Компьютерная академия «ШАГ»

Днепропетровский филиал

Кафедра Разработки программного обеспечения

КУРСОВОЙ ПРОЕКТ

«Zireael Messenger»

Студентов группы: ЕКО-16-П-1  
Евсеенко Н.В.

Самарского В.О.

Научный руководитель:  
Шаптала М.В.

Днепр 2018 г.

# Аннотация

Пояснительная записка содержит описание алгоритма программы «Zireael Messenger» и сервиса для работы с приложением. Чтобы облегчить работу с программой, в содержание введена структура системы и руководство пользователя.

Пояснительная записка оформлена в соответствии с предъявляемыми требованиями. Записка выполнена на 30 листах формата А4. Содержание пояснительной записки соответствует заданию, а разделы соответствуют содержанию.

Оглавление

[Аннотация 2](#_Toc533401261)

[Оглавление 3](#_Toc533401262)

[Введение 4](#_Toc533401263)

[Требования к функциональным характеристикам клиент-приложения 5](#_Toc533401264)

[Требования к техническим характеристикам 6](#_Toc533401265)

[Требования к программным характеристикам 7](#_Toc533401266)

[Проектирование системы 8](#_Toc533401267)

[Проектирование интерфейса 8](#_Toc533401268)

[Проектирование классов для клиент-приложения 10](#_Toc533401269)

[Проектирование базы данных 13](#_Toc533401270)

[Проектирование сервиса 17](#_Toc533401271)

[Функции сервиса 18](#_Toc533401272)

[Руководство пользователя 23](#_Toc533401273)

[Дополнения 30](#_Toc533401274)

# Введение

Темой данного курсового проекта является разработка простого приложения для обмена сообщениями и сервиса для обработки информации и взаимодействия с БД.

Чем доступней для людей становятся компьютерные технологии, тем больше появляется непрофессиональных пользователей. Поэтому основной задачей каждого программиста является создание интуитивно понятного интерфейса программы. Это задача так же будет выполнена в этом курсовом проекте.

Целью курсового проекта является:

* научиться создавать качественное и надежное программное обеспечение для обработки большого количества данных
* разработать техническую документацию к разработанному программному обеспечению
* Получить и закрепить навыки в работе с такими технологиями как WPF, WCF, Entity Framework, ADO.NET
* Получить и закрепить навыки в работе с пакетом программ Microsoft Visual Studio 2017 и Microsoft SQL Server

Результатом курсового проекта должна быть полноценная база данных для хранения данных, решающая поставленную задачу, приложение для работы с этой базой данных, а также техническая документация к указанному программному обеспечению.

# Требования к функциональным характеристикам клиент-приложения

В процессе курсового проектирования необходимо разработать ПО для обмена сообщениями. Данное программное обеспечение должно иметь простой и удобный интерфейс.

Так же необходим сервис для обработки информации и взаимодействия с базой данных.

Программное обеспечение должно выполнять следующие функции:

* Регистрация новых пользователей
* Вход в аккаунт
* Восстановление пароля
* Отправка текстовых сообщений
* Чтение сообщений
* Отправка файлов любого формата и размера
* Создание групп до 1000 пользователей
* Добавление контактов
* Блокировка пользователей
* Настройка программы и профиля

Сервис должен выполнять следующие функции:

* Обработка запросов клиентов
* Работа с базой данных
* Уведомление пользователям об изменении профиля(на почту)

# Требования к техническим характеристикам

Для работы программного обеспечения необходимо, что бы используемые вами технические средства соответствовали ниже перечисленным характеристикам:

* Компьютер типа IBM PC или ноутбук
* Процессор Pentium Celeron (Pentium IV и выше), AMD Athlon (и выше), с тактовой частотой не меньше 1,6 ГГц
* 1024 МБ ОЗ
* HHD или SSD со свободным местом не меньше 100 МБ
* Устройства ввода и манипуляций
* Видеоадаптер с 512 МБ памяти и более и поддержкой Microsoft DirectX 9 с драйвером WDDM
* Персональный компьютер должен быть подключен к глобальной сети интернет

# Требования к программным характеристикам

Для эксплуатации данного программного обеспечения необходимо, что бы на компьютере были установлены:

* операционная версия Windows 8+
* .NET Framework 4.6.1+
* Для работы компьютер должен быть подключен к глобальной сети Интернет

Для работы сервиса необходимо:

* операционная версия Windows 8+
* NET Framework 4.6.1+
* MS SQL Server
* Сервер должен быть подключен к глобальной сети Интернет

# Проектирование системы

## Проектирование интерфейса

Основной задачей каждого программиста является создание интуитивно понятного интерфейса программы, потому что удобный интерфейс всегда сделает работу пользователя приятней.

