МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»

(НГТУ)

Институт. Институт радиоэлектроники и информационных технологий .

Кафедра . Компьютерные технологии в проектировании и производстве .

**ОТЧЕТ**

**по прохождению** **производственной** **практики**

Направление подготовки/специальность: Информационные системы и технологии (09.03.02)

Образовательная программа: Распределенные информационные системы

Выполнил:

Студент гр. Шеварихин П.С.

*(группа) (подпись практиканта)*

Руководитель практики от кафедры

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Садков В.Д.

*(ученые звание и степень) (подпись)*

Отчет защищен с оценкой: \_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_20\_\_ г.

**Содержание**

Индивидуальное задание на производственную практику…………………….3

Совместный рабочий график (план)…………………………………………….6

Основная часть отчета……………………………………………………………7

1. Выбор темы ВКР………………………………………………………..7
   1. Описание предметной области…………………………………….7
   2. Актуальность выбранной темы……………………………………8
   3. Обоснование выбора C++ вместо Phyton/Qt……………………..9
   4. Выбор СУБД………………………………………………………11
   5. Перспективы ВКР…………………………………………………13
2. Анализ и Разработка Технического задания (ТЗ)…………………...15
   1. Описание предметной области…………………………………...15
   2. Цель и назначение…………………………………………………15
   3. Требования к защите информации……………………………….16
   4. Функциональные требования…………………………………….17
   5. Количественные требования………………………..……………18
   6. Требования к аппаратному и системному ПО…………………..19
   7. Сравнение аналоговых систем……………………………………19
   8. Архитектура………………………………………………………..20
   9. Тестирования……………………………………………………....20
   10. Требования к документации………………………………..21

Заключение………………………………………………………………………22

Список используемых источников……………………………………………..24

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ИМ. Р.Е. АЛЕКСЕЕВА»

(НГТУ)

# ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ

Студента гр. \_\_\_22-КТ\_\_\_\_\_\_ Ф.И.О.\_\_\_Шеварихин Павел Сергеевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Направление подготовки/специальность**: Информационные системы и технологии (09.03.02)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Образовательная программа**: Распределенные информационные системы\_\_\_\_\_\_\_\_\_\_\_\_\_

**Место прохождения практики** \_\_\_\_\_\_\_Кафедра “КТПП”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Время прохождения практики**

Дата начала практики «30» июня 2025 г.

Дата окончания практики «26» июля 2025 г.

**Тема индивидуального задания:**

Поиск темы выпускной квалификационной работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Содержание практики**

Во время прохождения практики студент обязан:

Ознакомиться: ГОСТ Р 56920-2016 «Требования к техническим заданиям на разработку ПО», ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы», правилами оформления ТЗ по ГОСТ 34.602-89.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Изучить: SIP, WebRTC, Opus, VP9/H.264, MTProto, асинхронные сокеты TCP/UDP, таймеры, работа с потоками.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнить следующие виды работ по приобретению практических навыков:

Собрать материал по теме индивидуального задания (выпускной квалификационной работы) для подготовки отчета по практике, выбрать тему ВКР, Составить ТЗ, включающее требования к серверу, клиенту и тестированию.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Должность на практике: практикант\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Планируемые результаты обучения при прохождении практики

|  |  |  |  |
| --- | --- | --- | --- |
| Планируемые результаты освоения образовательной программы | Планируемые результаты обучения при прохождении практики | | |
| **Код компетенции** | **Знать** | **Уметь** | **Владеть** |
| УК-1 | Нормативные требования к ТЗ (ГОСТ 34.602-89, ГОСТ Р 56920-2016), архитектуру VoIP-решений (WebRTC, SIP, кодеки Opus/VP9),  уязвимости мессенджеров (например, атаки на MTProto в Telegram) | Анализировать недостатки аналогов (VK Messenger, telegram), формулировать требования к безопасности и производительности | Навыками критического анализа и аргументации выбора технологий |
| УК-2 | Жизненный цикл ПО, методы оценки рисков | Планировать этапы разработки, рассчитывать ресурсы сервера | Методами проектного управления |
| УК-3 | Навыки работы в коллективе | Распределять задачи, проводить код-ревью | Git, командной разработкой |
| УК-4 | Правила оформления технической документации | Составлять отчеты по практике, защищать ТЗ перед руководителем | Навыками публичных выступлений, навыками деловой переписки и презентации идей |
| ОПК-2 | Методы декомпозиции, паттерны проектирования | Разбивать ТЗ на подпункты, оптимизировать задачи | UML-диаграммами, алгоритмами оценки сложности. |
| ОПК-8 | Современные исследования в области сетевых технологий и VoIP. | Применять научные данные для обоснования выбора технологий | Методами поиска и анализа научных источников. |
| ПК-2 | Методологию сравнения аналогов | Проводить сравнительный анализ, выделять ключевые параметры для ТЗ. | Навыками работы с инструментами тестирования JMeter, Selenium |

