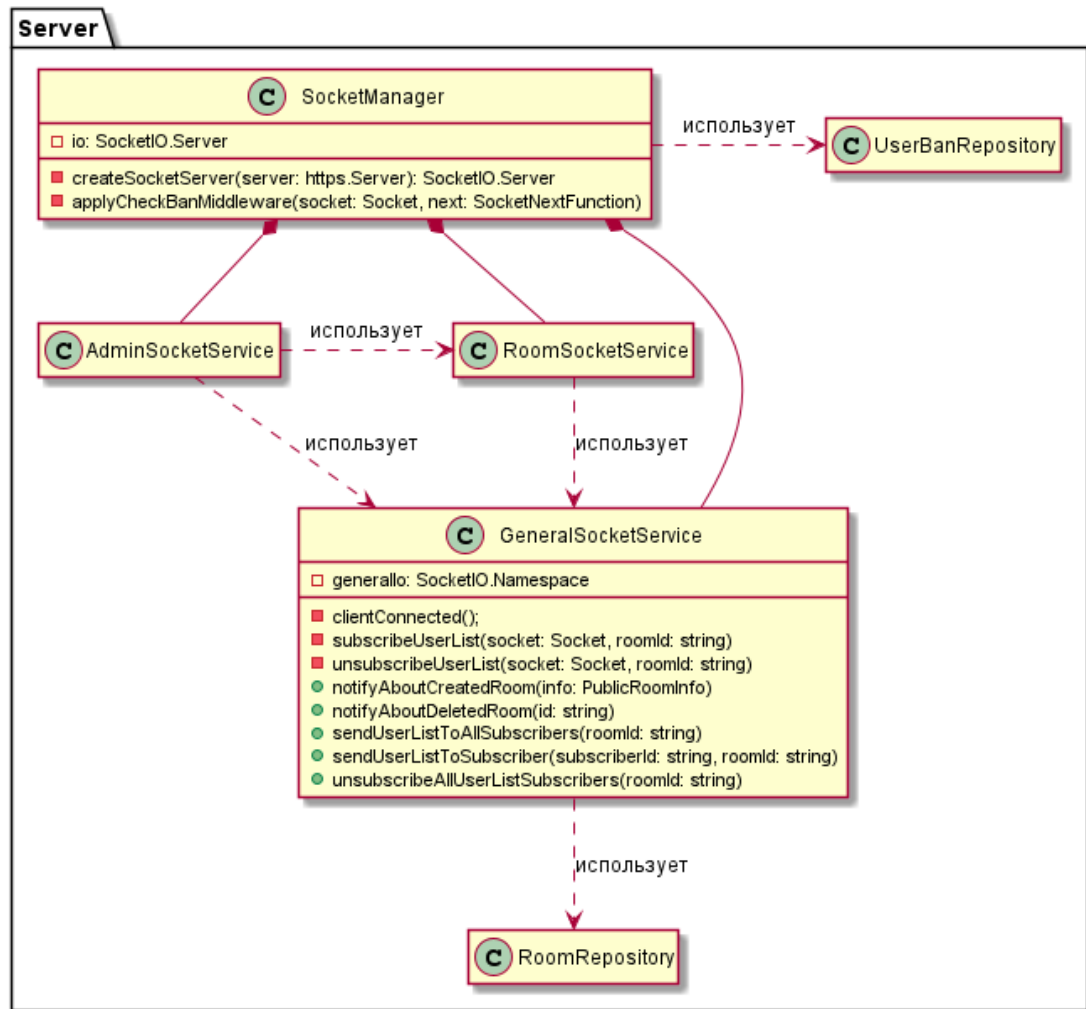


Рисунок П5.10 – Диаграмма классов серверной части – репозитории данных – аккаунты пользователей

Инв. № подл.	Подп. и дата
Взам. инв. №	Инв. № дубл.
Подп. и дата	
Инв. № подл.	