Поэтому при разработки интерфейса программы были использованы принципы Material Design

Material design базируется на тактильной реальности, его вдохновением стало изучение чернил с бумагой, он технологически продвинут, а также на 100% открыт для воображения и магии

Компания Google выработала ключевое правило — с особой тщательностью следить за потребностями пользователей, и, отталкиваясь от них, корректировать дизайн. Material Design — это настоящий тренд, благодаря которому совершенствуется UX, что неизменно вызывает желание вновь и вновь пользоваться определенным ресурсом или приложением.

Принципы Material Design

Принцип №1: material — это, в первую очередь, метафора, объединяющая корректную организацию пространства, а также движения объекта в нем. Данный дизайн неизменно строится на так называемой тактильной реальности, так как пользователи могут почувствовать грань между плоскостями, которой нет.

Принцип №2: всегда интуитивно понятные поверхности. Это значит, что грани с плоскостями всегда обязаны точно передавать смыслы визуального характера. То есть ощущения появляются на подсознательном уровне, благодаря теням, границам.

Принцип №3: взаимодействие объектов. Важно абсолютно все: свет, поверхность и ее отображение, движение. Все это становится необходимой базой для последующего взаимодействия объектов. Все должно выглядеть по максимуму реалистично, как будто действие происходит не на экране, а прямо перед пользователем.

Принцип №4: применяется единый адаптивный дизайн, но на каждом устройстве будет отражаться свой взгляд на одни и те же элементы, при этом все виды адаптированы к конкретным устройствам, но неизменными остаются цвета, иконки и пространственные отношения.

Принцип №5: он акцентирует внимание на графике, цвете и шрифтах. Дизайн должен быть максимально естественным и логичным, все элементы – сбалансированы и гармоничны по отношению друг к другу, чтобы у «юзеру» было комфортно и он понимал, что ему необходимо делать здесь и сейчас!

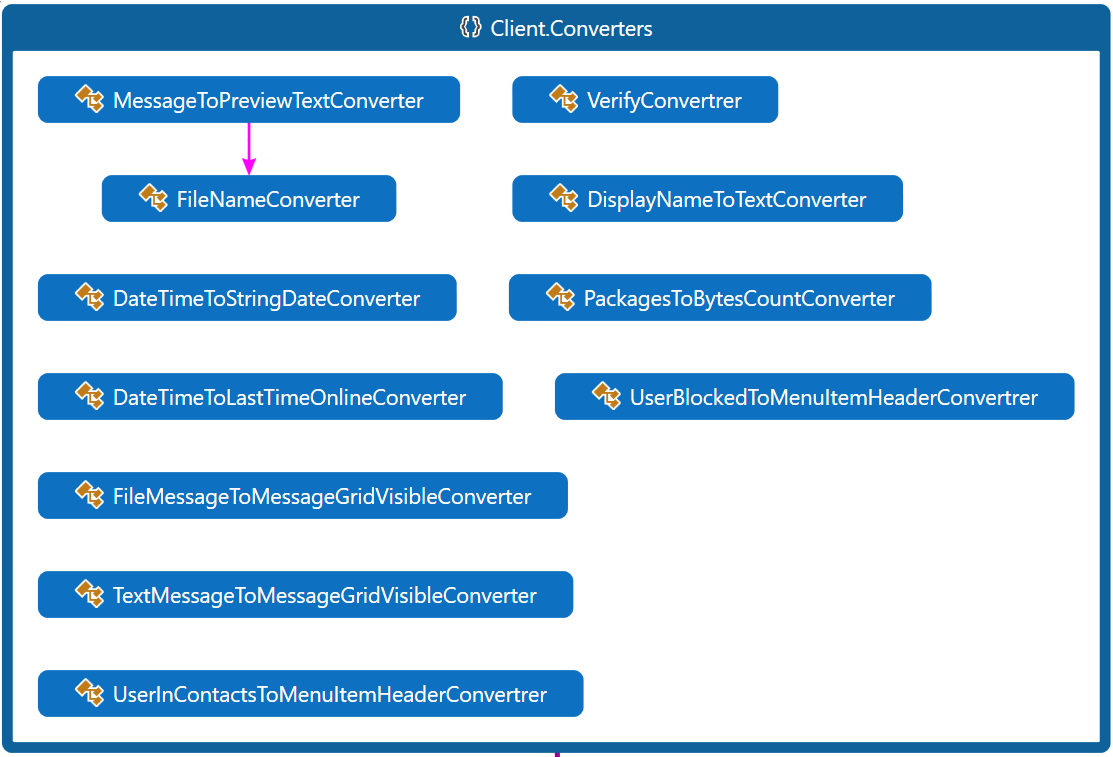
Принцип №6: первоначальность действия. По факту, это самая суть Материал Дизайна, так как посредством действий пользователя меняется дизайн, чтобы облегчить все задачи относительно эксплуатации web-ресурса или приложения.