Результаты освоения обучающимся компетенций при прохождении практики оцениваются по итогам защиты отчета по прохождению практики, с учетом выполнения индивидуального задания и отзыва (характеристики) о прохождении практики на предприятии.

Руководитель практики от кафедры

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_Садков В.Д.

*(ученые звание и степень) (подпись)*

Задание на практику получил:

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_ Шеварихин П.С.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г.

**СОВМЕСТНЫЙ РАБОЧИЙ ГРАФИК (ПЛАН)**

**ПРОВЕДЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ**

Студента гр. \_\_\_\_\_\_22-КТ\_\_\_\_\_\_\_ Ф.И.О.\_\_\_\_\_\_\_\_\_Шеварихин П.С\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| №№ п/п | Разделы (этапы) практики | Сроки выполнения  с \_\_\_\_\_ по\_\_\_\_\_\_ | Отметка о выполнении (подпись руководителя практики\*) |
| 1. | **Подготовительный (организационный) этап** |  |  |
| 1.1. | Проведение собрания студентов; получение индивидуального задания и путевки на практику |  |  |
| 1.2 | Оформление пропуска на предприятие |  |  |
| 1.3 | Прохождение инструктажа по технике безопасности |  |  |
| 2. | **Производственный этап** |  |  |
| 2.1 | Знакомство со структурой предприятия, его подразделениями, цехами, отделами |  |  |
| 2.2 | Знакомство с научно-исследовательской деятельностью предприятия. |  |  |
| 2.3 | Знакомство с организацией производственных и технологических процессов |  |  |
| 2.4 | Знакомство с работой группы BI |  |  |
| 2.5. | Приобретение навыков работы в должности консультанта |  |  |
| 2.6. | Выполнение индивидуального задания: |  |  |
|  | 1. Выбор темы ВКР |  |  |
|  | 2. Разработка технического задания (ТЗ) |  |  |
|  | 3.Планирование работ по реализации |  |  |
| 3. | Заключительный этап |  |  |
| 3.1 | Анализ и обобщение полученной информации |  |  |
| 3.2 | Написание отчета по практике |  |  |

\* На этапах 1.1, 3.1, 3.2 отметку о выполнении ставит руководитель практики от кафедры, на этапах 1.2, 1.3, 2 – руководитель практики от предприятия.

Руководитель практики от кафедры

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_Садков В.Д.

*(ученые звание и степень) (подпись)*

**Основная часть отчета**

1. **Выбор темы ВКР**

**1.1 Описание предметной области**

В эпоху цифровой трансформации коммуникационные технологии становятся критически важным элементом как личного, так и профессионального взаимодействия. Современные мессенджеры перестали быть просто инструментами для обмена сообщениями, превратившись в комплексные платформы, сочетающие текстовую, голосовую и видеосвязь с повышенными требованиями к безопасности и удобству использования.

Мессенджер на Qt/C++ представляет собой кроссплатформенное решение, предназначенное для обеспечения защищенной коммуникации между пользователями. В отличие от популярных централизованных сервисов, данная разработка предлагает гибкую архитектуру, позволяющую работать как через промежуточные серверы, так и в полностью децентрализованном P2P-режиме. Особое внимание уделяется вопросам конфиденциальности - все передаваемые данные защищаются сквозным шифрованием с использованием современных криптографических алгоритмов.