Изм.	Лист	№ документа	Подпись	Дата



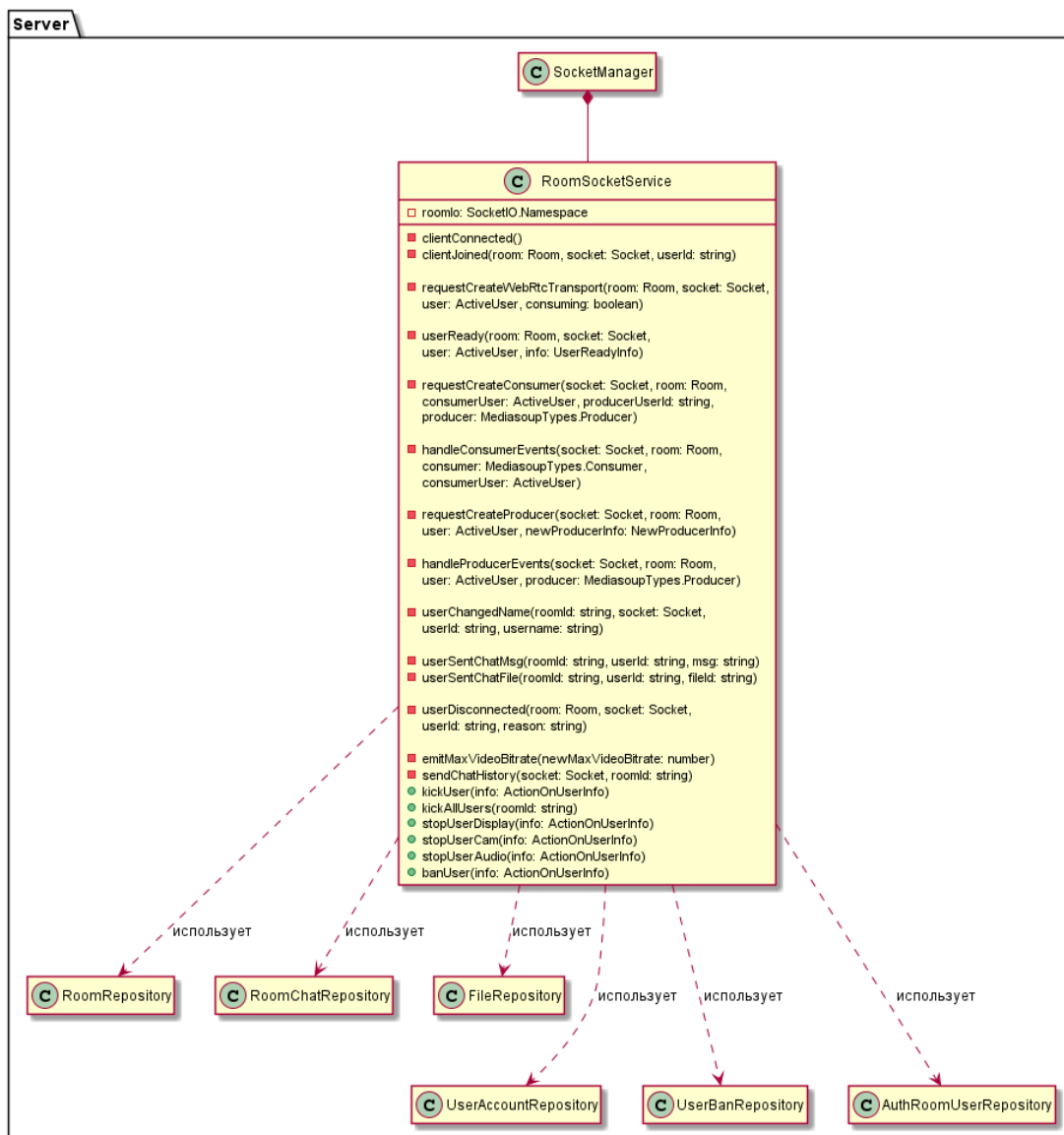


Рисунок П5.13 – Диаграмма классов серверной части – сокеты – часть 3

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата					
Изм.	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022				
					Лист				116

