Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Северо-Восточный федеральный университет имени М.К. Аммосова»

Институт математики и информатики

Кафедра информационных технологий

КУРСОВАЯ РАБОТА

на тему:

РАЗРАБОТКА МЕССЕНДЖЕРА

Выполнил(а): студент(ка) \_\_ курса

группы \_\_\_\_\_\_\_ ИМИ СВФУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.И. Полянский

подпись, дата

Проверил (а): к.ф.-м.н., доцент каф.

\_\_\_\_\_\_\_\_\_\_\_\_ ИМИ СВФУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Эверстов

подпись, дата

Якутск 2022

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc104483863)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc104483864)

[1.1 Состояние рынка мессенджеров 5](#_Toc104483865)

[1.2 Обзор аналогов 7](#_Toc104483866)

[1.2.1 Telegram 7](#_Toc104483867)

[1.2.2 WhatsApp 9](#_Toc104483868)

[1.3 Обзор инструментов разработки 11](#_Toc104483869)

[1.3.1 Язык программирования Python 12](#_Toc104483870)

[1.3.2 Среда разработки PyCharm 14](#_Toc104483871)

[1.3.3 Язык программирования C# 14](#_Toc104483872)

[1.3.4 Среда разработки Visual Studio 2019 15](#_Toc104483873)

[1.3.5 Система управления базами данных MySQL 16](#_Toc104483874)

[2 Разработка мессенджера «Plum» 17](#_Toc104483875)

[2.1 Требования к ПО 17](#_Toc104483876)

[2.2 Модель классов 19](#_Toc104483877)

[2.3 Диаграмма прецедентов 20](#_Toc104483878)

[2.4 Архитектура ПО 21](#_Toc104483879)

[2.5 Описание ПО 23](#_Toc104483880)

[2.6 Описание серверной части 31](#_Toc104483881)

[2.6.1 Авторизация и регистрация 31](#_Toc104483882)

[2.6.2 Контакты 32](#_Toc104483883)

[2.6.3 Сообщения 33](#_Toc104483884)

[2.6.4 Авторизованный пользователь 34](#_Toc104483885)

[2.6.5 Пользователи 35](#_Toc104483886)

[ЗАКЛЮЧЕНИЕ 36](#_Toc104483887)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 37](#_Toc104483888)

# ВВЕДЕНИЕ

Сегодня информация является очень востребованным ресурсом. Мы хотим быть в курсе мировых событий, хотим знать, что происходит в нашем родном городе, в учебном заведении, на работе или в кругу наших родных и близких. Помимо получения информации, мы и сами хотим ей делиться с окружающими, однако это сложно осуществить, когда тот, с кем мы хотим связаться, находится в другой точке города или на другом конце планеты. Поэтому ныне очень актуальным инструментом передачи информации стали мессенджеры.

Мессенджеры – это специальные приложения, с помощью которых люди могут мгновенно обмениваться текстовыми сообщениями, видео, изображениями, документами или какими-то иными видами информации в режиме реального времени, находясь в чате. Чат [2] – это понятие, относящееся к процессу обмена сообщениями в Интернете. В чате обычно участвуют два или более людей, которые общаются с помощью программного обеспечения, хранящего сообщения и передающего их между пользователями. Этим программным обеспечением может выступать и мессенджер.

Как было упомянуто выше, мессенджеры способны хранить отправляемые сообщения и это является их большим преимуществом. Таким образом, например, телефонные звонки с использованием сотовой связи на их фоне становятся менее актуальными. Хотя многие мессенджеры в том числе поддерживают и возможность аудио- и видеозвонков с использованием сети Интернет.

Хотелось бы сказать и о том, насколько популярны мессенджеры на сегодняшний день. В частности, самым известным на данный момент мессенджером является WhatsApp. По состоянию на октябрь 2021 года [3] два миллиарда пользователей ежемесячно использовали эту программу. В феврале 2014 года компания Facebook, переименованная в 2021 году в Meta (признана экстремистской на территории РФ), приобрела это мобильное приложение за 19 миллиардов долларов.

Программисты довольно много времени проводят за компьютером, а не за экраном смартфона. Также есть и другие люди, для которых важно оставаться в курсе последних событий, не переводя свое внимание на телефон. В этом может помочь десктопное приложение. Десктопной называется программа, которая устанавливается непосредственно на компьютер пользователя. Беря это во внимание, а также учитывая описанные выше факторы, можно считать актуальной разработку десктопного мессенджера.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Состояние рынка мессенджеров

Согласно исследованию компании «Deloitte» [4] о потреблении медиаконтента в России за 2021 год, в 73% случаев люди используют Интернет для доступа к мессенджерам — это наиболее популярная цель посещения сети.

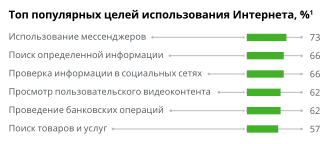


Рисунок 1 – статистика использования Интернета для различных целей в 2021 году по данным «Deloitte»

Тенденция применения мессенджеров претерпевает высокий рост. Об этом нам говорит исследование 2019 года от того же источника [5], в котором указано, что лишь в 53% случаев люди используют Интернет для доступа к мессенджерам. Таким образом, потребность в мессенджерах среди жителей России за 2 года возросла на целых 20%, что является серьезным показателем. Безусловно, на это в немалой степени повлияли условия пандемии – мы реже стали видеться в традиционном формате и стали чаще использовать технологии дистанционного общения.



Рисунок 2 – три наиболее востребованных функций смартфона согласно исследованию компании «Deloitte» за 2019 год

Конечно же, мессенджеры востребованы не только в России. Одними из самых загружаемых приложений для смартфонов в мире являются именно мессенджеры. По данным аналитической платформы «Statista» [1] за 2021 год, приложения «WhatsApp», «Telegram» и «Snapchat» были скачаны пользователями соответственно: 395, 329 и 327 миллиона раз, занимая таким образом 4, 5 и 6 место в списке самых скачиваемых приложений для смартфонов, что видно на рисунке 3.

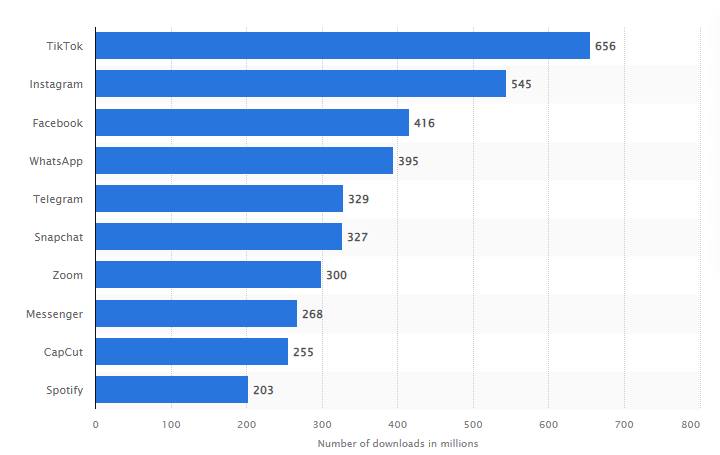


Рисунок 3 Список самых загружаемых приложений в мире за 2021 год

Говоря о прибыльности этих приложения, можно привести официальные данные о доходах Snapchat [6], которые на момент 2021 года составили 1.3 миллиарда долларов. Касательно Telegram и WhatsApp, сложно что-либо говорить. Telegram старается строго придерживаться своей политики бесплатного доступа и отсутствия встроенной рекламы (хотя это не запрещает зарабатывать на рекламе создателям контента), потому почти не приносит прибыли и существует на личные деньги его создателя – Павла Дурова, а также деньги третьих лиц. WhatsApp же не выкладывает в свободный доступ данные о своих доходах, но по оценкам специалистов журнала Forbes [7] в 2020 году приложение принесло владельцам 5.5 миллиардов долларов.

Подводя итоги, можно с уверенностью сказать, что общество нуждается в инструментах общения на расстоянии, из-за чего мессенджеры на сегодняшний день являются одними из самых востребованных приложений и безусловно приносят своим владельцам огромную прибыль, поэтому их разработка является достаточно актуальной проблемой

## Обзор аналогов

В предыдущих разделах уже упоминались такие мессенджеры как Telegram и WhatsApp. Среди самых популярных мессенджеров именно они имеют версии для десктопных устройств, поэтому рассмотрим их поподробнее в этом разделе.

### Telegram

Telegram – кроссплатформенное бесплатное приложение для мгновенного обмена сообщениями. Создателями являются братья Николай и Павел Дуровы. Им пользуются около 500 млн пользователей. Пользовательская часть приложения для Windows написана на языке программирования C++. Специально для этого приложения Николаем Дуровым и другими разработчиками Telegram был создан протокол MTProto, использующий алгоритм шифрования AES, протокол Диффи-Хеллмана для обмена RSA ключами, а также ряд хэш-функций. Также для обеспечения большей анонимности можно создавать анонимные чаты, использующие технологию end-to-end шифрования, обеспечивающего почти полную безопасность при обмене сообщениями[8][9].

Преимущества и особенности Telegram для пользователей состоят в следующем.

В Telegram присутствует возможность совершения аудио- и видеозвонков, создания групповых чатов, отправки голосовых и видео сообщений, отправка медиа и документов (причем их размер не ограничен).

Также Telegram имеет открытый API (application programming interface) для создания ботов. Боты – это программы, созданные сторонними разработчиками, использующими API-запросы к серверам Telegram для доступа к его функциям. Боты могут выполнять определенные функции автоматически или по заданному расписанию. Многие частные или даже государственные предприятия создают своих ботов в Telegram для того, чтобы дать возможность пользователям, не выходя из приложения воспользоваться предлагаемыми услугами, такими как, например, покупка товаров.

Помимо всего прочего, в Telegram любой пользователь может создать собственный канал, на который смогут подписываться другие пользователи и таким образом следить за сообщениями в этом канале. Это могут быть новостные каналы или каналы, связанные с вашим кругом интересов.

Еще одной особенностью приложения является то, что при авторизации на новом устройстве вам не нужно будет загружать какие-либо бэкап-файле. Вся ваша история переписок (за исключением переписок в анонимных чатах, упомянутых выше) хранится в защищенном облаке.

В список преимуществ можно включить и следующие вещи: групповые чаты в Telegram могут насчитывать до 50000 участников, связываться с другими пользователями можно как через номер телефона, так и через псевдоним, задаваемый в настройках. Еще пользователи Telegram активно пользуются бесплатными стикерами, которые могут быть созданы любым пользователем.

В общем-то этот мессенджер, я бы сказал, полностью удовлетворяет любые потребности пользователя и при этом сохраняет приятный, современный интерфейс, поэтому недостатки в нем выделить довольно сложно. Критики чаще всего обращают внимание на некоторые недостатки протокола MTProto, связанные с возможными уязвимостями. Также для регистрации Telegram не требует ввода пароля, требуется только ваш номер телефона, а код для аутентификации придет в виде СМС. Это создает самую явную угрозу – СМС могут быть перехвачены злоумышленниками различными путями. Конечно, если вы обычный пользователь, то вам не за чем беспокоиться ни об одной из этих угроз. Однако для крупных бизнесменов или влиятельных политиков, которым важна анонимность и приватность, эти аргументы могут быть существенными.

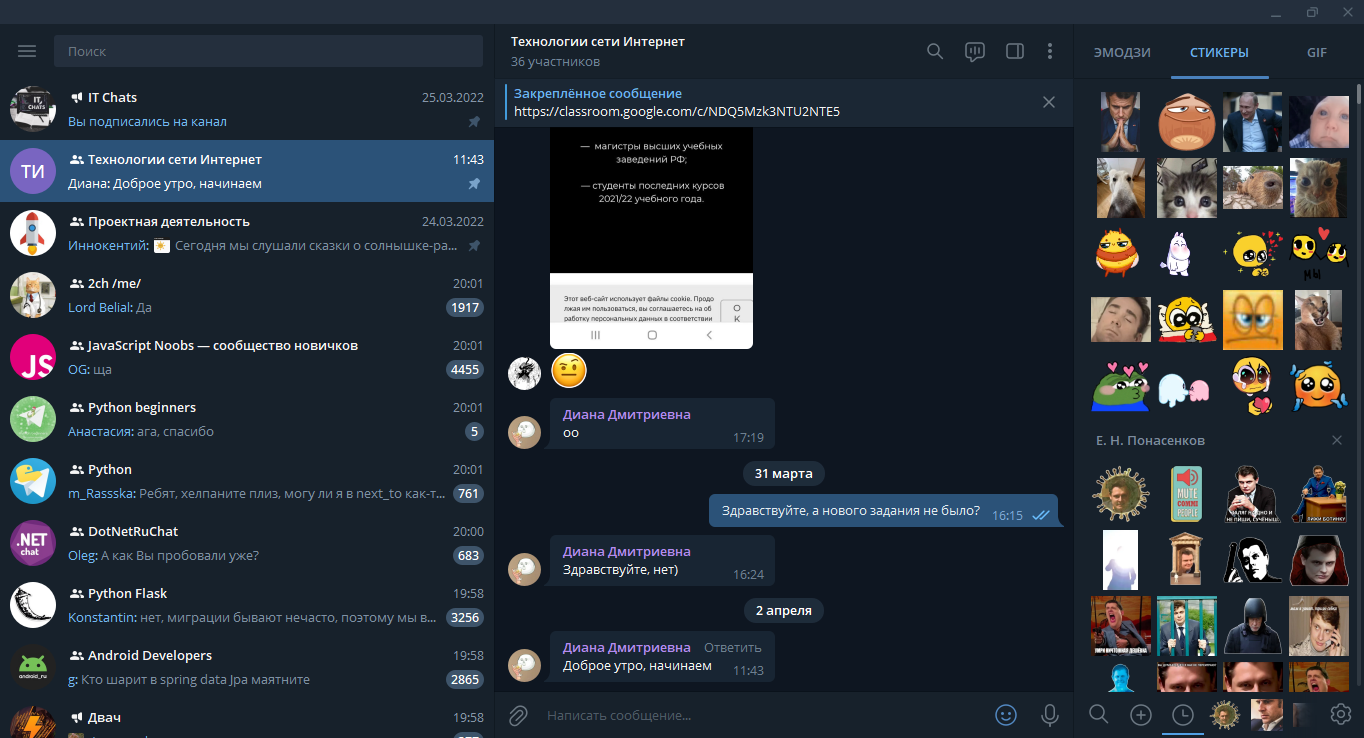


Рисунок 4 Интерфейс десктопной версии Telegram на ОС Windows

### WhatsApp

WhatsApp – бесплатный кроссплатформенный мессенджер для мгновенного обмена сообщениями, принадлежащий компании Meta. Насчитывает более 5 млрд человек в числе пользователей. Обмен сообщениями осуществляется при помощи XMPP-протокола на ejabberd-сервере, написанном на языке Erlang. Клиентская часть для десктопных устройств доступна через браузер и реализована на языке PHP.[10]

По сравнению с Telegram имеет более скудный список возможностей и особенностей.

Помимо текстовых сообщения, возможно обмениваться также и голосовыми сообщениями. Можно отправлять различные медиа файлы и документы. В приложении пользователи могут создавать группы и приглашать в них других пользователей. Также, как и в Telegram, некоторые чаты можно закидывать в архив.

Приложение автоматически синхронизирует список пользователей, с которыми вы можете связаться, с телефонной книгой телефона. Такая же функция есть у Telegram, она возможна благодаря тому, что регистрация в обоих мессенджерах происходит при помощи номера телефона.

В мессенджере есть возможность создавать так называемые «статусы», которые могут содержать в себе текст, картинку или непродолжительное видео. Подобная функция, называемая «историями», присутствует во многих популярных социальных сетях и впервые появилась в социальной сети Instagram (признана экстремистской на территории РФ).

Помимо перечисленных преимуществ в приложении присутствует и ряд недостатков. Первым является то, что сам мессенджер довольно сильно устарел. Связано это с тем, что обновляется и дополняется новыми функциями он крайне редко. Например, в нем до сих пор нет возможности закреплять какое-либо сообщение в групповом чате, хотя такая возможность есть абсолютно во всех популярных мессенджерах и социальных сетях. Также из-за нечастого обновления страдает и безопасность мессенджера. К примеру, в 2018 году телефон одного из богатейших людей мира, владельца компании Amazon, Джеффа Безоса, был взломан из-за уязвимости, которая была давно известна, но за исправление которой WhatsApp долго не брался. Еще вы не можете войти в аккаунты на нескольких смартфонах одновременно, так что если у вас есть планшет или второй телефон, то придется пользоваться веб-версией.

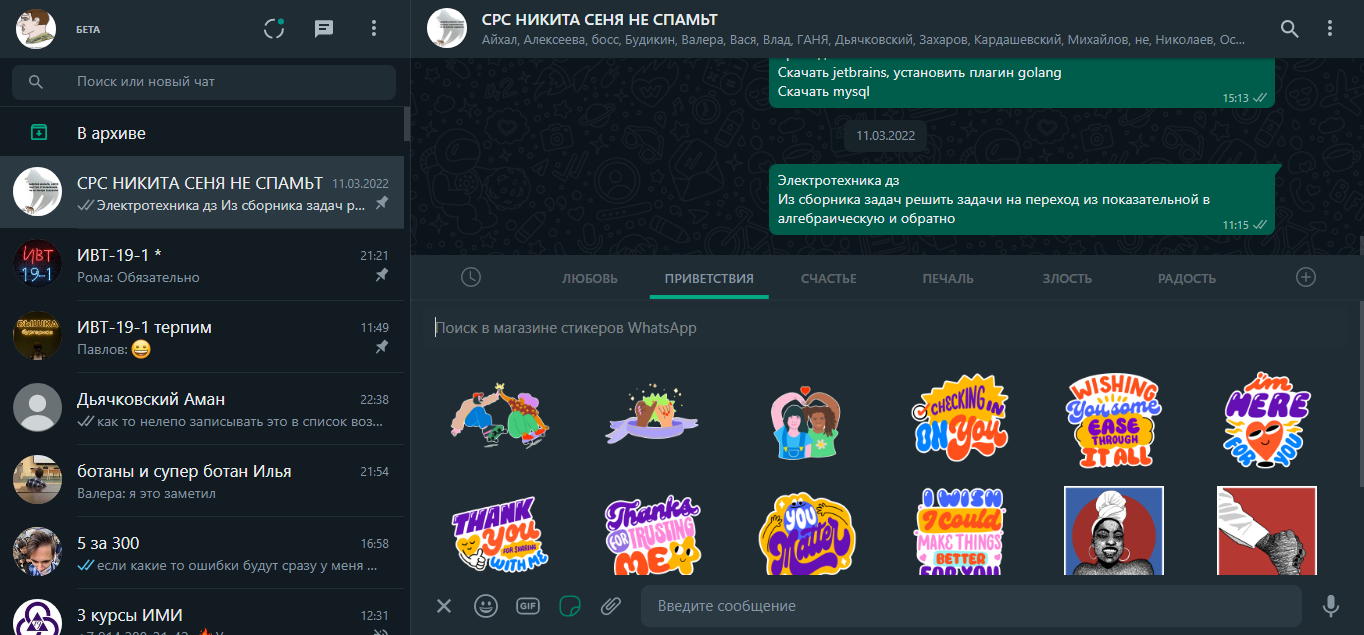


Рисунок 5 Интерфейс десктопной веб-версии WhatsApp на ОС Windows

## Обзор инструментов разработки

На текущий момент программная архитектура «клиент-сервер» наиболее востребована при создании программ, работающих с базами данных (БД), поскольку позволяет равномерно разделять вычислительную нагрузку между отдельными компонентами системы.[12]

Клиент-серверная архитектура подразумевает разделение прав компьютеров, входящих в сеть, на доступ к информационным ресурсам. Как правило, при данном подходе на поставщика услуг – сервер, ложатся обязанности по выполнению прикладных функций, касаемых данной задачи, а также функций управления ресурсами (БД, файловая система). На стороне заказчика услуг – клиента, остается компонент программы, отвечающий за интерфейс, функцией которого является ввод и отображение данных.

Особенности по работе с клиент-серверами:

1. Запуск запросов к серверам;

2. Ожидание ответа;

3. Получение ответа;

4. Соединение с небольшим числом серверов одновременно;

5. Как правило, взаимодействует непосредственно с конечными пользователями через любой пользовательский интерфейс;

6. Использование сетевых ресурсов [13]

Клиентская и серверная части требуют разных инструментальных подходов. В данной работе для разработки серверной части приложения был использован язык программирования (ЯП) Python 3, а для разработки клиенткой части ЯП C#. Для хранения информации на сервере использовалась система управления базами данных (СУБД) MySQL. Разработка серверной части проводилась в интегрированной среде разработки (IDE) PyCharm, а разработка клиентского интерфейса велась в IDE Visual Studio 2019.

IDE – совокупность программных средств, позволяющих эффективно вести разработку программного обеспечения. Обычно в числе инструментов имеет текстовый редактор, компилятор, интерпретатор, отладчик и прочие решения.

### Язык программирования Python

Python 3 – интерпретируемый высокоуровневый объектно-ориентированный ЯП с динамической типизацией. Он хорошо подходит для написания небольших приложений за кратчайшее время, так как имеет простой, лаконичный и человек читаемый синтаксис, а также располагает огромным количеством различных библиотек и фреймворков, пригодных для любых бизнес-задач.

Библиотека в программировании – это набор готовых подпрограмм для реализации какой-либо функциональности. Фреймворк – по сути, это большой набор библиотек, требующий от разработчика придерживаться определенной архитектуры.

Чаще всего язык Python используется для создания моделей машинного обучения, анализа данных и разработки серверных приложений. Для разработки своих сервисов его используют такие it-гиганты как Uber, Google, Instagram (признан экстремистским на территории РФ) и прочие известные бренды [14].

В данной работе для реализации серверной части приложения был выбран такой инструмент как Flask. Flask — микрофреймворк для создания веб-приложений, использующий инструменты сетевой библиотеки Werkzeug, а также Python-like язык шаблонов Jinja2. Предоставляет разработчику базовые возможности для создания веб-приложений. Часто используется для создания небольших веб-приложений.

Werkzeug — это инструментарий для WSGI, стандартного интерфейса Python между веб-приложениями и различными серверами [21].

Преимущества Flask:

1. сервер разработки и отладчик;
2. использование шаблонизатора Jinja2;
3. поддержка безопасных файлов cookie;
4. поддержка Unicode;
5. подробная актуальная документация;
6. расширения, создаваемые в том числе и сторонними разработчиками, для дополнительных функций

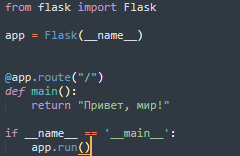


Рисунок 6 Пример минимального приложения на Flask

### Среда разработки PyCharm

PyCharm – это IDE для языка Python, создаваемая командой разработчиков JetBrains. Имеет широкий комплект средств, необходимых для эффективного написания кода. Распространяется в двух вариантах: платном и бесплатном. В настоящей работе используется платная версия продукта – PyCharm Professional Edition. Эта версия ПО предоставляет более широкий функционал, и в том числе возможность вести многоязычную разработку приложений. Эта среда поддерживает в том числе и фреймворк Flask. [19]

Изображение выглядит как текст, снимок экрана, монитор, внутренний

Автоматически созданное описание

Рисунок 7 Интерфейс среды разработки PyCharm

### Язык программирования C#

C# (C Sharp) – компилируемый высокоуровневый объектно-ориентированный ЯП со статической типизацией. Используется для разработки различных приложений под платформу .NET. Является прямым потомком двух наиболее успешных языков в мире – C++ и C. Также этот язык близко связан с Java, но часто они различаются в деталях реализации каких-либо технологий.

В списке своих инструментов имеет популярный фреймворк Windows Forms для разработки GUI(graphical user interface – графический интерфейс пользователя) приложений под ОС Windows, который и был использован в данной работе. Благодаря своей особенности такой как технология делегатов и событий, C# удалось реализовать через библиотеку Windows Forms простой, интегрированный и логически непротиворечивый способ управления разработкой Windows-приложений [17].

Делегаты в C# — это тип, объявляющий сигнатуру какого-либо метода. Создавая экземпляр делегата, можно указать для него метод или множество методов, которые будут вызываться при обращении к данному экземпляру.

События – особые типы делегатов. Для их экземпляров вне класса можно только привязывать методы, но обращение (вызов) к этим событиям может производиться только внутри класса, где это событие было объявлено.

### Среда разработки Visual Studio 2019

Microsoft Visual Studio 2019 – многофункциональная интегрированная среда разработки, разрабатываемая компанией Microsoft. Позволяет создавать консольные приложения, приложения с графическим интерфейсом (включая приложения Windows Forms), игры и веб-приложения на различных языках программирования, в том числе и на C#.

Имеет встроенный отладчик и редактор GUI-форм, поддерживает технологию IntelliSense, позволяет проводить простой рефакторинг кода.

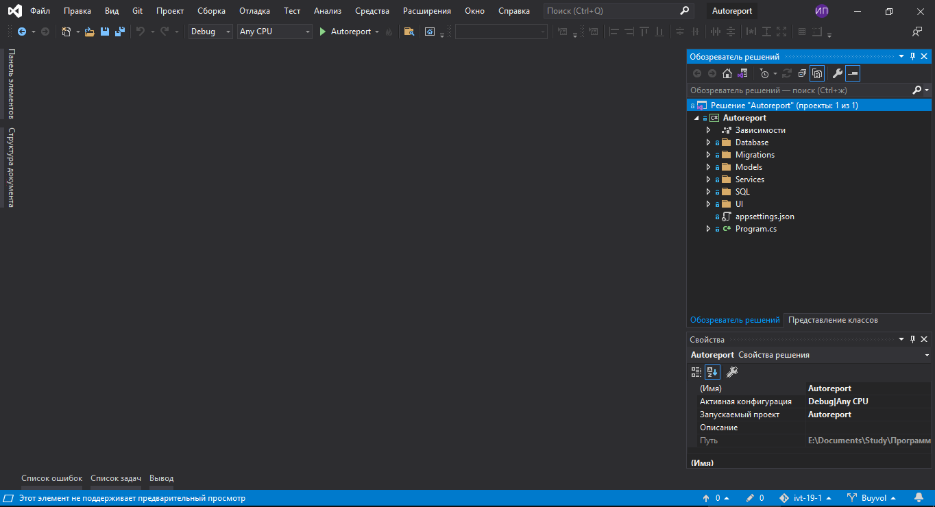


Рисунок 8 Интерфейс среды разработки Visual Studio 2019

### Система управления базами данных MySQL

MySQL – свободная реляционная система управления базами данных. Реляционными называют модели данных, основанные на понятии отношений (ralation). Эти отношения также неформально называют таблицами.

Эта СУБД является особенно хорошим решением для небольших приложений. Она обладает высокой надежностью и эффективностью, что делает ее выбором номер два среди всех реляционных СУБД, о чем говорит исследование «Statista» по состоянию на январь 2022 [20].