Разработка мессенджера требует комплексного подхода, объединяющего:

* Сетевые технологии для обеспечения стабильного соединения
* Криптографические методы защиты данных
* Механизмы синхронизации между устройствами
* Оптимизацию передачи мультимедийного контента
* Создание интуитивно понятного пользовательского интерфейса

**1.2 Актуальность выбранной темы**

Разработка мессенджера на Qt/C++ является чрезвычайно актуальной по следующим причинам:

Потребность в приватности: В условиях роста киберугроз и массового сбора данных пользователи все чаще ищут альтернативные решения с гарантией конфиденциальности. Сквозное шифрование становится обязательным требованием, а не дополнительной опцией.

Кроссплатформенность: Современные пользователи работают на устройствах с разными операционными системами. Решение на Qt позволяет обеспечить единый пользовательский опыт на Windows, Linux и macOS без потери функциональности.

Производительность: C++ в сочетании с Qt Framework обеспечивает высокую эффективность работы при минимальном потреблении ресурсов, что особенно важно для мобильных устройств и компьютеров со скромными характеристиками.

Гибкость архитектуры: Возможность работы в различных режимах (клиент-серверный, P2P) делает решение универсальным для разных сценариев использования - от корпоративной коммуникации до личного общения.

Открытость технологий: В отличие от проприетарных решений, открытая архитектура позволяет проводить независимый аудит безопасности и адаптировать систему под конкретные требования.

Таким образом, разработка мессенджера на Qt/C++ сочетает в себе техническую новизну, практическую значимость и соответствует современным тенденциям в области защищенных коммуникаций. На основании анализа предметной области и её актуальности была выбрана следующая тема для выпускной квалификационной работы: "Разработка кроссплатформенного мессенджера с защищенной коммуникацией на базе Qt/C++".

**1.3. Обоснование выбора C++ вместо Python/Qt**

Язык C++ был выбран для разработки мессенджера по следующим ключевым причинам:

1. Предыдущий опыт и уверенное владение

В течение 3 лет обучения я активно использовал C++ в курсовых проектах, включая:

* Разработку многопоточных приложений (std::thread, std::async).
* Работу с сетевыми библиотеками (Boost.Asio, QtNetwork).
* Оптимизацию производительности (аллокации памяти, SIMD-инструкции).

Это позволяет сосредоточиться на архитектуре мессенджера, а не на изучении нового языка.

2. Производительность и низкоуровневый контроль

Для мессенджера с поддержкой голосовой связи и шифрования критичны:

* Минимальные задержки: C++ исключает накладные расходы интерпретатора (в отличие от Python).
* Эффективная работа с сетью: Прямое управление сокетами через QtNetwork или Boost.Asio.
* Обработка аудио: Использование нативных библиотек (WebRTC, Opus) без оберток.
* Безопасность: Ручное управление памятью исключает риски утечек (актуально для E2E-шифрования).

3. Совместимость с необходимыми технологиями

Qt Framework: Нативная поддержка C++ (QML для интерфейса, QtNetwork для коммуникации).

* Криптография: Интеграция с OpenSSL (нативный C API).
* Мультимедиа: Библиотеки FFmpeg (C/C++) для кодирования голоса/видео.
* Кроссплатформенность: Сборка под Windows, Linux, macOS без зависимостей от интерпретатора.

1. Ограничения Python для данной задачи

* Зависимость от оберток: Например, PyQt — это лишь binding к C++-библиотеке, что усложняет отладку.
* Высокие задержки: GIL (Global Interpreter Lock) осложняет многопоточную обработку аудио.
* Проблемы с производительностью: Даже с Cython сложно достичь скорости нативного C++.
* Слабая поддержка VoIP: Библиотеки вроде pyaudio уступают WebRTC по функционалу.

5. Карьерные перспективы

C++ востребован в:

* Разработке высоконагруженных систем (мессенджеры, игровые сервера).
* Embedded- и IoT-устройствах (где Python неприменим).
* Опыт работы с Qt/C++ повышает конкурентоспособность в компаниях типа Telegram, Discord, JetBrains.

Разработка современного мессенджера, поддерживающего текстовую и голосовую связь в реальном времени, предъявляет строгие требования к производительности, стабильности работы и эффективности использования системных ресурсов. В данном контексте язык программирования C++ демонстрирует неоспоримые преимущества перед другими вариантами, в частности перед Python, что делает его оптимальным выбором для реализации данного проекта.

