МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра системного анализа и автоматического управления

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ВИДЕО-АУДИО КОНФЕРЕНЦИЙ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 5 курса 551 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Ковтунова Александра Алексеевича

Научный руководитель
к. ф.-м. н., доцент _____ И. Е. Тананко

Заведующий кафедрой
к. ф.-м. н., доцент _____ И. Е. Тананко

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы.

Звонки стали неотъемлемой частью жизни человека за последние пол века. Без них невозможно представить современный мир. За последние 10 лет, вместе с популяризацией удаленной работы, сервисы звонков шагнули вперед, перейдя от использования классической телекомуникационной сети к использованию сети интернет. Часто технические компании разрабатывают свой аналог для привлечения пользователей к своей экосистеме или же для личных нужд.

Современные сервисы аудио-видео связи используются в образовательной сфере, корпоративной среде, игровых сообществах и социальных платформах. При этом существующие решения не всегда позволяют гибко адаптировать функциональность под требования конкретного проекта, а также могут иметь ограничения, связанные с производительностью, безопасностью или зависимостью от сторонних экосистем. В связи с этим разработка собственного сервиса аудио-видео звонков является актуальной задачей, поскольку позволяет изучить современные технологии передачи медиаданных в реальном времени и реализовать масштабируемое решение, отвечающее современным требованиям к качеству связи и удобству использования.

Цель бакалаврской работы – создание сервиса, с помощью которого можно совершать аудио-видео звонки.

Поставленная цель определила **следующие задачи**:

1. Описать назначение программного продукта.
2. Выделить основные сущности.
3. Спроектировать UML-диаграммы классов и диаграммы последовательностей.
4. Спроектировать архитектуру программного продукта.
5. Реализовать серверную часть приложения.
6. Реализовать клиентскую часть приложения.

Методологические основы разработки веб-приложений для видео- и аудиоконференций представлены в работах Э. Таненбаума и М. ван Стеена [1], С. Ньюмана [2], Р. Мартина [3], М. Клеппмана [4], В. Ф. Шаньгина [5], В. Г. Олифера и Н. А. Олифер [6], М. Фаулера [7] и Г. Буча [8].

Теоритическая значимость бакалаврской работы. Теоретическая значимость работы заключается в систематизации и анализе подходов к разработке

веб-приложений для видео- и аудиоконференций с использованием современных технологий распределённых систем, WebRTC-коммуникаций и микросервисной архитектуры. В работе рассматриваются методы организации обмена медиаданными в реальном времени, обеспечения масштабируемости, отказоустойчивости и безопасности веб-приложений, а также подходы к проектированию клиент-серверного взаимодействия в условиях высокой нагрузки.

Практическая значимость бакалаврской работы. Практическая значимость работы заключается в разработке веб-приложения для проведения видео- и аудиоконференций, обеспечивающего обмен медиаданными в реальном времени, управление пользователями и конференциями, а также безопасное взаимодействие между клиентской и серверной частями системы. Разработанное приложение может быть использовано в образовательной сфере, корпоративной среде и для организации удалённого взаимодействия пользователей.

Структура и объем работы. Бакалаврская работа состоит из введения и 2 разделов, заключения, списка использованных источников и 1 приложения. Общий объем работы – 47 страниц, из них 36 страниц – основное содержание, включая 25 рисунков, цифровой носитель в качестве приложения, список использованных источников информации – 26 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Проектирование» посвящён анализу предметной области, формированию требований к веб-сервису видео-аудио конференций и построению объектной модели системы в нотации UML. Разрабатываемый программный продукт предназначен для организации удалённых встреч с обменом аудио- и видеоданными, текстовым чатом и демонстрацией экрана в режиме реального времени.

В подразделе 1.1 определено назначение программного продукта и выделены группы пользователей (акторы): участник комнаты, организатор встречи и администратор встречи. Организатор создаёт комнату и обладает полным контролем над ней; администратор назначается организатором и имеет расширенные, но ограниченные права модерации. На основании сценариев использования построена диаграмма прецедентов (рисунок 1), отражающая регистрацию и авторизацию, управление профилем, восстановление пароля, создание и подключение к комнате, работу с медиаустройствами, чат, управление участниками, назначение ролей и завершение встречи.

