ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ Г. МОСКВЫ

«КОЛЛЕДЖ ПРЕДПРИНИМАТЕЛЬСТВА №11»

ЦЕНТР ИНФОРМАЦИОННО–КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Допустить к защите

Заместитель директора по содержанию и ИКТ

ГАПОУ КП № 11

\_\_\_\_\_\_\_\_\_\_\_ Адилова К.О.

« » мая 2021 г.

**ДИПЛОМНАЯ РАБОТА**

**Разработка и внедрение корпоративного мессенджера в компании Neovox (ООО “Ньюконтакт”)**

по специальности: **09.02.07 Информационные системы и программирование**

|  |  |
| --- | --- |
| **Выполнил:**  студент группы ИСиП-33  Семенова Арина Александровна    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись | **Научный руководитель:** преподаватель Центра ИКТ  Макиевский Станислав Евгеньевич    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись |

Москва, 2021г

## ОГЛАВЛЕНИЕ

[**Введение** 3](#_Toc64555416)

[**Глава 1. Проектирование и разработка корпоративного мессенджера для компании Neovox (ООО “Ньюконтакт”)** 6](#_Toc64555417)

[**1.1. Общие сведения** 6](#_Toc64555418)

[**1.1.1. Описание предметной области** 6](#_Toc64555419)

[**1.1.2. Анализ требований к информационной системе** 8](#_Toc64555420)

[**1.1.3. Требования к интерфейсу** 8](#_Toc64555421)

[**1.1.4. Анализ существующих приложений, используемых компанией на данный момент** 9](#_Toc64555422)

[**1.2.** **Обзор технологий** 11](#_Toc64555423)

[**1.2.1.** **Архитектура клиент-сервер** 11](#_Toc64555424)

[**1.2.2.** **Протокол TCP/IP. Передача данных с помощью Socket** 12](#_Toc64555425)

[**1.2.3.** **Выбор среды разработки** 13](#_Toc64555426)

[**1.2.4.** **Выбор платформы для разработки** 13](#_Toc64555427)

[**1.2.5.** **Язык программирования C#** 14](#_Toc64555428)

[**1.2.6.** **Диаграммы классов, UseCase, ER** 14](#_Toc64555429)

[**Глава 2. Разработка и внедрение корпоративного мессенджера** 16](#_Toc64555430)

[**2.1. Разработка приложения сервера мессенджера** 17](#_Toc64555431)

[**2.2. Разработка приложения клиента мессенджера** 18](#_Toc64555432)

[**2.2. Разработка базы данных** 21](#_Toc64555433)

[**2.3. Разработка программной части приложения** 21](#_Toc64555434)

[**2.4. Разработка визуального интерфейса приложения** 21](#_Toc64555435)

# **Введение**

Компания Neovox входит в ТОП-10 крупнейших аутсорсинговых контактных центров России. Клиентами Neovox являются крупнейшие российские и зарубежные компании: государственные организации, ведущие банки, интернет-магазины, страховые и логистические компании. В компании работает более 3000 человек. Внутри компании проводится обучение сотрудников, которое осуществляется с помощью мессенджера Skype. Отчет о проделанной работе необходимо сдавать в форме сообщения в WhatsApp, технические специалисты используют в своей работе Telegram.

Рассмотрим процесс работы специалистов Центра на примере конкретной ситуации: у клиента «Почта России» возникли проблемы с удержанием посылки на таможне. Так как «Почта России» является клиентом Neovox, поступающие звонки на горячую линию находятся в компетенции Neovox. Звонок принимает оператор и фиксирует дату начала разговора во внутреннем ПО компании. В ходе разговора с клиентом оператор определяет тип проблемы и в зависимости от него:

А) Оформляет заявку на перенаправление задачи техническому специалисту (в системе Neovox и копией в чате Telegram с техническими специалистами)

Б) Решает проблему сам (связывается с супервайзером с помощью WhatsApp в случае если проблема требует нестандартного решения)