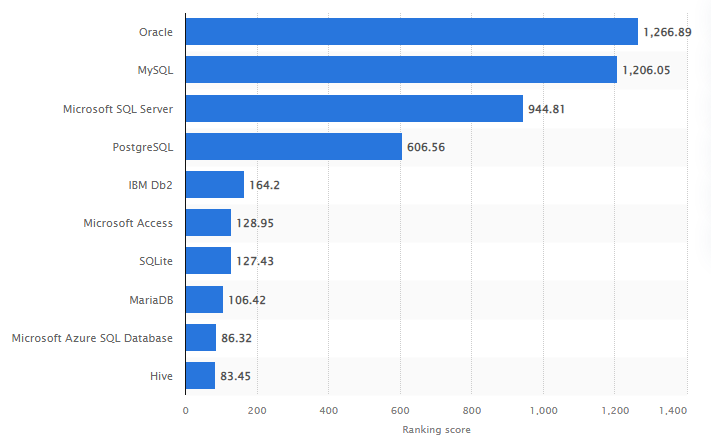


Рисунок 9 Рейтинг наиболее популярных реляционных СУБД

Она обладает рядом следующих преимуществ:

1. многопоточна;
2. бесплатна;
3. обладает оптимизация связей;
4. поддерживает записи фиксированной и динамичной длины;
5. гибкая поддержка различных типов данных;
6. быстрая работа;
7. хорошая поддержка на хостингах;

# Разработка мессенджера «Plum»

## Требования к ПО

Для определения того, готов ли разрабатываемый продукт, необходимо предъявить ряд требований к его свойствам и различным качествам. Для данного ПО был составлен следующий список требований.

Система должна хранить информацию о:

1. Пользователях
   1. Логин
      1. Не менее 3 символов и не более 16
      2. Только буквы латинского алфавита, цифры и символы: \*\_()#!&
   2. Короткое имя
      1. По умолчанию должен использоваться указанный пользователем логин
      2. Не менее 3 символов и не более 16
      3. Только буквы латинского алфавита, цифры и символы: \*\_ ()#!&
   3. Контакты
   4. Номер телефона
   5. Хэш пароля
      1. Алгоритм хэширования pbkdf2:sha256
   6. Текстовый статус
      1. Не более 139 символов
   7. Дата последнего посещения
   8. Общедоступность информации о телефоне
2. Сообщениях
   1. Автор сообщения
   2. Адресат сообщения
   3. Дата отправки
   4. Текст сообщения
      1. Не более 1000 символов и не менее 1

Для использования приложения пользователь должен быть зарегистрирован и аутентифицирован в системе.

Регистрация пользователя:

1. Ввод логина
2. Ввод короткого имени
3. Ввод пароля:
   1. Не менее 6 символов и не более 32
   2. Только буквы латинского алфавита и цифры
   3. Храниться в системе должен в виде хэша

Авторизация пользователя:

1. Ввод логина
2. Ввод пароля

Настройка профиля:

1. Смена логина
2. Смена короткого имени
3. Задать/сменить номер телефона
4. Настроить общедоступность информации о номера телефона
5. Смена пароля
6. Установка/изменение текстового статуса

Просмотр информацию о профилях других пользователей:

1. Короткое имя
2. Номер телефона (если просматриваемый пользователь сделал это поле общедоступным)
3. Текстовый статус
4. Активен ли сейчас пользователь

Пользователь должен иметь возможность отправлять текстовые сообщения другим пользователям.

Пользователю должен быть доступен список его контактов:

1. Можно добавлять других пользователей в контакты
2. Удалить пользователя из списка контактов

Пользователь должен иметь возможность искать других пользователей:

1. Поиск по короткому имени, логину или по номеру телефона
   1. При поиске по номеру телефона, должны быть возвращены только те пользователи, которые сделали доступ к своему номеру общедоступным

Пользователю должен быть доступен просмотр сообщений от других пользователей:

1. Короткое имя отправителя
2. Текст сообщения
3. Дата отправки сообщения

## Модель классов

В данной работе взаимодействие будет осуществляться между двумя классами объектов – пользователями. А взаимодействовать они будут посредством отправки друг другу сообщений. Поэтому (опираясь также на требования) структура классов будет иметь вид, изображенный на рисунке 10. Данная диаграмма была построена при помощи программы PowerDesigner.

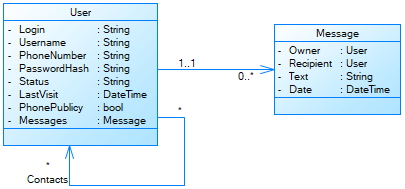


Рисунок 10 Диаграмма классов

Стрелочный переход от класса User к классу Message обозначает связь между этими классами. Эта связь ассоциативная с типом «один-ко-многим». Также имеет место стрелочный переход от User к самому себе со связью «много-ко-многим». Имя этой связи Contacts, то есть список контактов.

## Диаграмма прецедентов

Диаграмма прецедентов (use cases) описывает возможные варианты взаимодействия ПО и пользователя. С помощью нее могут быть описаны требования к взаимодействию пользователя и программы.

Пользователи и другие факторы, влияющие на систему извне, называются актерами и обозначаются на диаграмме в виде фигуры человека.

Отношения актеров с прецедентами, а также отношения прецедентов друг с другом подразделяются на 4 вида:

1. ассоциации – обозначает инициацию прецедента актером; изображается сплошной линией
2. включение – взаимосвязь базового прецедента с «внутренним», который будет инициализирован, если был выполнен базовый прецедент. Изображается пунктирной линией со стрелкой, указывающей на внутренний прецедент;
3. расширение – такая взаимосвязь показывает, что внутренний прецедент будет инициализирован только при каких-то условиях. Изображается в виде пунктирной линии со стрелкой, указывающей на базовый прецедент;
4. обобщение – создает связь между прецедентом и его частными случаями. Изображается как сплошная линия со стрелкой, указывающей на обобщающий прецедент

Учитывая требования, диаграмма будет иметь следующую структуру прецедентов:

1. Авторизация – ввод логина и пароля для предоставления доступа к приложению
2. Регистрация – ввод логина и пароля для их сохранения в системе
3. Аутентификация – связана включением с прецедентами «Авторизация» и «Регистрация», проверяет подлинность логина и пароля, а также их соответствие требованиям
4. Изменение профиля – редактирование личной информации пользователя
5. Выбор собеседника из списка контактов
6. Удалить – связан с прецедентом из предыдущего пункта расширяющей связью
7. Выбор собеседника из поиска по короткому имени, логину или номеру телефона
8. Добавить в список контактов - связан с прецедентом из предыдущего пункта расширяющей связью
9. Отправить сообщение

## Архитектура ПО

На рисунке 11 представлена диаграмма архитектуры разрабатываемого приложения. В ней на вершине расположен элемент, обозначающий саму программу - PlumMessenger. От этого элемента стрелки тянутся к формам, через которые приложение ведет диалог с пользователем. Это следующие формы:

1. Authorization – форма авторизации (и регистрации);
2. MainWindow – главное окно;
3. Search – форма поиска;
4. User Info – форма просмотра информации о пользователе
5. Edit Profile – форма модификации данных об авторизованном пользователе

От форм стрелки идут к компонентам клиентской части. Эти компоненты собирают в себе свойства и методы для взаимодействия с серверной частью. Приложение включает следующие компоненты:

1. AuthRequest – компонент, совершающий к серверу запросы, связанные с авторизацией и регистрацией;
2. MessageRequest – компонент, запрашивающий у сервера сообщения и отправляющий сообщения на сервер;
3. ContactRequest – компонент, совершающий к серверу запросы, связанные с получением списка контактов авторизованного пользователя и добавлением/удалением пользователей из этого списка;
4. UserRequest – компонент, отвечающий за запросы к серверу, связанные с работой с пользователями, такой как: получение информации о стороннем пользователе, получение информации об авторизованном пользователе, модификация информации об авторизованном пользователе.

От всех этих компонентов стрелки ведут к одной системе – серверу. Сервер в свою очередь также разделен на следующие компоненты:

1. authorization – авторизация, регистрация
2. message – отправка/получение сообщений
3. contact – получение/добавление/удаление контактов
4. users – получение информации о пользователях
5. user – получение информации об авторизованном пользователе и ее модификация

От этих компонентов стрелки сходятся на объекте базы данных, которая хранит необходимую информацию о пользователях, сообщениях и контактах.

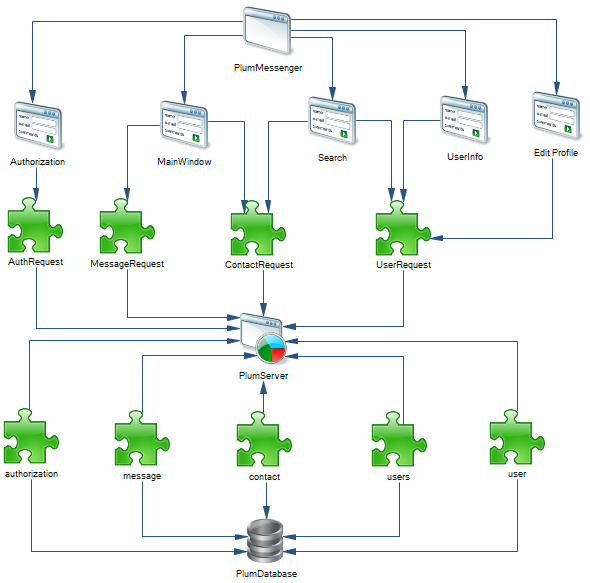


Рисунок 11 Архитектура ПО

## Описание ПО

При запуске приложения перед пользователем отображается окно авторизации [Рисунок 12].

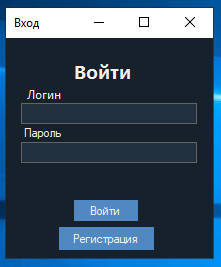


Рисунок 12 Окно авторизации

Из этого окна пользователь может перейти в окно регистрации [Рисунок 13].

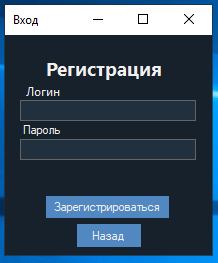


Рисунок 13 Окно регистрации

Зарегистрируем пользователя с логином sadscream [Рисунок 14]. При попытке установить, например, его пароль на не соответствующий требованиям – 123, пользователю будет сообщено об ошибке, так как этот пароль короче 6 символов.

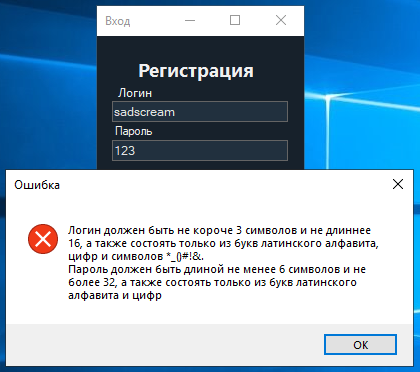


Рисунок 14 Аутентификация с ошибкой

Если указанные данные соответствуют требованиям, то пользователь будет успешно зарегистрирован в системе [Рисунок 15].

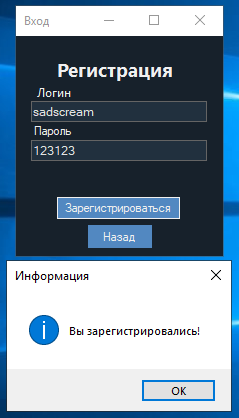


Рисунок 15 Успешная аутентификация

Теперь в окне входа можно ввести учетные данные и авторизоваться, после чего пользователь будет переведен в основное окно программы [Рисунок 16]. Красным на рисунке отмечена область, в которой отображено имя пользователя сверху и его логин через символ «@» снизу. Зеленым отмечена панель поиска, а оранжевым – панель со списком контактов. Голубым выделена «Кнопка редактирования профиля».

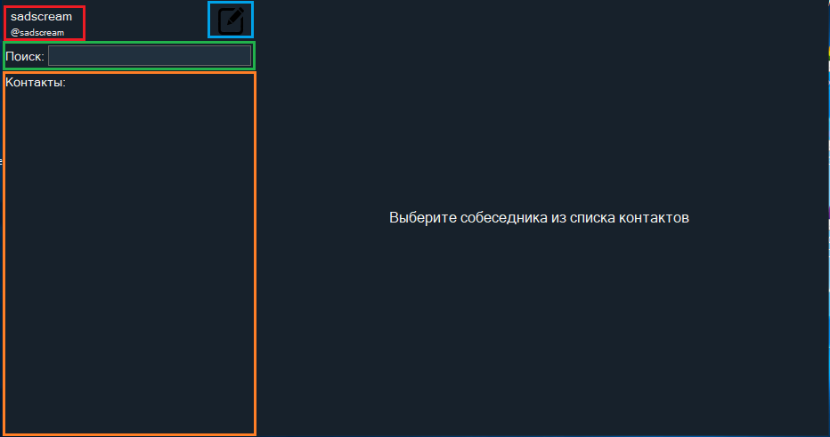


Рисунок 16 Основное окно

Зарегистрируем еще одного пользователя с логином saddy. При этом во время регистрации при попытке взять уже занятый логин «sadscream», будет выведено соответствующее сообщение об ошибке [Рисунок 17].

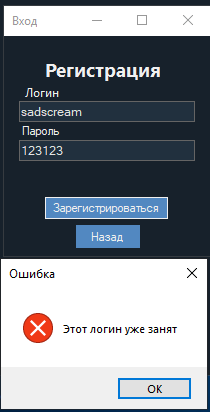


Рисунок 17 Логин уже занят

Теперь от лица пользователя «sadscream» попробуем найти только что зарегистрированного пользователя [Рисунок 18]. Для этого в поле поиска вобьем «sad» - префикс логина saddy. В качестве результата поиск вернет пользователя с соответствующим логином.

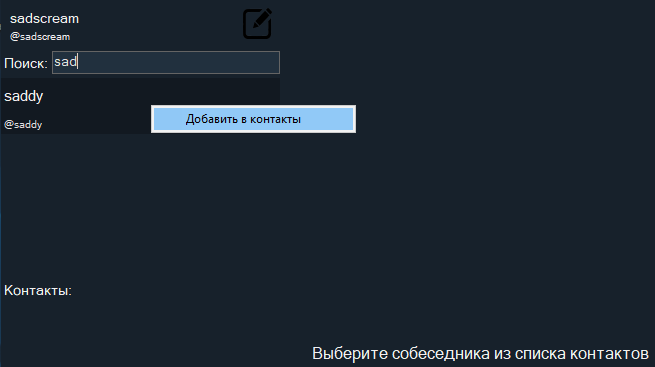


Рисунок 18 Поиск пользователя

При нажатии по этому объекту правой кнопкой мыши, будет доступно контекстное меню с опцией «Добавить в контакты», как видно из рисунка. После нажатия по ней пользователь будет добавлен в список контактов [Рисунок 19]. При выборе этого контакта из списка, будет открыта панель чата. Красным на рисунке выделена «Панель информации о пользователе», где отображается имя собеседника, а также сведения о том, активен ли он сейчас. Если пользователь не проявлял никакой активности за последние 5 минут, то его состояние отметится как Offline, а иначе – Online. Зеленым выделена область ввода и отправки сообщений.

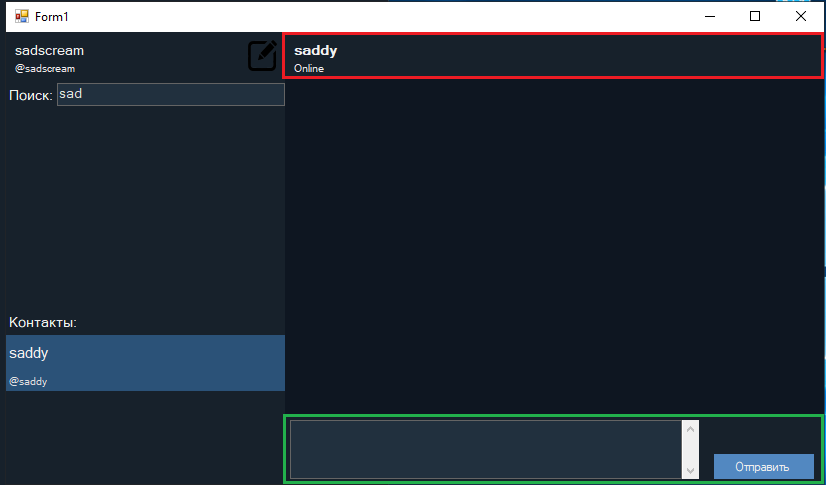


Рисунок 19 Пользователь добавлен в список контактов

В то же время с точки зрения пользователя saddy в его список контактов был добавлен пользователь sadscream [Рисунок 20].

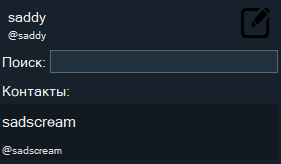


Рисунок 20 Список контактов пользователя

Пускай эти пользователи перекинутся парой сообщений [Рисунок 21].

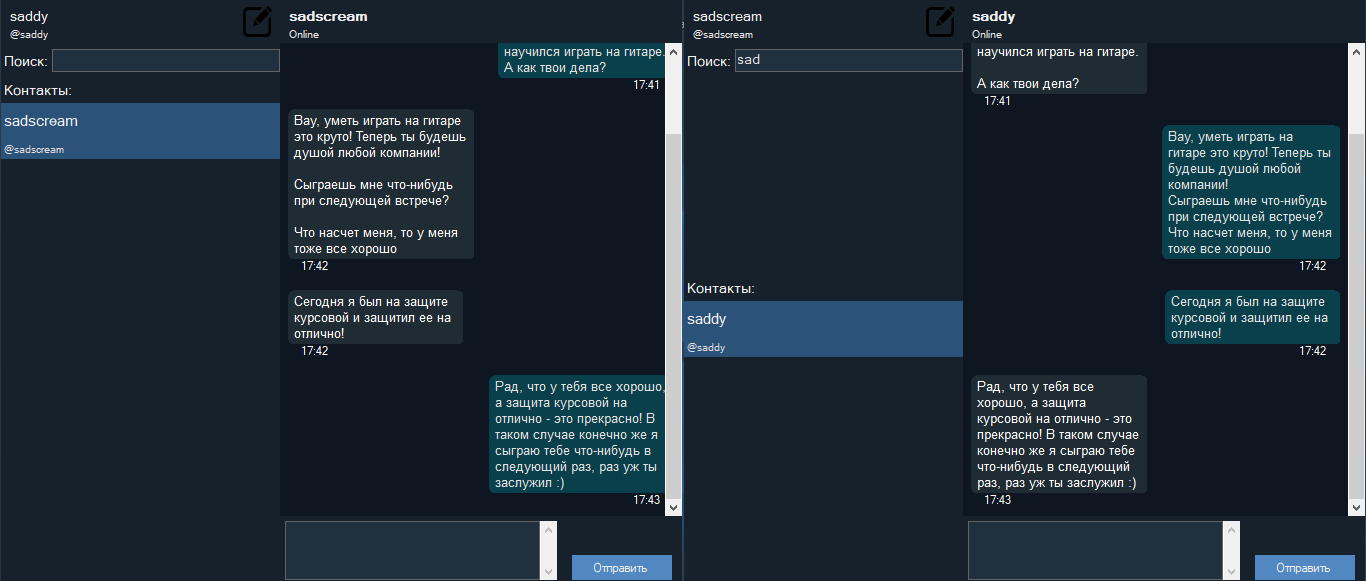


Рисунок 21 Обмен сообщениями

Если выйти из чата с пользователем, то на объекте контакта будет отображено количество новых сообщений, которые еще не были прочитаны [Рисунок 22]. Красным обведены два новых сообщения от пользователя sadscream, которые еще не были прочитаны пользователем saddy. Пользователь saddy может видеть, что ему пришло два новых сообщений, в области, обрисованной зеленым на рисунке.

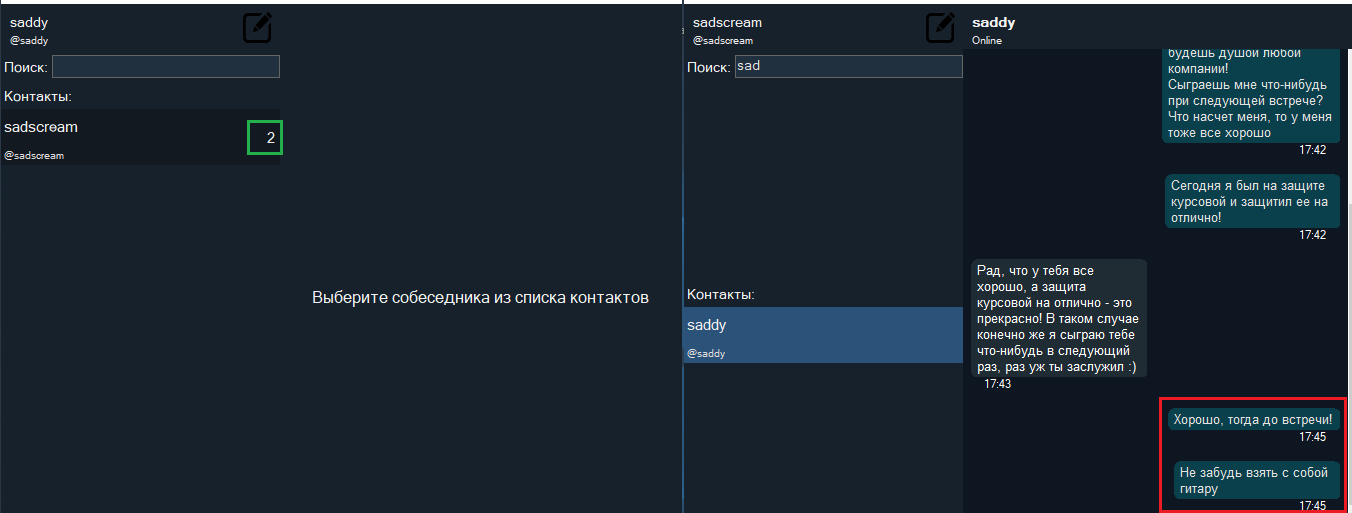


Рисунок 22 Непрочитанные сообщения

При нажатии по «Панели информации о пользователе» (выделена красным на рисунке 19) пользователю sadscream будет предоставлена доступная информация о пользователе saddy [Рисунок 23].

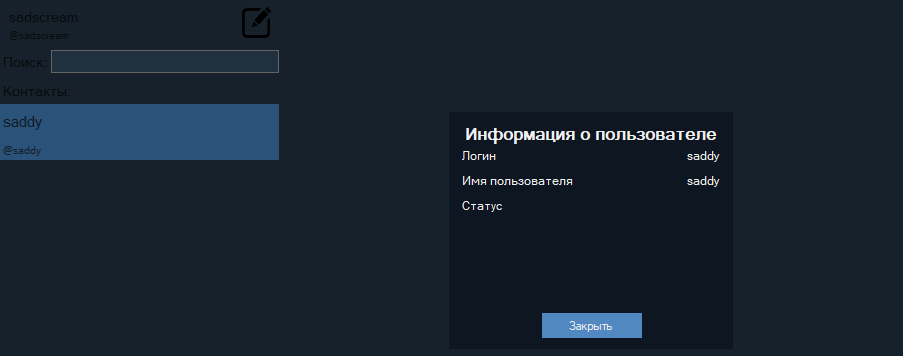


Рисунок 23 Информация о пользователе

Пользователь может изменять информацию о себе. Для этого ему нужно нажать на «Кнопку редактирования профиля» (выделена голубым на рисунке 16). После этого перед ним будет открыто окно редактирования информации о себе [Рисунок 24].

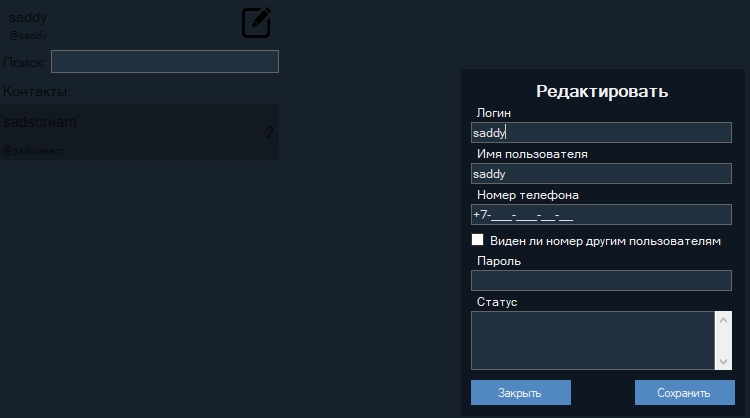


Рисунок 24 Окно редактирования

Немного отредактируем данные. Имя пользователя поменяем на «Илья», номер телефона изменим на «+7-924-111-11-11», а также подтвердим видимость номера для других пользователей. В статусе напишем «Привет, мир!» [Рисунок 25].

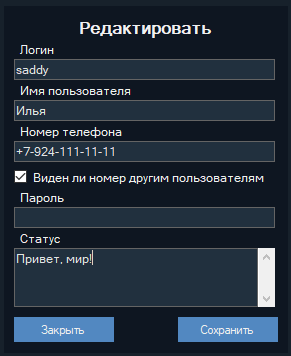


Рисунок 25 Редактирование профиля

Теперь нажмем кнопку «Сохранить». После этого информация о пользователе обновится и для прочих пользователей также старая информация о пользователе saddy будет заменена обновленной [Рисунок 26].

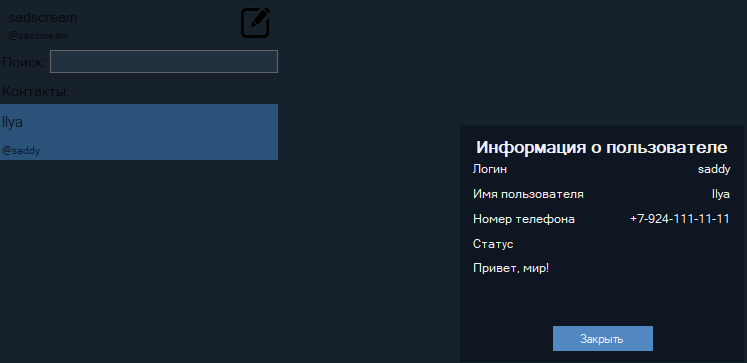


Рисунок 26 Обновленная информация о пользователе

При нажатии правой кнопкой мыши по пользователю из списка контактов будет открыто контекстное меню, в котором пользователь может удалить контакт [Рисунок 27].

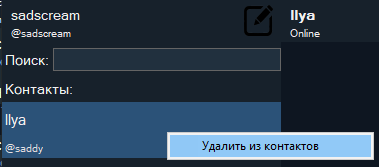


Рисунок 27 Удаление контакта

После нажатия пользователь saddy будет удален из списка контактов пользователя sadscream. А пользователь sadscream в свою очередь – из списка контактов saddy [Рисунок 28].



Рисунок 28 Результат удаления контакта

Попробуем от лица пользователя sadscream найти пользователя saddy, но теперь уже по первым двум буквам имени [Рисунок 29].