Рисунок 1 – Диаграмма прецедентов

Подраздел 1.2 посвящён функциональным и нефункциональным требованиям. Сформулированы четырнадцать сценариев использования с детализацией ожидаемого поведения системы. Среди функциональных требований: регистрация по электронной почте с подтверждением кода; авторизация с временной блокировкой IP при трёх и более неудачных попытках входа; редактирование профиля (отображаемое имя, аватар, пароль); восстановление пароля по ссылке со сроком действия до шести часов; создание комнаты с названием, настройками доступа и опциональным паролем; подключение по ссылке с проверкой

прав; управление микрофоном и камерой; текстовый чат с фильтрацией сообщений; отображение участников и активного спикера; трансляция экрана; модерация участников (исключение, блокировка, управление потоками); назначение ролей; завершение встречи организатором. Нефункциональные требования задают подключение к комнате не более чем за три секунды, задержку передачи данных не свыше 500 мс, поддержку до ста активных участников в одной комнате, хранение паролей в хешированном виде (алгоритм Argon2), использование протоколов HTTPS и WSS, автоматическое переподключение при разрыве соединения и возможность горизонтального масштабирования серверов обработки видео.

Подраздел 1.3 описывает выделение основных сущностей предметной области методикой CRC-диаграмм (Class Responsibility Collaboration). Понятие виртуальная комната (Room) обозначает пространство встречи, объединяющее участников и хранящее параметры доступа и жизненный цикл сессии. Участник комнаты (Room Member) связывает учётную запись пользователя с конкретной комнатой и фиксирует роль и состояние подключения. Роль и права (Role, Permission) реализуют модель разграничения доступа RBAC (Role-Based Access Control): организатор, администратор и участник обладают различным набором разрешений. Медиа-сессия (Media Session) отвечает за аудио-, видеопотоки и трансляцию экрана. Дополнительно выделены сущности чат, сообщение, сервис аутентификации и сервис уведомлений; для каждой определены ответственность и объекты сотрудничества.

В подразделе 1.4 построены диаграммы взаимодействия (последовательностей) для ключевых сценариев: регистрация пользователя (взаимодействие интерфейса, сервиса аутентификации и сервиса уведомлений), авторизация, изменение профиля, восстановление пароля, создание и подключение к комнате с учётом альтернативных ветвей (несуществующая комната, отсутствие прав, ввод пароля), обмен сообщениями в чате, передача аудио-видео потока и трансляция экрана, назначение ролей организатором. Диаграммы последовательности в UML позволяют описать динамику обмена сообщениями между пользовательским интерфейсом, сервисами аутентификации, управления комнатами, чата и медиасервисом.

Подраздел 1.5 содержит UML-диаграмму классов (рисунок 2), объединяющую результаты CRC-анализа и диаграмм взаимодействия. На ней формали-

зованы атрибуты, методы и отношения между сущностями, что завершает этап структурного проектирования.