Для коммуникации сотрудники используют несколько мессенджеров, необходимые для передачи важной информации, файлов, создания как аудио, так и видеоконференций. Определение пути решения проблемы происходит во время звонка с клиентом, это означает, что оператор обязан потратить минимальное количество времени на то, чтобы определить, как помочь человеку на линии. Использование нескольких программ в данном случае нецелесообразно, так как необходимо уметь быстро ориентироваться и помнить, какой мессенджер служит для решения определенной задачи, что снижает эффективность новых сотрудников и увеличивает количество времени, потраченного на решение проблемы клиента. За каждый из мессенджеров отвечает отдельный штат сотрудников, который так же подразделяется по городам на меньшие штаты, что также является дополнительным расходом ресурсов компании. Часто так же аккаунты сотрудников, зарегистрированных в мессенджерах, не подразделяются на рабочие и основные. То есть в контакт-листе присутствуют не имеющие к работе отношения контакты (семья, друзья и так далее). Возникновение проблем с серверами одного мессенджера может повлечь за собой задержки в работе сотрудников, непредвиденные технические проблемы, которые так же могут не зависеть от технических специалистов компании. По требованию заказчика мессенджер должен быть доступен только для сотрудников компании.

Решить эти проблемы можно несколькими способами, первым из которых является продолжение использования сторонних мессенджеров. Данный способ является неактуальным и неудобным для заказчика по ряду причин, основной из которых является зависимость стабильной работы компании от серверов и технических возможностей организации-создателя этих мессенджеров. Логика достаточно проста: если случается поломка на сервере Skype, то технические специалисты компании Neovox не могут повлиять на её исправление. Так же доступ к чатам могут заполучить не только сотрудники компании, но и сторонние лица, что может привести к утечке данных.

**Проблема:** Низкая эффективность сотрудников и увеличенная трата ресурсов компании при организации работы с использованием нескольких мессенджеров;

**Цель:** Оптимизация труда сотрудников и расходов компании путем разработки корпоративного мессенджера.

**Задачи:**

1. Анализ организации и процесса обмена информацией;
2. Поиск и сравнительный анализ аналогичных приложений, позволяющих организовать работу контактного центра;
3. Анализ требований заказчика;
4. Анализ и учет системных требований;
5. Описание технологии разработки нового мессенджера;
6. Составление модели мессенджера;
7. Составление диаграмм UseCase, ERD, диаграммы классов;
8. Выбор и описание инструментов для создания мессенджера;
9. Разработка базы данных клиент-серверного приложения;
10. Разработка интерфейса программы (FrontEnd);
11. Разработка программной части (BackEnd);
12. Тестирование программы на наличие ошибок;
13. Описание экономической эффективности проекта;
14. Ввод программы в эксплуатацию.

**Объект дипломной работы:** Компания Neovox (ООО “Ньконтакт”)

**Предмет:** Корпоративный мессенджер на основе архитектуры клиент-сервер

В результате дипломной работы планируется спроектировать и разработать мессенджер, с помощью которого будет оптимизирован рабочий процесс и сокращены расходы ресурсов компании.

# **Глава 1. Проектирование и разработка корпоративного мессенджера для компании Neovox (ООО “Ньюконтакт”)**

## **1.1. Общие сведения**

## **1.1.1. Описание предметной области**

Система Call-Центра состоит из штата сотрудников и комплекса определенных технических средств. Деятельность контактного центра заключается в обработке письменных запросов в чате и голосовых запросов по телефону. Необходимо не просто принимать звонки, а принимать их правильно, за короткое время уметь узнать и зафиксировать максимальный объем данных, предоставляемых клиентом. Как правило, звонки принимаются операторами. Правильная интонация, диалог по скрипту, оперативное решение проблемы клиента - все это лишь малая часть того, что должен соблюдать оператор контактного центра. До звонка, во время звонка и после него оператор обязан сопровождать свою работу короткими отчетами с помощью мессенджера WhatsApp. В диалоге с супервайзером оператор до начала работы согласовывает линию, на которую он должен выйти, во время работы обращается с уточнением вопросов клиента, если их решение нельзя найти в скрипте и оно отличается от стандартного, после звонка докладывает его результат супервайзеру. В обязанности супервайзера входит сопровождение группы операторов дистанционно, контроль их действий и прослушивание записей звонков с фиксацией ошибок оператора и последующим разбором этих ошибок с ним. Согласно требованиям компании, все общение со специалистами IT-отдела ведется исключительно через Telegram и в случае технических неполадок нужно оперативно среагировать и как можно быстрее донести информацию о поломке до специалистов именно через этот мессенджер. Специалист IT-отдела должен незамедлительно реагировать на поступающие запросы от сотрудников Центра и устранять технические поломки с минимальными затратами временных ресурсов компании. С руководителями проекта и лицами, стоящими выше супервайзера связаться можно исключительно через Skype с помощью видеоконференции, так как до руководства отдела доходят обычно только очень сложные и важные вопросы, для решения которых необходима полная включенность в разговор, не ограниченная текстовыми сообщениями.