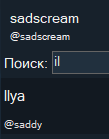


Рисунок 29 Поиск по имени

Или же произведем поиск по первым 4 цифрам номера телефона [Рисунок 30].

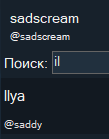


Рисунок 30 Поиск по номеру телефона

## Описание серверной части

В соответствии с архитектурой ПО, сервер разделен на 5 компонентов, предоставляющих URL-маршруты для различных запросов.

### Авторизация и регистрация

Предоставляет два маршрута: /api/sign-up и /api/sign-in.

Запрос к /sign-up/ требует HTTP-метод POST. В теле запроса должен содержаться JSON-объект с полями «login» - логин и «password» - пароль. В случае удачной регистрации будет возвращен JSON-объект с полем «ok» равным True.

В случае неудачи будет возвращен JSON с полем «ok» равным False. Также в возвращаемом JSON будет поле «message» со словесным описанием возникшей ошибки. Если неудача вызвана тем, что пользователь в данный момент авторизован, то HTTP-код ошибки будет равен 403. Если регистрируемый логин уже занят, то код ошибки будет 409. Если логин или пароль не соответствуют требованиям, то код ошибки будет 422.

Запрос к /sign-in/ требует метод GET. Для аутентификации используется Basic-метод, поэтому логин и пароль должны передаваться в HTTP-заголовке «Authorization» в виде пары «логин:пароль» в кодировке Base64 указанием метода аутентификации. В случае удачной авторизации будет возвращен JSON с полем «ok» равным True. Такой же результат будет возвращен при авторизации, если авторизация уже произведена. В таком случае JSON также будет иметь поле «message» с описанием ошибки.

При ошибке будет возвращен JSON с полями «ok» = False и «message» с соответствующим сообщением об ошибке. В случае если ошибка вызвана тем, что введенный логин не зарегистрирован в системе или пароль не соответствует паролю учетной записи с таким логином, то HTTP-код ошибки будет равен 403. Если для аутентификации будет использован метод, отличный от Basic-методы, то код ошибки будет равен 401.

### Контакты

Данный компонент предоставляет маршруты /api/contacts/ и /api/contacts/<user\_id>, где вместо <user\_id> должен быть передан уникальный идентификатор пользователя.

При GET-запросе к /contacts/ будет возвращен список всех контактов, которые имеются у авторизованного в данный момент пользователя, в виде JSON-объекта с полями «ok» = True и «contacts», где contacts является списком из словарей, содержащих в себе поля с информацией о каждом контакте.

Если запрос совершен не авторизованным пользователем, то будет возвращен JSON-объект с полем «ok» = False и HTTP-кодом ошибки 401. Это же касается всех остальных перечисленных ниже URL-маршрутов. Также при обращении к этому и всем нижеперечисленным маршрутам в базе данных будет обновляться информация о дате последнего посещения пользователя на дату запроса соответственно.

При POST-запросе к /contacts/ необходима передача в теле запроса JSON-объекта, содержащего единственное поле «user\_id» - идентификатором некоторого пользователя. Пользователь с таким идентификатором будет добавлен в список контактов текущего авторизованного пользователя. Также текущий пользователь будет добавлен в список контактов добавляемого пользователя. При удачном добавлении будет возвращен JSON с полем «ok» = True.

При неудачной попытке будет возвращен JSON с полем «ok» = False и полем «message», содержащим соответствующее сообщение об ошибке. Если неудача вызвана попыткой добавить в контакты самого себя, то HTTP-код ошибки будет равен 405. Если добавляемый пользователь не будет равен, то код будет 404.

При DELETE-запросе к маршруту /contacts/<user\_id> из списка контактов текущего пользователя будет удален пользователь с передаваемым идентификатором, а в качестве ответ возвращен JSON с полем «ok» = True. Если пользователь с таким идентификатором не будет найден, то поле «ok» в JSON ответе примет значение False, а в поле «message» будет указано соответствующее сообщение об ошибке. HTTP-код ошибки в таком случае – 404.

### Сообщения

Данный модуль предоставляет 2 маршрута: /api/messages/ и /api/messages/history/<user\_id>/, где <user\_id> - идентификатор пользователя.

При POST-запросе к /messages/ в теле запроса должен быть передан JSON с полями recipient\_id, text, date. Здесь recipient\_id – идентификатор пользователя. Пользователю с таким идентификатором мы передаем сообщение. Поле text – текст сообщения, а поле date – дата отправки сообщения в виде числа с плавающей точкой - UNIX отметки времени. В случае успеха информация о том, что текущий пользователь передал пользователю с идентификатором «recipient\_id» сообщение, будет сохранена. При этом если пользователь, которому назначается сообщение, отсутствует в списке контактов текущего пользователя, то он будет в него добавлен. В качестве ответа будет возвращен JSON с полем «ok» = True.

При неудаче поле «ok» JSON-ответа будет равно False, а в поле «message» будет храниться соответствующее сообщение об ошибке. Если ошибка вызвана несоответствием сообщения требованиям, HTTP-код ошибки будет равен 422. Если пользователь с идентификатором «recipient\_id» не найден, то код ошибки будет 404.

При GET-запросе к /messages / будет возвращен список всех сообщений, присланных текущему пользователю и которые еще не были им прочитаны. В JSON ответе будет поле «ok» = True и поле «messages», которое является списком из словарей с информацией о сообщениях.

При GET-запросе по маршруту /messages/history/<user\_id> будут возвращены все сообщения от пользователя с идентификатором равным «user\_id», которые предназначены текущему пользователю. При этом все возвращаемые сообщения будут помечены как прочитанные. В случае успеха в JSON-ответе поле «ok» будет равно True, а в поле «messages» будет содержаться список словарей с информацией о сообщениях.

Если не будет найден пользователь с соответствующим идентификатором, то будет возвращен JSON-ответ с полем «ok» равным True и полем «message» с соответствующим текстом ошибки. Код ошибки будет 404.

### Авторизованный пользователь

Этот компонент предоставляет один маршрут - /api/user/

При GET-запросе возвращает JSON с информацией об авторизованном в данный момент пользователе с полем «ok» = True и прочими полями, содержащими информацию о пользователе.

При POST-запросе в JSON теле запроса необходимо передать любые из следующих полей: login, username, phone\_number, phone\_visibility, password, status. Эти поля соответствуют: логину, псевдониму, номеру телефона, опции общедоступности номера телефона, паролю и текстовому статусу пользователя. При успешном исполнении у текущего пользователя будут обновлены соответствующие поля на то, что было передано в запросе. Ответ будет содержать JSON с полем «ok» = True.

В случае ошибки в JSON ответе поле «ok» будет равно False, а в поле «message» будет текст соответствующей ошибки. Если ошибка вызвана тем, что переданный логин уже занят другим пользователем систему, то возвращаемый HTTP-код ошибки будет равен 409. В любом другом случае, если какое-либо поле не соответствует требованиям, код ошибки будет равен 422.

### Пользователи

В этом компоненте представлен один маршрут - /api/users/search/<query>, где query – строка, по которой будет производиться поиск среди пользователей.

По этому маршруту можно совершить только GET-запрос. В соответствии с переданной строкой query будет искаться пользователи, логин, псевдоним или номер которых начинается с тех же символов, что и строка query, вне зависимости от регистра букв. Для поиска по номеру будут также игнорироваться все символы, кроме цифр. В ответе будет возвращен JSON-объект с полем «ok» = True и полем «users», содержащим в себе список из словарей, состоящих из информации о пользователях, удовлетворяющих критериям поиска. Пользователи, которые находятся в списке контактов текущего пользователя будут проигнорированы.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения работы был проведен анализ состояния рынка мессенджеров, который показал, что мессенджеры могут приносить большую прибыль, а также пользуются большой популярностью среди населения, что отражает актуальность выбранной темы.

Для решения поставленной задачи был произведен обзор существующих аналогов, выбраны подходящие инструменты, а также был спроектирован ряд требований. В результате удалось разработать готовый продукт, удовлетворяющий всем требованиям и реализующий все необходимые функции.

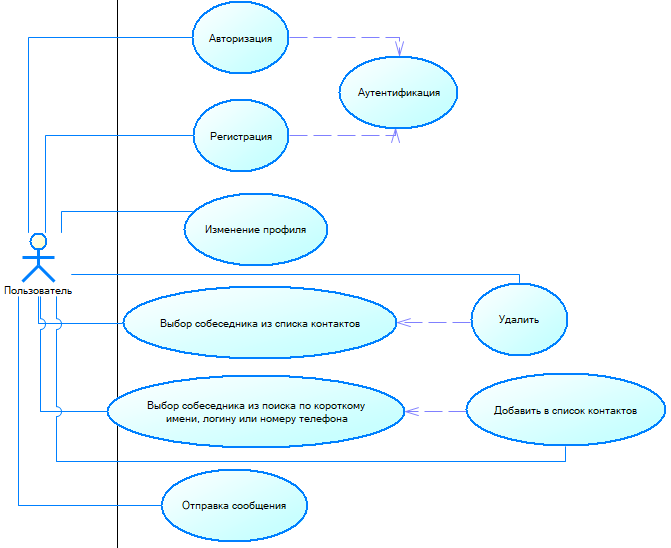
Мессенджер разрабатывался как клиент-серверное приложение. Для реализации серверной части ПО была выбрана среда разработки PyCharm, язык программирования Python с фреймворком Flask и для хранения информации была выбрана база данных MySQL. Клиентская часть разрабатывалась в среде VisualStudio на языке C# с использованием библиотеки Windows Forms.

В итоге основная цель работы была достигнута – был разработан мессенджер под платформу ОС Windows, предоставляющий все необходимые для общения возможности. В дальнейшем в приложение планируется вносить ряд доработок и оптимизаций, а также пополнять список возможностей как на стороне клиента, так и на стороне сервера.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аналитическая платформа «Statista» [Электронный ресурс] : Leading mobile apps worldwide in 2021 – 26.02.2022. Режим доступа - https://www.statista.com/statistics/1285960/top-downloaded-mobile-apps-worldwide/
2. Свободная энциклопедия Википедия [Электронный ресурс] : Чат – 26.02.2022. Режим доступа – <https://ru.wikipedia.org/wiki/Чат_(программа)>
3. Официальный блог WhatsApp [Электронный ресурс] : Два миллиарда пользователей: конфиденциальная связь по всему миру – 26.02.2022. Режим доступа - <https://blog.whatsapp.com/two-billion-users-connecting-the-world-privately>
4. Исследовательский центр компании «Делойт» в СНГ [Электронный ресурс] : Медиапотребление в России – 2021 – 26.02.2022. Режим доступа - <https://www2.deloitte.com/content/dam/Deloitte/ru/Documents/technology-media-telecommunications/russian/mediaconsupmtion-russia-2021.pdf>
5. Исследовательский центр компании «Делойт» в СНГ [Электронный ресурс] : Медиапотребление в России – 2019 – 26.02.2022. Режим доступа - <https://www2.deloitte.com/ru/ru/pages/technology-media-and-telecommunications/articles/media-consumption-in-russia-2019.html>
6. Официальный сайт Snapchat [Электронный ресурс] : Официальная презентация Snap Inc. для инвесторов – 26.02.2022. Режим доступа - <https://s25.q4cdn.com/442043304/files/doc_financials/2021/q4/Q4'21-Earnings-Slides.pdf>
7. Официальный сайт журнала Forbes [Электронный ресурс] : How Much Revenue Can WhatsApp Generate? – 26.02.2022. Режим доступа - <https://www.forbes.com/sites/greatspeculations/2017/11/10/how-much-revenue-can-whatsapp-generate/?sh=7364e7e02f2c>
8. Свободная энциклопедия Википедия [Электронный ресурс] : Telegram – 26.02.2022. Режим доступа - <https://ru.wikipedia.org/wiki/Telegram>
9. Свободная энциклопедия Википедия [Электронный ресурс] : MTProto – 26.02.2022. Режим доступа - https://ru.wikipedia.org/wiki/MTProto
10. Свободная энциклопедия Википедия [Электронный ресурс] : WhatsApp – 26.02.2022. Режим доступа - https://ru.wikipedia.org/wiki/WhatsApp
11. Новостное издание «The Guardian» [Электронный ресурс] : Jeff Bezos hack: Amazon boss's phone 'hacked by Saudi crown prince' – 26.02.2022. Режим доступа - <https://www.theguardian.com/technology/2020/jan/21/amazon-boss-jeff-bezoss-phone-hacked-by-saudi-crown-prince>
12. Материалы VIII Международной студенческой научной конференции «Студенческий научный форум» [Электронный ресурс] : Маркин Е.И., Рябова К.М., Артюшина Е.А. РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ АРХИТЕКТУРЫ «КЛИЕНТ-СЕРВЕР» - 29.04.2022. Режим доступа: <https://scienceforum.ru/2016/article/2016021199>
13. Калмыков, А. Ю. Архитектура клиент-сервер / А. Ю. Калмыков, Н. В. Синкевич, В. В. Денисов // Прошлое, настоящее и будущее Российской цивилизации : материалы всероссийской научно-практической конференции, Ставрополь, 28–29 апреля 2016 года. – Ставрополь: Общество с ограниченной ответственностью "Издательско-информационный центр "Фабула", 2016. – С. 164-168. – EDN WECXIF.
14. Платформа для исследования инструментов разработки IT-компаний [Электронный ресурс] : Companies using Python – 01.05.2022. Режим доступа - <https://stackshare.io/python>
15. Свободная энциклопедия Википедия [Электронный ресурс] : Python – 01.05.2022. Режим доступа - <https://ru.wikipedia.org/wiki/Python>
16. Свободная энциклопедия Википедия [Электронный ресурс] : C Sharp – 01.05.2022. Режим доступа - <https://ru.wikipedia.org/wiki/C_Sharp>
17. Шилдт Г. Полный справочник по C#/Г //Шилдт,–М.: Издательский дом «Вильямс. – 2004.
18. Официальная документация по C# [Электронный ресурс] : Различия между делегатами и событиями – 01.05.2022. Режим доступа - <https://docs.microsoft.com/ru-ru/dotnet/csharp/distinguish-delegates-events>
19. Официальный сайт JetBrains [Электронный ресурс] : PyCharm – 01.05.2022. Режим доступа - <https://www.jetbrains.com/ru-ru/pycharm/>
20. Аналитическая платформа «Statista» [Электронный ресурс] : Ranking of the most popular relational database management systems worldwide, as of January 2022 – 01.05.2022. Режим доступа - <https://www.statista.com/statistics/1131568/worldwide-popularity-ranking-relational-database-management-systems/>
21. Разработка веб-приложений с использованием Flask на языке Python / Под ред. Мигель Гринберг – O’REILLY/ДМК Пресс, 2014. С. 6–10.

Приложение А. Диаграмма прецедентов



Приложение Б. Код сервер

Конфигурация. Файл config.py

USERNAME = "root"

PASSWORD = "1234"

HOST = "127.0.0.1:3306"

DB\_NAME = "plum\_db"

APPNAME = "Messenger"

SECRET\_KEY = "test secret key"

Модуль Flask-приложения. Файл app.py

**import** datetime

**from** flask **import** Flask

**from** flask\_migrate **import** Migrate

**from** flask\_login **import** LoginManager, current\_user

**from** sqlalchemy **import** exc

**from** config **import** \*

**from** models.db\_context **import** db, User

**from** tools.response **import** make\_response

**from** api **import** user, users, contact, message, authorization

app = Flask(\_\_name\_\_)

app.config['SQLALCHEMY\_DATABASE\_URI'] = f'mysql://{USERNAME}:{PASSWORD}@{HOST}/{DB\_NAME}'

app.config['SECRET\_KEY'] = SECRET\_KEY

app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False

db.init\_app(app)

login\_manager = LoginManager()

login\_manager.init\_app(app)

migrate = Migrate(app, db, compare\_type=True)

app.register\_blueprint(user.user\_api, url\_prefix="/api")

app.register\_blueprint(users.users\_api, url\_prefix="/api")

app.register\_blueprint(contact.contact\_api, url\_prefix="/api")

app.register\_blueprint(message.msg\_api, url\_prefix="/api")

app.register\_blueprint(authorization.auth\_api, url\_prefix="/api")

@login\_manager.user\_loader

**def** load\_user(user\_id):

**try**:

**return** db.session.query(User).get(user\_id)

**except** exc.OperationalError:

**return** db.session.query(User).get(user\_id)

@app.before\_request

**def** update\_last\_visit():

**if** current\_user.is\_authenticated:

current\_user.last\_visit = datetime.datetime.utcnow()

db.session.commit()

@app.route("/", methods=['POST', 'GET'])

**def** main\_page():

**return** {"ok": True}

**def** unauthorized():

**return** make\_response({"ok": False}, 401)

login\_manager.unauthorized\_callback = unauthorized

**if** \_\_name\_\_ == "\_\_main\_\_":

app.run()

Модели таблиц базы данных. Файл db\_context.py

**import** datetime

**from** flask\_sqlalchemy **import** SQLAlchemy

**from** sqlalchemy.dialects.mysql **import** DATETIME

**from** flask\_login **import** UserMixin

**from** werkzeug.security **import** generate\_password\_hash, check\_password\_hash

db = SQLAlchemy()

user\_contact\_association = db.Table('user\_contact', db.Model.metadata,

db.Column('user\_id', db.ForeignKey('user.user\_id'), index=True),

db.Column('contact\_id', db.ForeignKey('user.user\_id')),

db.UniqueConstraint('user\_id', 'contact\_id', name='unique\_contacts')

)

**class** User(db.Model, UserMixin):

user\_id = db.Column(db.Integer, primary\_key=True)

login = db.Column(db.String(128), nullable=False, unique=True)

username = db.Column(db.String(128), nullable=False)

phone\_number = db.Column(db.String(16), nullable=True, unique=True)

phone\_visibility = db.Column(db.Boolean, nullable=False, default=False)

password\_hash = db.Column(db.String(256), nullable=False)

status = db.Column(db.String(256), nullable=True)

last\_visit = db.Column(db.DateTime, nullable=False, default=datetime.datetime.now())

messages = db.relationship('Message', cascade="all,delete", backref='owner', lazy=True, foreign\_keys="[Message.owner\_id]")

contacts = db.relationship('User',cascade="all,delete",secondary=user\_contact\_association, primaryjoin=user\_id==user\_contact\_association.c.user\_id, secondaryjoin=user\_id==user\_contact\_association.c.contact\_id)

**def** add\_contact(self, contact):

**if** contact **not** **in** self.contacts:

self.contacts.append(contact)

contact.contacts.append(self)

**def** remove\_contact(self, contact):

**if** contact **in** self.contacts:

self.contacts.remove(contact)

contact.contacts.remove(self)

**def** get\_id(self):

**return** self.user\_id

**def** set\_password(self, password):

self.password\_hash = generate\_password\_hash(password)

**def** check\_password(self, password):

**return** check\_password\_hash(self.password\_hash, password)

**class** Message(db.Model):

message\_id = db.Column(db.Integer, primary\_key=True)

owner\_id = db.Column(db.Integer,db.ForeignKey("user.user\_id",ondelete='CASCADE'))

recipient\_id = db.Column(db.Integer,db.ForeignKey("user.user\_id",ondelete='CASCADE'))

recipient = db.relationship("User", foreign\_keys=[recipient\_id])

text = db.Column(db.String(4096))

date = db.Column(DATETIME(fsp=6), nullable=False)

read = db.Column(db.Boolean, nullable=False, default=False)

Функция формирования ответа. Файл response.py

**from** flask **import** make\_response

**from** json **import** dumps

**def** json\_response(data: dict, code: int = 200):

response = make\_response(dumps(data, ensure\_ascii=False), code)

response.headers["Content-Type"] = "application/json"

**return** response

Компонент авторизации. Файл authorization.py

**import** datetime, time

**from** flask **import** Blueprint, request

**from** flask\_login **import** login\_required, current\_user, logout\_user, login\_user

**from** api.users **import** check\_login, check\_password

**from** tools.response **import** \*

**from** models.db\_context **import** db, User, Message

auth\_api = Blueprint('auth\_api', \_\_name\_\_)

@auth\_api.route('/sign-up/', methods=['POST'])

**def** sign\_up():

data = {"ok": False}

**if** current\_user.is\_authenticated:

data["message"] = "Вы авторизованы"

**return** json\_response(data, 403)

r = request.json

login = r['login']

password = r['password']

user\_obj = db.session.query(User).filter(User.login == login).first()

**if** user\_obj:

data["message"] = "Этот логин уже занят"

**return** json\_response(data, 409)

**if** check\_login(login) **and** check\_password(password):

user = User(login=login, username=login)

user.set\_password(password)

db.session.add(user)

db.session.commit()

data["ok"] = True

**return** json\_response(data)

data["message"] = ("Логин должен быть не короче 3 символов и не длиннее 16, а также состоять только из букв латинского алфавита, цифр и символов \*\_()#!&.**\n**Пароль должен быть длиной не менее 6 символов и не более 32, а также состоять только из букв латинского алфавита и цифр")

**return** json\_response(data, 422)

@auth\_api.route('/sign-in/', methods=['GET'])

**def** sign\_in():

data = {"ok": False}

**if** current\_user.is\_authenticated:

data["ok"] = True

data["message"] = "Вы уже авторизованы"

**return** json\_response(data, 200)

**if** request.authorization:

login = request.authorization.username

password = request.authorization.password

user\_obj = db.session.query(User).filter(User.login == login).first()

**if** user\_obj **and** user\_obj.check\_password(password):

login\_user(user\_obj)

data["ok"] = True

data["user\_id"] = user\_obj.user\_id

**return** json\_response(data)

data["message"] = "Неправильный логин или пароль"

**return** json\_response(data, 403)

data["message"] = "BasicAuth needs"

**return** json\_response(data, 401)

@auth\_api.route('/logout/')

@login\_required

**def** logout():

logout\_user()

**return** json\_response({"ok": True})

Компонент отвечающий за контакты. Файл contact.py

**import** datetime, time

**from** flask **import** Blueprint, request

**from** flask\_login **import** login\_required, current\_user

**from** typing **import** List

**from** api.user **import** user\_to\_json

**from** tools.response **import** \*

**from** models.db\_context **import** db, User, Message

contact\_api = Blueprint('contact\_api', \_\_name\_\_)

@contact\_api.route("/contacts/", methods=['GET'])

@login\_required

**def** get\_contacts():

data = {"ok": True,"contacts": list(map(user\_to\_json, current\_user.contacts))}

**return** data

@contact\_api.route("/contacts/", methods=['POST'])

@login\_required

**def** add\_contact():

data = {"ok": False}

contact\_id = request.json["user\_id"]

**if** current\_user.user\_id == contact\_id:

data["message"] = "Нельзя добавлять в контакты самого себя"

**return** json\_response(data, 405)

contact = db.session.query(User).filter(User.user\_id == contact\_id).first()

**if** contact:

current\_user.add\_contact(contact)

db.session.commit()

data["ok"] = True

**return** json\_response(data)

data["message"] = "Пользователь не найден"

**return** json\_response(data, 404)

@contact\_api.route("/contacts/<int:user\_id>", methods=['DELETE'])

@login\_required

**def** delete\_contact(user\_id):

data = {"ok": False}

contact = db.session.query(User).filter(User.user\_id == user\_id).first()

**if** contact:

current\_user.remove\_contact(contact)

db.session.commit()

data["ok"] = True

**return** json\_response(data)

data["message"] = "Пользователь не найден"

**return** json\_response(data, 404)

Компонент сообщений. Файл message.py

**import** datetime

**from** sqlalchemy **import** or\_, and\_, desc, asc

**from** sqlalchemy.orm **import** contains\_eager

**from** flask **import** Blueprint, request

**from** flask\_login **import** login\_required, current\_user, logout\_user

**from** typing **import** List

**from** tools.response **import** \*

**from** models.db\_context **import** db, User, Message

msg\_api = Blueprint('msg\_api', \_\_name\_\_)

@msg\_api.route("/messages/", methods=['POST'])

@login\_required

**def** send\_message():

data = {"ok": False}

r = request.json

text = r["text"]

**if** **not** check\_message(text):

data["message"] = "Сообщение не может содержать 0 или более 1000 символов."