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата

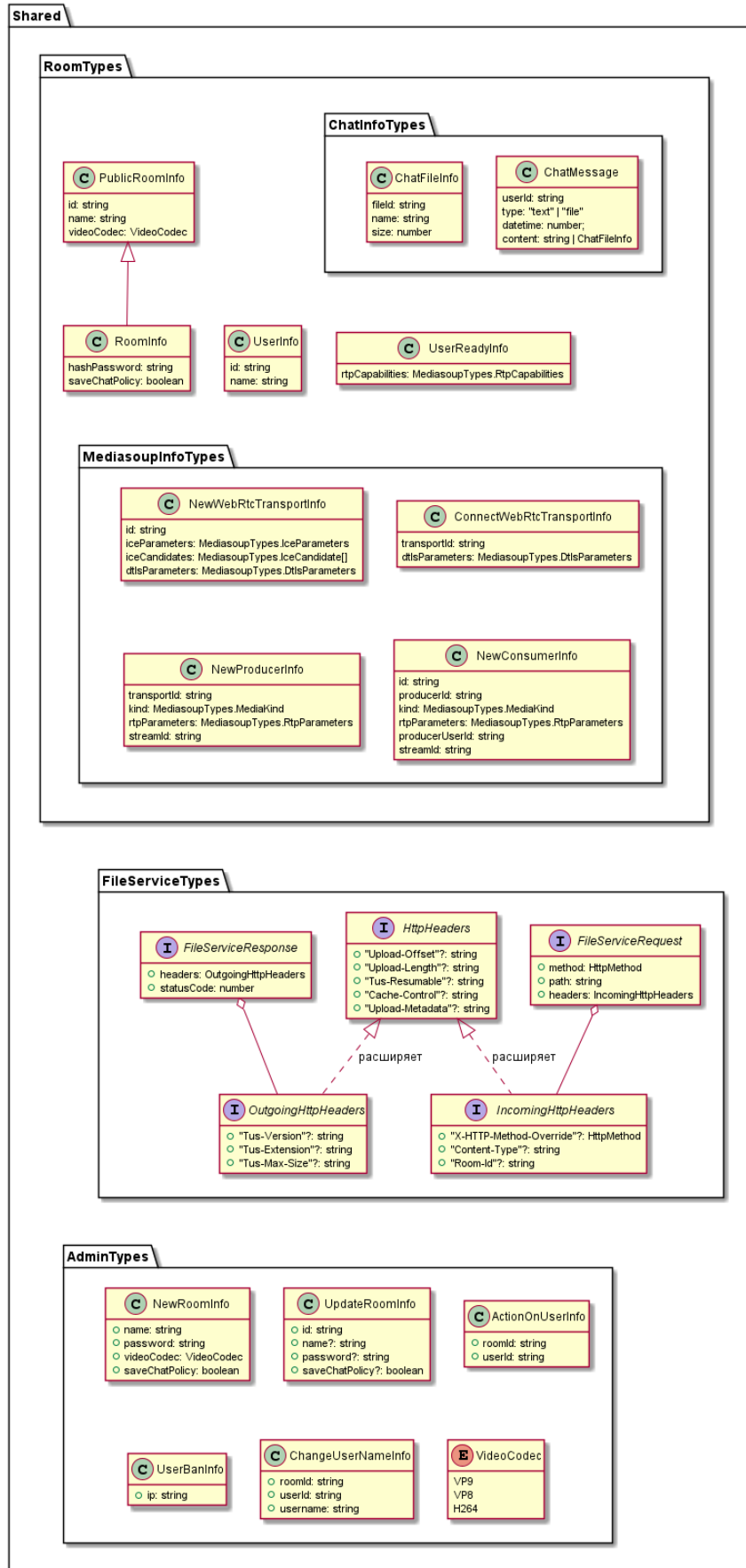


Рисунок П5.14 – Диаграмма классов общих типов

Изм	Лист	№ документа	Подпись	Дата