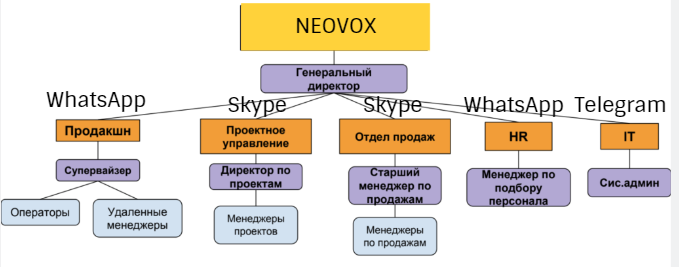


Рис.1 Организационная структура компании

На данной схеме изображена организационная структура компании и указаны мессенджеры, с помощью которых осуществляется деятельность разных отделов и их специалистов. Держать такую схему в голове достаточно сложно, особенно для новых сотрудников. Часто операторы, начавшие работать недавно, теряют очень много времени на то, чтобы написать специалисту о проблеме клиента, что доставляет дискомфорт и самому клиенту, так как увеличивается время ожидания на линии. Использование именно разных мессенджеров оправдывается тем, что в одном имеются функции, которых нет в другом. Супервайзеры в Центре отвечают за организацию работы операторов и корректировку рабочего плана, между собой и с операторами им необходимо связываться в WhatsApp. Технические специалисты же большое внимание уделяют Telegram. Данный метод работы с использованием нескольких мессенджеров сильно замедляет процесс решения не только технических проблем, но и проблем клиента, который обращается с определенным вопросом.

## **1.1.2. Анализ требований к информационной системе**

Корпоративный мессенджер должен представлять собой удобную и понятную для рядового пользователя ПК программу, которая позволит обмениваться файлами, быстрыми сообщениями (текстовыми, голосовыми или видео), помогающую в решении рабочих задач. Кроме того, в этой программе должны быть реализованы следующие функции:

* Сортировка пользователей по городам (отдельные чат-комнаты для городов для сообщения информации на целый штат сотрудников, работающих в определенном офисе)
* Просмотр профиля сотрудника (для получения дополнительной контактной информации)
* Осуществление доступа к чату администрации только по паролю, это исключит возможность передачи конфиденциальной информации третьим лицам
* Осуществление доступа к чату только для сотрудников компании
* Отсутствие в контакт-листе посторонних контактов, не связанных с деятельностью компании
* Нетребовательность ресурсов, возможность установки мессенджера на малопроизводительные системы
* Возможность запуска сервера только администратором компании, это является дополнительной гарантией безопасности и защиты от утечки информации, так как взаимодействие с «ядром» программы будет осуществляться только системным администратором, работающим в компании
* Возможность подключения одним пользователем к нескольким чатам, так как один человек может быть задействованным в нескольких проектах одновременно
* Возможность прикреплять файлы внутри чата
* Поиск контактов по имени или должности для быстрого доступа к нужному контакту

## **1.1.3. Требования к интерфейсу**

Интерфейс мессенджера должен быть интуитивно понятным и простым в освоении для пользователей любого возраста и уровня владения ПК, что поможет компании исключить дополнительную затрату ресурсов на обучение сотрудников использованию мессенджера. Стилевое оформление не должно содержать броских и ярких акцентов, отвлекающий сотрудников от их рабочих обязанностей. Оформление должно придерживаться строго трех базовых цветов логотипа компании. Отсутствие лишних броских деталей позволит сотрудникам концентрироваться только на поставленных рабочих задачах.