**return** json\_response(data, 422)

date = datetime.datetime.fromtimestamp(r["date"])

recipient = db.session.query(User).filter(User.user\_id == r["recipient\_id"]).first()

**if** recipient **and** recipient != current\_user:

**if** recipient **not** **in** current\_user.contacts:

current\_user.add\_contact(recipient)

message\_obj = Message(owner=current\_user, recipient=recipient, text=text, date=date)

db.session.add(message\_obj)

db.session.commit()

data["ok"] = True

**return** json\_response(data)

data["message"] = "Пользователь не найден"

**return** json\_response(data, 404)

@msg\_api.route("/messages/", methods=['GET'])

@login\_required

**def** get\_new\_messages():

data = {"ok": True}

list\_messages: List[Message] = db.session.query(Message).filter(

and\_(Message.recipient == current\_user, Message.read.is\_(False), Message.owner\_id.in\_(u.user\_id **for** u **in** current\_user.contacts))).all()

data["messages"] = list(map(message\_to\_json, list\_messages))

**return** json\_response(data)

@msg\_api.route("/messages/history/<int:user\_id>", methods=['GET'])

@login\_required

**def** get\_contact\_messages(user\_id):

data = {"ok": False}

user\_obj: User = db.session.query(User).filter(User.user\_id == user\_id).first()

**if** **not** user\_obj:

data["message"] = "Пользователя с таким идентификатором не существует"

**return** json\_response(data, 404)

list\_messages\_query = db.session.query(Message).filter(or\_(and\_(Message.owner == user\_obj,Message.recipient == current\_user,Message.owner\_id.in\_(u.user\_id **for** u **in** current\_user.contacts)),and\_(Message.recipient == user\_obj,Message.owner == current\_user,Message.recipient\_id.in\_(u.user\_id **for** u **in** current\_user.contacts)))).order\_by(asc(Message.date))

list\_messages: List[Message] = list\_messages\_query.all()

data["ok"] = True

data["messages"] = list(map(**lambda** message: message\_to\_json(message, True **if** message.owner != current\_user **else** False), list\_messages))

db.session.commit()

**return** json\_response(data)

**def** message\_to\_json(message\_obj: Message, set\_read: bool = False) -> dict:

message\_obj.read = set\_read **and** **not** message\_obj.read

data = {"message\_id": message\_obj.message\_id,"owner\_id": message\_obj.owner\_id,"recipient\_id": message\_obj.recipient\_id,"text": message\_obj.text,"date": message\_obj.date.strftime("%d.%m.%Y, %H:%M:%S.%f"),"read": message\_obj.read}

**return** data

**def** check\_message(text: str): **return** 1 <= len(text) <= 1000

Компонент, отвечающий за пользователя. Файл user.py

**import** datetime, time

**from** flask **import** Blueprint, request

**from** flask\_login **import** login\_required, current\_user

**from** typing **import** List

**from** api.users **import** user\_to\_json, check\_login, check\_username, check\_password, check\_field\_length

**from** tools.response **import** \*

**from** models.db\_context **import** db, User

user\_api = Blueprint('user\_api', \_\_name\_\_)

@user\_api.route("/user/", methods=['GET'])

@login\_required

**def** get\_current\_user():

data = {"ok": True}

data.update(user\_to\_json(current\_user))

**return** json\_response(data)

@user\_api.route("/user/", methods=['POST'])

@login\_required

**def** update\_fields():

data = {"ok": False}

json: dict = request.json

field\_func = {"login": change\_login,"username": change\_username,"phone\_number": change\_phone\_number,"phone\_visibility": change\_phone\_visibility,"password": change\_password,"status": change\_status}

changed = False

**for** field **in** field\_func.keys():

**if** json.get(field) **is** **not** None **and** \

(**not** hasattr(current\_user, field) **or** json.get(field) != current\_user.\_\_getattr\_\_(field)):

**if** field == "password" **and** (len(json.get("password")) == 0 **or** current\_user.check\_password(json.get("password"))):

**continue**

changed = True

res = field\_func[field](json.get(field), data)

**if** res != 200:

db.session.commit()

**return** json\_response(data, res)

**if** **not** changed:

data["ok"] = True

data["message"] = "Поля уже обновлены"

db.session.commit()

**return** json\_response(data)

**def** change\_login(new\_login, data) -> int:

user\_obj = db.session.query(User).filter(User.login == new\_login).first()

**if** user\_obj:

data["message"] = "Этот логин уже используется"

**return** 409

**elif** check\_login(new\_login):

current\_user.login = new\_login

data["ok"] = True

**return** 200

data["message"] = ("Неправильный формат логина. Логин должно быть не короче 3 символов и не длиннее 16, а также состоять из букв латинского алфавита, цифр или символов \*\_()#!&")

**return** 422

**def** change\_username(new\_username, data):

**if** check\_username(new\_username):

current\_user.username = new\_username

data["ok"] = True

**return** 200

data["message"] = ("Неправильный формат имени. Имя должно быть не короче 3 символов и не длиннее 16, а также состоять из букв латинского алфавита, цифр или символов \*\_ ()#!&")

**return** 422

**def** change\_phone\_number(new\_phone\_number, data):

**if** check\_field\_length(new\_phone\_number, 11, 16):

current\_user.phone\_number = new\_phone\_number

data["ok"] = True

**return** 200

data["message"] = ("Неправильный номер телефона. Номер телефона не может быть короче 11 символов, а также не должен быть длиннее 16")

**return** 422

**def** change\_phone\_visibility(new\_phone\_visibility, data):

current\_user.phone\_visibility = new\_phone\_visibility

data["ok"] = True

**return** 200

**def** change\_password(new\_password, data):

**if** check\_password(new\_password):

current\_user.set\_password(new\_password)

data["ok"] = True

**return** 200

data["message"] = ("Неправильный формат пароля. Пароль должен быть длиной не менее 6 символов и не более 32, а также состоять только из букв латинского алфавита и цифр")

**return** 422

**def** change\_status(new\_status, data):

**if** check\_field\_length(new\_status, max\_len=139):

current\_user.status = new\_status

data["ok"] = True

**return** 200

data["message"] = "Статус не может быть длиннее 139 символов"

**return** 422

Компонент, отвечающий за пользователей. Файл users.py

**import** datetime, time

**from** flask **import** Blueprint, request

**from** sqlalchemy **import** or\_, and\_, desc, asc

**from** flask\_login **import** login\_required, current\_user

**from** typing **import** List

**from** sqlalchemy.sql.functions **import** user

**from** string **import** ascii\_letters

**from** tools.response **import** \*

**from** models.db\_context **import** db, User

users\_api = Blueprint('users\_api', \_\_name\_\_)

@users\_api.route("/users/", methods=['GET'])

@login\_required

**def** get\_users():

list\_users = db.session.query(User).all()

data = {"ok": True,"users": list(map(user\_to\_json, list\_users))}

**return** data

@users\_api.route("/users/search/<string:query>", methods=['GET'])

@login\_required

**def** search\_users\_by\_query(query):

**def** remove\_non\_digit(s: str):

**return** ''.join(filter(str.isdigit, s))

users: List[User] = db.session.query(User).filter(and\_(User.user\_id != current\_user.user\_id,User.user\_id.not\_in(u.user\_id **for** u **in** current\_user.contacts), or\_(

User.username.like(f"{query}%"),

User.login.like(f"{query}%"), and\_(User.phone\_visibility.is\_(True),User.phone\_number.regexp\_replace('[^[:digit:]]', '').like(f"{remove\_non\_digit(query)}%"))))).all()

data = {"ok": True,"users": list(map(user\_to\_json, users))}

**return** json\_response(data)

**def** user\_to\_json(user\_obj: User) -> dict:

data = {"user\_id": user\_obj.user\_id,"login": user\_obj.login,"username": user\_obj.username,"phone\_visibility": user\_obj.phone\_visibility,"status": user\_obj.status,"last\_visit": user\_obj.last\_visit.strftime("%d.%m.%Y, %H:%M")}

**if** user\_obj.user\_id == current\_user.user\_id **or** user\_obj.phone\_visibility:

data.update({"phone\_number": user\_obj.phone\_number})

**return** data

**def** check\_field\_length(check\_str: str, min\_len: int = 0, max\_len: int = 1000):

**return** min\_len <= len(check\_str) <= max\_len

**def** check\_password(password: str):

**return** check\_field\_length(password, 6, 32) **and** \

all(c.isdigit() **or** c **in** ascii\_letters **for** c **in** password)

**def** check\_username(username: str):

**return** check\_login(username, "\*\_ ()#!&")

**def** check\_login(login: str, symbol\_alphabet: str = None):

**if** symbol\_alphabet **is** None:

symbol\_alphabet = "\*\_()#!&"

**return** check\_field\_length(login, 3, 16) **and** all(c.isdigit() **or** c **in** ascii\_letters **or** c **in** symbol\_alphabet **for** c **in** login)

Приложение Б. Код приложения

Точка входа в приложение. Файл Program.cs

**using** System;

**using** System.Windows.Forms;

**namespace** PlumMessenger {

**static** **class** Program {

[STAThread]

**static** **void** Main() {

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(**false**);

Application.Run(new MainF());}}}

Класс пользователя. Файл User.cs

**using** Newtonsoft.Json.Linq;

**using** System;

**using** System.Globalization;

**namespace** PlumMessenger.Models {

**public** **class** User {

**public** **event** EventHandler EditEvent;

**private** **string** login;

**private** **string** username;

**private** **string** phoneNumber;

**private** **string** status;

**private** DateTime lastVisit;

**private** **bool** phonePublicy;

**public** **string** password;

**public** **int** Id { **get**; **set**; }

**public** **string** Login{**get**{**return** login;}

**set**{

**if** (**value** == login) **return**;

EditEvent?.Invoke(**this**, EventArgs.Empty);

login = **value**;}}

**public** **string** Username{**get**{**return** username;}

**set**{

**if** (**value** == username) **return**;

EditEvent?.Invoke(**this**, EventArgs.Empty);

username = **value**;}}

**public** **string** PhoneNumber{**get**{**return** phoneNumber;}

**set**{

**if** (**value** == phoneNumber) **return**;

EditEvent?.Invoke(**this**, EventArgs.Empty);

phoneNumber = **value**;}}

**public** **string** Status{**get**{**return** status;}

**set**{

**if** (**value** == status) **return**;

EditEvent?.Invoke(**this**, EventArgs.Empty);

status = **value**;}}

**public** DateTime LastVisit{**get**{**return** lastVisit;}

**set**{

EditEvent?.Invoke(**this**, EventArgs.Empty);

lastVisit = **value**;}}

**public** **bool** PhonePublicy{**get**{**return** phonePublicy;}

**set**{

**if** (**value** == phonePublicy) **return**;

EditEvent?.Invoke(**this**, EventArgs.Empty);

phonePublicy = **value**;}}

**public** **static** User FromJson(JObject json, **bool** ignorePhonePublicy = **false**){

User contact = new User();

contact.Id = (**int**)json["user\_id"];

contact.Login = (**string**)json["login"];

contact.Username = (**string**)json["username"];

contact.PhonePublicy = (**bool**)json["phone\_visibility"];

**if** (contact.PhonePublicy || ignorePhonePublicy){

contact.PhoneNumber = (**string**)json["phone\_number"];}

contact.Status = (**string**)json["status"];

DateTime dt = DateTime.ParseExact((**string**)json["last\_visit"], "%dd.%MM.%yyyy, %HH:%mm", CultureInfo.CurrentCulture);

contact.LastVisit = dt;

**return** contact;}

**internal** **static** JObject ToJson(User currentUser){

JObject json = new JObject();

json["user\_id"] = currentUser.Id;

json["login"] = currentUser.Login;

json["username"] = currentUser.Username;

json["phone\_visibility"] = currentUser.PhonePublicy;

json["phone\_number"] = currentUser.PhoneNumber;

json["status"] = currentUser.Status;

json["password"] = currentUser.password;

**return** json;}

**public** **bool** IsOnline(){

**if** ((DateTime.UtcNow - LastVisit).TotalMinutes > 5) **return** **false**;

**return** **true**;}

**public** **static** **void** UpdateFromUser(User to, User **from**){

to.Login = **from**.Login;

to.Username = **from**.Username;

to.PhonePublicy = **from**.PhonePublicy;

to.phoneNumber = **from**.PhoneNumber;

to.Status = **from**.Status;

to.LastVisit = **from**.LastVisit;}}}

Класс контакта. Файл Contact.cs

**using** Newtonsoft.Json.Linq;

**using** System;

**using** System.Collections.Generic;

**using** System.Collections.ObjectModel;

**using** System.Collections.Specialized;

**using** System.Globalization;

**using** System.Linq;

**namespace** PlumMessenger.Models{

**public** **class** Contact : User{

**public** **event** EventHandler MessageAddedEvent;

**public** **event** EventHandler UnreadMessageAddedEvent;

**public** **event** EventHandler UnreadMessageRemovedEvent;

**private** ObservableCollection<Message> unreadMessages = new ObservableCollection<Message>();

**private** ObservableCollection<Message> messages = new ObservableCollection<Message>();

**public** Contact() : **base**(){

messages.CollectionChanged += MessagesModifiedEvent;

unreadMessages.CollectionChanged += UnreadMessagesModifiedEvent;

MessageAddedEvent += MessageAdded;}

**public** ObservableCollection<Message> GetUnreadMessages() {**return** unreadMessages;}

**public** ObservableCollection<Message> GetMessages(){**return** messages;}

**public** **void** SetUnreadMessages(List<Message> newMessages){

**this**.unreadMessages = new ObservableCollection<Message>(newMessages);}

**public** **void** SetMessages(List<Message> messages){

**this**.messages = new ObservableCollection<Message>(messages);}

**private** **void** MessageAdded(**object** sender, EventArgs e){

NotifyCollectionChangedEventArgs addEvent = (NotifyCollectionChangedEventArgs)e;

**foreach** (Message message **in** addEvent.NewItems){

**if** (unreadMessages.**Select**(x => x.Id).Contains(message.Id)){

unreadMessages.**Remove**(unreadMessages.First(x => x.Id == message.Id));}}}

**private** **void** MessagesModifiedEvent(**object** sender, NotifyCollectionChangedEventArgs e){

**if** (e.Action == NotifyCollectionChangedAction.**Add**){

MessageAddedEvent?.Invoke(**this**, e);}

}

**private** **void** UnreadMessagesModifiedEvent(**object** sender, NotifyCollectionChangedEventArgs e){

**if** (e.Action == NotifyCollectionChangedAction.**Add**){

UnreadMessageAddedEvent?.Invoke(**this**, e);}

**else** **if** (e.Action == NotifyCollectionChangedAction.**Remove**){

UnreadMessageRemovedEvent?.Invoke(**this**, e);}}

**public** **static** Contact FromJson(JObject json){

Contact contact = new Contact();

contact.Id = (**int**)json["user\_id"];

contact.Login = (**string**)json["login"];

contact.Username = (**string**)json["username"];

contact.PhonePublicy = (**bool**)json["phone\_visibility"];

**if** (contact.PhonePublicy){

contact.PhoneNumber = (**string**)json["phone\_number"];}

contact.Status = (**string**)json["status"];

DateTime dt = DateTime

.ParseExact((**string**)json["last\_visit"], "%dd.%MM.%yyyy, %HH:%mm", CultureInfo.CurrentCulture);

contact.LastVisit = dt;

**return** contact;}}}

Класс сообщений. Файл Message.cs

**using** Newtonsoft.Json.Linq;

**using** System;

**namespace** PlumMessenger.Models{

**public** **class** Message{

**public** **event** EventHandler ReadEvent;

**private** **bool** read = **false**;

**public** **int** Id { **get**; **set**; }

**public** **int** OwnerId { **get**; **set**; }

**public** User Owner { **get**; **set**; }

**public** **int** RecipientId { **get**; **set**; }

**public** User Recipient { **get**; **set**; }

**public** **string** Text { **get**; **set**; }

**public** DateTime Date { **get**; **set**; }

**public** **bool** Read {**get**{ **return** read;}

**set**{

**if** (**value**) ReadEvent?.Invoke(**this**, EventArgs.Empty);

read = **value**;}}

**internal** **static** Message MessageFromJson(JObject json){

Message message = new Message();

message.Id = (**int**)json["message\_id"];

message.OwnerId = (**int**)json["owner\_id"];

message.RecipientId = (**int**)json["recipient\_id"];

message.Text = (**string**)json["text"];

DateTime dt = DateTime.ParseExact((**string**)json["date"], "dd.MM.yyyy, HH:mm:ss.ffffff", **null**);

message.Date = dt;

message.Read = (**bool**)json["read"];

**return** message;

}}}

Класс инициализации Http клиента. Файл Base.cs

**using** System;

**using** System.Net.Http;

**using** System.Net.Http.Headers;

**using** System.Net;

**namespace** PlumMessenger.Connection.Requests{

**public** **class** **Base**{

**protected** **Base**(**string** host) {

**string** Host = "http://127.0.0.1:5000";

**if** (ClientCreator.Client == **null**){

Host = host == **null** ? Host : host;

CookieContainer cookieContainer = new CookieContainer();

HttpClientHandler clienthandler = new HttpClientHandler {

AllowAutoRedirect = **true**,

UseCookies = **true**,

CookieContainer = cookieContainer

};

ClientCreator.Client = new HttpClient(clienthandler);

Client.BaseAddress = new Uri(Host);

Client.DefaultRequestHeaders.Accept.Clear();

Client.DefaultRequestHeaders.Accept.**Add**(new MediaTypeWithQualityHeaderValue("application/json"));

}

}

**private** **sealed** **class** ClientCreator{

**private** **static** HttpClient client = **null**;

**public** **static** HttpClient Client { **get** { **return** client; } **set** { client = **value**; } }

}

**public** **static** HttpClient Client{

**get** { **return** ClientCreator.Client; }

}

}

}

Родительский класс для запросов. Файл DefaultRequest.cs

**using** Newtonsoft.Json.Linq;

**using** PlumMessenger.Connection.Errors;

**using** System.Net;

**using** System.Net.Http;

**using** System.Text;

**using** System.Threading.Tasks;

**namespace** PlumMessenger.Connection.Requests{

**public** **abstract** **class** DefaultRequest : **Base**{

**protected** **string** ApiTemplate { **get**; **set**; } = "api/{0}";

**public** DefaultRequest(**string** host) : **base**(host){ }

**protected** **async** Task<JObject> GetJsonFromHttpContent(HttpContent content){

**string** result = **await** content.ReadAsStringAsync();

**return** JObject.Parse(result);

}

**protected** **async** **void** ConfirmAuthorization(HttpResponseMessage response){

**if** (!response.IsSuccessStatusCode){

JObject json = **await** GetJsonFromHttpContent(response.Content);

**if** (response.StatusCode == HttpStatusCode.Unauthorized){

**throw** new Unauthorized(json["message"].ToString());

}}}}}

Компонент запросов авторизации. Файл AuthRequest.cs

**using** System;

**using** System.Text;

**using** System.Threading.Tasks;

**using** System.Net.Http;

**using** System.Net.Http.Headers;

**using** PlumMessenger.Connection.Errors;

**using** Newtonsoft.Json.Linq;

**using** System.Net;

**namespace** PlumMessenger.Connection.Requests{

**public** **class** AuthRequest : DefaultRequest{

**public** **static** **bool** LoggedIn = **false**;

**string** ApiSignUp { **get**; **set**; }

**string** ApiSignIn { **get**; **set**; }

**string** ApiLogout { **get**; **set**; }

**public** AuthRequest(**string** host = **null**) : **base**(host){

ApiSignUp = **String**.Format(ApiTemplate, "sign-up/");

ApiSignIn = **String**.Format(ApiTemplate, "sign-in/");

ApiLogout = **String**.Format(ApiTemplate, "logout/");

}

**public** **async** Task SignUpMethod(**string** login, **string** password){

HttpResponseMessage response = **await** Client.PostAsJsonAsync(ApiSignUp, new {login = login,password = password});

**if** (!response.IsSuccessStatusCode){

JObject json = **await** GetJsonFromHttpContent(response.Content);

**if** ((**int**)response.StatusCode == 422){

**throw** new IncorrectLoginOrPasswordFormat(json["message"].ToString());

}

**else** **if** (response.StatusCode == HttpStatusCode.Conflict){

**throw** new LoginAlreadyTaken(json["message"].ToString());

}

**else** **if** (response.StatusCode == HttpStatusCode.Forbidden){

**throw** new Authorized(json["message"].ToString());

}

}

}

**public** **async** Task<**int**> SignInMethod(**string** login, **string** password){

**var** byteArray = Encoding.ASCII.GetBytes($"{login}:{password}");

Client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Basic", Convert.ToBase64String(byteArray));

HttpResponseMessage response = **await** Client.GetAsync(ApiSignIn);

JObject json = **await** GetJsonFromHttpContent(response.Content);

**if** (!response.IsSuccessStatusCode){

**if** (response.StatusCode == HttpStatusCode.Forbidden){

**throw** new InvalidLoginOrPassword(json["message"].ToString());

}

}

**else** **if** (json["message"] != **null**){

**throw** new Authorized(json["message"].ToString());

}

LoggedIn = **true**;

Client.DefaultRequestHeaders.Authorization = **null**;

**return** (**int**)json["user\_id"];

}

**public** **async** Task LogoutMethod(**string** login, **string** password){

HttpResponseMessage response = **await** Client.GetAsync(ApiLogout);

ConfirmAuthorization(response);

LoggedIn = **false**;

}

}

}

Компонент запросов контактов. Файл ContactRequest.cs

**using** Newtonsoft.Json.Linq;

**using** PlumMessenger.Connection.Errors;

**using** PlumMessenger.Models;

**using** System;

**using** System.Collections.Generic;

**using** System.Net;

**using** System.Net.Http;

**using** System.Threading.Tasks;

**namespace** PlumMessenger.Connection.Requests{

**public** **class** ContactsRequest : DefaultRequest{

**string** ApiContacts { **get**; **set**; }

**string** ApiContactsWithParam { **get**; **set**; }

**public** ContactsRequest(**string** host = **null**) : **base**(host){

ApiContacts = **String**.Format(ApiTemplate, "contacts/");

ApiContactsWithParam = **String**.Format(ApiTemplate, "contacts/{0}");

}

**public** **async** Task<List<Contact>> GetContacts(){

List<Contact> contacts = new List<Contact>();

HttpResponseMessage response = **await** Client.GetAsync(ApiContacts);

JObject json = **await** GetJsonFromHttpContent(response.Content);

**foreach** (JObject contact **in** json["contacts"]){

Contact parsedContact = Contact.FromJson(contact);

contacts.**Add**(parsedContact);

}

**return** contacts;

}

**public** **async** Task AddContact(**int** userId){

HttpResponseMessage response = **await** Client.PostAsJsonAsync(ApiContacts, new { user\_id = userId });

JObject json = **await** GetJsonFromHttpContent(response.Content);

**if** (!response.IsSuccessStatusCode){

**if** (response.StatusCode == HttpStatusCode.NotFound){

**throw** new UserNotFound(json["message"].ToString());

}

**else** **if** (response.StatusCode == HttpStatusCode.MethodNotAllowed){

**throw** new UserNotFound(json["message"].ToString());

}

}

}

**public** **async** Task DeleteContact(**int** userId){

HttpResponseMessage response = **await** Client.DeleteAsync(**String**.Format(ApiContactsWithParam, userId));

JObject json = **await** GetJsonFromHttpContent(response.Content);

**if** (!response.IsSuccessStatusCode){

**if** (response.StatusCode == HttpStatusCode.NotFound){

**throw** new UserNotFound(json["message"].ToString());

}}}}

}

Компонент запросов сообщений. Файл MessageRequest.cs

**using** Newtonsoft.Json.Linq;

**using** PlumMessenger.Models;

**using** System;

**using** System.Collections.Generic;

**using** System.Net;

**using** System.Net.Http;

**using** System.Threading.Tasks;

**using** PlumMessenger.Connection.Errors;

**namespace** PlumMessenger.Connection.Requests{

**class** MessagesRequest : DefaultRequest{

**string** ApiNewMessages { **get**; **set**; }

**string** ApiMessages { **get**; **set**; }

**public** MessagesRequest(**string** host = **null**) : **base**(host){

ApiNewMessages = **String**.Format(ApiTemplate, "messages/");

ApiMessages = **String**.Format(ApiTemplate, "messages/history/{0}");

}

**public** **async** Task<**bool**> SendMessage(**int** userId, **string** text){

HttpResponseMessage response = **await** Client.PostAsJsonAsync(ApiNewMessages, new{recipient\_id = userId,text = text,date = (DateTime.UtcNow - new DateTime(1970, 1, 1)).TotalSeconds});

**if** (!response.IsSuccessStatusCode){

JObject json = **await** GetJsonFromHttpContent(response.Content);

**if** (response.StatusCode == HttpStatusCode.NotFound){

**throw** new RecipientDoesntExist(json["message"].ToString());

}

**return** **false**;

}

**return** **true**;

}

**public** **async** Task<List<Message>> GetMessages(**int** userId){

List<Message> messages = new List<Message>();

HttpResponseMessage response = **await** Client.GetAsync(**String**.Format(ApiMessages, userId));

JObject json = **await** GetJsonFromHttpContent(response.Content);

**foreach** (JObject message **in** json["messages"]){

Message parsedMessage = Message.MessageFromJson(message);

messages.**Add**(parsedMessage);

}

**return** messages;

}

**public** **async** Task<List<Message>> GetNewMessages(){

List<Message> messages = new List<Message>();

HttpResponseMessage response = **await** Client.GetAsync(ApiNewMessages);

JObject json = **await** GetJsonFromHttpContent(response.Content);

**foreach** (JObject message **in** json["messages"]){

Message parsedMessage = Message.MessageFromJson(message);

messages.**Add**(parsedMessage);

}

**return** messages;

}

}

}

Компонент запросов пользователей. Файл UserRequest.cs

**using** Newtonsoft.Json.Linq;

**using** PlumMessenger.Models;

**using** System;

**using** System.Collections.Generic;

**using** System.Net.Http;

**using** System.Threading.Tasks;

**namespace** PlumMessenger.Connection.Requests{

**class** UserRequest : DefaultRequest{

**public** **static** **int** CurrentUserId;

**public** **static** User CurrentUser;

**string** ApiUser { **get**; **set**; }

**string** ApiUsers { **get**; **set**; }

**public** UserRequest(**string** host = **null**) : **base**(host){

ApiUser = **String**.Format(ApiTemplate, "user/");

ApiUsers = **String**.Format(ApiTemplate, "users/search/{0}");

}

**public** **async** Task<User> GetUser(){

HttpResponseMessage response = **await** Client.GetAsync(**String**.Format(ApiUser));

JObject json = **await** GetJsonFromHttpContent(response.Content);

**if** (CurrentUser == **null**)CurrentUser = User.FromJson(json, **true**);

**else**{

User u = User.FromJson(json, **true**);

User.UpdateFromUser(CurrentUser, u);

}

**return** CurrentUser;

}

**public** **async** Task<**bool**> EditUser(User modified){

HttpResponseMessage response = **await** Client.PostAsJsonAsync(ApiUser, User.ToJson(modified));

JObject json = **await** GetJsonFromHttpContent(response.Content);

**if** (!response.IsSuccessStatusCode){

**throw** new Exception(json["message"].ToString());

}

**else** **if** (json["message"] != **null**){**return** **false**;}

**return** **true**;

}

**public** **async** Task<List<Contact>> SearchUsers(**string** query){

List<Contact> users = new List<Contact>();

HttpResponseMessage response = **await** Client.GetAsync(**String**.Format(ApiUsers, query));

JObject json = **await** GetJsonFromHttpContent(response.Content);

**foreach** (JObject contact **in** json["users"]){

Contact parsedContact = Contact.FromJson(contact);

users.**Add**(parsedContact);

}

**return** users;

}

}

}

Исключения, возможные при авторизации. Файл AuthErrors.cs

**using** System;

**namespace** PlumMessenger.Connection.Errors{

**class** LoginAlreadyTaken : Exception{**public** LoginAlreadyTaken(**string** message) : **base**(message) { }}

**class** Unauthorized : Exception{**public** Unauthorized(**string** message) : **base**(message) { }}

**class** Authorized : Exception{**public** Authorized(**string** message) : **base**(message) { }}

**class** IncorrectLoginOrPasswordFormat : Exception{**public** IncorrectLoginOrPasswordFormat(**string** message) : **base**(message) { }}

**class** InvalidLoginOrPassword : Exception{**public** InvalidLoginOrPassword(**string** message) : **base**(message) { }}}

Исключения, возможные при работе с контактами. Файл ContactErrors.cs

**using** System;

**namespace** PlumMessenger.Connection.Errors{

**class** SelfAdd : Exception{**public** SelfAdd(**string** message) : **base**(message) { }}

**class** UserNotFound : Exception{**public** UserNotFound(**string** message) : **base**(message) { }}}

Исключения, возможные при работе с сообщениями. Файл MessageErrors.cs

**using** System;

**namespace** PlumMessenger.Connection.Errors{

**class** RecipientDoesntExist : Exception{**public** RecipientDoesntExist(**string** message) : **base**(message) { }}

}

Класс пуллинга данных с сервера. Файл Puller.cs

**using** PlumMessenger.Models;

**using** PlumMessenger.Connection.Requests;

**using** System;

**using** System.Collections.Generic;

**using** System.Collections.ObjectModel;

**using** System.Collections.Specialized;

**using** System.Linq;

**using** System.Threading;

**using** System.Threading.Tasks;

**namespace** PlumMessenger.Connection{

**class** Puller{

**public** **static** ObservableCollection<Contact> AvailableContacts = new ObservableCollection<Contact>();

**public** **event** EventHandler AddContact;

**public** **event** EventHandler DeleteContact;

Thread loopThread;