ВКРБ 09.03.04.059.2022

ПРИЛОЖЕНИЕ 6

ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ ЗАХВАТА МЕДИАПОТОКА

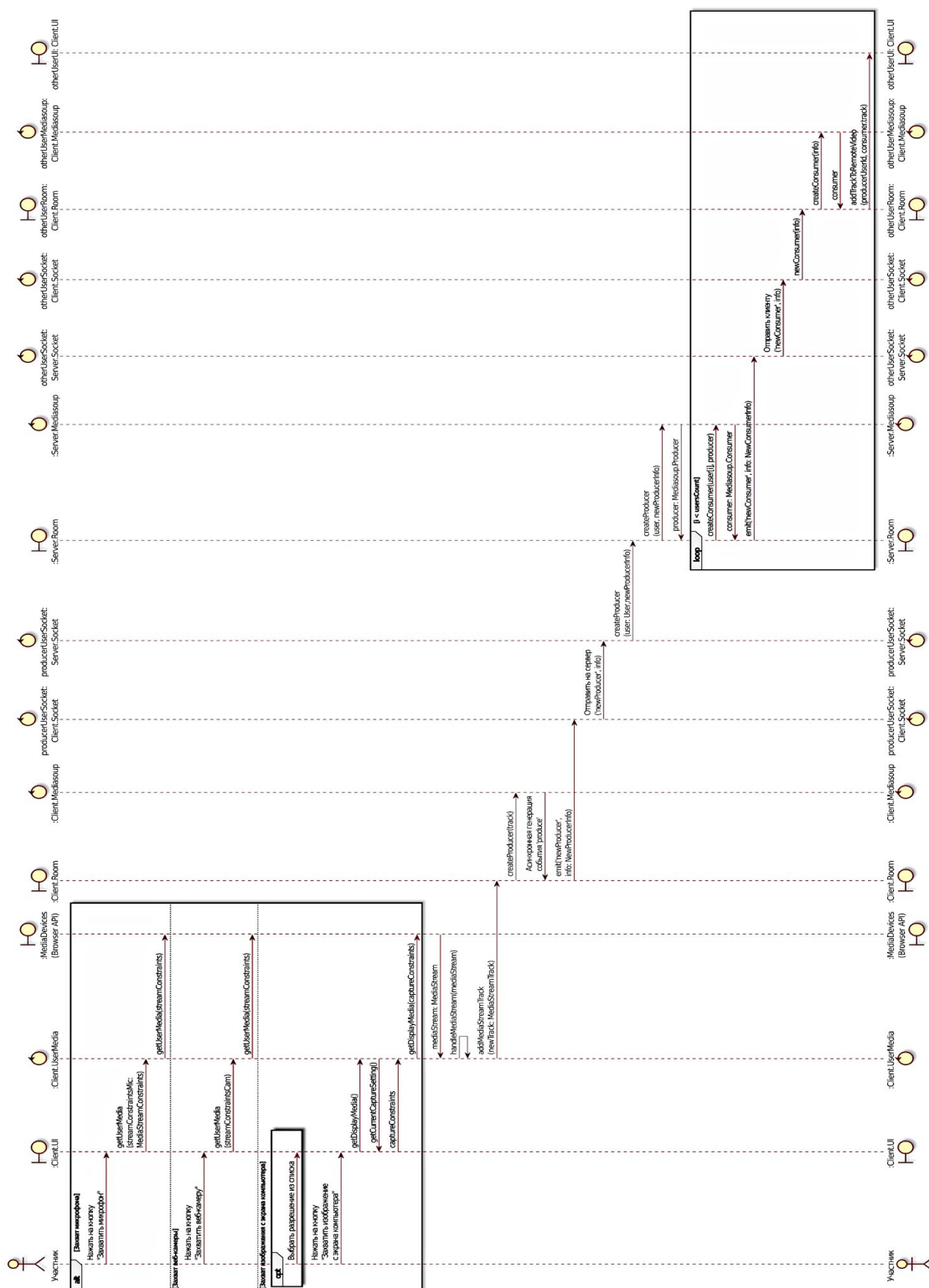
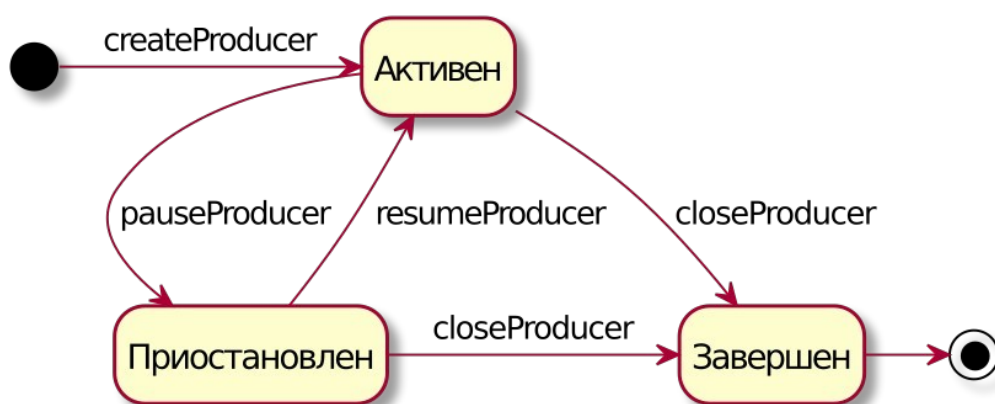