Принцип №7: пользователь неизменно инициатор всех происходящих изменений, совершая определенные действия. Все начинается с «общения» с интерфейсом.

Принцип №8: анимационная хореография. Действия каждого объекта по отношению к другому синхронно. Объекты, находящиеся в поле пользовательского зрения, не нарушают базис даже после трансформации и перемещения.

Принцип №9: именно движение является обоснованием смысла. Каждый переход в интерфейсе обязан быть понятным и эффективным. Движения объектов существуют для привлечения внимания, неся в себе смысл.

## Проектирование классов для клиент-приложения



*public class DateTimeToLastTimeOnlineConverter : IValueConverter*

конвертирует DateTime:LastTimeOnline пользователя в string «был онлайн …»

*public class DateTimeToStringDateConverter : IValueConverter*

конвертирует время отправки сообщения в нужный формат:

hh:mm или dd:MM:yy hh:mm

*public class FileMessageToMessageGridVisibleConverter : IValueConverter*

показывает или скрывает Grid в зависимости от типа сообщения(текст/файл)

*public class FileNameConverter: IValueConverter*

обрезает большие названия файлов (максимум 25 символов)

*public class MessageToPreviewTextConverter : IValueConverter*

в зависимости от типа сообщения выводит текст сообщения или названия фала

*public class DisplayNameToTextConverter : IValueConverter*

обрезает большие имена

*public class PackagesToBytesCountConverter : IMultiValueConverter*

конвертирует количество байт из класса Packages в удобный формат

*public class TextMessageToMessageGridVisibleConverter : IValueConverter*

показывает или скрывает Grid в зависимости от типа сообщения(текст/файл)

*public class UserBlockedToMenuItemHeaderConvertrer : IValueConverter*

если пользователь заблокирован возвращает «Разблокировать» иначе «Заблокировать»