**private** **static** **bool** looping = **false**;

**public** **bool** Looping {**get** {**return** looping;} **set** {looping = **value**;}}

**public** **static** **int** ObservingUserId = -1;

**public** **static** **readonly** UserRequest userRequest = new UserRequest();

**public** **static** **readonly** ContactsRequest contactRequest = new ContactsRequest();

**public** **static** **readonly** MessagesRequest messagesRequest = new MessagesRequest();

**private** **static** **double** contactsRequestTiming = 3000.0;

**private** **static** **double** unreadMessagesRequestTiming = 650.0;

**private** **static** **double** messagesRequestTiming = 150.0;

**public** Puller(){

AvailableContacts.CollectionChanged += ContactsChanged;

}

**private** **void** ContactsChanged(**object** sender, NotifyCollectionChangedEventArgs e){

**if** (e.Action == NotifyCollectionChangedAction.**Add**){

AddContact.Invoke(e.NewItems[0], e);

}**else** **if** (e.Action == NotifyCollectionChangedAction.**Remove**){

DeleteContact.Invoke(e.OldItems[0], e);

}}

**public** **void** StartPulling(){

Looping = **true**;

loopThread = new Thread(Listener);

loopThread.IsBackground = **true**;

loopThread.Start();

}

**public** **static** **async** **void** Listener(){

DateTime startContacts = DateTime.UtcNow,

startNewMessages = DateTime.UtcNow,

startCurrentMessages = DateTime.UtcNow;

**double** spanContacts = contactsRequestTiming,

spanNewMessages = unreadMessagesRequestTiming,

spanCurrentMessages = messagesRequestTiming;

**while** (looping){

**if** (spanContacts >= contactsRequestTiming){

**await** GetContacts();

startContacts = DateTime.UtcNow;}

**if** (spanNewMessages >= unreadMessagesRequestTiming){

**await** GetNewMessages();

startNewMessages = DateTime.UtcNow;}

**if** (spanCurrentMessages >= messagesRequestTiming && ObservingUserId != -1){

**await** GetCurrentMessages(ObservingUserId);

startCurrentMessages = DateTime.UtcNow;}

spanContacts = (DateTime.UtcNow - startContacts).TotalMilliseconds;

spanNewMessages = (DateTime.UtcNow - startNewMessages).TotalMilliseconds;

spanCurrentMessages = (DateTime.UtcNow - startCurrentMessages).TotalMilliseconds;}}

**private** **static** **async** Task GetContacts(){

IEnumerable<**int**> AvailableContactsIds = AvailableContacts.**Select**(x => x.Id);

List<Contact> contacts = **await** contactRequest.GetContacts();

**foreach** (Contact contact **in** contacts){

**if** (AvailableContactsIds.Contains(contact.Id))

User.UpdateFromUser((User)AvailableContacts.First(x => x.Id == contact.Id), (User)contact);

**else**

AvailableContacts.**Add**(contact);}

**var** exceptedContactsIds = AvailableContacts.**Select**(x => x.Id).Except(contacts.**Select**(x => x.Id));

**if** (exceptedContactsIds.Count() > 0){

**var** exceptedContacts = AvailableContacts.**Where**(x => exceptedContactsIds.Contains(x.Id)).ToList();

**foreach** (Contact contact **in** exceptedContacts){AvailableContacts.**Remove**(contact);}}}

**private** **static** **async** Task GetNewMessages(){

List<Message> messages = **await** messagesRequest.GetNewMessages();

**foreach** (Message message **in** messages){

Contact messageOwner = AvailableContacts.FirstOrDefault(x => x.Id == message.OwnerId);

**if** (messageOwner == **null**) **continue**;

**if** (messageOwner.GetUnreadMessages().**Select**(x => x.Id).Contains(message.Id) ||

message.OwnerId == UserRequest.CurrentUserId)

**continue**;

message.Owner = messageOwner;

messageOwner.GetUnreadMessages().**Add**(message);}}

**private** **static** **async** Task GetCurrentMessages(**int** userId){

List<Message> messages = **await** messagesRequest.GetMessages(userId);

**foreach** (Message message **in** messages){

Contact contact;

Contact messageOwner = AvailableContacts.FirstOrDefault(x => x.Id == message.OwnerId);

Contact messageRecipient = AvailableContacts.FirstOrDefault(x => x.Id == message.RecipientId);

**if** (messageOwner != **null**){

contact = messageOwner;

**if** (messageOwner.GetMessages().**Select**(x => x.Id).Contains(message.Id)){**continue**;}

}**else** **if** (messageRecipient != **null**){

contact = messageRecipient;

**if** (messageRecipient.GetMessages().**Select**(x => x.Id).Contains(message.Id)){**continue**; }

}

**else** **continue**;

**if** (messageRecipient == **null** && message.RecipientId == UserRequest.CurrentUserId){

message.Owner = messageOwner;

message.RecipientId = UserRequest.CurrentUserId;

messageOwner.GetMessages().**Add**(message);

}**else** **if** (messageRecipient != **null** && messageRecipient.Id == userId){

message.Recipient = messageRecipient;

message.OwnerId = UserRequest.CurrentUserId;

messageRecipient.GetMessages().**Add**(message);

}}

}

}

}

Класс панели контакта. Файл ContactElement.cs

**using** System;

**using** System.Collections.Generic;

**using** System.Data;

**using** System.Drawing;

**using** System.Linq;

**using** System.Threading.Tasks;

**using** System.Windows.Forms;

**using** PlumMessenger.Models;

**namespace** PlumMessenger{

**public** **partial** **class** ContactElement : UserControl{

**bool** selected = **false**;

**public** Contact contact;

**string** loginTemplate = "@{0}";

**public** **event** EventHandler Clicked;

**public** **event** MouseEventHandler RightClicked;

**public** **event** EventHandler MouseIn;

**public** **event** EventHandler MouseOut;

**public** Func<**int**, Task> ContextMenuAction { **get**; **set**; }

Color selectedColor = Color.FromArgb(43, 82, 120);

Color hoveredColor = Color.FromArgb(32, 43, 54);

Color defaultColor = Color.FromArgb(18, 25, 33); *// Transparent;*

**public** **bool** Selected{**get**{**return** selected;}

**set**{

**if** (**value**)SetColorSelected();

**else** SetColorDefault();

selected = **value**;}

}

**public** ContactElement(){InitializeComponent();}

**public** ContactElement(Contact contact){

InitializeComponent();

**this**.contact = contact;

contact.EditEvent += ContactInfoChangedEvent;

contact.UnreadMessageAddedEvent += NewMessageEvent;

contact.UnreadMessageRemovedEvent += NewMessageEvent;

UpdateFields();

}

**public** **void** OpenContextMenuFromSearch(**object** sender, MouseEventArgs e){

ContextMenu = new ContextMenu();

MenuItem itemAdd = new MenuItem("Добавить в контакты");

**if** (ContextMenuAction != **null**)

itemAdd.Click += **async** (o, \_) => **await** ContextMenuAction(contact.Id);

ContextMenu.MenuItems.**Add**(itemAdd);

ContextMenu.Show(**this**, e.Location);

}

**public** **void** OpenContextMenuDefault(**object** sender, MouseEventArgs e){

ContextMenu = new ContextMenu();

MenuItem itemRemove = new MenuItem("Удалить из контактов");

**if** (ContextMenuAction != **null**)

itemRemove.Click += **async** (o, \_) => **await** ContextMenuAction(contact.Id);

ContextMenu.MenuItems.**Add**(itemRemove);

ContextMenu.Show(**this**, e.Location);

}

**public** **void** SetColorSelected(){BackColor = selectedColor;}

**public** **void** SetColorDefault(){BackColor = defaultColor;}

**private** **void** DispatchMouseEvents(){

**this**.MouseClick += ((o, e) => {

**if** (e.Button == MouseButtons.Right){

RightClicked?.Invoke(**this**, e);

**return**;

}

Clicked?.Invoke(**this**, EventArgs.Empty);

});

**this**.MouseEnter += (o, e) => MouseIn?.Invoke(**this**, EventArgs.Empty);

**this**.MouseLeave += (o, e) => MouseOut?.Invoke(**this**, EventArgs.Empty);

**foreach** (Control c **in** GetNestedControls<Control>(**this**)){

c.MouseClick += ((o, e) => {

**if** (e.Button == MouseButtons.Right){

RightClicked?.Invoke(**this**, e);

**return**;

}

Clicked?.Invoke(**this**, EventArgs.Empty);

});

c.MouseEnter += (o, e) => **this**.MouseIn?.Invoke(**this**, EventArgs.Empty);

c.MouseLeave += (o, e) => **this**.MouseOut?.Invoke(**this**, EventArgs.Empty);

}

}

**private** IEnumerable<ControlType> GetNestedControls<ControlType>(Control control){

**if** (control.Controls == **null** || control.Controls.Count == 0)

**return** Enumerable.Empty<ControlType>();

**return** control.Controls.OfType<ControlType>().Concat(control.Controls.Cast<Control>().SelectMany(x => GetNestedControls<ControlType>(x)));

}

**private** **void** UpdateFields(){

usernameLabel.Text = contact.Username;

loginLabel.Text = **String**.Format(loginTemplate, contact.Login);

}

**private** **void** ContactInfoChangedEvent(**object** sender, EventArgs e){UpdateFields();}

**private** **void** NewMessageEvent(**object** sender, EventArgs e){

**int** currentCount = ((Contact)sender).GetUnreadMessages().Count();

**this**.Invoke(new MethodInvoker(() => {

**if** (currentCount > 0) newMessagesCountLabel.Text = currentCount.ToString();

**else** newMessagesCountLabel.Text = "";

}));

}

**private** **void** ContactElement\_Load(**object** sender, EventArgs e){

DispatchMouseEvents();

MouseIn += ContactElement\_MouseEnter;

MouseOut += ContactElement\_MouseLeave;

}

**private** **void** ContactElement\_MouseEnter(**object** sender, EventArgs e){

**if** (selected)**return**;

BackColor = hoveredColor;

}

**private** **void** ContactElement\_MouseLeave(**object** sender, EventArgs e){

**if** (selected)**return**;

BackColor = defaultColor;

}

}

}

Класс дизайна панели контактов. Файл ContactElement.Designer.cs

**namespace** PlumMessenger{

**partial** **class** ContactElement{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){components.Dispose();}

**base**.Dispose(disposing);

}

**private** **void** InitializeComponent(){

**this**.usernameLabel = new System.Windows.Forms.Label();

**this**.newMessagesCountLabel = new System.Windows.Forms.Label();

**this**.dataPanel = new System.Windows.Forms.Panel();

**this**.loginLabel = new System.Windows.Forms.Label();

**this**.dataPanel.SuspendLayout();

**this**.SuspendLayout();

**this**.usernameLabel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.usernameLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.usernameLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.usernameLabel.Location = new System.Drawing.Point(0, 0);

**this**.usernameLabel.Name = "usernameLabel";

**this**.usernameLabel.Size = new System.Drawing.Size(161, 34);

**this**.usernameLabel.TabIndex = 0;

**this**.usernameLabel.Text = "username";

**this**.usernameLabel.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;

**this**.newMessagesCountLabel.Dock = System.Windows.Forms.DockStyle.Right;

**this**.newMessagesCountLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.newMessagesCountLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.newMessagesCountLabel.Location = new System.Drawing.Point(161, 0);

**this**.newMessagesCountLabel.Name = "newMessagesCountLabel";

**this**.newMessagesCountLabel.Size = new System.Drawing.Size(42, 56);

**this**.newMessagesCountLabel.TabIndex = 1;

**this**.newMessagesCountLabel.TextAlign = System.Drawing.ContentAlignment.MiddleRight;

**this**.dataPanel.Controls.**Add**(**this**.loginLabel);

**this**.dataPanel.Controls.**Add**(**this**.usernameLabel);

**this**.dataPanel.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.dataPanel.Location = new System.Drawing.Point(0, 0);

**this**.dataPanel.Name = "dataPanel";

**this**.dataPanel.Size = new System.Drawing.Size(161, 56);

**this**.dataPanel.TabIndex = 2;

**this**.loginLabel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.loginLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 7F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.loginLabel.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.loginLabel.Location = new System.Drawing.Point(0, 34);

**this**.loginLabel.Name = "loginLabel";

**this**.loginLabel.Size = new System.Drawing.Size(161, 22);

**this**.loginLabel.TabIndex = 1;

**this**.loginLabel.Text = "@login";

**this**.loginLabel.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.None;

**this**.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(18)))), ((**int**)(((**byte**)(25)))), ((**int**)(((**byte**)(33)))));

**this**.Controls.**Add**(**this**.dataPanel);

**this**.Controls.**Add**(**this**.newMessagesCountLabel);

**this**.Margin = new System.Windows.Forms.Padding(0, 0, 0, 5);

**this**.Name = "ContactElement";

**this**.Size = new System.Drawing.Size(203, 56);

**this**.Load += new System.EventHandler(**this**.ContactElement\_Load);

**this**.dataPanel.ResumeLayout(**false**);

**this**.ResumeLayout(**false**);}

**private** System.Windows.Forms.Label usernameLabel;

**private** System.Windows.Forms.Label newMessagesCountLabel;

**private** System.Windows.Forms.Panel dataPanel;

**private** System.Windows.Forms.Label loginLabel;}}

Класс панели информации о контакте. Файл ContactInfoElement.cs

**using** PlumMessenger.Models;

**using** System;

**using** System.Collections.Generic;

**using** System.ComponentModel;

**using** System.Data;

**using** System.Drawing;

**using** System.Linq;

**using** System.Text;

**using** System.Threading.Tasks;

**using** System.Windows.Forms;

**namespace** PlumMessenger.Controls{

**public** **partial** **class** ContactInfoElement : UserControl{

**public** **event** EventHandler CloseClickedEvent;

**private** Contact contact;

**public** ContactInfoElement(Contact contact){InitializeComponent();**this**.contact = contact;}

**private** **void** SetFields(){

loginLabel.Text = contact.Login;

userNameLabel.Text = contact.Username;

phoneNumberLabel.Text = contact.PhoneNumber;

**if** (contact.PhonePublicy){

phoneNumberPanel.Visible = **true**;

phoneNumberLabel.Text = contact.PhoneNumber;

} **else** phoneNumberPanel.Visible = **false**;

statusLabel.Text = contact.Status;

}

**private** **void** closeBtn\_Click(**object** sender, EventArgs e){

CloseClickedEvent.Invoke(**this**, EventArgs.Empty);

}

**private** **void** ContactInfoElement\_Load(**object** sender, EventArgs e){

contact.EditEvent += CurrentUserEdited;

SetFields();

}

**private** **void** CurrentUserEdited(**object** sender, EventArgs e){

SetFields();

}

}

}

Класс дизайна панели инорфмации о контакте. Файл ContactInfoElement.Designer.cs

**namespace** PlumMessenger.Controls{

**partial** **class** ContactInfoElement{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){

components.Dispose();

}

**base**.Dispose(disposing);

}

**private** **void** InitializeComponent(){

**this**.label4 = new System.Windows.Forms.Label();

**this**.panel5 = new System.Windows.Forms.Panel();

**this**.statusLabel = new System.Windows.Forms.Label();

**this**.closeBtn = new System.Windows.Forms.Button();

**this**.panel3 = new System.Windows.Forms.Panel();

**this**.panel4 = new System.Windows.Forms.Panel();

**this**.panel2 = new System.Windows.Forms.Panel();

**this**.panel9 = new System.Windows.Forms.Panel();

**this**.label5 = new System.Windows.Forms.Label();

**this**.phoneNumberPanel = new System.Windows.Forms.Panel();

**this**.phoneNumberLabel = new System.Windows.Forms.Label();

**this**.label3 = new System.Windows.Forms.Label();

**this**.panel7 = new System.Windows.Forms.Panel();

**this**.userNameLabel = new System.Windows.Forms.Label();

**this**.label7 = new System.Windows.Forms.Label();

**this**.panel6 = new System.Windows.Forms.Panel();

**this**.loginLabel = new System.Windows.Forms.Label();

**this**.label1 = new System.Windows.Forms.Label();

**this**.panel1 = new System.Windows.Forms.Panel();

**this**.panel5.SuspendLayout();

**this**.panel3.SuspendLayout();

**this**.panel4.SuspendLayout();

**this**.panel2.SuspendLayout();

**this**.panel9.SuspendLayout();

**this**.phoneNumberPanel.SuspendLayout();

**this**.panel7.SuspendLayout();

**this**.panel6.SuspendLayout();

**this**.panel1.SuspendLayout();

**this**.SuspendLayout();

**this**.label4.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label4.ForeColor = System.Drawing.SystemColors.Control;

**this**.label4.Location = new System.Drawing.Point(0, 0);

**this**.label4.Name = "label4";

**this**.label4.Size = new System.Drawing.Size(264, 25);

**this**.label4.TabIndex = 14;

**this**.label4.Text = "Информация о пользователе";

**this**.label4.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

**this**.panel5.Controls.**Add**(**this**.label4);

**this**.panel5.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel5.Location = new System.Drawing.Point(0, 0);

**this**.panel5.Name = "panel5";

**this**.panel5.Size = new System.Drawing.Size(264, 25);

**this**.panel5.TabIndex = 3;

**this**.statusLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.statusLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.statusLabel.Location = new System.Drawing.Point(0, 24);

**this**.statusLabel.Name = "statusLabel";

**this**.statusLabel.Size = new System.Drawing.Size(261, 65);

**this**.statusLabel.TabIndex = 10;

**this**.statusLabel.Text = "";

**this**.closeBtn.Anchor = ((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Left | System.Windows.Forms.AnchorStyles.Right)));

**this**.closeBtn.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(82)))), ((**int**)(((**byte**)(136)))), ((**int**)(((**byte**)(193)))));

**this**.closeBtn.FlatAppearance.BorderSize = 0;

**this**.closeBtn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

**this**.closeBtn.ForeColor = System.Drawing.SystemColors.Control;

**this**.closeBtn.Location = new System.Drawing.Point(83, 0);

**this**.closeBtn.Name = "closeBtn";

**this**.closeBtn.Size = new System.Drawing.Size(100, 25);

**this**.closeBtn.TabIndex = 0;

**this**.closeBtn.Text = "Закрыть";

**this**.closeBtn.UseVisualStyleBackColor = **false**;

**this**.closeBtn.Click += new System.EventHandler(**this**.closeBtn\_Click);

**this**.panel3.Controls.**Add**(**this**.closeBtn);

**this**.panel3.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel3.Location = new System.Drawing.Point(0, 191);

**this**.panel3.Name = "panel3";

**this**.panel3.Size = new System.Drawing.Size(264, 25);

**this**.panel3.TabIndex = 2;

**this**.panel4.Controls.**Add**(**this**.panel3);

**this**.panel4.Controls.**Add**(**this**.panel2);

**this**.panel4.Controls.**Add**(**this**.panel5);

**this**.panel4.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.panel4.Location = new System.Drawing.Point(10, 10);

**this**.panel4.Name = "panel4";

**this**.panel4.Size = new System.Drawing.Size(264, 217);

**this**.panel4.TabIndex = 14;

**this**.panel2.Controls.**Add**(**this**.panel9);

**this**.panel2.Controls.**Add**(**this**.phoneNumberPanel);

**this**.panel2.Controls.**Add**(**this**.panel7);

**this**.panel2.Controls.**Add**(**this**.panel6);

**this**.panel2.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel2.Location = new System.Drawing.Point(0, 25);

**this**.panel2.Name = "panel2";

**this**.panel2.Size = new System.Drawing.Size(264, 166);

**this**.panel2.TabIndex = 1;

**this**.panel9.Controls.**Add**(**this**.statusLabel);

**this**.panel9.Controls.**Add**(**this**.label5);

**this**.panel9.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel9.Location = new System.Drawing.Point(0, 75);

**this**.panel9.Name = "panel9";

**this**.panel9.Size = new System.Drawing.Size(264, 90);

**this**.panel9.TabIndex = 14;

**this**.label5.AutoSize = **true**;

**this**.label5.Dock = System.Windows.Forms.DockStyle.Left;

**this**.label5.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label5.ForeColor = System.Drawing.SystemColors.Control;

**this**.label5.Location = new System.Drawing.Point(0, 0);

**this**.label5.Name = "label5";

**this**.label5.Size = new System.Drawing.Size(47, 15);

**this**.label5.TabIndex = 11;

**this**.label5.Text = "Статус";

**this**.phoneNumberPanel.Controls.**Add**(**this**.phoneNumberLabel);

**this**.phoneNumberPanel.Controls.**Add**(**this**.label3);

**this**.phoneNumberPanel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.phoneNumberPanel.Location = new System.Drawing.Point(0, 50);

**this**.phoneNumberPanel.Name = "phoneNumberPanel";

**this**.phoneNumberPanel.Size = new System.Drawing.Size(264, 25);

**this**.phoneNumberPanel.TabIndex = 13;

**this**.phoneNumberLabel.AutoSize = **true**;

**this**.phoneNumberLabel.Dock = System.Windows.Forms.DockStyle.Right;

**this**.phoneNumberLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.phoneNumberLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.phoneNumberLabel.Location = new System.Drawing.Point(145, 0);

**this**.phoneNumberLabel.Name = "phoneNumberLabel";

**this**.phoneNumberLabel.Size = new System.Drawing.Size(119, 15);

**this**.phoneNumberLabel.TabIndex = 5;

**this**.phoneNumberLabel.Text = "aaaaaaaaaaaaaaaa";

**this**.label3.AutoSize = **true**;

**this**.label3.Dock = System.Windows.Forms.DockStyle.Left;

**this**.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label3.ForeColor = System.Drawing.SystemColors.Control;

**this**.label3.Location = new System.Drawing.Point(0, 0);

**this**.label3.Name = "label3";

**this**.label3.Size = new System.Drawing.Size(109, 15);

**this**.label3.TabIndex = 4;

**this**.label3.Text = "Номер телефона";

**this**.panel7.Controls.**Add**(**this**.userNameLabel);

**this**.panel7.Controls.**Add**(**this**.label7);

**this**.panel7.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel7.Location = new System.Drawing.Point(0, 25);

**this**.panel7.Name = "panel7";

**this**.panel7.Size = new System.Drawing.Size(264, 25);

**this**.panel7.TabIndex = 12;

**this**.userNameLabel.AutoSize = **true**;

**this**.userNameLabel.Dock = System.Windows.Forms.DockStyle.Right;

**this**.userNameLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.userNameLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.userNameLabel.Location = new System.Drawing.Point(145, 0);

**this**.userNameLabel.Name = "userNameLabel";

**this**.userNameLabel.Size = new System.Drawing.Size(119, 15);

**this**.userNameLabel.TabIndex = 2;

**this**.userNameLabel.Text = "aaaaaaaaaaaaaaaa";

**this**.label7.AutoSize = **true**;

**this**.label7.Dock = System.Windows.Forms.DockStyle.Left;

**this**.label7.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label7.ForeColor = System.Drawing.SystemColors.Control;

**this**.label7.Location = new System.Drawing.Point(0, 0);

**this**.label7.Name = "label7";

**this**.label7.Size = new System.Drawing.Size(118, 15);

**this**.label7.TabIndex = 3;

**this**.label7.Text = "Имя пользователя";

**this**.panel6.Controls.**Add**(**this**.loginLabel);

**this**.panel6.Controls.**Add**(**this**.label1);

**this**.panel6.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel6.Location = new System.Drawing.Point(0, 0);

**this**.panel6.Name = "panel6";

**this**.panel6.Size = new System.Drawing.Size(264, 25);

**this**.panel6.TabIndex = 11;

**this**.loginLabel.AutoSize = **true**;

**this**.loginLabel.Dock = System.Windows.Forms.DockStyle.Right;

**this**.loginLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.loginLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.loginLabel.Location = new System.Drawing.Point(145, 0);

**this**.loginLabel.Name = "loginLabel";

**this**.loginLabel.Size = new System.Drawing.Size(119, 15);

**this**.loginLabel.TabIndex = 3;

**this**.loginLabel.Text = "aaaaaaaaaaaaaaaa";

**this**.label1.AutoSize = **true**;

**this**.label1.Dock = System.Windows.Forms.DockStyle.Left;

**this**.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label1.ForeColor = System.Drawing.SystemColors.Control;

**this**.label1.Location = new System.Drawing.Point(0, 0);

**this**.label1.Name = "label1";

**this**.label1.Size = new System.Drawing.Size(41, 15);

**this**.label1.TabIndex = 0;

**this**.label1.Text = "Логин";

**this**.panel1.Anchor = System.Windows.Forms.AnchorStyles.None;

**this**.panel1.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(14)))), ((**int**)(((**byte**)(22)))), ((**int**)(((**byte**)(33)))));

**this**.panel1.Controls.**Add**(**this**.panel4);

**this**.panel1.Location = new System.Drawing.Point(12, 16);

**this**.panel1.Name = "panel1";

**this**.panel1.Padding = new System.Windows.Forms.Padding(10);

**this**.panel1.Size = new System.Drawing.Size(284, 237);

**this**.panel1.TabIndex = 1;

**this**.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

**this**.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.Controls.**Add**(**this**.panel1);

**this**.Name = "ContactInfoElement";

**this**.Size = new System.Drawing.Size(309, 268);

**this**.Load += new System.EventHandler(**this**.ContactInfoElement\_Load);

**this**.panel5.ResumeLayout(**false**);

**this**.panel3.ResumeLayout(**false**);

**this**.panel4.ResumeLayout(**false**);

**this**.panel2.ResumeLayout(**false**);

**this**.panel9.ResumeLayout(**false**);

**this**.panel9.PerformLayout();

**this**.phoneNumberPanel.ResumeLayout(**false**);

**this**.phoneNumberPanel.PerformLayout();

**this**.panel7.ResumeLayout(**false**);

**this**.panel7.PerformLayout();

**this**.panel6.ResumeLayout(**false**);

**this**.panel6.PerformLayout();

**this**.panel1.ResumeLayout(**false**);

**this**.ResumeLayout(**false**);

}

**private** System.Windows.Forms.Label label4;

**private** System.Windows.Forms.Panel panel5;

**private** System.Windows.Forms.Label statusLabel;

**private** System.Windows.Forms.Button closeBtn;

**private** System.Windows.Forms.Panel panel3;

**private** System.Windows.Forms.Panel panel4;

**private** System.Windows.Forms.Panel panel2;

**private** System.Windows.Forms.Label label3;

**private** System.Windows.Forms.Label userNameLabel;

**private** System.Windows.Forms.Label label1;

**private** System.Windows.Forms.Panel panel1;

**private** System.Windows.Forms.Panel panel9;

**private** System.Windows.Forms.Panel phoneNumberPanel;

**private** System.Windows.Forms.Panel panel7;

**private** System.Windows.Forms.Panel panel6;

**private** System.Windows.Forms.Label label5;

**private** System.Windows.Forms.Label label7;

**private** System.Windows.Forms.Label phoneNumberLabel;

**private** System.Windows.Forms.Label loginLabel;

}

}

Класс панели авторизации. Файл LoginElement.cs

**using** System;

**using** System.Windows.Forms;

**namespace** PlumMessenger.Controls{

**public** **partial** **class** LoginElement : UserControl{

**public** **event** EventHandler LoginClicked;

**public** **event** EventHandler RegistrationClicked;

**public** LoginElement(){InitializeComponent();}

**public** **string** GetLogin(){**return** loginText.Text;}

**public** **string** GetPassword(){**return** passwordText.Text;}

**private** **void** loginBtn\_Click(**object** sender, EventArgs e){LoginClicked.Invoke(**this**, EventArgs.Empty);}

**private** **void** goToRegistration\_Click(**object** sender, EventArgs e){RegistrationClicked.Invoke(**this**, EventArgs.Empty);}}