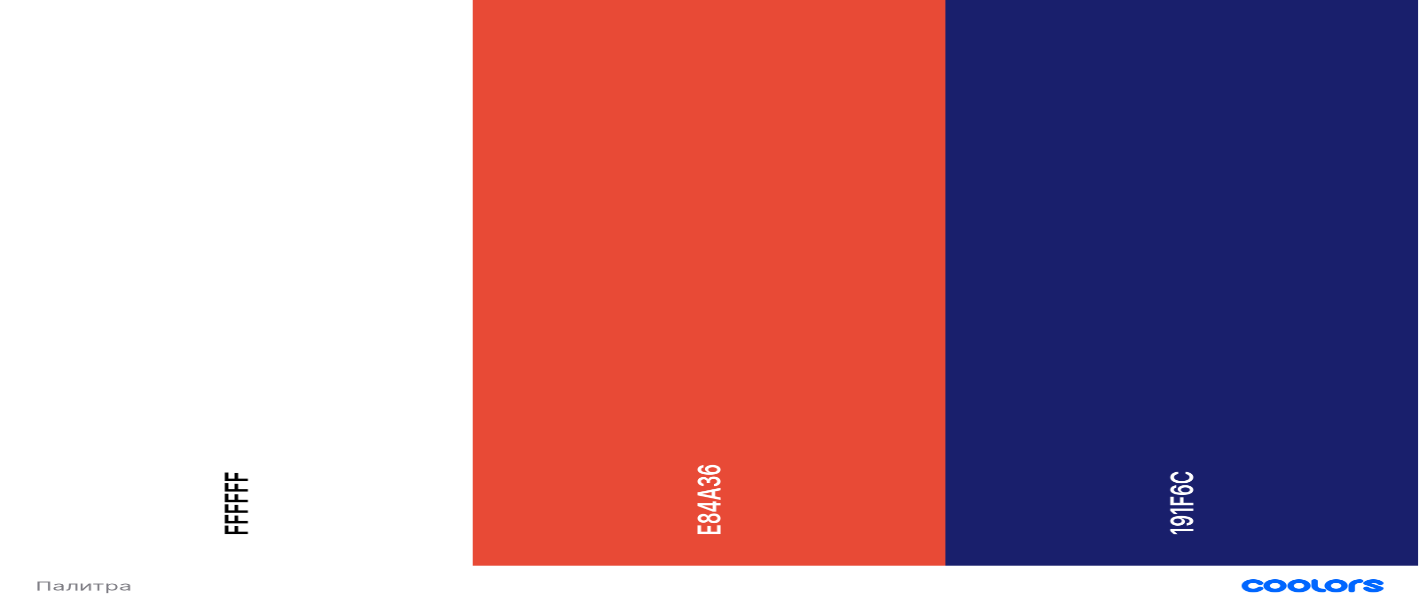


Рис.2 Цветовая палитра стиля оформления приложения

## **1.1.4. Анализ существующих приложений, используемых компанией на данный момент**

Рассмотрим принцип работы корпоративного мессенджера на примерах уже существующих приложений от наиболее популярных к менее популярным.

Пример 1. Telegram

Telegram является одним из наиболее распространенных мессенджеров, используемых среди различных компаний. Это удобное приложение, имеющее как мобильную версию, так и версию для компьютера. Поддержка нескольких платформ и высокая степень защиты за счёт особенностей протокола шифрования данных MTProto делает его приоритетным выбором среди пользователей. Так же одним из преимуществ является user friendly интерфейс, позволяющий пользователю кастомизировать приложение на свой вкус с помощью смены цвета фона, иконки и некоторых других элементов интерфейса. Мессенджер имеет функцию отправки голосовых, текстовых и видеосообщений, но для высокого качества связи требуется стабильное интернет-соединение. Резюмируя, можно выделить следующие достоинства и недостатки:

**Достоинства:**

* Возможность передавать большие файлы (по сравнению с другими мессенджерами) размером до 1,5гб;
* Возможность совершать видеозвонки;
* Высокая степень безопасности;
* Быстрая передача данных;
* Возможность создавать папки для чатов и закреплять некоторые выше других, что упрощает процесс сортировки, если чатов много;
* Бесплатен;

**Недостатки:**

* Отсутствие возможности совершать групповые звонки;
* Доступ к каналам чата может иметь абсолютно любой человек, имеющий аккаунт в Telegram;
* Общедоступность в нашем случае это подвергает общедоступный аккаунт риску спама и нежелательной информации, мешающей работе).

Пример 2. Skype

Skype является одним из самых первых корпоративных месенджеров, разработанных, в большей степени, для совершения звонков. В приложении присутствует набор стандартных функций: аудиозвонки, видеозвонки, демонстрация экрана, обмен быстрыми сообщениями. Кроме того, есть дополнительные опции, доступные для взаимодействия: групповые и приватные чаты, обмен небольшими файлами.

**Достоинства:**

* Доступен для большинства платформ(iOS, Mac, Linux, а приложение поддерживается на Windows, Windows Phone, Android ) ;
* Возможность отправлять файлы большинства доступных форматов;
* Возможность создавать групповые чаты с сохраняющейся историей сообщений.