Таким образом, C++ был выбран как наиболее подходящий язык для реализации мессенджера благодаря уникальной комбинации производительности, контроля над системными ресурсами, богатой экосистеме специализированных библиотек и долгосрочной актуальности на рынке разработки программного обеспечения. Этот выбор позволяет создать решение, которое не только соответствует современным требованиям к функциональности и производительности, но и обладает потенциалом для дальнейшего развития и масштабирования.

* 1. **Выбор СУБД**

При разработке мессенджера, выбор системы управления базами данных играет критическую роль. Одним из наиболее подходящих вариантов для таких систем является Microsoft SQL Server, который обладает несколькими значимыми преимуществами.

― Надежность и безопасность. SQL Server обеспечивает высокий уровень безопасности данных благодаря механизмам авторизации, контроля доступа и шифрования. Это гарантирует защиту данных от несанкционированного доступа и обеспечивает резервное копирование и восстановление данных, что критически важно для сохранности информации.

― Высокая производительность. SQL Server отличается оптимизированным движком запросов, который обеспечивает быстрые и эффективные операции с базой данных. Возможности оптимизации запросов, такие как индексы и представления, дополнительно улучшают производительность запросов к базе данных. Это особенно важно для систем, которые обрабатывают данные и требуют высокоскоростного доступа к ним.

― Простота в использовании. SQL Server имеет пользовательский интерфейс с интуитивно понятным дизайном баз данных, что делает его простым в использовании даже для начинающих разработчиков.

― Широкая поддержка. SQL Server является одной из самых популярных систем управления базами данных, что обеспечивает широкую поддержку со стороны сообщества разработчиков и доступ к большому количеству документации и ресурсам. Это существенно упрощает процесс разработки и поддержки системы.

― Интеграция с другими продуктами Microsoft. SQL Server отлично интегрируется с другими продуктами Microsoft, такими как Azure, SharePoint и Excel. Это упрощает работу с данными и их обмен между различными приложениями, что особенно полезно для бизнес-процессов, требующих взаимодействия между различными системами.

― Масштабирование и отказоустойчивость. SQL Server позволяет масштабировать систему в зависимости от потребностей, обеспечивая высокую производительность даже при одновременном подключении большого количества пользователей. Технологии кластеризации и автоматического создания контрольных точек обеспечивают отказоустойчивость и возможность быстрого восстановления данных в случае сбоя.

Выбор SQL Server как СУБД для мессенджера обеспечит надежную, производительную и безопасную базу данных, которая будет легко интегрироваться с другими системами и поддерживаться широким сообществом разработчиков.

**1.5. Перспективы ВКР**

Разрабатываемый мессенджер на Qt/C++ имеет значительные перспективы как с точки зрения дальнейшего развития проекта, так и с позиции его практического применения. Реализованные в ходе работы решения открывают несколько направлений для совершенствования и внедрения.

**Перспективы научного развития:**

1. Углубленное исследование гибридной архитектуры сети может быть продолжено в следующих направлениях:

* Разработка адаптивного алгоритма выбора протокола передачи данных в зависимости от качества сети
* Исследование методов компенсации потерь пакетов при использовании UDP
* Создание математической модели для прогнозирования задержек в различных сетевых конфигурациях

1. Оптимизация работы с памятью может быть расширена:

* Разработка специализированных аллокаторов для разных компонентов системы
* Исследование влияния различных стратегий управления памятью на производительность
* Создание универсального решения для работы с памятью в реальном времени

1. Исследования в области безопасности могут включать:

* Разработку новых методов защиты от атак типа "человек посередине"
* Исследование постквантовых алгоритмов шифрования
* Создание системы верификации участников коммуникации

**Перспективы практического применения:**

1. Коммерциализация проекта:

* Создание корпоративной версии с поддержкой серверной инфраструктуры
* Разработка мобильных клиентов на основе Qt для Android/iOS
* Интеграция с существующими VoIP-решениями

1. Развитие экосистемы:

* Создание SDK для сторонних разработчиков
* Разработка системы плагинов для расширения функционала
* Поддержка дополнительных кодеков и форматов данных

1. Оптимизация производительности:

* Портация критичных участков кода на SIMD-инструкции
* Разработка специализированных версий для встраиваемых систем
* Оптимизация энергопотребления на мобильных платформах

**Перспективы развития технологий:**

1. Интеграция с новыми технологиями:

* Поддержка WebAssembly для веб-версии клиента
* Использование машинного обучения для улучшения качества звука
* Реализация функций дополненной реальности в видеочате

1. Масштабирование архитектуры:

* Разработка кластерной версии серверной части
* Реализация механизмов балансировки нагрузки
* Создание системы геораспределенных серверов

1. Исследовательские направления:

* Анализ поведения системы в условиях нестабильных сетей
* Исследование методов снижения энергопотребления
* Разработка новых подходов к синхронизации мультимедийных потоков

1. Анализ и Разработка Технического задания (ТЗ)
   1. **Описание предметной области**

Разрабатываемый мессенджер на Qt/C++ представляет собой современное коммуникационное решение, предназначенное для безопасного обмена сообщениями и организации голосовой связи между пользователями. Основная цель разработки - создание высокопроизводительного и защищенного приложения с минимальными задержками передачи данных, работающего на различных платформах.

**2.2 Цель и назначение**

В основу разработки мессенджера на Qt/C++ легла цель создания современного, безопасного и производительного решения для коммуникации, которое объединяет в себе следующие ключевые аспекты:

― реализация защищенного обмена сообщениями с использованием современных алгоритмов шифрования, обеспечивающих конфиденциальность переписки;

― создание системы голосовой связи с минимальной задержкой передачи данных, что особенно важно для комфортного общения в реальном времени;

― разработка кроссплатформенного решения, работающего под различными операционными системами без потери функциональности;

― оптимизация сетевого взаимодействия для обеспечения стабильной работы даже при нестабильном интернет-соединении;

― внедрение механизмов синхронизации истории сообщений между различными устройствами пользователя.

Создание данного мессенджера позволит:

― обеспечить пользователей современным и безопасным средством коммуникации;

― предложить альтернативу существующим мессенджерам с открытым исходным кодом;

― продемонстрировать возможности Qt Framework для создания сложных сетевых приложений;

― реализовать инновационные подходы к организации защищенной передачи данных.

Особое внимание уделяется следующим аспектам:

― интеграции различных протоколов передачи данных для обеспечения максимальной надежности соединения;

― созданию интуитивно понятного пользовательского интерфейса, адаптируемого под разные платформы;

― реализации системы уведомлений, работающей во всех популярных операционных системах;

― разработке гибкой системы настроек и персонализации;

― обеспечению совместимости с различными системами аутентификации пользователей.

**2.3 Требования к защите информации**

Мессенджер на Qt/C++ должен обеспечивать комплексную защиту передаваемых и хранимых данных. Для этого реализуются следующие меры безопасности:

Шифрование передаваемых данных. Все сообщения и голосовые данные должны шифроваться с использованием алгоритмов AES-256 или XChaCha20-Poly1305. Ключи шифрования генерируются для каждого сеанса связи индивидуально.

Защита от несанкционированного доступа. Реализация системы аутентификации с обязательной двухфакторной проверкой. Использование механизма одноразовых паролей для критически важных операций.

Безопасное хранение данных. Локальная база сообщений на устройствах пользователей должна быть зашифрована. Пароли хранятся только в хэшированном виде с использованием современных алгоритмов хэширования.

Защита от сетевых атак. Реализация механизмов защиты от MITM-атак, перехвата сессии, подмены пакетов. Использование подписей для всех передаваемых пакетов данных.

Контроль целостности системы. Регулярное обновление криптографических библиотек. Мониторинг уязвимостей в используемых компонентах. Автоматическая проверка целостности исполняемых файлов.

Резервное копирование. Возможность создания зашифрованных резервных копий истории сообщений с последующим хранением в защищенном облачном хранилище.