}

Класс дизайна панели авторизации. Файл LoginElement.Designer.cs

**namespace** PlumMessenger.Controls{

**partial** **class** LoginElement{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){components.Dispose();}

**base**.Dispose(disposing);

}

**private** **void** InitializeComponent(){

**this**.label = new System.Windows.Forms.Label();

**this**.login\_label = new System.Windows.Forms.Label();

**this**.loginText = new System.Windows.Forms.TextBox();

**this**.passwordText = new System.Windows.Forms.TextBox();

**this**.password\_label = new System.Windows.Forms.Label();

**this**.panel1 = new System.Windows.Forms.Panel();

**this**.loginBtn = new System.Windows.Forms.Button();

**this**.goToRegistration = new System.Windows.Forms.Button();

**this**.panel1.SuspendLayout();

**this**.SuspendLayout();

**this**.label.AutoSize = **true**;

**this**.label.Font = new System.Drawing.Font("Segoe UI", 14F, System.Drawing.FontStyle.Bold);

**this**.label.ForeColor = System.Drawing.SystemColors.Control;

**this**.label.Location = new System.Drawing.Point(51, 9);

**this**.label.Name = "label";

**this**.label.Size = new System.Drawing.Size(69, 25);

**this**.label.TabIndex = 10;

**this**.label.Text = "Войти";

**this**.login\_label.AutoSize = **true**;

**this**.login\_label.Font = new System.Drawing.Font("Segoe UI", 9F);

**this**.login\_label.ForeColor = System.Drawing.SystemColors.Control;

**this**.login\_label.Location = new System.Drawing.Point(3, 0);

**this**.login\_label.Name = "login\_label";

**this**.login\_label.Size = new System.Drawing.Size(41, 15);

**this**.login\_label.TabIndex = 6;

**this**.login\_label.Text = "Логин";

**this**.loginText.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.loginText.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.loginText.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.loginText.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.loginText.Location = new System.Drawing.Point(0, 16);

**this**.loginText.MaxLength = 16;

**this**.loginText.Name = "loginText";

**this**.loginText.Size = new System.Drawing.Size(176, 21);

**this**.loginText.TabIndex = 3;

**this**.passwordText.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.passwordText.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.passwordText.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.passwordText.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.passwordText.Location = new System.Drawing.Point(0, 55);

**this**.passwordText.MaxLength = 32;

**this**.passwordText.Name = "passwordText";

**this**.passwordText.PasswordChar = '\*';

**this**.passwordText.Size = new System.Drawing.Size(176, 21);

**this**.passwordText.TabIndex = 5;

**this**.password\_label.AutoSize = **true**;

**this**.password\_label.ForeColor = System.Drawing.SystemColors.Control;

**this**.password\_label.Location = new System.Drawing.Point(0, 39);

**this**.password\_label.Name = "password\_label";

**this**.password\_label.Size = new System.Drawing.Size(45, 13);

**this**.password\_label.TabIndex = 4;

**this**.password\_label.Text = "Пароль";

**this**.panel1.Controls.**Add**(**this**.loginBtn);

**this**.panel1.Controls.**Add**(**this**.login\_label);

**this**.panel1.Controls.**Add**(**this**.loginText);

**this**.panel1.Controls.**Add**(**this**.passwordText);

**this**.panel1.Controls.**Add**(**this**.password\_label);

**this**.panel1.Location = new System.Drawing.Point(3, 37);

**this**.panel1.Name = "panel1";

**this**.panel1.Size = new System.Drawing.Size(176, 134);

**this**.panel1.TabIndex = 12;

**this**.loginBtn.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(82)))), ((**int**)(((**byte**)(136)))), ((**int**)(((**byte**)(193)))));

**this**.loginBtn.FlatAppearance.BorderSize = 0;

**this**.loginBtn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

**this**.loginBtn.ForeColor = System.Drawing.SystemColors.Control;

**this**.loginBtn.Location = new System.Drawing.Point(53, 113);

**this**.loginBtn.Name = "loginBtn";

**this**.loginBtn.Size = new System.Drawing.Size(64, 22);

**this**.loginBtn.TabIndex = 7;

**this**.loginBtn.Text = "Войти";

**this**.loginBtn.UseVisualStyleBackColor = **false**;

**this**.loginBtn.Click += new System.EventHandler(**this**.loginBtn\_Click);

**this**.goToRegistration.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(82)))), ((**int**)(((**byte**)(136)))), ((**int**)(((**byte**)(193)))));

**this**.goToRegistration.FlatAppearance.BorderSize = 0;

**this**.goToRegistration.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

**this**.goToRegistration.ForeColor = System.Drawing.SystemColors.Control;

**this**.goToRegistration.Location = new System.Drawing.Point(41, 177);

**this**.goToRegistration.Name = "goToRegistration";

**this**.goToRegistration.Size = new System.Drawing.Size(95, 23);

**this**.goToRegistration.TabIndex = 11;

**this**.goToRegistration.Text = "Регистрация";

**this**.goToRegistration.UseVisualStyleBackColor = **false**;

**this**.goToRegistration.Click += new System.EventHandler(**this**.goToRegistration\_Click);

**this**.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

**this**.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.Controls.**Add**(**this**.label);

**this**.Controls.**Add**(**this**.panel1);

**this**.Controls.**Add**(**this**.goToRegistration);

**this**.Name = "LoginElement";

**this**.Size = new System.Drawing.Size(182, 207);

**this**.panel1.ResumeLayout(**false**);

**this**.panel1.PerformLayout();

**this**.ResumeLayout(**false**);

**this**.PerformLayout();}

**private** System.Windows.Forms.Label label;

**private** System.Windows.Forms.Label login\_label;

**private** System.Windows.Forms.TextBox loginText;

**private** System.Windows.Forms.TextBox passwordText;

**private** System.Windows.Forms.Label password\_label;

**private** System.Windows.Forms.Panel panel1;

**private** System.Windows.Forms.Button loginBtn;

**private** System.Windows.Forms.Button goToRegistration;

}

}

Класс панели облачка сообщения от пользователя. Файл MeBubble.cs

**using** System;

**using** System.ComponentModel;

**using** System.Drawing;

**using** System.Windows.Forms;

**using** System.Drawing.Drawing2D;

**using** System.Drawing.Design;

**namespace** PlumMesseger{

**public** **partial** **class** MeBubble : UserControl{

**public** MeBubble(){InitializeComponent();}

[Editor("System.ComponentModel.Design.MultilineStringEditor, System.Design, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a", typeof(UITypeEditor))]

**public** **string** Body{**get**{**return** label1.Text;}

**set**{label1.Text = **value**;}}

**public** Cursor ChatTextCursor{**get**{**return** label1.Cursor;}

**set**{label1.Cursor = **value**;}}

**public** Color MsgColor{**get**{**return** label1.BackColor;}

**set**{label1.BackColor = **value**;}}

**public** Color MsgTextColor{**get**{**return** label1.ForeColor;}

**set**{label1.ForeColor = **value**;}}

**public** Color TimeColor{**get**{**return** time.ForeColor;}

**set**{time.ForeColor = **value**;}}

**public** **string** Time{**get**{**return** time.Text;}

**set**{time.Text = **value**;

SetTimeWidth();}

}

**private** **void** SetTimeWidth(){

time.Width = TextRenderer.MeasureText(time.Text, time.Font).Width;

}

**private** **void** Bubble\_Load(**object** sender, EventArgs e){}

Panel messageBottom = new Panel();

Label time = new Label();

**bool** isAdded = **false**;

**protected** **override** **void** OnPaint(PaintEventArgs e){

label1.MaximumSize = new Size((**this**.Width - 21)/2, 0);

label1.Width = **this**.Width - 21;

**if** (label1.Height < 1){

**this**.MinimumSize = new Size(0, 11+15);

**this**.Height = 11 + 15;

}**else**{

**this**.MinimumSize = new Size(0, label1.Height + 10 + 15);

**this**.Height = label1.Height + 10 + 15;

}

**if** (!isAdded){

messageBottom.Size = new Size(0, 15);

messageBottom.Dock = DockStyle.Bottom;

messageBottom.Padding = new Padding(0, 0, 10, 0);

messageBottom.BackColor = Color.Transparent;

messageBottom.ForeColor = Color.White;

time.Dock = DockStyle.Right;

SetTimeWidth();

messageBottom.Controls.**Add**(time);

**this**.Controls.**Add**(messageBottom);

isAdded = **true**;

}

GraphicsPath path = RoundedRectangle.Create(label1.ClientRectangle,5, RoundedRectangle.RectangleCorners.All);

label1.Region = new Region(path);

**base**.OnPaint(e);

}

**public** **delegate** **void** ChatTextClick(**object** sender, EventArgs e);

**public** **event** ChatTextClick OnChatTextClick;

**private** **void** label1\_Click(**object** sender, EventArgs e){

**if** (OnChatTextClick != **null**){OnChatTextClick.Invoke(sender, e);}

}

}

}

Класс дизайна панели облачка от пользователя. Файл MeBubble.Designer.cs

**namespace** PlumMesseger{

**partial** **class** MeBubble{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){components.Dispose();}

**base**.Dispose(disposing);

}

**private** **void** InitializeComponent(){

**this**.label1 = new System.Windows.Forms.Label();

**this**.SuspendLayout();

**this**.label1.AutoSize = **true**;

**this**.label1.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(9)))), ((**int**)(((**byte**)(64)))), ((**int**)(((**byte**)(75)))));

**this**.label1.Cursor = System.Windows.Forms.Cursors.IBeam;

**this**.label1.Dock = System.Windows.Forms.DockStyle.Right;

**this**.label1.Font = new System.Drawing.Font("Arial", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(0)));

**this**.label1.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.label1.Location = new System.Drawing.Point(176, 5);

**this**.label1.MaximumSize = new System.Drawing.Size(140, 0);

**this**.label1.MinimumSize = new System.Drawing.Size(70, 0);

**this**.label1.Name = "label1";

**this**.label1.Padding = new System.Windows.Forms.Padding(3);

**this**.label1.Size = new System.Drawing.Size(139, 150);

**this**.label1.TabIndex = 1;

**this**.label1.Text = "";

**this**.label1.Click += new System.EventHandler(**this**.label1\_Click);

**this**.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

**this**.AutoSize = **true**;

**this**.BackColor = System.Drawing.Color.Transparent;

**this**.Controls.**Add**(**this**.label1);

**this**.MinimumSize = new System.Drawing.Size(0, 41);

**this**.Name = "MeBubble";

**this**.Padding = new System.Windows.Forms.Padding(0, 5, 5, 5);

**this**.Size = new System.Drawing.Size(320, 193);

**this**.Load += new System.EventHandler(**this**.Bubble\_Load);

**this**.ResumeLayout(**false**);

**this**.PerformLayout();

}

**private** System.Windows.Forms.Label label1;

}

}

Класс панели облачка сообщения пользователю. Файл YouBubble.cs

**using** System;

**using** System.ComponentModel;

**using** System.Drawing;

**using** System.Windows.Forms;

**using** System.Drawing.Drawing2D;

**using** System.Drawing.Design;

**namespace** PlumMesseger{

**public** **partial** **class** YouBubble : UserControl{

**public** YouBubble(){InitializeComponent();}

[Editor("System.ComponentModel.Design.MultilineStringEditor, System.Design, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a", typeof(UITypeEditor))]

**public** **string** Body{**get**{**return** label1.Text.Replace(Environment.NewLine,"**\n**");}

**set**{label1.Text = **value**.Replace("**\n**", Environment.NewLine);}}

**public** Cursor ChatTextCursor{**get**{**return** label1.Cursor;}

**set**{label1.Cursor = **value**;}}

**public** Color MsgColor{**get**{**return** label1.BackColor;}

**set**{label1.BackColor = **value**;}}

**public** Color MsgTextColor{**get**{**return** label1.ForeColor;}

**set**{label1.ForeColor = **value**;}}

**public** Color TimeColor{**get**{**return** time.ForeColor;}

**set**{time.ForeColor = **value**;}}

**public** **string** Time{**get**{**return** time.Text;}

**set**{time.Text = **value**;SetTimeWidth();}}

**private** **void** SetTimeWidth(){

time.Width = TextRenderer.MeasureText(time.Text, time.Font).Width;

}

Panel messageBottom = new Panel();

Label time = new Label();

**bool** isAdded = **false**;

**private** **void** YouBubble\_Load(**object** sender, EventArgs e){}

**protected** **override** **void** OnPaint(PaintEventArgs e){

label1.MaximumSize = new Size((**this**.Width - 21)/2, 0);

label1.Width = **this**.Width - 21;

**if** (label1.Height < 1){

**this**.MinimumSize = new Size(0, 11 + 15);

**this**.Height = 11 + 15;

}**else**{

**this**.MinimumSize = new Size(0, label1.Height + 10 + 15);

**this**.Height = label1.Height + 10 + 15;

}

**if** (!isAdded){

messageBottom.Size = new Size(0, 15);

messageBottom.Dock = DockStyle.Bottom;

messageBottom.Padding = new Padding(10, 0, 0, 0);

messageBottom.BackColor = Color.Transparent;

messageBottom.ForeColor = Color.White;

time.Dock = DockStyle.Left;

SetTimeWidth();

messageBottom.Controls.**Add**(time);

**this**.Controls.**Add**(messageBottom);

isAdded = **true**;}

GraphicsPath path = RoundedRectangle.Create(label1.ClientRectangle, 5, RoundedRectangle.RectangleCorners.All);

label1.Region = new Region(path);

**base**.OnPaint(e);}

**public** **delegate** **void** ChatImageClick(**object** sender, EventArgs e);

**public** **delegate** **void** ChatTextClick(**object** sender, EventArgs e);

**public** **event** ChatTextClick OnChatTextClick;

**private** **void** label1\_Click(**object** sender, EventArgs e){

**if** (OnChatTextClick != **null**){OnChatTextClick.Invoke(sender, e);}

}}}

Класс дизайна панели облачка сообщения пользователю. Файл YouBubble.Designer.cs

**namespace** PlumMesseger{

**partial** **class** YouBubble{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){components.Dispose();}

**base**.Dispose(disposing);

}

**private** **void** InitializeComponent(){

**this**.label1 = new System.Windows.Forms.Label();

**this**.SuspendLayout();

**this**.label1.AutoSize = **true**;

**this**.label1.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(32)))), ((**int**)(((**byte**)(44)))), ((**int**)(((**byte**)(51)))));

**this**.label1.Cursor = System.Windows.Forms.Cursors.IBeam;

**this**.label1.Dock = System.Windows.Forms.DockStyle.Left;

**this**.label1.Font = new System.Drawing.Font("Arial", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(0)));

**this**.label1.ForeColor = System.Drawing.Color.White;

**this**.label1.Location = new System.Drawing.Point(5, 5);

**this**.label1.MaximumSize = new System.Drawing.Size(140, 0);

**this**.label1.MinimumSize = new System.Drawing.Size(70, 0);

**this**.label1.Name = "label1";

**this**.label1.Padding = new System.Windows.Forms.Padding(3);

**this**.label1.Size = new System.Drawing.Size(139, 150);

**this**.label1.TabIndex = 3;

**this**.label1.Text = "";

**this**.label1.Click += new System.EventHandler(**this**.label1\_Click);

**this**.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

**this**.BackColor = System.Drawing.Color.Transparent;

**this**.Controls.**Add**(**this**.label1);

**this**.Name = "YouBubble";

**this**.Padding = new System.Windows.Forms.Padding(5, 5, 0, 5);

**this**.Size = new System.Drawing.Size(320, 169);

**this**.Load += new System.EventHandler(**this**.YouBubble\_Load);

**this**.ResumeLayout(**false**);

**this**.PerformLayout();}

**private** System.Windows.Forms.Label label1;

}

}

Класс панели регистрации. Файл RegistrationElement.cs

**using** System;

**using** System.Windows.Forms;

**namespace** PlumMessenger.Controls{

**public** **partial** **class** RegistrationElement : UserControl{

**public** **event** EventHandler RegistrationClicked;

**public** **event** EventHandler LoginClicked;

**public** RegistrationElement(){InitializeComponent();}

**public** **string** GetLogin(){**return** loginText.Text;}

**public** **string** GetPassword(){**return** passwordText.Text;}

**private** **void** registrationBtn\_Click(**object** sender, EventArgs e){RegistrationClicked.Invoke(**this**, EventArgs.Empty);}

**private** **void** goToLoginBtn\_Click(**object** sender, EventArgs e){LoginClicked.Invoke(**this**, EventArgs.Empty);}

}

}

Класс дизайна панели регистрации. Файл RegistrationElement.Designer.cs

**namespace** PlumMessenger.Controls{

**partial** **class** RegistrationElement{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){components.Dispose();}

**base**.Dispose(disposing);}

**private** **void** InitializeComponent(){

**this**.label = new System.Windows.Forms.Label();

**this**.login\_label = new System.Windows.Forms.Label();

**this**.loginText = new System.Windows.Forms.TextBox();

**this**.passwordText = new System.Windows.Forms.TextBox();

**this**.password\_label = new System.Windows.Forms.Label();

**this**.panel1 = new System.Windows.Forms.Panel();

**this**.registrationBtn = new System.Windows.Forms.Button();

**this**.goToLoginBtn = new System.Windows.Forms.Button();

**this**.panel1.SuspendLayout();

**this**.SuspendLayout();

**this**.label.AutoSize = **true**;

**this**.label.Font = new System.Drawing.Font("Segoe UI", 14F, System.Drawing.FontStyle.Bold);

**this**.label.ForeColor = System.Drawing.SystemColors.Control;

**this**.label.Location = new System.Drawing.Point(24, 9);

**this**.label.Name = "label";

**this**.label.Size = new System.Drawing.Size(128, 25);

**this**.label.TabIndex = 8;

**this**.label.Text = "Регистрация";

**this**.login\_label.AutoSize = **true**;

**this**.login\_label.Font = new System.Drawing.Font("Segoe UI", 9F);

**this**.login\_label.ForeColor = System.Drawing.SystemColors.Control;

**this**.login\_label.Location = new System.Drawing.Point(3, 0);

**this**.login\_label.Name = "login\_label";

**this**.login\_label.Size = new System.Drawing.Size(41, 15);

**this**.login\_label.TabIndex = 6;

**this**.login\_label.Text = "Логин";

**this**.loginText.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.loginText.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.loginText.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.loginText.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.loginText.Location = new System.Drawing.Point(0, 16);

**this**.loginText.MaxLength = 16;

**this**.loginText.Name = "loginText";

**this**.loginText.Size = new System.Drawing.Size(176, 21);

**this**.loginText.TabIndex = 3;

**this**.passwordText.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.passwordText.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.passwordText.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.passwordText.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.passwordText.Location = new System.Drawing.Point(0, 55);

**this**.passwordText.MaxLength = 32;

**this**.passwordText.Name = "passwordText";

**this**.passwordText.Size = new System.Drawing.Size(176, 21);

**this**.passwordText.TabIndex = 5;

**this**.password\_label.AutoSize = **true**;

**this**.password\_label.ForeColor = System.Drawing.SystemColors.Control;

**this**.password\_label.Location = new System.Drawing.Point(0, 39);

**this**.password\_label.Name = "password\_label";

**this**.password\_label.Size = new System.Drawing.Size(45, 13);

**this**.password\_label.TabIndex = 4;

**this**.password\_label.Text = "Пароль";

**this**.panel1.Controls.**Add**(**this**.registrationBtn);

**this**.panel1.Controls.**Add**(**this**.login\_label);

**this**.panel1.Controls.**Add**(**this**.loginText);

**this**.panel1.Controls.**Add**(**this**.passwordText);

**this**.panel1.Controls.**Add**(**this**.password\_label);

**this**.panel1.Location = new System.Drawing.Point(3, 37);

**this**.panel1.Name = "panel1";

**this**.panel1.Size = new System.Drawing.Size(176, 134);

**this**.panel1.TabIndex = 9;

**this**.registrationBtn.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(82)))), ((**int**)(((**byte**)(136)))), ((**int**)(((**byte**)(193)))));

**this**.registrationBtn.FlatAppearance.BorderSize = 0;

**this**.registrationBtn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

**this**.registrationBtn.ForeColor = System.Drawing.SystemColors.Control;

**this**.registrationBtn.Location = new System.Drawing.Point(26, 112);

**this**.registrationBtn.Name = "registrationBtn";

**this**.registrationBtn.Size = new System.Drawing.Size(123, 22);

**this**.registrationBtn.TabIndex = 7;

**this**.registrationBtn.Text = "Зарегистрироваться";

**this**.registrationBtn.UseVisualStyleBackColor = **false**;

**this**.registrationBtn.Click += new System.EventHandler(**this**.registrationBtn\_Click);

**this**.goToLoginBtn.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(82)))), ((**int**)(((**byte**)(136)))), ((**int**)(((**byte**)(193)))));

**this**.goToLoginBtn.FlatAppearance.BorderSize = 0;

**this**.goToLoginBtn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

**this**.goToLoginBtn.ForeColor = System.Drawing.SystemColors.Control;

**this**.goToLoginBtn.Location = new System.Drawing.Point(60, 177);

**this**.goToLoginBtn.Name = "goToLoginBtn";

**this**.goToLoginBtn.Size = new System.Drawing.Size(64, 23);

**this**.goToLoginBtn.TabIndex = 8;

**this**.goToLoginBtn.Text = "Назад";

**this**.goToLoginBtn.UseVisualStyleBackColor = **false**;

**this**.goToLoginBtn.Click += new System.EventHandler(**this**.goToLoginBtn\_Click);

**this**.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

**this**.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.Controls.**Add**(**this**.goToLoginBtn);

**this**.Controls.**Add**(**this**.label);

**this**.Controls.**Add**(**this**.panel1);

**this**.Name = "RegistrationElement";

**this**.Size = new System.Drawing.Size(182, 207);

**this**.panel1.ResumeLayout(**false**);

**this**.panel1.PerformLayout();

**this**.ResumeLayout(**false**);

**this**.PerformLayout();}

**private** System.Windows.Forms.Label label;

**private** System.Windows.Forms.Label login\_label;

**private** System.Windows.Forms.TextBox loginText;

**private** System.Windows.Forms.TextBox passwordText;

**private** System.Windows.Forms.Label password\_label;

**private** System.Windows.Forms.Panel panel1;

**private** System.Windows.Forms.Button registrationBtn;

**private** System.Windows.Forms.Button goToLoginBtn;

}

}

Класс отрисовки облачка сообщения. Файл Shaping.cs

**using** System.Drawing;

**using** System.Drawing.Drawing2D;

**namespace** PlumMesseger{

**public** **abstract** **class** RoundedRectangle{

**public** **enum** RectangleCorners{

None = 0, TopLeft = 1, TopRight = 2, BottomLeft = 4, BottomRight = 8,

All = TopLeft | TopRight | BottomLeft | BottomRight

}

**public** **static** GraphicsPath Create(**int** x, **int** y, **int** width, **int** height,**int** radius, RectangleCorners corners){

**int** xw = x + width;

**int** yh = y + height;

**int** xwr = xw - radius;

**int** yhr = yh - radius;

**int** xr = x + radius;

**int** yr = y + radius;

**int** r2 = radius \* 2;

**int** xwr2 = xw - r2;

**int** yhr2 = yh - r2;

GraphicsPath p = new GraphicsPath();

p.StartFigure();

**if** ((RectangleCorners.TopLeft & corners) == RectangleCorners.TopLeft){

p.AddArc(x, y, r2, r2, 180, 90);

}

**else**{

p.AddLine(x, yr, x, y);

p.AddLine(x, y, xr, y);

}

p.AddLine(xr, y, xwr, y);

**if** ((RectangleCorners.TopRight & corners) == RectangleCorners.TopRight){

p.AddArc(xwr2, y, r2, r2, 270, 90);

}

**else**{

p.AddLine(xwr, y, xw, y);

p.AddLine(xw, y, xw, yr);

}

p.AddLine(xw, yr, xw, yhr);

**if** ((RectangleCorners.BottomRight & corners) == RectangleCorners.BottomRight){

p.AddArc(xwr2, yhr2, r2, r2, 0, 90);

}**else**{

p.AddLine(xw, yhr, xw, yh);

p.AddLine(xw, yh, xwr, yh);

}

p.AddLine(xwr, yh, xr, yh);

**if** ((RectangleCorners.BottomLeft & corners) == RectangleCorners.BottomLeft){

p.AddArc(x, yhr2, r2, r2, 90, 90);

}**else**{

p.AddLine(xr, yh, x, yh);

p.AddLine(x, yh, x, yhr);

}

p.AddLine(x, yhr, x, yr);

p.CloseFigure();

**return** p;

}

**public** **static** GraphicsPath Create(Rectangle rect, **int** radius, RectangleCorners c){ **return** Create(rect.X, rect.Y, rect.Width, rect.Height, radius, c); }

}

}

Класс панели редактирования информации. Файл UserEditElement.cs

**using** PlumMessenger.Connection.Requests;

**using** PlumMessenger.Models;

**using** System;

**using** System.Windows.Forms;

**namespace** PlumMessenger.Controls{

**public** **partial** **class** UserEditElement : UserControl{

**public** **event** EventHandler SaveClickedEvent;

**public** **event** EventHandler CloseClickedEvent;

User modifiedUser = new User();

**public** UserEditElement(){

InitializeComponent();}

**private** **void** closeBtn\_Click(**object** sender, EventArgs e){CloseClickedEvent?.Invoke(**this**, e);}

**private** **async** **void** saveBtn\_Click(**object** sender, EventArgs e){

SaveClickedEvent?.Invoke(**this**, e);

UserRequest ur = new UserRequest();

**try**{

SetModifiedUserFields();

**if** (**await** ur.EditUser(modifiedUser)){

**await** ur.GetUser();

MessageBox.Show("Информация обновлена", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

} **else**{

MessageBox.Show("Не было внесено изменений", "Внимание", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}}

**catch** (Exception exc){

MessageBox.Show(exc.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}}

**private** **void** SetModifiedUserFields(){

modifiedUser.Login = loginTextBox.Text;

modifiedUser.Username = userNameTextBox.Text;