**Недостатки:**

* Низкая степень безопасности;
* Медленная скорость работы, загрузки и передачи данных;
* Низкое качество видео во время видеозвонка с более, чем пятью пользователями;
* Большое количество отвлекающей рекламы внутри приложения;
* Версия для бизнеса предоставляется платно.

Пример 3. Discord

Изначально Discord был разработан как мессенджер для геймеров, но сейчас он также нашел применение и в бизнес-сфере. Это обусловлено его широким функционалом, в который входит: возможность создавать чаты для групп, видеоконференции, возможность создавать чат-комнаты с доступом по персональному приглашению или паролю, заблокированные для посторонних пользователей. Основным и наиболее весомым преимуществом этого мессенджера является возможность создания голосовых каналов, с помощью которых можно быстро и доступно обмениваться короткой информацией, не собирая для этого большую видеоконференцию. К каналу можно настроить как неограниченное количество пользователей для подключения, так и ограничить его, например, до 10 человек.

**Достоинства:**

* Простой и понятный для рядового пользователя интерфейс;
* Высокая скорость передачи данных;
* Возможность проведения групповых видеоконференций, но с ограничением;
* Высокий уровень безопасности.

**Недостатки:**

* Ограничение на количество участников видеозвонка – не более 10 человек;
* Интеграция только с сервисами для геймеров;
* Много отвлекающих элементов интерфейса и функций, предназначенных для взаимодействия с компьютерной игрой, что сильно отвлекает внимание.

## **Обзор технологий**

## **Архитектура клиент-сервер**

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты (потребители этих функций).

Практические реализации такой архитектуры называются клиент-серверными технологиями.

Клиент – это в данном случае мессенджер. Как правило компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам. Компьютер (или программу), управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер...). Клиент и сервер какого-либо ресурса могут находится как в рамках одной вычислительной системы, так и на различных компьютерах, связанных сетью. Основной принцип технологии "клиент-сервер" заключается в разделении функций приложения на три группы:

* ввод и отображение данных (взаимодействие с пользователем);
* прикладные функции, характерные для данной предметной области;
* функции управления ресурсами (файловой системой, базой данных и т.д.)

Также стоит заметить, что в основе взаимодействия клиент-сервер лежит принцип того, что такое взаимодействие начинает клиент, сервер лишь отвечает клиенту и сообщает о том может ли он предоставить услугу клиенту и если может, то на каких условиях. Данная концепция взаимодействия была разработана в первую очередь для того, чтобы разделить нагрузку между участниками процесса обмена информацией, а также для того, чтобы разделить программный код поставщика и заказчика.

**Преимуществом** модели взаимодействия клиент-сервер является то, что программный код клиентского приложения и серверного разделен. Если мы говорим про локальные компьютерные сети, то к преимуществам архитектуры клиент-сервер можно отнести пониженные требования к машинам клиентов, так как большая часть вычислительных операций будет производиться на сервере, а также архитектура клиент-сервер довольно гибкая и позволяет администратору сделать локальную сеть более защищенной.

**К недостаткам** модели взаимодействия клиент-сервер можно отнести то, что стоимость серверного оборудования значительно выше клиентского. Сервер должен обслуживать специально обученный и подготовленный человек. Если в локальной сети ломается сервер, то и клиенты не смогут работать (в качестве частного случая можно привести пример: мощности сервера не всегда хватает, чтобы удовлетворить запросы клиентов.

В качестве заключения можно акцентировать внимание на том, что архитектура клиент-сервер не делит компьютеры на только клиент или только сервер, а скорее позволяет распределить нагрузку и разделить функционал между клиентской частью и серверной.

## **Протокол TCP/IP. Передача данных с помощью Socket**

Протоколом называется набор правил, задающих форматы сообщений и процедуры, которые позволяют компьютерам и прикладным программам обмениваться информацией. Эти правила соблюдаются каждым компьютером в сети, в результате чего любой хост-получатель может понять отправленное ему сообщение. Набор протоколов TCP/IP можно рассматривать как многоуровневую структуру. TCP обеспечивает доставку передаваемого сообщения получателю.

В .NET за работу с TCP отвечают три класса из пространства имён System.Net.Sockets:

* Socket — обеспечивает базовый функционал TCP и UDP сокетов.
* TcpListener — этот класс обеспечивает функционал TCP сервера;
* TcpClient – этот класс работает как TCP клиент. С его помощью осуществляется передача данных от клиента серверу и, как ни странно наоборот.

Для передачи данных в моем случае будет использоваться socket. Socket - это механизм обмена сообщениями между процессами, при этом эти процессы могут происходить на различных компьютерах, которые объединены в одну сеть.