*public class UserInContactsToMenuItemHeaderConvertrer : IValueConverter*

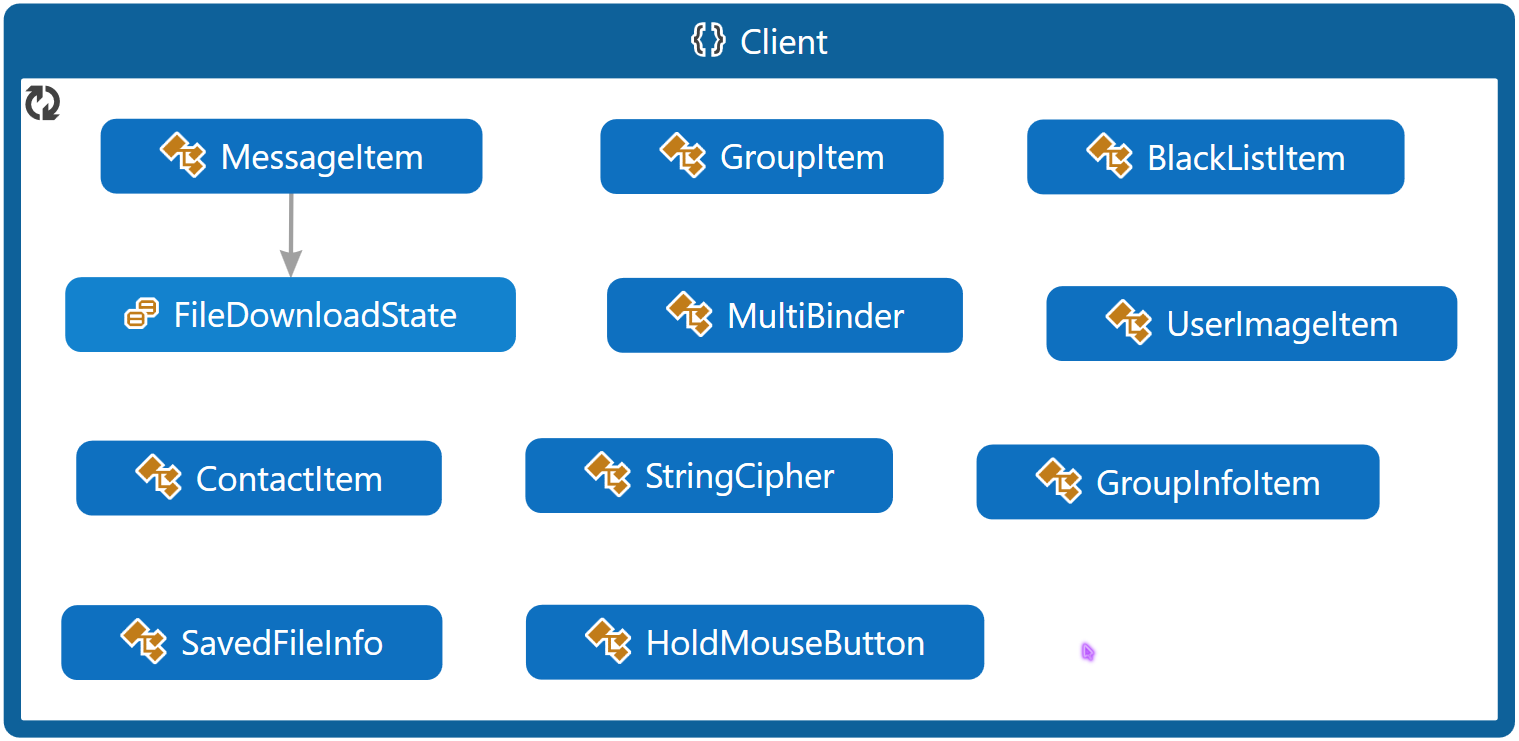
если пользователь есть в контактах возвращает «Удалить из контактов» иначе «Добавить в контакты»