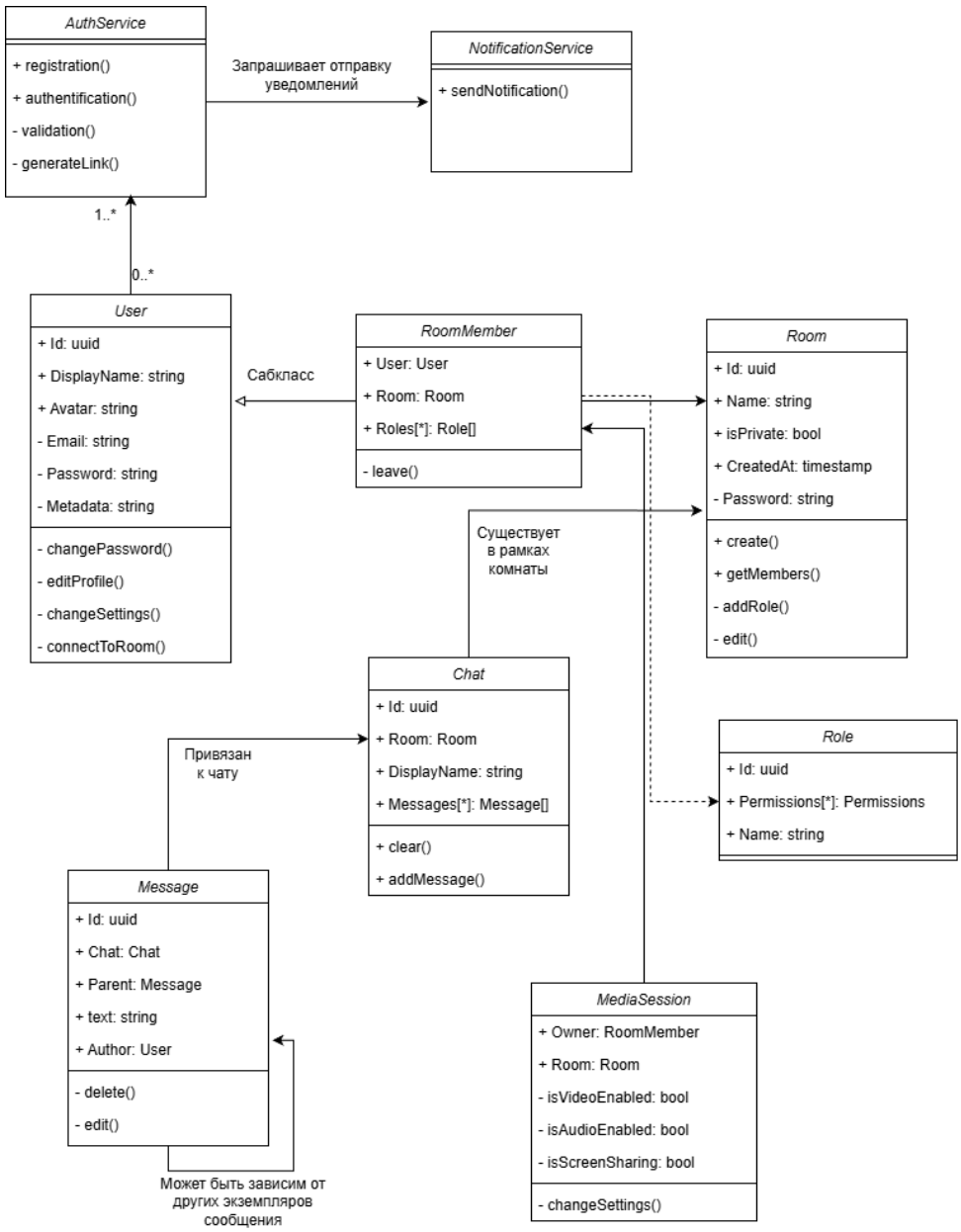

Рисунок 2 – Диаграмма классов

Теоретической основой раздела служат подходы инженерии требований и объектно-ориентированного анализа, описание клиент-серверных и распределённых систем, принципы безопасности веб-приложений (хеширование паролей, TLS, контроль доступа на основе ролей, ограничение частоты запросов), а также технологии передачи данных в реальном времени: протокол WebSocket для событий и чата и WebRTC для медиапотоков с привлечением выделенного медиа-сервера (SFU).

В результате проектирования определено назначение сервиса, сформу-

лированы функциональные и нефункциональные требования, построены CRC-карты, диаграммы взаимодействия и диаграмма классов. Полученная модель обеспечивает прослеживаемость от пользовательских сценариев к структуре данных и служит основой для выбора архитектуры и программной реализации, изложенных во втором разделе.

Второй раздел «Реализация» посвящён разработке архитектуры и программного кода веб-сервиса видео-аудио конференций. Основной вклад автора работы заключается в самостоятельной реализации серверной части (HTTP API и WebSocket), клиентского веб-приложения, интеграции с медиа-сервером и внедрении механизмов безопасности.

Подраздел 2.1 описывает архитектуру системы с использованием модели C4. На контекстном уровне система взаимодействует с участниками, организаторами и администраторами встреч, почтовым сервисом уведомлений и браузером пользователя. Контейнерная диаграмма (рисунок 3) включает веб-приложение на Vue.js, API-сервер и WebSocket-сервер на Rust, медиа-сервер LiveKit, базу данных PostgreSQL и внешний сервис отправки электронной почты. Компонентный уровень детализирует API-сервер (контроллеры аутентификации, комнат и участников; сервисы бизнес-логики; репозитории) и WebSocket-сервер (обработчик соединений, менеджер комнат, обработчик сообщений, менеджер медиа, эмиттер событий). Взаимодействие организовано по принципу разделения ответственности: контроллеры обрабатывают запросы, сервисы реализуют правила предметной области, репозитории абстрагируют доступ к данным.

В подразделе 2.2 обоснован выбор технологического стека. Для серверной части выбран язык Rust с фреймворком Actix-web и асинхронной средой Tokio в силу высокой производительности, безопасности памяти на этапе компиляции и эффективной обработки множества одновременных WebSocket-соединений. СУБД PostgreSQL обеспечивает ACID-транзакции, поддержку JSONB и масштабирование репликацией. Клиент реализован на Vue.js 3 с TypeScript, Pinia, Vue Router и Vite.

Подраздел 2.3 содержит описание реализованной серверной части. API-сервер обрабатывает регистрацию, вход, обновление JWT-токенов, восстановление пароля, управление профилем, создание и подключение к комнате, отправку и получение сообщений чата, модерацию участников и назначение ролей.