## **Выбор среды разработки**

Для разработки дипломного проекта я выбрала Visual Studio 2019. Microsoft Visual Studio - это интегрированная среда разработки (IDE) от Microsoft. Она имеет бесплатную версию, интуитивно понятна в использовании, содержит в себе все необходимые инструменты для программирования не только функциональной части приложения, но и обеспечивает комфорт в создании графического интерфейса. В программе есть удобный редактор кода, который содержит инструменты для работы с языками, подсвечивает ошибки, синтаксис, типы данных, быстрый компилятор, функция автоматизированной сборки, обозреватель классов, виртуальный сервер для среды, позволяющий запускать сайты прямо из среды. Также Visual Studio значительно сокращает время написания программной части приложения своими подсказками и уже заранее добавленными благодаря интеллектуальной системе частями кода, которые нет необходимости писать вручную с нуля.

## **Выбор платформы для разработки**

Windows Presentation Foundation (WPF) — это платформа пользовательского интерфейса для создания клиентских приложений для настольных систем. Выбор конкретно этой платформы можно обосновать следующими преимуществами: платформа разработки WPF поддерживает широкий набор компонентов для разработки приложений, включая документы и безопасность, привязки данных, модель приложения, элементы управления, ресурсы, графику, макет. Эта платформа является частью платформы .NET, и так как в процессе обучения я уже создавала приложения в .NET с помощью ASP.NET и Windows Forms, принципы программирования на WPF мне являются знакомыми. WPF использует расширяемый язык разметки для приложений (XAML), что подразумевает возможность декларативного определения графического интерфейса, предоставляющем альтернативу программному созданию графики и элементов управления. Также в WPF все элементы измеряются единицах, независимых от устройства, следовательно, приложения на WPF свободно масштабируются под экраны с любым доступным разрешением.

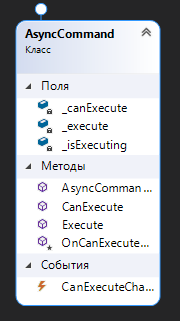
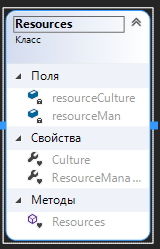
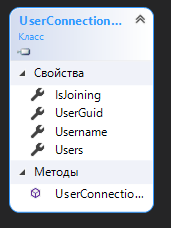
## **Язык программирования C#**

Данный язык использует принципы объектно-ориентированного программирования. Это подразумевает сначала создание конструкций на основе предметной области, а потом реализацию взаимодействия между ними. Это упрощает процесс создания определенных функций. Так же в языке присутствует упрощенный синтаксис и компилятор, можно использовать уже готовые конструкции при разработке, вместо того, чтобы писать огромное количество строчек кода. В результате код становится легко читаемым и скорость разработки приложения значительно повышается. Одним из самых существенных плюсов является наличие большого количества шаблонов и библиотек, помогающих в разработке с помощью необходимых решений из NuGet.

## **Диаграммы классов, UseCase, ER**

Класс представляет собой шаблон для создания объектов, обеспечивающий начальные значения состояний: реализацию поведения методов и полей, инициализацию полей-переменных. То есть класс описывает возможные свойства объектов или то, какими они могут быть. Класс, по сути, это концепт, который описывает состояние (атрибуты) и поведение (методы). У каждого атрибута есть свой тип, к каждого метод — своя сигнатура, но в диаграмме классов только имя класса является формой к обязательному заполнению. Имя класса пишется в самой верхней строке, затем идут атрибуты класса, их типы записываются после двоеточия и, методы пишутся в нижнем делении. Тип, возвращаемый методом, записывается после двоеточия в самом конце сигнатуры метода.

**Диаграмма классов** может использоваться на разных этапах жизненного цикла разработки программного обеспечения, моделируя диаграммы классов постепенно с разных точек зрения по мере продвижения по уровням детализации.

Примеры таблиц диаграммы классов из дипломного проекта.

**Диаграмма вариантов использования (UseCase diagram)** представляет собой графическую схему, которая описывает действующие лица и варианты использования, а также спецификации, являющиеся текстовым описанием определенных последовательностей действий (потока событий), которые в дальнейшем осуществляет пользователь при работе с программой.