Рисунок П6 – Диаграмма последовательности захвата медиапотока



```
stateDiagram-v2
    [*] --> Приостановлен : createConsumer
    Приостановлен --> Активен : resumeConsumer
    Активен --> Приостановлен : pauseConsumer
    Активен --> Завершен : closeConsumer
    Приостановлен --> Завершен : closeConsumer
    Завершен --> [*]
```

Рисунок П7.2 – Диаграмма состояний медиапотока-потребителя на сервере

```

stateDiagram-v2
    [*] --> Активен : resumeConsumer
    Активен --> [*] : pauseConsumer
    Активен --> Завершен : closeConsumer
    [*] --> Завершен : closeConsumer
    Завершен --> [*]
  
```

ДИАГРАММЫ АКТИВНОСТЕЙ МЕДИАПОТОКА

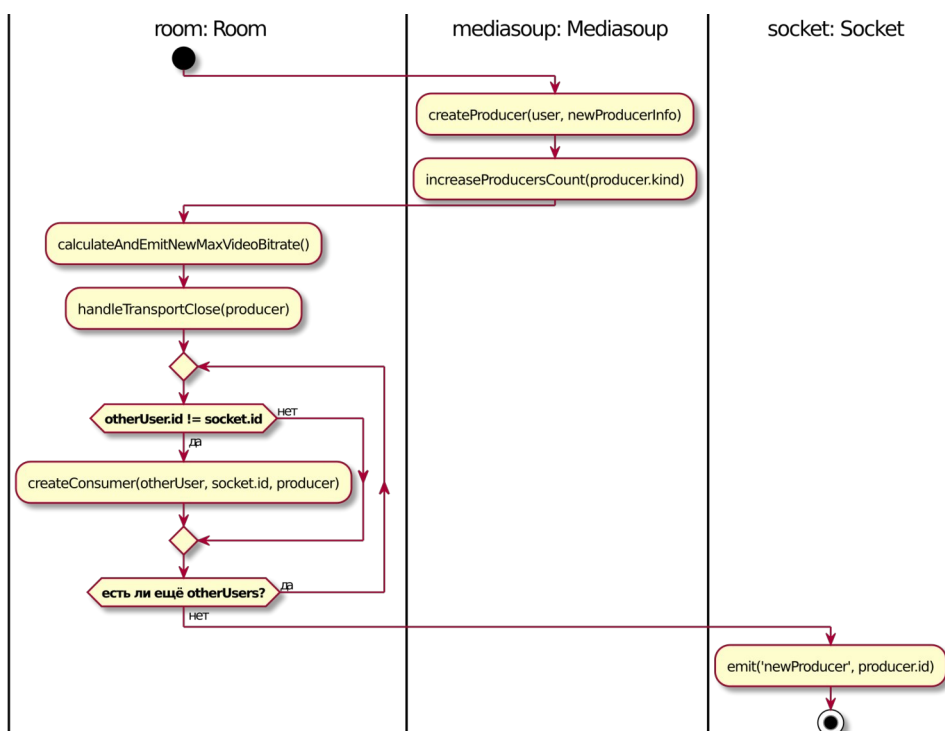


Рисунок П8.1 – Диаграмма активностей при создании медиапотока-производителя на сервере