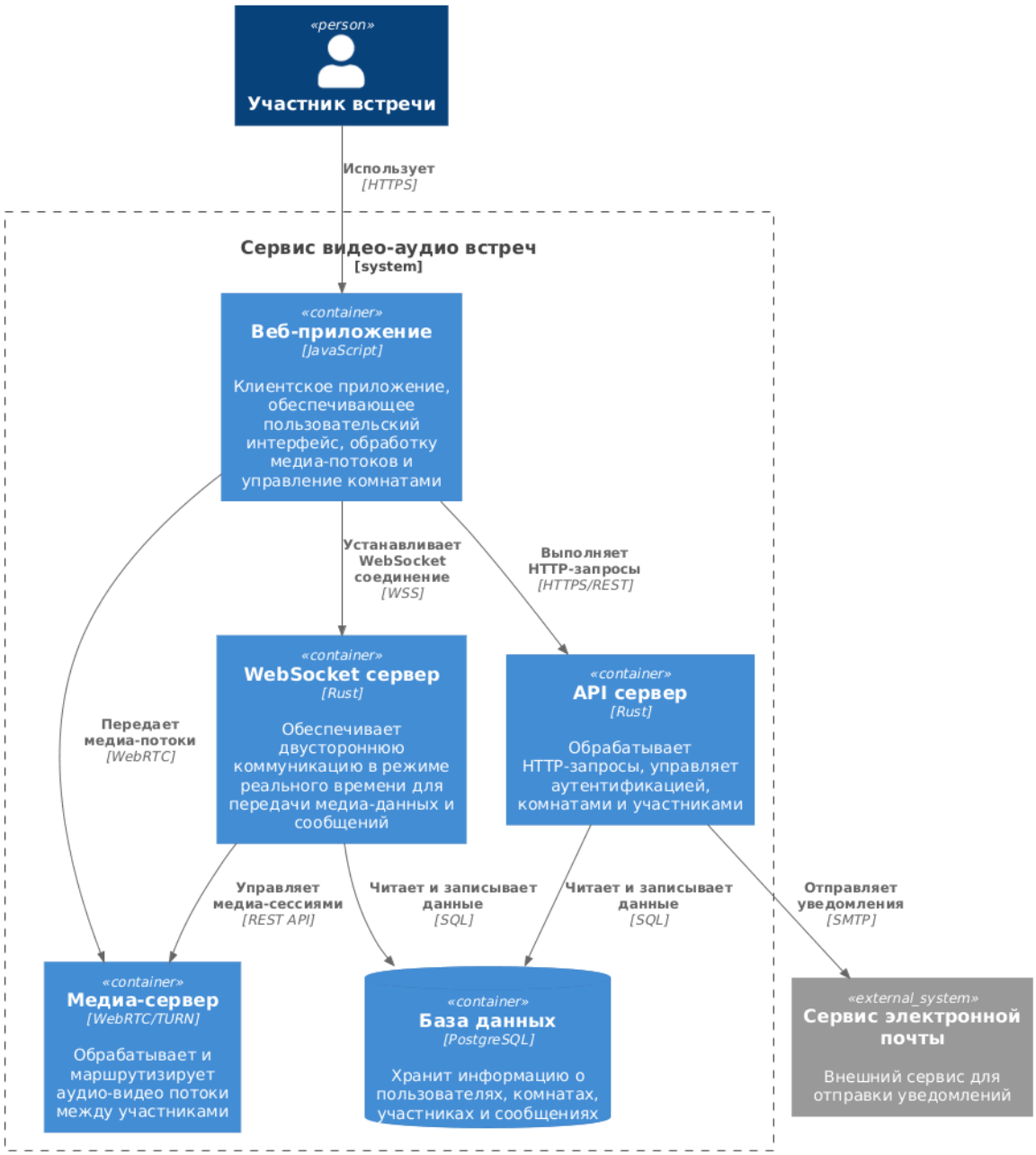

Рисунок 3 – Контейнерная диаграмма системы

Публичные маршруты /api/v1/auth доступны без аутентификации; остальные защищены middleware, верифицирующим Bearer-токен. Пароли хешируются алгоритмом Argon2id; при создании комнаты инициализируются роли, генерируется токен LiveKit; при подключении (join_room) проверяются пароль комнаты, бан и лимит участников. Модерационные действия проходят через сервис прав с проверкой разрешений и приоритета ролей. Для защиты от перебора паролей реализован ограничитель частоты запросов по IP и email. WebSocket-сервер доставляет в реальном времени события подключения участников, сообщения чата, изменения ролей и состояния медиапотоков.