Основными элементами диаграммы являются Актеры(участники), прецеденты (варианты взаимодействия) и сущности. Участник - это множество логически связанных ролей, которые исполняются при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Участником может быть человек или другая подсистема, система или класс, которые представляют нечто за пределами сущности. Графически участник изображается “человечком”. Прецедент – это описание множества последовательных событий (включая возможные варианты), которые выполняет система, приводящие к ожидаемому участником результату. Прецедент представляет собой поведение сущности, описывая взаимодействие между актерами и системой. Прецедент не показывает процесс достижения определенного результата, а только какое конкретно действие выполняется. Прецеденты обозначаются в виде эллипса, внутри которого указано его название или само действие.



Пример первой UseCase диаграммы из дипломного проекта

ER-диаграмма – это схема «сущность-связь, представляющая собой разновидность блок-схемы, показывающая, как различные «сущности» (люди, объекты, концепции и так далее) логически связаны между собой внутри системы. ER-диаграммы в большинстве случаев применяются для проектирования и отладки реляционных баз данных в информационных системах для бизнеса. ER-диаграммы (или ER-модели) полагаются на стандартный набор фигур, включая прямоугольники, овалы, ромбы и соединительные линии, для изображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи — глаголов.

# **Глава 2. Разработка и внедрение корпоративного мессенджера**

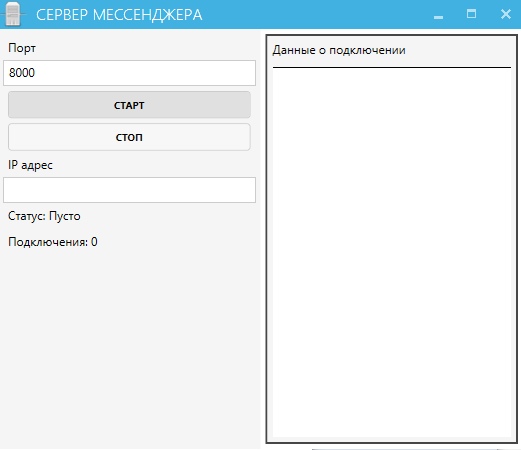
Для корректной работы мессенджера и гарантии его запуска в локальной сети предприятия необходимо разработать два приложения с отличающимся функционалом. Первое приложения для сервера будет под ответственностью технического специалиста компании, который будет включать его и проводить техническое обслуживание.

Второе приложение предназначено непосредственно для клиентов, то есть рядовых сотрудников компании, подключающихся к самому чату с помощью функций, реализованных внутри окна системы.

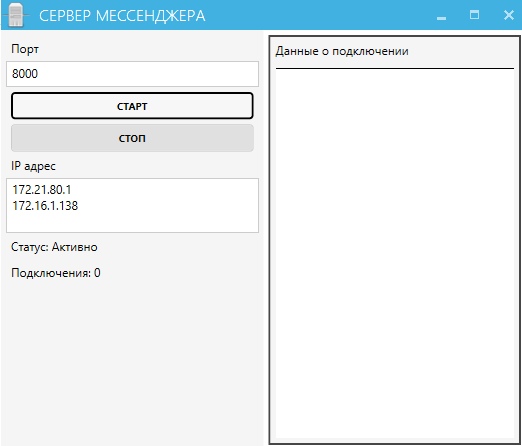
## **2.1. Разработка приложения сервера мессенджера**

Для обеспечения работы мессенджера необходимо предоставить локальный сервер, который может быть запущен только техническим специалистом компании, что снижает риск перебоев и неожиданных поломок, так как система будет контролироваться непосредственно сотрудником предприятия.

Приложение сервера мессенджера представляет собой окно с интуитивно понятным и простым интерфейсом. Для того, чтобы запустить сервер достаточно указать порт (он указан в сетевых свойствах локальной сети и компьютера) и нажать на кнопку «Старт».



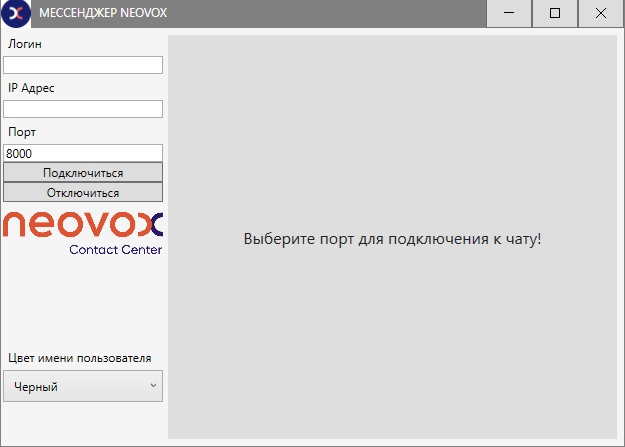
После нескольких секунд ожидания в колонке справа появятся данные о текущем подключении и состоянии сервера на данный момент, а в строке IP-адрес отобразится текущий адрес компьютера в локальной сети. Так как сервер запущен на этом компьютере, то подключиться к нему возможно через IP-адрес самого компьютера.