**2.4 Функциональные требования**

Обмен сообщениями:

* Отправка и получение текстовых сообщений
* Поддержка форматированного текста
* Передача файлов любого типа
* Возможность цитирования сообщений
* Редактирование отправленных сообщений

Голосовая связь:

* Осуществление голосовых вызовов
* Поддержка групповых звонков
* Регулировка громкости
* Возможность отключения микрофона

Управление контактами:

* Добавление/удаление контактов
* Создание групп и каналов
* Категоризация контактов
* Поиск по списку контактов

Настройки и безопасность:

* Управление параметрами приватности
* Настройка уведомлений
* Управление историей сообщений
* Настройки внешнего вида

Дополнительные функции:

* Синхронизация между устройствами
* Поиск по истории сообщений
* Возможность блокировки пользователей
* Система статусов и присутствия

**2.5 Количественные требования**

Производительность системы:

* Время доставки сообщений: не более 500 мс
* Задержка голосовой связи: не более 150 мс
* Время отклика интерфейса: не более 100 мс
* Время запуска приложения: не более 2 секунд

Масштабируемость:

* Максимальное количество контактов: не ограничено
* Максимальный размер группы: 500 участников
* Максимальный размер передаваемого файла: 2 ГБ
* Минимальная скорость соединения: 64 кбит/с

Хранение данных:

* Объем локальной базы сообщений: до 50 ГБ
* Срок хранения истории: настраиваемый
* Частота автоматического резервного копирования: ежедневно

**2.6 Требования к аппаратному и системному ПО**

Основные требования к аппаратному обеспечению:

― Процессор: Intel Core i5.

― Оперативная память: минимум 8 ГБ.

― Диск: минимум 32 ГБ.

― Сетевое подключение.

Основные требования к программному обеспечению:

― Операционная система Windows 10/11.

**2.7 Сравнение аналоговых систем**

|  |  |  |  |
| --- | --- | --- | --- |
| Система | Задержка текста | Задержка голоса | Особенности |
| Telegram | 100-300 мс | 200-400 мс | Централизованные серверы, MTProto |
| Signal | 150-250 мс | 180-350 мс | E2E-шифрование, открытый исходных код |
| WhatsApp | 100-200 мс | 150-300 мс | Проприетарное шифрование |
| Matrix | 200-500 мс | 350-500 мс | Децентрализованная архитектура |

Таблица 1 – Сравнение с существующими мессенджерами

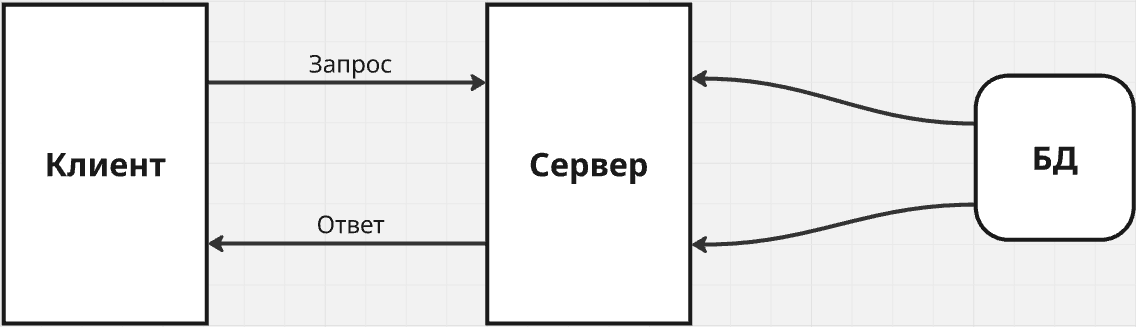
**2.8 Архитектура**

Рисунок 1 – клиент-серверная архитектура

* 1. **Тестирование**

**Методики тестирования**

* Функциональное тестирование
* Нагрузочное тестирование
* Тестирование качества звука
* Кросс-платформенное тестирование

**Ожидаемые результаты**

Производительность:

* Задержка передачи сообщений: <100 мс
* Задержка голосовой связи: <150 мс
* Время отклика интерфейса: <50 мс

Ресурсоемкость:

* CPU-нагрузка сервера: <50% при пиковой нагрузке
* Потребление RAM клиентом: <500 МБ
* Потребление трафика голосового звонка: <64 кбит/с на канал

Качество связи:

* MOS-оценка качества голоса: ≥4.0
* Уровень потерь пакетов: <1%
* Время восстановления соединения: <2 сек
  1. **Требования к документации**

**Пояснительная записка:**

Описание архитектуры

Листинг кода

Листинг классов

Графики тестирования

**Руководство пользователя:**

Скриншоты интерфейса

Пошаговая инструкция по использованию

Пошаговая инструкция по настройке

**Заключение**

Работа над мессенджером на Qt/C++ стала для меня важным этапом профессионального становления, позволившим соединить теоретические знания с практическими навыками разработки сложных сетевых приложений. В процессе проектирования системы я глубоко изучил архитектурные особенности современных мессенджеров, что дало возможность осознанно подходить к выбору технологических решений на каждом этапе разработки. Особое внимание я уделил анализу механизмов обеспечения безопасности данных, что потребовало детального изучения криптографических алгоритмов и принципов сетевой безопасности.