**if** (phoneTextBox.MaskCompleted)modifiedUser.PhoneNumber = phoneTextBox.Text;

modifiedUser.PhonePublicy = phoneVisibilityCheckBox.Checked;

modifiedUser.password = passwordTextBox.Text;

modifiedUser.Status = statusTextBox.Text;}

**private** **void** SetFields(){

loginTextBox.Text = UserRequest.CurrentUser.Login;

userNameTextBox.Text = UserRequest.CurrentUser.Username;

phoneTextBox.Text = UserRequest.CurrentUser.PhoneNumber;

phoneVisibilityCheckBox.Checked = UserRequest.CurrentUser.PhonePublicy;

passwordTextBox.Text = **string**.Empty;

statusTextBox.Text = UserRequest.CurrentUser.Status;}

**private** **void** UserEditElement\_Load(**object** sender, EventArgs e){

User.UpdateFromUser(modifiedUser, UserRequest.CurrentUser);

UserRequest.CurrentUser.EditEvent += CurrentUserEdited;

SetFields();

}

**private** **void** CurrentUserEdited(**object** sender, EventArgs e){SetFields();}

}}

Файл дизайна панели редактирования. Файл UserEditElement.Designer.cs

**namespace** PlumMessenger.Controls{

**partial** **class** UserEditElement{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){components.Dispose();}

**base**.Dispose(disposing);}

**private** **void** InitializeComponent(){

**this**.panel1 = new System.Windows.Forms.Panel();

**this**.panel4 = new System.Windows.Forms.Panel();

**this**.panel3 = new System.Windows.Forms.Panel();

**this**.saveBtn = new System.Windows.Forms.Button();

**this**.closeBtn = new System.Windows.Forms.Button();

**this**.panel2 = new System.Windows.Forms.Panel();

**this**.phoneVisibilityCheckBox = new System.Windows.Forms.CheckBox();

**this**.phoneTextBox = new System.Windows.Forms.MaskedTextBox();

**this**.statusTextBox = new System.Windows.Forms.TextBox();

**this**.label6 = new System.Windows.Forms.Label();

**this**.passwordTextBox = new System.Windows.Forms.TextBox();

**this**.label5 = new System.Windows.Forms.Label();

**this**.label3 = new System.Windows.Forms.Label();

**this**.userNameTextBox = new System.Windows.Forms.TextBox();

**this**.label2 = new System.Windows.Forms.Label();

**this**.loginTextBox = new System.Windows.Forms.TextBox();

**this**.label1 = new System.Windows.Forms.Label();

**this**.panel5 = new System.Windows.Forms.Panel();

**this**.label4 = new System.Windows.Forms.Label();

**this**.panel1.SuspendLayout();

**this**.panel4.SuspendLayout();

**this**.panel3.SuspendLayout();

**this**.panel2.SuspendLayout();

**this**.panel5.SuspendLayout();

**this**.SuspendLayout();

**this**.panel1.Anchor = System.Windows.Forms.AnchorStyles.None;

**this**.panel1.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(14)))), ((**int**)(((**byte**)(22)))), ((**int**)(((**byte**)(33)))));

**this**.panel1.Controls.**Add**(**this**.panel4);

**this**.panel1.Location = new System.Drawing.Point(3, 3);

**this**.panel1.Name = "panel1";

**this**.panel1.Padding = new System.Windows.Forms.Padding(10);

**this**.panel1.Size = new System.Drawing.Size(284, 347);

**this**.panel1.TabIndex = 0;

**this**.panel4.Controls.**Add**(**this**.panel3);

**this**.panel4.Controls.**Add**(**this**.panel2);

**this**.panel4.Controls.**Add**(**this**.panel5);

**this**.panel4.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.panel4.Location = new System.Drawing.Point(10, 10);

**this**.panel4.Name = "panel4";

**this**.panel4.Size = new System.Drawing.Size(264, 327);

**this**.panel4.TabIndex = 14;

**this**.panel3.Controls.**Add**(**this**.saveBtn);

**this**.panel3.Controls.**Add**(**this**.closeBtn);

**this**.panel3.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel3.Location = new System.Drawing.Point(0, 301);

**this**.panel3.Name = "panel3";

**this**.panel3.Size = new System.Drawing.Size(264, 25);

**this**.panel3.TabIndex = 2;

**this**.saveBtn.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(82)))), ((**int**)(((**byte**)(136)))), ((**int**)(((**byte**)(193)))));

**this**.saveBtn.Dock = System.Windows.Forms.DockStyle.Right;

**this**.saveBtn.FlatAppearance.BorderSize = 0;

**this**.saveBtn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

**this**.saveBtn.ForeColor = System.Drawing.SystemColors.Control;

**this**.saveBtn.Location = new System.Drawing.Point(164, 0);

**this**.saveBtn.Name = "saveBtn";

**this**.saveBtn.Size = new System.Drawing.Size(100, 25);

**this**.saveBtn.TabIndex = 1;

**this**.saveBtn.Text = "Сохранить";

**this**.saveBtn.UseVisualStyleBackColor = **false**;

**this**.saveBtn.Click += new System.EventHandler(**this**.saveBtn\_Click);

**this**.closeBtn.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(82)))), ((**int**)(((**byte**)(136)))), ((**int**)(((**byte**)(193)))));

**this**.closeBtn.Dock = System.Windows.Forms.DockStyle.Left;

**this**.closeBtn.FlatAppearance.BorderSize = 0;

**this**.closeBtn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

**this**.closeBtn.ForeColor = System.Drawing.SystemColors.Control;

**this**.closeBtn.Location = new System.Drawing.Point(0, 0);

**this**.closeBtn.Name = "closeBtn";

**this**.closeBtn.Size = new System.Drawing.Size(100, 25);

**this**.closeBtn.TabIndex = 0;

**this**.closeBtn.Text = "Закрыть";

**this**.closeBtn.UseVisualStyleBackColor = **false**;

**this**.closeBtn.Click += new System.EventHandler(**this**.closeBtn\_Click);

**this**.panel2.Controls.**Add**(**this**.phoneVisibilityCheckBox);

**this**.panel2.Controls.**Add**(**this**.phoneTextBox);

**this**.panel2.Controls.**Add**(**this**.statusTextBox);

**this**.panel2.Controls.**Add**(**this**.label6);

**this**.panel2.Controls.**Add**(**this**.passwordTextBox);

**this**.panel2.Controls.**Add**(**this**.label5);

**this**.panel2.Controls.**Add**(**this**.label3);

**this**.panel2.Controls.**Add**(**this**.userNameTextBox);

**this**.panel2.Controls.**Add**(**this**.label2);

**this**.panel2.Controls.**Add**(**this**.loginTextBox);

**this**.panel2.Controls.**Add**(**this**.label1);

**this**.panel2.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel2.Location = new System.Drawing.Point(0, 25);

**this**.panel2.Name = "panel2";

**this**.panel2.Size = new System.Drawing.Size(264, 276);

**this**.panel2.TabIndex = 1;

**this**.phoneVisibilityCheckBox.AutoSize = **true**;

**this**.phoneVisibilityCheckBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.phoneVisibilityCheckBox.ForeColor = System.Drawing.SystemColors.Control;

**this**.phoneVisibilityCheckBox.Location = new System.Drawing.Point(0, 127);

**this**.phoneVisibilityCheckBox.Name = "phoneVisibilityCheckBox";

**this**.phoneVisibilityCheckBox.Size = new System.Drawing.Size(257, 19);

**this**.phoneVisibilityCheckBox.TabIndex = 13;

**this**.phoneVisibilityCheckBox.Text = "Виден ли номер другим пользователям";

**this**.phoneVisibilityCheckBox.UseVisualStyleBackColor = **true**;

**this**.phoneTextBox.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.phoneTextBox.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.phoneTextBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.phoneTextBox.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.phoneTextBox.Location = new System.Drawing.Point(0, 100);

**this**.phoneTextBox.Mask = "+7-000-000-00-00";

**this**.phoneTextBox.Name = "phoneTextBox";

**this**.phoneTextBox.Size = new System.Drawing.Size(261, 21);

**this**.phoneTextBox.TabIndex = 12;

**this**.statusTextBox.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.statusTextBox.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.statusTextBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.statusTextBox.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.statusTextBox.Location = new System.Drawing.Point(0, 207);

**this**.statusTextBox.MaxLength = 139;

**this**.statusTextBox.Multiline = **true**;

**this**.statusTextBox.Name = "statusTextBox";

**this**.statusTextBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;

**this**.statusTextBox.Size = new System.Drawing.Size(261, 59);

**this**.statusTextBox.TabIndex = 11;

**this**.label6.AutoSize = **true**;

**this**.label6.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label6.ForeColor = System.Drawing.SystemColors.Control;

**this**.label6.Location = new System.Drawing.Point(3, 189);

**this**.label6.Name = "label6";

**this**.label6.Size = new System.Drawing.Size(47, 15);

**this**.label6.TabIndex = 10;

**this**.label6.Text = "Статус";

**this**.passwordTextBox.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.passwordTextBox.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.passwordTextBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.passwordTextBox.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.passwordTextBox.Location = new System.Drawing.Point(0, 166);

**this**.passwordTextBox.MaxLength = 32;

**this**.passwordTextBox.Name = "passwordTextBox";

**this**.passwordTextBox.Size = new System.Drawing.Size(261, 21);

**this**.passwordTextBox.TabIndex = 9;

**this**.label5.AutoSize = **true**;

**this**.label5.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label5.ForeColor = System.Drawing.SystemColors.Control;

**this**.label5.Location = new System.Drawing.Point(3, 148);

**this**.label5.Name = "label5";

**this**.label5.Size = new System.Drawing.Size(51, 15);

**this**.label5.TabIndex = 8;

**this**.label5.Text = "Пароль";

**this**.label3.AutoSize = **true**;

**this**.label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label3.ForeColor = System.Drawing.SystemColors.Control;

**this**.label3.Location = new System.Drawing.Point(3, 82);

**this**.label3.Name = "label3";

**this**.label3.Size = new System.Drawing.Size(109, 15);

**this**.label3.TabIndex = 4;

**this**.label3.Text = "Номер телефона";

**this**.userNameTextBox.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.userNameTextBox.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.userNameTextBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.userNameTextBox.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.userNameTextBox.Location = new System.Drawing.Point(0, 59);

**this**.userNameTextBox.MaxLength = 16;

**this**.userNameTextBox.Name = "userNameTextBox";

**this**.userNameTextBox.Size = new System.Drawing.Size(261, 21);

**this**.userNameTextBox.TabIndex = 3;

**this**.label2.AutoSize = **true**;

**this**.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label2.ForeColor = System.Drawing.SystemColors.Control;

**this**.label2.Location = new System.Drawing.Point(3, 41);

**this**.label2.Name = "label2";

**this**.label2.Size = new System.Drawing.Size(118, 15);

**this**.label2.TabIndex = 2;

**this**.label2.Text = "Имя пользователя";

**this**.loginTextBox.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.loginTextBox.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.loginTextBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.loginTextBox.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.loginTextBox.Location = new System.Drawing.Point(0, 18);

**this**.loginTextBox.MaxLength = 16;

**this**.loginTextBox.Name = "loginTextBox";

**this**.loginTextBox.Size = new System.Drawing.Size(261, 21);

**this**.loginTextBox.TabIndex = 1;

**this**.label1.AutoSize = **true**;

**this**.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label1.ForeColor = System.Drawing.SystemColors.Control;

**this**.label1.Location = new System.Drawing.Point(3, 0);

**this**.label1.Name = "label1";

**this**.label1.Size = new System.Drawing.Size(41, 15);

**this**.label1.TabIndex = 0;

**this**.label1.Text = "Логин";

**this**.panel5.Controls.**Add**(**this**.label4);

**this**.panel5.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel5.Location = new System.Drawing.Point(0, 0);

**this**.panel5.Name = "panel5";

**this**.panel5.Size = new System.Drawing.Size(264, 25);

**this**.panel5.TabIndex = 3;

**this**.label4.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label4.ForeColor = System.Drawing.SystemColors.Control;

**this**.label4.Location = new System.Drawing.Point(0, 0);

**this**.label4.Name = "label4";

**this**.label4.Size = new System.Drawing.Size(264, 25);

**this**.label4.TabIndex = 14;

**this**.label4.Text = "Редактировать";

**this**.label4.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

**this**.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

**this**.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.Controls.**Add**(**this**.panel1);

**this**.Name = "UserEditElement";

**this**.Size = new System.Drawing.Size(290, 354);

**this**.Load += new System.EventHandler(**this**.UserEditElement\_Load);

**this**.panel1.ResumeLayout(**false**);

**this**.panel4.ResumeLayout(**false**);

**this**.panel3.ResumeLayout(**false**);

**this**.panel2.ResumeLayout(**false**);

**this**.panel2.PerformLayout();

**this**.panel5.ResumeLayout(**false**);

**this**.ResumeLayout(**false**);}

**private** System.Windows.Forms.Panel panel1;

**private** System.Windows.Forms.Label label1;

**private** System.Windows.Forms.Panel panel2;

**private** System.Windows.Forms.TextBox loginTextBox;

**private** System.Windows.Forms.TextBox statusTextBox;

**private** System.Windows.Forms.Label label6;

**private** System.Windows.Forms.TextBox passwordTextBox;

**private** System.Windows.Forms.Label label5;

**private** System.Windows.Forms.Label label3;

**private** System.Windows.Forms.TextBox userNameTextBox;

**private** System.Windows.Forms.Label label2;

**private** System.Windows.Forms.MaskedTextBox phoneTextBox;

**private** System.Windows.Forms.CheckBox phoneVisibilityCheckBox;

**private** System.Windows.Forms.Panel panel3;

**private** System.Windows.Forms.Button closeBtn;

**private** System.Windows.Forms.Button saveBtn;

**private** System.Windows.Forms.Panel panel4;

**private** System.Windows.Forms.Panel panel5;

**private** System.Windows.Forms.Label label4;}}

Форма авторизации. Файл Login.cs

**using** System;

**using** System.Windows.Forms;

**using** PlumMessenger.Connection.Requests;

**namespace** PlumMessenger{

**public** **partial** **class** Login : Form{

AuthRequest authApi;

**public** Login(){

InitializeComponent();

authApi = new AuthRequest();}

**private** **void** Login\_FormClosing(**object** sender, FormClosingEventArgs e){

**if** (!AuthRequest.LoggedIn){Application.Exit();}

}

**private** **async** **void** login\_Click(**object** sender, EventArgs e){

**try**{

UserRequest.CurrentUserId = **await** authApi.SignInMethod(loginElement.GetLogin(), loginElement.GetPassword());

UserRequest ur = new UserRequest();

**await** ur.GetUser();

Close();

}**catch** (Exception exc){

MessageBox.Show(exc.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

**private** **async** **void** registration\_Click(**object** sender, EventArgs e){

**try**{

**await** authApi.SignUpMethod(registrationElement.GetLogin(), registrationElement.GetPassword());

MessageBox.Show("Вы зарегистрировались!", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information);

}**catch** (Exception exc){

MessageBox.Show(exc.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}}

**private** **void** Login\_Load(**object** sender, EventArgs e){

loginElement.RegistrationClicked += GoToRegistration;

registrationElement.LoginClicked += GoToLogin;

loginElement.LoginClicked += login\_Click;

registrationElement.RegistrationClicked += registration\_Click;}

**private** **void** GoToRegistration(**object** sender, EventArgs e){

loginElement.Visible = **false**;

registrationElement.Visible = **true**;

}

**private** **void** GoToLogin(**object** sender, EventArgs e){

loginElement.Visible = **true**;

registrationElement.Visible = **false**;

}}}

Дизайн формы авторизации. Файл Login.Designer.cs

**namespace** PlumMessenger{

**partial** **class** Login{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){components.Dispose();}

**base**.Dispose(disposing);

}

**private** **void** InitializeComponent(){

**this**.loginElement = new PlumMessenger.Controls.LoginElement();

**this**.registrationElement = new PlumMessenger.Controls.RegistrationElement();

**this**.SuspendLayout();

**this**.loginElement.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.loginElement.Location = new System.Drawing.Point(12, 12);

**this**.loginElement.Name = "loginElement";

**this**.loginElement.Size = new System.Drawing.Size(187, 207);

**this**.loginElement.TabIndex = 0;

**this**.registrationElement.Location = new System.Drawing.Point(12, 12);

**this**.registrationElement.Name = "registrationElement";

**this**.registrationElement.Size = new System.Drawing.Size(182, 207);

**this**.registrationElement.TabIndex = 1;

**this**.registrationElement.Visible = **false**;

**this**.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

**this**.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.ClientSize = new System.Drawing.Size(207, 220);

**this**.Controls.**Add**(**this**.registrationElement);

**this**.Controls.**Add**(**this**.loginElement);

**this**.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;

**this**.KeyPreview = **true**;

**this**.Name = "Login";

**this**.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

**this**.Text = "Вход";

**this**.FormClosing += new System.Windows.Forms.FormClosingEventHandler(**this**.Login\_FormClosing);

**this**.Load += new System.EventHandler(**this**.Login\_Load);

**this**.ResumeLayout(**false**);}

**private** Controls.LoginElement loginElement;

**private** Controls.RegistrationElement registrationElement;

}}

Главная форма. Файл MainF.cs

**using** PlumMessenger.Models;

**using** PlumMessenger.Connection.Requests;

**using** System;

**using** System.Collections.Generic;

**using** System.Data;

**using** System.Drawing;

**using** System.Linq;

**using** System.Threading.Tasks;

**using** System.Windows.Forms;

**using** System.Collections.Specialized;

**using** PlumMessenger.Connection;

**using** PlumMesseger;

**using** PlumMessenger.Controls;

**namespace** PlumMessenger{

**public** **partial** **class** MainF : Form{

ContactElement lastContactElement;

Contact currentContact;

**public** **static** ContactsRequest contacts;

Puller puller;

**int** WaitingUserId = -1;

**public** MainF(){InitializeComponent();}

**private** **void** MainF\_Load(**object** sender, EventArgs e){

Login login = new Login();

login.ShowDialog();

SetEditButtonHoveredListeners();

SetMinInfoHeaderClickedListeners();

searchPanelWrapper.Visible = **false**;

chatWrapper.Visible = **false**;

puller = new Puller();

puller.AddContact += AddContact;

puller.DeleteContact += DeleteContact;

puller.DeleteContact += searchTextBox\_TextChanged;

UserRequest.CurrentUser.EditEvent += CurrentUserEdited;

UpdateCurrentUserInfoLabels();

puller.StartPulling();

}

**private** **void** MainF\_Shown(**object** sender, EventArgs e){**this**.label1.Focus();}

**private** **void** SetEditButtonHoveredListeners(){

**foreach** (Control c **in** GetNestedControls<Control>(editUserInfoPanel)){

c.MouseEnter += editUserInfoPanel\_MouseEnter;

c.MouseLeave += editUserInfoPanel\_MouseLeave;

}}

**private** **void** SetMinInfoHeaderClickedListeners(){

**foreach** (Control c **in** GetNestedControls<Control>(contactMinInfoPanel)){

c.Click += contactMinInfoPanel\_Click;

}}

**private** **void** CurrentUserEdited(**object** sender, EventArgs e){UpdateCurrentUserInfoLabels();}

**private** **void** UpdateCurrentUserInfoLabels(){

userNameLabel.Text = UserRequest.CurrentUser.Username;

userLoginLabel.Text = "@"+UserRequest.CurrentUser.Login;

}

**private** **void** DeleteContact(**object** sender, EventArgs e){

NotifyCollectionChangedEventArgs args = (NotifyCollectionChangedEventArgs)e;

**this**.Invoke(new MethodInvoker(() => {contactsPanel.Controls.**Remove**(contactsPanel.Controls.Cast<ContactElement>().First(x => x.contact == (Contact)sender));

**if** (currentContact == (Contact)sender){

chatWrapper.Visible = **false**;

lastContactElement.Selected = **false**;

currentContact = **null**;

lastContactElement = **null**;}}));}

**private** **void** AddContact(**object** sender, EventArgs e){

NotifyCollectionChangedEventArgs args = (NotifyCollectionChangedEventArgs)e;

**this**.Invoke(new MethodInvoker(() => {

**if** (args.Action == NotifyCollectionChangedAction.**Add**){

Contact contact = (Contact)sender;

ContactElement element = new ContactElement(contact);

element.Dock = DockStyle.Top;

contactsPanel.RowCount = contactsPanel.RowCount + 1;

contactsPanel.RowStyles.**Add**(new RowStyle(SizeType.AutoSize));

contactsPanel.Controls.**Add**(element);

element.Clicked += ContactClicked;

element.RightClicked += element.OpenContextMenuDefault;

element.ContextMenuAction = (x =>{

Console.WriteLine($"{lastContactElement} {element}");

**return** Puller.contactRequest.DeleteContact(x);});

**if** (contact.Id == WaitingUserId){

ContactClicked(element, EventArgs.Empty);

searchTextBox\_TextChanged(**null**, EventArgs.Empty);

WaitingUserId = -1;}}}));}

**private** **void** ContactClicked(**object** sender, EventArgs e){

**this**.Invoke(new MethodInvoker(() => {

ContactElement clickedContactElement = (ContactElement)sender;

Contact contact = clickedContactElement.contact;

**if** (currentContact == **null**)

SetChatVisibility(**true**);

**if** (lastContactElement != **null**){

**if** (lastContactElement == clickedContactElement)

**return**;

lastContactElement.Selected = **false**;}

lastContactElement = clickedContactElement;

clickedContactElement.Selected = **true**;

Puller.ObservingUserId = -1;

chatPanel.Controls.Clear();

chatPanel.RowStyles.Clear();

chatPanel.ColumnStyles.Clear();

**if** (currentContact != **null**){EditCurrentContactListeners(**false**);}

currentContact = contact;

EditCurrentContactListeners(**true**);

ChatHeaderToCurrentContact();

LoadMessages();

Puller.ObservingUserId = contact.Id;}));}

**private** **void** EditCurrentContactListeners(**bool** **add**){

**if** (**add**){

currentContact.MessageAddedEvent += AddMessage;

currentContact.EditEvent += СurrentContactEdited;

} **else**{

currentContact.EditEvent -= СurrentContactEdited;

currentContact.MessageAddedEvent -= AddMessage;

}}

**private** **void** SetChatVisibility(**bool** state){chatWrapper.Visible = state;}

**private** **void** СurrentContactEdited(**object** sender, EventArgs e){ChatHeaderToCurrentContact();}

**private** **void** ChatHeaderToCurrentContact(){

nameLabel.Text = currentContact.Username;

onlineLabel.Text = currentContact.IsOnline() ? "Online" : "Offline";}

**private** **void** LoadMessages(){

**foreach** (Models.Message message **in** currentContact.GetMessages()){AddMessageBubble(message);}}

**private** **void** AddMessage(**object** sender, EventArgs e){

**this**.Invoke(new MethodInvoker(() => {

NotifyCollectionChangedEventArgs messageEvent = (NotifyCollectionChangedEventArgs)e;

**foreach** (Models.Message message **in** messageEvent.NewItems){AddMessageBubble(message);}}));}

**private** **void** AddMessageBubble(Models.Message message){

UserControl bubble;

**string** dateFormat = "{0:HH:mm}";

**if** ((DateTime.UtcNow - message.Date).TotalDays >= 1.0){dateFormat = "{0:dd.MM HH:mm}";}

**if** (message.OwnerId == UserRequest.CurrentUserId){

bubble = new MeBubble();

((MeBubble)bubble).Body = message.Text;

((MeBubble)bubble).Time = **String**.Format(dateFormat, message.Date);}**else**{

bubble = new YouBubble();

((YouBubble)bubble).Body = message.Text;

((YouBubble)bubble).Time = **String**.Format(dateFormat, message.Date);}

bubble.AutoSize = **false**;

bubble.Dock = DockStyle.Fill;

bubble.TabIndex = 0;

chatPanel.RowCount = chatPanel.RowCount + 1;

chatPanel.RowStyles.**Add**(new RowStyle(SizeType.AutoSize));

chatPanel.Controls.**Add**(bubble);

}

**private** **void** MainF\_FormClosing(**object** sender, FormClosingEventArgs e){

**if** (puller != **null**)puller.Looping = **false**;}

**private** **async** **void** sendBtn\_Click(**object** sender, EventArgs e){

**if** (!**String**.IsNullOrWhiteSpace(messageTextBox.Text)){

**if** (!Puller.AvailableContacts.Contains(currentContact)){WaitingUserId = currentContact.Id;}

**try**{

**var** res = **await** Puller.messagesRequest

.SendMessage(currentContact.Id, messageTextBox.Text.Trim(' '));

**if** (res)messageTextBox.Clear();

**else**

MessageBox.Show("Не доставлено", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}**catch** (Exception exc){

MessageBox.Show(exc.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);}}}

**private** **void** chatPanel\_ControlAdded(**object** sender, ControlEventArgs e){

chatPanelWrapperL2.ScrollControlIntoView(e.Control);}

**private** **void** editUserInfo\_Click(**object** sender, EventArgs e){

UserEditElement userEditElement = new UserEditElement();

userEditElement.CloseClickedEvent += PopUpControlCloseClicked;

**this**.Controls.**Add**(userEditElement);

userEditElement.Dock = DockStyle.Fill;

userEditElement.BringToFront();

leftPanelWrapper.Enabled = **false**;

}

**private** **void** PopUpControlCloseClicked(**object** sender, EventArgs e){

**this**.Controls.**Remove**((Control)sender);

((Control)sender).Dispose();

leftPanelWrapper.Enabled = **true**;

}

**private** **void** editUserInfoPanel\_MouseEnter(**object** sender, EventArgs e){

editUserInfoPanel.BackColor = Color.FromArgb(32, 43, 54);}

**private** **void** editUserInfoPanel\_MouseLeave(**object** sender, EventArgs e){

editUserInfoPanel.BackColor = Color.FromArgb(23, 33, 43);}

**private** IEnumerable<ControlType> GetNestedControls<ControlType>(Control control){

**if** (control.Controls == **null** || control.Controls.Count == 0)**return** Enumerable.Empty<ControlType>();

**return** control.Controls.OfType<ControlType>().Concat(control.Controls.Cast<Control>().SelectMany(x => GetNestedControls<ControlType>(x)));}

**private** **void** contactMinInfoPanel\_Click(**object** sender, EventArgs e){**if** (currentContact == **null**)**return**;

ContactInfoElement contactInfo = new ContactInfoElement(currentContact);

contactInfo.CloseClickedEvent += PopUpControlCloseClicked;

**this**.Controls.**Add**(contactInfo);

contactInfo.Dock = DockStyle.Fill;

contactInfo.BringToFront();

leftPanelWrapper.Enabled = **false**;

}

**private** **async** **void** searchTextBox\_TextChanged(**object** sender, EventArgs e){

**if** (searchTextBox.Text.Length == 0){

searchPanelWrapper.Visible = **false**;

searchPanel.Controls.Clear();

searchPanel.RowStyles.Clear();

searchPanel.ColumnStyles.Clear();

**return**;

} **else**{

searchPanelWrapper.Visible = **true**;