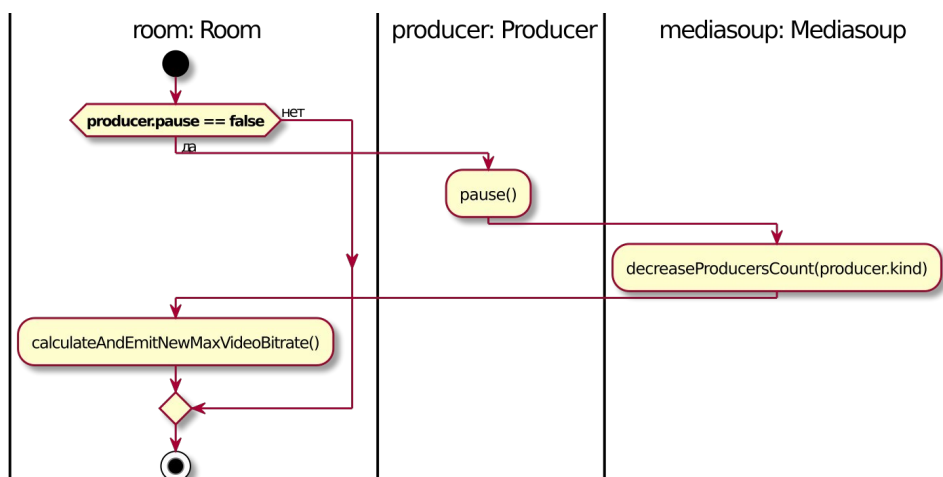


Рисунок П8.2 – Диаграмма активностей при приостановке медиапотока-производителя на сервере

Инов. № подл.	Подп. и дата
Взам. инв. №	Инв. № дубл
Подп. и дата	Подп. и дата

Изм	Лист	№ документа	Подпись	Дата
-----	------	-------------	---------	------

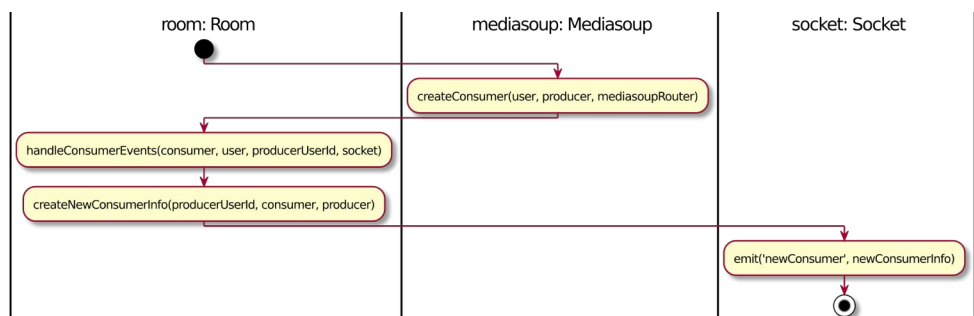


Рисунок П8.3 – Диаграмма активностей при создании медиапотока-потребителя на сервере

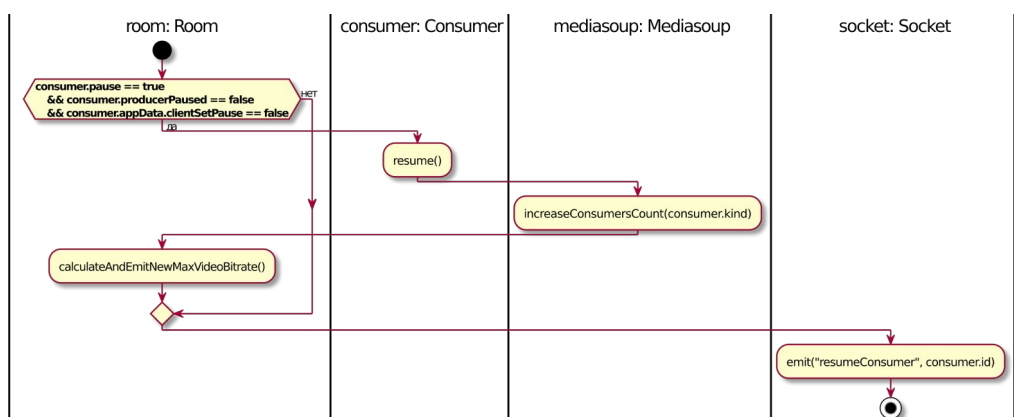
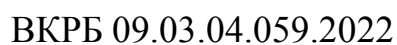


Рисунок П8.4 – Диаграмма активностей при снятии с паузы медиапотока-потребителя на сервере

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл	Подп. и дата
Изм	Лист	№ документа	Подпись	Дата
ВКРБ 09.03.04.059.2022				
Лист 121				