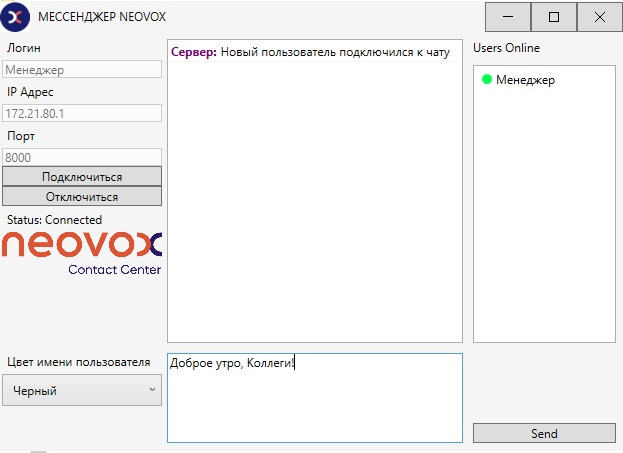
Отобразившийся IP-адрес необходимо скопировать в буфер обмена для дальнейшего использования. Сервер готов принимать сообщения только в состоянии «Активен», поэтому перед запуском клиента приложения необходимо убедиться, что в графе «Статус» указан именно этот параметр.

2.2. Разработка приложения клиента мессенджера

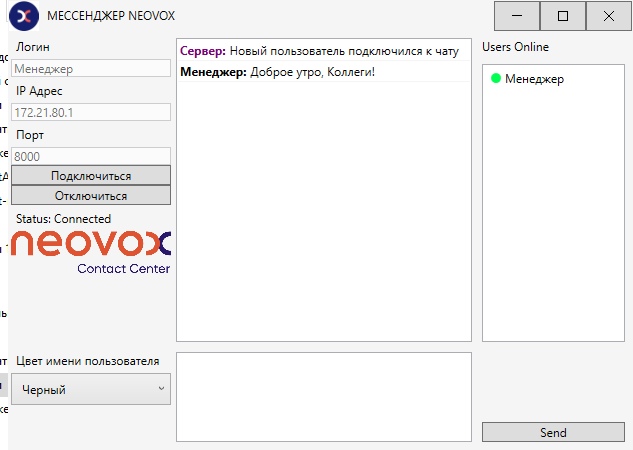
После запуска приложения самого мессенджера, скопированный IP-адрес из приложения сервера необходимо вставить в строку IP-адрес, указать логин и цвет логина. В приложении реализована функция выделения логинов различными цветами, каждый цвет соответствует занимаемой сотрудником должности (например, черный-менеджер, красный-супервайзер, зеленый-оператор и так далее). После этого необходимо нажать кнопку «Подключиться» и ожидать присоединения к чат-комнате.



После успешного подключения к чат-комнате, пользователь видит основное окно чата, где отображаются отправленные сообщения, справа может отследить сотрудников, находящихся в данный момент «В сети» (они отмечены зеленым значком) и в нижнем поле пользователь может написать свое сообщение и отправить его после нажатия на кнопку «Send» или клавиши Enter на клавиатуре. При каждом подключении нового пользователя в чат приходит оповещение о том, что он зашел в чат-комнату.



Для того, чтобы выйти из приложения, достаточно нажать на кнопку «Отключиться» и пользователь попадет на главный экран, где уже самостоятельно может завершить работу с мессенджером путем закрытия его через панель управления, как стандартное приложение.



## **2.2. Разработка базы данных**

## **2.3. Разработка программной части приложения**

## **2.4. Разработка визуального интерфейса приложения**

**2.5. Настройка локального сервера**

**2.6. Авторизация администраторов и назначение функций для их учетных записей**

**2.7 Тестирование приложения**

**2.8 Ввод приложения в эксплуатацию**

**ГЛАВА 3. ТЕХНИКО–ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ**

**ГЛАВА 4. ТЕХНИКА БЕЗОПАСНОСТИ**

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ЛИТЕРАТУРЫ**

**ПРИЛОЖЕНИЯ**