**string** searchString = searchTextBox.Text;

List<Contact> searchedUsers = **await** SearchUsers(searchString);

searchPanel.Controls.Clear();

searchPanel.RowStyles.Clear();

searchPanel.ColumnStyles.Clear();

FillSearchPanel(searchedUsers);

}}

**private** **void** FillSearchPanel(List<Contact> searchedUsers){

**foreach** (Contact contact **in** searchedUsers){

**this**.Invoke(new MethodInvoker(() => {

ContactElement element = new ContactElement(contact);

element.Dock = DockStyle.Top;

searchPanel.RowCount = searchPanel.RowCount + 1;

searchPanel.RowStyles.**Add**(new RowStyle(SizeType.AutoSize));

searchPanel.Controls.**Add**(element);

element.Clicked += ContactClicked;

element.RightClicked += element.OpenContextMenuFromSearch;

element.ContextMenuAction = ((x) =>{

WaitingUserId = contact.Id;

**return** Puller.contactRequest.AddContact(x);

});}));}}

**private** **async** Task<List<Contact>> SearchUsers(**string** searchString){

**return** **await** Puller.userRequest.SearchUsers(searchString);}}}

Дизайн главной формы. Файл MainF.Designer.cs

**namespace** PlumMessenger{

**partial** **class** MainF{

**private** System.ComponentModel.IContainer components = **null**;

**protected** **override** **void** Dispose(**bool** disposing){

**if** (disposing && (components != **null**)){components.Dispose();}

**base**.Dispose(disposing);}

**private** **void** InitializeComponent(){

**this**.leftPanelWrapper = new System.Windows.Forms.Panel();

**this**.contactsPanelWrapper = new System.Windows.Forms.Panel();

**this**.contactsPanel = new System.Windows.Forms.TableLayoutPanel();

**this**.contactsLabelPanel = new System.Windows.Forms.Panel();

**this**.contactsLabel = new System.Windows.Forms.Label();

**this**.searchPanelWrapper = new System.Windows.Forms.Panel();

**this**.searchLabelPanel = new System.Windows.Forms.Panel();

**this**.panel1 = new System.Windows.Forms.Panel();

**this**.searchTextBox = new System.Windows.Forms.TextBox();

**this**.label1 = new System.Windows.Forms.Label();

**this**.leftPanelHeader = new System.Windows.Forms.Panel();

**this**.userInfoPanel = new System.Windows.Forms.Panel();

**this**.userLoginLabel = new System.Windows.Forms.Label();

**this**.userNameLabel = new System.Windows.Forms.Label();

**this**.editUserInfoPanel = new System.Windows.Forms.Panel();

**this**.editUserInfo = new System.Windows.Forms.PictureBox();

**this**.contentPanelWrapper = new System.Windows.Forms.Panel();

**this**.chatWrapper = new System.Windows.Forms.Panel();

**this**.chatPanelWrapperL2 = new System.Windows.Forms.Panel();

**this**.chatPanel = new System.Windows.Forms.TableLayoutPanel();

**this**.messagePanelWrapper = new System.Windows.Forms.Panel();

**this**.messagePanel = new System.Windows.Forms.Panel();

**this**.messageTextBox = new System.Windows.Forms.TextBox();

**this**.sendButtonPanel = new System.Windows.Forms.Panel();

**this**.sendBtn = new System.Windows.Forms.Button();

**this**.contactMinInfoPanel = new System.Windows.Forms.Panel();

**this**.onlineLabel = new System.Windows.Forms.Label();

**this**.nameLabel = new System.Windows.Forms.Label();

**this**.label2 = new System.Windows.Forms.Label();

**this**.searchPanel = new System.Windows.Forms.TableLayoutPanel();

**this**.leftPanelWrapper.SuspendLayout();

**this**.contactsPanelWrapper.SuspendLayout();

**this**.contactsLabelPanel.SuspendLayout();

**this**.searchPanelWrapper.SuspendLayout();

**this**.searchLabelPanel.SuspendLayout();

**this**.panel1.SuspendLayout();

**this**.leftPanelHeader.SuspendLayout();

**this**.userInfoPanel.SuspendLayout();

**this**.editUserInfoPanel.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(**this**.editUserInfo)).BeginInit();

**this**.contentPanelWrapper.SuspendLayout();

**this**.chatWrapper.SuspendLayout();

**this**.chatPanelWrapperL2.SuspendLayout();

**this**.messagePanelWrapper.SuspendLayout();

**this**.messagePanel.SuspendLayout();

**this**.sendButtonPanel.SuspendLayout();

**this**.contactMinInfoPanel.SuspendLayout();

**this**.SuspendLayout();

**this**.leftPanelWrapper.Controls.**Add**(**this**.contactsPanelWrapper);

**this**.leftPanelWrapper.Controls.**Add**(**this**.contactsLabelPanel);

**this**.leftPanelWrapper.Controls.**Add**(**this**.searchPanelWrapper);

**this**.leftPanelWrapper.Controls.**Add**(**this**.searchLabelPanel);

**this**.leftPanelWrapper.Controls.**Add**(**this**.leftPanelHeader);

**this**.leftPanelWrapper.Dock = System.Windows.Forms.DockStyle.Left;

**this**.leftPanelWrapper.Location = new System.Drawing.Point(0, 0);

**this**.leftPanelWrapper.Name = "leftPanelWrapper";

**this**.leftPanelWrapper.Size = new System.Drawing.Size(279, 489);

**this**.leftPanelWrapper.TabIndex = 0;

**this**.contactsPanelWrapper.AutoScroll = **true**;

**this**.contactsPanelWrapper.Controls.**Add**(**this**.contactsPanel);

**this**.contactsPanelWrapper.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.contactsPanelWrapper.Location = new System.Drawing.Point(0, 303);

**this**.contactsPanelWrapper.Name = "contactsPanelWrapper";

**this**.contactsPanelWrapper.Size = new System.Drawing.Size(279, 186);

**this**.contactsPanelWrapper.TabIndex = 1;

**this**.contactsPanel.AutoSize = **true**;

**this**.contactsPanel.ColumnCount = 1;

**this**.contactsPanel.ColumnStyles.**Add**(new System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 100F));

**this**.contactsPanel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.contactsPanel.Location = new System.Drawing.Point(0, 0);

**this**.contactsPanel.Name = "contactsPanel";

**this**.contactsPanel.RowCount = 5;

**this**.contactsPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.contactsPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.contactsPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.contactsPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.contactsPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.contactsPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F));

**this**.contactsPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F));

**this**.contactsPanel.Size = new System.Drawing.Size(279, 0);

**this**.contactsPanel.TabIndex = 2;

**this**.contactsLabelPanel.AutoSize = **true**;

**this**.contactsLabelPanel.Controls.**Add**(**this**.contactsLabel);

**this**.contactsLabelPanel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.contactsLabelPanel.Location = new System.Drawing.Point(0, 276);

**this**.contactsLabelPanel.Name = "contactsLabelPanel";

**this**.contactsLabelPanel.Size = new System.Drawing.Size(279, 27);

**this**.contactsLabelPanel.TabIndex = 5;

**this**.contactsLabel.AutoSize = **true**;

**this**.contactsLabel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.contactsLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.contactsLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.contactsLabel.Location = new System.Drawing.Point(0, 0);

**this**.contactsLabel.Name = "contactsLabel";

**this**.contactsLabel.Padding = new System.Windows.Forms.Padding(0, 5, 0, 5);

**this**.contactsLabel.Size = new System.Drawing.Size(76, 27);

**this**.contactsLabel.TabIndex = 3;

**this**.contactsLabel.Text = "Контакты:";

**this**.searchPanelWrapper.AutoScroll = **true**;

**this**.searchPanelWrapper.Controls.**Add**(**this**.searchPanel);

**this**.searchPanelWrapper.Dock = System.Windows.Forms.DockStyle.Top;

**this**.searchPanelWrapper.Location = new System.Drawing.Point(0, 78);

**this**.searchPanelWrapper.Name = "searchPanelWrapper";

**this**.searchPanelWrapper.Size = new System.Drawing.Size(279, 198);

**this**.searchPanelWrapper.TabIndex = 1;

**this**.searchLabelPanel.AutoSize = **true**;

**this**.searchLabelPanel.Controls.**Add**(**this**.panel1);

**this**.searchLabelPanel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.searchLabelPanel.Location = new System.Drawing.Point(0, 47);

**this**.searchLabelPanel.Name = "searchLabelPanel";

**this**.searchLabelPanel.Size = new System.Drawing.Size(279, 31);

**this**.searchLabelPanel.TabIndex = 4;

**this**.panel1.AutoSize = **true**;

**this**.panel1.Controls.**Add**(**this**.searchTextBox);

**this**.panel1.Controls.**Add**(**this**.label1);

**this**.panel1.Dock = System.Windows.Forms.DockStyle.Top;

**this**.panel1.Location = new System.Drawing.Point(0, 0);

**this**.panel1.Name = "panel1";

**this**.panel1.Size = new System.Drawing.Size(279, 31);

**this**.panel1.TabIndex = 5;

**this**.searchTextBox.Anchor = System.Windows.Forms.AnchorStyles.None;

**this**.searchTextBox.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.searchTextBox.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.searchTextBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.searchTextBox.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.searchTextBox.Location = new System.Drawing.Point(51, 4);

**this**.searchTextBox.Margin = new System.Windows.Forms.Padding(0);

**this**.searchTextBox.MaxLength = 16;

**this**.searchTextBox.Name = "searchTextBox";

**this**.searchTextBox.Size = new System.Drawing.Size(228, 23);

**this**.searchTextBox.TabIndex = 4;

**this**.searchTextBox.TextChanged += new System.EventHandler(**this**.searchTextBox\_TextChanged);

**this**.label1.AutoSize = **true**;

**this**.label1.Dock = System.Windows.Forms.DockStyle.Top;

**this**.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label1.ForeColor = System.Drawing.SystemColors.Control;

**this**.label1.Location = new System.Drawing.Point(0, 0);

**this**.label1.Name = "label1";

**this**.label1.Padding = new System.Windows.Forms.Padding(0, 7, 0, 7);

**this**.label1.Size = new System.Drawing.Size(52, 31);

**this**.label1.TabIndex = 3;

**this**.label1.Text = "Поиск:";

**this**.leftPanelHeader.AutoSize = **true**;

**this**.leftPanelHeader.Controls.**Add**(**this**.userInfoPanel);

**this**.leftPanelHeader.Controls.**Add**(**this**.editUserInfoPanel);

**this**.leftPanelHeader.Dock = System.Windows.Forms.DockStyle.Top;

**this**.leftPanelHeader.Location = new System.Drawing.Point(0, 0);

**this**.leftPanelHeader.Name = "leftPanelHeader";

**this**.leftPanelHeader.Size = new System.Drawing.Size(279, 47);

**this**.leftPanelHeader.TabIndex = 2;

**this**.userInfoPanel.AutoSize = **true**;

**this**.userInfoPanel.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.userInfoPanel.Controls.**Add**(**this**.userLoginLabel);

**this**.userInfoPanel.Controls.**Add**(**this**.userNameLabel);

**this**.userInfoPanel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.userInfoPanel.Location = new System.Drawing.Point(0, 0);

**this**.userInfoPanel.Name = "userInfoPanel";

**this**.userInfoPanel.Size = new System.Drawing.Size(234, 47);

**this**.userInfoPanel.TabIndex = 3;

**this**.userLoginLabel.AutoSize = **true**;

**this**.userLoginLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 7F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.userLoginLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.userLoginLabel.Location = new System.Drawing.Point(6, 29);

**this**.userLoginLabel.Margin = new System.Windows.Forms.Padding(3, 0, 3, 5);

**this**.userLoginLabel.Name = "userLoginLabel";

**this**.userLoginLabel.Size = new System.Drawing.Size(35, 13);

**this**.userLoginLabel.TabIndex = 1;

**this**.userLoginLabel.Text = "label4";

**this**.userNameLabel.AutoSize = **true**;

**this**.userNameLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.userNameLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.userNameLabel.Location = new System.Drawing.Point(6, 9);

**this**.userNameLabel.Name = "userNameLabel";

**this**.userNameLabel.Size = new System.Drawing.Size(46, 17);

**this**.userNameLabel.TabIndex = 0;

**this**.userNameLabel.Text = "label3";

**this**.editUserInfoPanel.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.editUserInfoPanel.Controls.**Add**(**this**.editUserInfo);

**this**.editUserInfoPanel.Dock = System.Windows.Forms.DockStyle.Right;

**this**.editUserInfoPanel.Location = new System.Drawing.Point(234, 0);

**this**.editUserInfoPanel.Name = "editUserInfoPanel";

**this**.editUserInfoPanel.Padding = new System.Windows.Forms.Padding(8);

**this**.editUserInfoPanel.Size = new System.Drawing.Size(45, 47);

**this**.editUserInfoPanel.TabIndex = 4;

**this**.editUserInfoPanel.MouseEnter += new System.EventHandler(**this**.editUserInfoPanel\_MouseEnter);

**this**.editUserInfoPanel.MouseLeave += new System.EventHandler(**this**.editUserInfoPanel\_MouseLeave);

**this**.editUserInfo.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.editUserInfo.Image = global::PlumMessenger.Properties.Resources.edit;

**this**.editUserInfo.Location = new System.Drawing.Point(8, 8);

**this**.editUserInfo.Name = "editUserInfo";

**this**.editUserInfo.Size = new System.Drawing.Size(29, 31);

**this**.editUserInfo.SizeMode = System.Windows.Forms.PictureBoxSizeMode.StretchImage;

**this**.editUserInfo.TabIndex = 2;

**this**.editUserInfo.TabStop = **false**;

**this**.editUserInfo.Click += new System.EventHandler(**this**.editUserInfo\_Click);

**this**.contentPanelWrapper.Controls.**Add**(**this**.chatWrapper);

**this**.contentPanelWrapper.Controls.**Add**(**this**.label2);

**this**.contentPanelWrapper.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.contentPanelWrapper.Location = new System.Drawing.Point(279, 0);

**this**.contentPanelWrapper.Name = "contentPanelWrapper";

**this**.contentPanelWrapper.Size = new System.Drawing.Size(648, 489);

**this**.contentPanelWrapper.TabIndex = 1;

**this**.chatWrapper.AutoSize = **true**;

**this**.chatWrapper.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(80)))), ((**int**)(((**byte**)(80)))), ((**int**)(((**byte**)(80)))));

**this**.chatWrapper.Controls.**Add**(**this**.chatPanelWrapperL2);

**this**.chatWrapper.Controls.**Add**(**this**.messagePanelWrapper);

**this**.chatWrapper.Controls.**Add**(**this**.contactMinInfoPanel);

**this**.chatWrapper.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.chatWrapper.Location = new System.Drawing.Point(0, 0);

**this**.chatWrapper.Name = "chatWrapper";

**this**.chatWrapper.Size = new System.Drawing.Size(648, 489);

**this**.chatWrapper.TabIndex = 5;

**this**.chatPanelWrapperL2.AutoScroll = **true**;

**this**.chatPanelWrapperL2.AutoSize = **true**;

**this**.chatPanelWrapperL2.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(14)))), ((**int**)(((**byte**)(22)))), ((**int**)(((**byte**)(33)))));

**this**.chatPanelWrapperL2.Controls.**Add**(**this**.chatPanel);

**this**.chatPanelWrapperL2.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.chatPanelWrapperL2.Location = new System.Drawing.Point(0, 45);

**this**.chatPanelWrapperL2.Name = "chatPanelWrapperL2";

**this**.chatPanelWrapperL2.Size = new System.Drawing.Size(648, 375);

**this**.chatPanelWrapperL2.TabIndex = 4;

**this**.chatPanel.AutoSize = **true**;

**this**.chatPanel.ColumnCount = 1;

**this**.chatPanel.ColumnStyles.**Add**(new System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 100F));

**this**.chatPanel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.chatPanel.Location = new System.Drawing.Point(0, 0);

**this**.chatPanel.Name = "chatPanel";

**this**.chatPanel.RowCount = 4;

**this**.chatPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.chatPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.chatPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.chatPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.chatPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.chatPanel.Size = new System.Drawing.Size(648, 0);

**this**.chatPanel.TabIndex = 3;

**this**.chatPanel.ControlAdded += new System.Windows.Forms.ControlEventHandler(**this**.chatPanel\_ControlAdded);

**this**.messagePanelWrapper.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.messagePanelWrapper.Controls.**Add**(**this**.messagePanel);

**this**.messagePanelWrapper.Controls.**Add**(**this**.sendButtonPanel);

**this**.messagePanelWrapper.Dock = System.Windows.Forms.DockStyle.Bottom;

**this**.messagePanelWrapper.Location = new System.Drawing.Point(0, 420);

**this**.messagePanelWrapper.Name = "messagePanelWrapper";

**this**.messagePanelWrapper.Size = new System.Drawing.Size(648, 69);

**this**.messagePanelWrapper.TabIndex = 3;

**this**.messagePanel.Controls.**Add**(**this**.messageTextBox);

**this**.messagePanel.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.messagePanel.Location = new System.Drawing.Point(0, 0);

**this**.messagePanel.Name = "messagePanel";

**this**.messagePanel.Padding = new System.Windows.Forms.Padding(5);

**this**.messagePanel.Size = new System.Drawing.Size(528, 69);

**this**.messagePanel.TabIndex = 5;

**this**.messageTextBox.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(48)))), ((**int**)(((**byte**)(62)))));

**this**.messageTextBox.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;

**this**.messageTextBox.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.messageTextBox.Font = new System.Drawing.Font("Microsoft Sans Serif", 11F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.messageTextBox.ForeColor = System.Drawing.SystemColors.ControlLight;

**this**.messageTextBox.Location = new System.Drawing.Point(5, 5);

**this**.messageTextBox.MaxLength = 1000;

**this**.messageTextBox.Multiline = **true**;

**this**.messageTextBox.Name = "messageTextBox";

**this**.messageTextBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;

**this**.messageTextBox.Size = new System.Drawing.Size(518, 59);

**this**.messageTextBox.TabIndex = 0;

**this**.sendButtonPanel.AutoSize = **true**;

**this**.sendButtonPanel.Controls.**Add**(**this**.sendBtn);

**this**.sendButtonPanel.Dock = System.Windows.Forms.DockStyle.Right;

**this**.sendButtonPanel.Location = new System.Drawing.Point(528, 0);

**this**.sendButtonPanel.MinimumSize = new System.Drawing.Size(120, 69);

**this**.sendButtonPanel.Name = "sendButtonPanel";

**this**.sendButtonPanel.Padding = new System.Windows.Forms.Padding(10, 0, 10, 5);

**this**.sendButtonPanel.Size = new System.Drawing.Size(120, 69);

**this**.sendButtonPanel.TabIndex = 4;

**this**.sendBtn.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(82)))), ((**int**)(((**byte**)(136)))), ((**int**)(((**byte**)(193)))));

**this**.sendBtn.Dock = System.Windows.Forms.DockStyle.Bottom;

**this**.sendBtn.FlatAppearance.BorderSize = 0;

**this**.sendBtn.FlatStyle = System.Windows.Forms.FlatStyle.Flat;

**this**.sendBtn.ForeColor = System.Drawing.SystemColors.Control;

**this**.sendBtn.Location = new System.Drawing.Point(10, 39);

**this**.sendBtn.Name = "sendBtn";

**this**.sendBtn.Size = new System.Drawing.Size(100, 25);

**this**.sendBtn.TabIndex = 1;

**this**.sendBtn.Text = "Отправить";

**this**.sendBtn.UseVisualStyleBackColor = **false**;

**this**.sendBtn.Click += new System.EventHandler(**this**.sendBtn\_Click);

**this**.contactMinInfoPanel.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.contactMinInfoPanel.Controls.**Add**(**this**.onlineLabel);

**this**.contactMinInfoPanel.Controls.**Add**(**this**.nameLabel);

**this**.contactMinInfoPanel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.contactMinInfoPanel.Location = new System.Drawing.Point(0, 0);

**this**.contactMinInfoPanel.Name = "contactMinInfoPanel";

**this**.contactMinInfoPanel.Size = new System.Drawing.Size(648, 45);

**this**.contactMinInfoPanel.TabIndex = 2;

**this**.contactMinInfoPanel.Click += new System.EventHandler(**this**.contactMinInfoPanel\_Click);

**this**.onlineLabel.AutoSize = **true**;

**this**.onlineLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 7F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.onlineLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.onlineLabel.Location = new System.Drawing.Point(6, 29);

**this**.onlineLabel.Name = "onlineLabel";

**this**.onlineLabel.Size = new System.Drawing.Size(35, 13);

**this**.onlineLabel.TabIndex = 1;

**this**.onlineLabel.Text = "label4";

**this**.nameLabel.AutoSize = **true**;

**this**.nameLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.nameLabel.ForeColor = System.Drawing.SystemColors.Control;

**this**.nameLabel.Location = new System.Drawing.Point(6, 9);

**this**.nameLabel.Name = "nameLabel";

**this**.nameLabel.Size = new System.Drawing.Size(52, 17);

**this**.nameLabel.TabIndex = 0;

**this**.nameLabel.Text = "label3";

**this**.label2.Dock = System.Windows.Forms.DockStyle.Fill;

**this**.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((**byte**)(204)));

**this**.label2.ForeColor = System.Drawing.SystemColors.Control;

**this**.label2.Location = new System.Drawing.Point(0, 0);

**this**.label2.Name = "label2";

**this**.label2.Size = new System.Drawing.Size(648, 489);

**this**.label2.TabIndex = 5;

**this**.label2.Text = "Выберите собеседника из списка контактов";

**this**.label2.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

**this**.searchPanel.AutoSize = **true**;

**this**.searchPanel.ColumnCount = 1;

**this**.searchPanel.ColumnStyles.**Add**(new System.Windows.Forms.ColumnStyle(System.Windows.Forms.SizeType.Percent, 100F));

**this**.searchPanel.Dock = System.Windows.Forms.DockStyle.Top;

**this**.searchPanel.Location = new System.Drawing.Point(0, 0);

**this**.searchPanel.Name = "searchPanel";

**this**.searchPanel.RowCount = 5;

**this**.searchPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.searchPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.searchPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.searchPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.searchPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle());

**this**.searchPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F));

**this**.searchPanel.RowStyles.**Add**(new System.Windows.Forms.RowStyle(System.Windows.Forms.SizeType.Absolute, 20F));

**this**.searchPanel.Size = new System.Drawing.Size(279, 0);

**this**.searchPanel.TabIndex = 3;

**this**.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

**this**.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

**this**.BackColor = System.Drawing.Color.FromArgb(((**int**)(((**byte**)(23)))), ((**int**)(((**byte**)(33)))), ((**int**)(((**byte**)(43)))));

**this**.ClientSize = new System.Drawing.Size(927, 489);

**this**.Controls.**Add**(**this**.contentPanelWrapper);

**this**.Controls.**Add**(**this**.leftPanelWrapper);

**this**.IsMdiContainer = **true**;

**this**.Name = "MainF";

**this**.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;

**this**.Text = "Form1";

**this**.FormClosing += new System.Windows.Forms.FormClosingEventHandler(**this**.MainF\_FormClosing);

**this**.Load += new System.EventHandler(**this**.MainF\_Load);

**this**.Shown += new System.EventHandler(**this**.MainF\_Shown);

**this**.leftPanelWrapper.ResumeLayout(**false**);

**this**.leftPanelWrapper.PerformLayout();

**this**.contactsPanelWrapper.ResumeLayout(**false**);

**this**.contactsPanelWrapper.PerformLayout();

**this**.contactsLabelPanel.ResumeLayout(**false**);

**this**.contactsLabelPanel.PerformLayout();

**this**.searchPanelWrapper.ResumeLayout(**false**);

**this**.searchPanelWrapper.PerformLayout();

**this**.searchLabelPanel.ResumeLayout(**false**);

**this**.searchLabelPanel.PerformLayout();

**this**.panel1.ResumeLayout(**false**);

**this**.panel1.PerformLayout();

**this**.leftPanelHeader.ResumeLayout(**false**);

**this**.leftPanelHeader.PerformLayout();

**this**.userInfoPanel.ResumeLayout(**false**);

**this**.userInfoPanel.PerformLayout();

**this**.editUserInfoPanel.ResumeLayout(**false**);

((System.ComponentModel.ISupportInitialize)(**this**.editUserInfo)).EndInit();

**this**.contentPanelWrapper.ResumeLayout(**false**);

**this**.contentPanelWrapper.PerformLayout();

**this**.chatWrapper.ResumeLayout(**false**);

**this**.chatWrapper.PerformLayout();

**this**.chatPanelWrapperL2.ResumeLayout(**false**);

**this**.chatPanelWrapperL2.PerformLayout();

**this**.messagePanelWrapper.ResumeLayout(**false**);

**this**.messagePanelWrapper.PerformLayout();

**this**.messagePanel.ResumeLayout(**false**);

**this**.messagePanel.PerformLayout();

**this**.sendButtonPanel.ResumeLayout(**false**);

**this**.contactMinInfoPanel.ResumeLayout(**false**);

**this**.contactMinInfoPanel.PerformLayout();

**this**.ResumeLayout(**false**);}

**private** System.Windows.Forms.Panel leftPanelWrapper;

**private** System.Windows.Forms.Panel searchPanelWrapper;

**private** System.Windows.Forms.Panel contactsPanelWrapper;

**private** System.Windows.Forms.TableLayoutPanel contactsPanel;

**private** System.Windows.Forms.Panel contentPanelWrapper;

**private** System.Windows.Forms.Panel contactMinInfoPanel;

**private** System.Windows.Forms.Panel messagePanelWrapper;

**private** System.Windows.Forms.Button sendBtn;

**private** System.Windows.Forms.TextBox messageTextBox;

**private** System.Windows.Forms.Panel sendButtonPanel;

**private** System.Windows.Forms.Panel messagePanel;

**private** System.Windows.Forms.TableLayoutPanel chatPanel;

**private** System.Windows.Forms.Panel chatPanelWrapperL2;

**private** System.Windows.Forms.Panel chatWrapper;

**private** System.Windows.Forms.Label label2;

**private** System.Windows.Forms.Panel searchLabelPanel;

**private** System.Windows.Forms.Label nameLabel;

**private** System.Windows.Forms.Label onlineLabel;

**private** System.Windows.Forms.Panel leftPanelHeader;

**private** System.Windows.Forms.Panel userInfoPanel;

**private** System.Windows.Forms.Label userLoginLabel;

**private** System.Windows.Forms.Label userNameLabel;

**private** System.Windows.Forms.PictureBox editUserInfo;

**private** System.Windows.Forms.Panel editUserInfoPanel;

**private** System.Windows.Forms.Panel contactsLabelPanel;

**private** System.Windows.Forms.Label contactsLabel;

**private** System.Windows.Forms.Panel panel1;

**private** System.Windows.Forms.Label label1;

**private** System.Windows.Forms.TextBox searchTextBox;

**private** System.Windows.Forms.TableLayoutPanel searchPanel;}