Подраздел 2.4 описывает клиентское приложение. Маршрутизация разделяет гостевые и защищённые страницы; при загрузке приложения сессия восстанавливается по refresh-токену. Формы входа, регистрации и подключения к комнате валидируются с помощью vue-validate и zod. Store аутентификации (Pinia) сохраняет токены в localStorage, пароли на клиенте не хранятся. Страница встречи последовательно выполняет HTTP-запрос подключения к комнате, установку WebSocket-соединения, загрузку истории чата и подключение к LiveKit через композабл useMedia (рисунок 4). Компоненты сетки участников, панели управления медиа и чата отображают видеопотоки, элементы управления и переписку. HTTP-клиент автоматически обновляет access-токен при ответе 401; элементы интерфейса согласованы с правами роли пользователя в комнате, окончательная проверка выполняется на сервере. Требование использования HTTPS и WSS обеспечивается при развёртывании за обратным прокси с терминацией TLS; секреты задаются через переменные окружения.

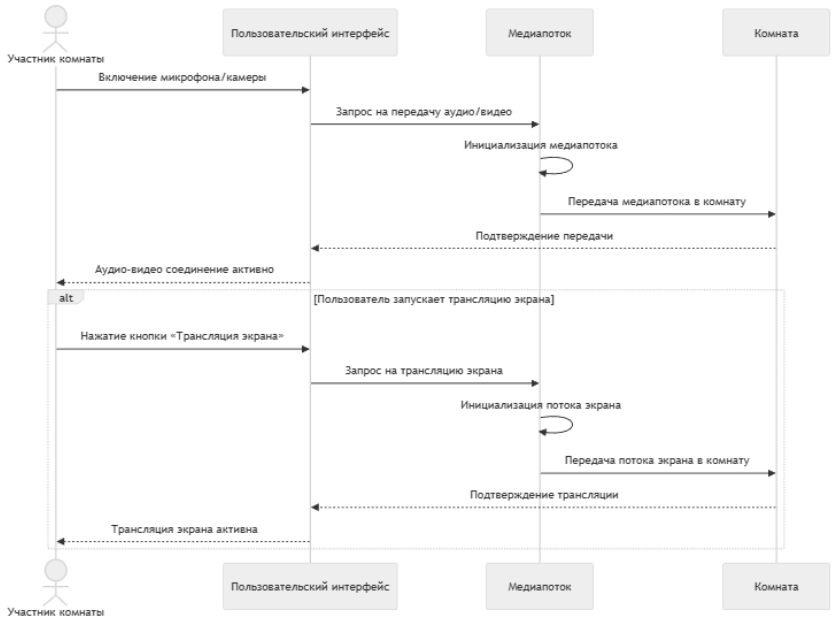

Рисунок 4 – Диаграмма взаимодействия при передаче аудио-видео потока

ЗАКЛЮЧЕНИЕ

В рамках данной работы был разработан веб-сервис для проведения аудио и видеовстреч. Для данного сервиса была разработана архитектура на основе фреймворка C4 обеспечивающая нормальное взаимодействие между модулями программы. Входе выполнения работы было изучено:

- Архитектурный фреймворк C4.
- Основные элементы языка Rust.
- Принцип работы P2P соединений.
- Фронтенд фреймворк Vue.

Основные источники информации:

1 *Эндрю, Т.* Распределённые системы. Принципы и парадигмы / Т. Эндрю, В. С. Маартен. — Питер, 2003. — 877 с.

2 *Сэм, Н.* Создание микросервисов. Проектирование масштабируемых систем / Н. Сэм. — Питер, 2022. — 624 с.

3 *Роберт, М.* Чистая архитектура. Искусство разработки программного обеспечения / М. Роберт. — Питер, 2018. — 352 с.

4 *Мартин, К.* Высоконагруженные приложения. Программирование, масштабирование, поддержка / К. Мартин. — Питер, 2018. — 640 с.

5 *Федорович, Ш. В.* Защита информации в компьютерных системах и сетях / Ш. В. Федорович. — ДМК Пресс, 2017. — 592 с.

6 *Григорьевич, О. В.* Компьютерные сети. Принципы, технологии, протоколы / О. В. Григорьевич, О. Н. Алексеевна. — Питер, 2020. — 1008 с.

7 *Мартин, Ф.* UML. Основы. Краткое руководство по стандартному языку объектного моделирования / Ф. Мартин. — Символ-Плюс, 2011. — 192 с.

8 *Гради, Б.* Объектно-ориентированный анализ и проектирование с примерами приложений / Б. Гради. — Вильямс, 2010. — 720 с.