Изм	Лист	№ документа	Подпись	Дата



ПРИЛОЖЕНИЕ 10

СОДЕРЖИМОЕ ФАЙЛА PACKAGE.JSON

```
{
  "name": "nostromo",
  "version": "0.3.0",
  "description": "WebRTC SFU Server for Video conferencing on
Node.js",
  "main": "dist/index.js",
  "scripts": {
    "start": "node dist/index.js",
    "build": "tsc -p src/tsconfig.json",
    "prepare": "npm run build"
  },
  "repository": {
    "type": "git",
    "url": "git+https://gitlab.com/SgAkErRu/nostromo.git"
  },
  "author": "SgAkErRu",
  "license": "BSD-3-Clause",
  "dependencies": {
    "cookie": "^0.5.0",
    "dotenv": "10.0.0",
    "express": "4.17.2",
    "jose": "^4.8.0",
    "mediasoup": "3.9.6",
    "nanoid": "3.2.0",
    "nostromo-web": "git+https://git@gitlab.com/sgakerru/nostromo-web.git#dev",
    "socket.io": "4.4.1"
  },
  "devDependencies": {
    "@types/express": "4.17.13",
    "@types/node": "12.20.47",
```

Инв. № подл.	Подп. и дата				Инв. № дубл	Подп. и дата				Взам. инв. №	Инв. № дубл				Подп. и дата									
<pre> }, "author": "SgAkErRu", "license": "BSD-3-Clause", "dependencies": { "cookie": "^0.5.0", "dotenv": "10.0.0", "express": "4.17.2", "jose": "^4.8.0", "mediasoup": "3.9.6", "nanoid": "3.2.0", "nostromo-web": "git+https://git@gitlab.com/sgakerru/nostromo-web.git#dev", "socket.io": "4.4.1" }, "devDependencies": { "@types/express": "4.17.13", "@types/node": "12.20.47", } }, "scripts": { "start": "node index.js" }, "repository": { "type": "git", "url": "https://github.com/sgakerru/nostromo-web.git" }, "keywords": ["websocket", "chat", "realtime"], "author": "Sergey Akhmedov", "license": "MIT" }</pre>																								
Изм					Лист					№ документа					Подпись					Дата				
ВКРБ 09.03.04.059.2022															Лист									
															123									

```

    "@typescript-eslint/eslint-plugin": "5.17.0",
    "@typescript-eslint/parser": "5.17.0",
    "eslint": "8.12.0",
    "nostromo-shared": "git+https://git@gitlab.com/sgakerru/nostromo-shared.git#dev",
    "typescript": "4.6.3"
  },
  "homepage": "https://gitlab.com/SgAkErRu/nostromo#readme",
  "bundleDependencies": true
}

```

Инв. № подл.	Подп. и дата				Взам. инв №	Инв. № дубл	Подп. и дата	
Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022			Лист
								124

АКТ О ВНЕДРЕНИИ

Государственное бюджетное
учреждение здравоохранения
Астраханской области

«Александро-Мариинская
областная клиническая больница»

г. Астрахань, ул. Татищева, д.2

ОГРН 1053000627690
ИНН 3016045958
КПП 301901001

АКТ О ВНЕДРЕНИИ
Программного комплекса для организации и
проведения видеоконференций

Настоящий акт составлен в том, что в ГБУЗ АО АМОКБ внедрен программный комплекс для организации и проведения видеоконференций, разработанный студентом Астраханского государственного технического университета Катуниным С.С. в рамках его выпускной квалификационной работы (ВКР). Руководитель ВКР к.т.н, доцент Белов С.В.

Зам. кабинета телемедицины



Сокольская Е. Г.

Зам. главного врача по организационно-методической работе



Захаров Д. А.



«16» мая 2022 г.

Рисунок П11 – Акт о внедрении программного комплекса

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата	ВКРБ 09.03.04.059.2022					Лист
										125
Изм	Лист	№ документа	Подпись	Дата						

ПРИЛОЖЕНИЕ 12

ПРОГРАММНЫЙ ПРОДУКТ НА ОПТИЧЕСКОМ НОСИТЕЛЕ

Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл	Подп. и дата							
					Изм	Лист	№ документа	Подпись	Дата	ВКРБ 09.03.04.059.2022	Лист
											126