Разработка технического задания стала для меня ценным опытом системного подхода к проектированию программного обеспечения. Мне удалось не только сформулировать четкие функциональные требования, но и научиться оценивать технические решения с точки зрения их производительности, надежности и масштабируемости. Работа над проектом потребовала углубленного изучения возможностей Qt Framework, особенно в части сетевого программирования и работы с мультимедиа, что значительно расширило мой профессиональный инструментарий как C++-разработчика.

Особую ценность представляет приобретенный опыт кросс-платформенной разработки, где приходилось учитывать особенности различных операционных систем и находить оптимальные решения для обеспечения одинаково качественной работы приложения на всех платформах. Взаимодействие с научным руководителем помогло мне развить навыки профессиональной аргументации технических решений и научиться критически оценивать собственные проектные предложения.

Работа над мессенджером позволила мне не только освоить новые технологии, но и понять важность комплексного подхода к созданию программных продуктов, где удобство пользователя, производительность и безопасность одинаково важны. Полученные знания и опыт создают прочную основу для моей дальнейшей профессиональной деятельности в области разработки сетевых приложений и систем реального времени. Этот проект стал важной ступенью в моем становлении как инженера-программиста, научив видеть не только техническую, но и пользовательскую сторону разработки программного обеспечения.

**Список использованных источников**

1. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы. — М.: Стандартинформ, 2020.
2. ГОСТ Р 56920-2016. Требования к техническим заданиям на разработку ПО. — М.: Росстандарт, 2016.
3. RFC 3550. RTP: A Transport Protocol for Real-Time Applications. IETF, 2003.
4. RFC 7587. RTP Payload Format for the Opus Codec. IETF, 2015.
5. Мейер Б. Объектно-ориентированное конструирование программных систем. — М.: ДМК Пресс, 2020. — 560 с.
6. Таненбаум Э., Вудхалл А. Современные операционные системы. — СПб.: Питер, 2022. — 1120 с.
7. Шлее М. \*Qt 6. Разработка приложений на C++\*. — СПб.: БХВ-Петербург, 2023. — 784 с.
8. Джосаттис Н. Стандартная библиотека C++: справочное руководство. — М.: Вильямс, 2021. — 1136 с.
9. Коплиен Д. \*C++17 STL. Стандартная библиотека шаблонов\*. — М.: ДМК Пресс, 2022. — 496 с.
10. Boost.Asio Documentation [Электронный ресурс]. — URL: <https://www.boost.org/doc/libs/>
11. WebRTC API Guide [Электронный ресурс]. — URL: <https://webrtc.org/>
12. Opus Codec Technical Documentation [Электронный ресурс]. — URL: <https://opus-codec.org/docs/>
13. Qt 6 Framework Documentation [Электронный ресурс]. — URL: https://doc.qt.io/
14. PortAudio API Reference [Электронный ресурс]. — URL: <http://www.portaudio.com/docs/>
15. Грибова Е.В. Анализ методов шифрования в современных мессенджерах // Информационные технологии. — 2023. — №5. — С. 45–52.
16. Иванов А.С. Оптимизация задержек в VoIP-системах на основе WebRTC // Труды университета. — 2022. — Т. 18, №3. — С. 112–120.
17. Rosenberg J. SIP: Session Initiation Protocol // IEEE Communications Magazine. — 2021. — Vol. 59, No. 4. — P. 78–84.
18. Разработка сетевых приложений на C++ с использованием Boost.Asio [Электронный ресурс]. — URL: <https://github.com/chriskohlhoff/asio>
19. Qt WebRTC Examples [Электронный ресурс]. — URL: <https://github.com/qt/qtwebrtc>
20. End-to-End Encryption in Messaging Apps: Best Practices [Электронный ресурс]. — OWASP, 2023.
21. Статья “Сети для самых маленьких” URL: https://linkmeup.gitbook.io/sdsm