*public class VerifyConvertrer : IMultiValueConverter*

если все переданные значения true возвращает true иначе false

*public class MultiBinder : MultiBinding*

вспомогательный класс для удобства использования мультибиндингов в xaml



*public enum FileDownloadState*

состояния загрузки файла

*public class HoldMouseButton*

событие на удержание кнопки мыши определенное время

*public static class StringCipher*

шифрование/дешифровка байт и текста System.Security.Cryptography

*public class BlackListItem : DependencyObject*

информация о пользователе в черном списке

*public class ContactItem : DependencyObject*

информация о контакте

*public class GroupInfoItem : DependencyObject*

информация о выделенном чате

*public class GroupItem : DependencyObject*

информация о чате

*public class MessageItem : DependencyObject*

информация о сообщении

*public class SavedFileInfo*

информация о сохраненном файле

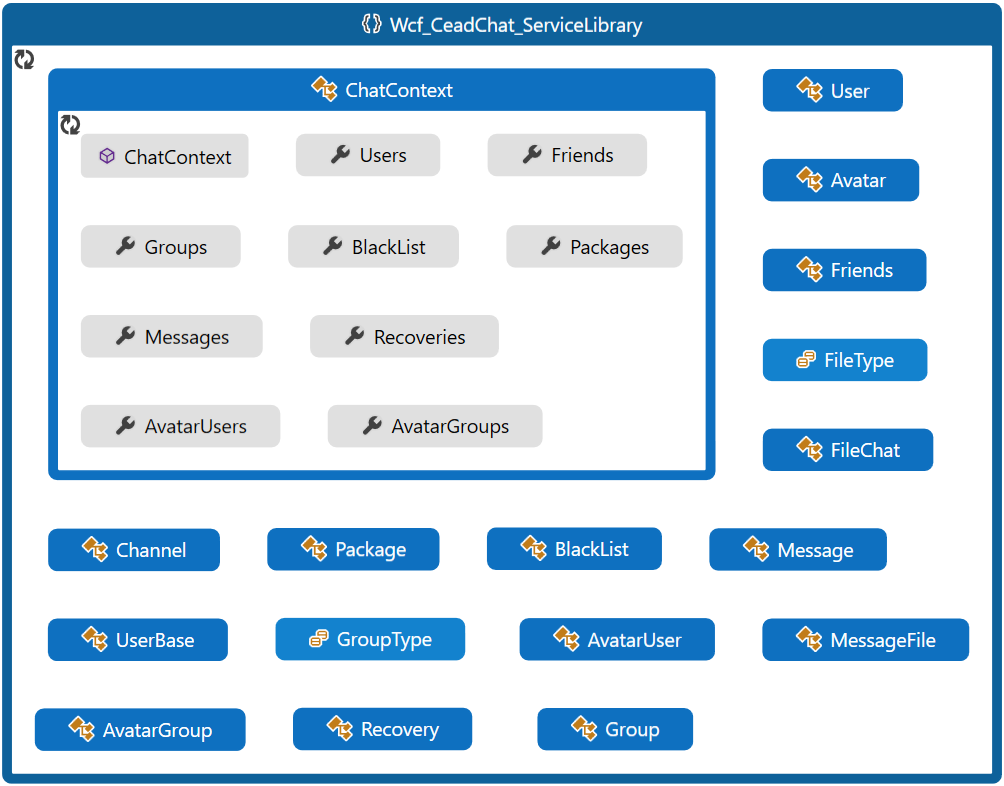
*public class UserImageItem*

информация об аватаре и BitmapSource

*public partial class MainWindow*

класс главного окна приложения

## Проектирование базы данных



Класс **UserBase** представляет модель «пользователь» и имеет следующие свойства:

* public int Id – уникальный идентификатор пользователя
* public string Login – логин пользователя
* public string DisplayName – имя пользователя
* public DateTime LastTimeOnline – время последнего онлайна
* public bool IsOnline – онлайн

Класс **User** дополняет модель «пользователь» и имеет следующие свойства:

* public string Email – почтовый адрес пользователя
* public string PasswordHash – хеш пароля (Argon2)
* public ICollection<Group> Groups – чаты в которых состоит пользователь
* public DateTime DateCreated – дата создания аккаунта
* public string Token - токен авторизации
* public DateTime TokenDate – дата создания токена авторизации

Класс **Avatar** представляет модель «аватар» и имеет следующие свойства:

* public int Id – уникальный идентификатор картинки
* public byte[] SmallData – картинка 64х64
* public byte[] BigData – картинка 512х512
* public string Format – формат картинки
* public DateTime DateTime – дата загрузки картинки

Класс **AvatarUser** наследуется от Avatar представляет модель «аватар пользователя» и имеет следующие свойства:

* public UserBase User – владелец аватара

Класс **AvatarGroup** наследуется от Avatar представляет модель «аватар чата» и имеет следующие свойства:

* public Group Group – чат которому присвоен аватар

Класс **Friends** представляет модель «контакты» и имеет следующие свойства:

* public int Id – уникальный идентификатор пары
* public User Sender – пользователь у которого в контактах User2
* public User User2 – добавленный в контакты

Класс **Package** представляет модель «часть файла» и имеет следующие свойства:

* public int Id – уникальный идентификатор пакета
* public int Number – номер по порядку
* public byte[] Data - байты

Класс **FileChat** представляет модель «файл» и имеет следующие свойства:

* public int Id – уникальный идентификатор файла
* public string Name – название файла
* public int Lenght – длинна файла
* public string Hash – md5 хеш файла
* public string FullPath – полный путь к файлу
* public int CountPackages – кол-во частей
* public int CountReadyPackages – кол-во загруженных частей
* public List<Package> Packages – части файла

Перечисление **FileType** представляет собой модель «тип файла» содержит:

* Image – изображение
* File – файл другого типа

Перечисление **GroupType** представляет собой модель «тип чата» содержит:

* SingleUser - диалог
* MultyUser – многопользовательский чат
* Channel – канал

Класс **Group** представляет собой модель «чат» содержит свойства:

* public int Id – уникальный идентификатор чата
* public UserBase Creator – создатель чата
* public bool IsVisible -видимость чата
* public string Name – название чата
* public ICollection<User> Users – пользователи в чате
* public ICollection<Message> Messages – сообщения в чате
* public GroupType Type – тип чата
* public Message LastMessage – последнее сообщение

Класс **Message** представляет собой модель «сообщение» содержит свойства:

* public int Id – уникальный идентификатор чата
* public bool IsVisible – видимость сообщения
* public string Text – текст сообщения
* public DateTime DateTime – время отправки
* public bool IsRead – прочитано
* public bool IsChanged - изменено
* public Group Group – чат в котором отправлено сообщение
* public UserBase Sender - отправитель

Класс **MessageFile** наследуется от Message, представляет собой модель «файловое сообщение» содержит свойства:

* public FileChat File - файл
* public FileType Type – тип файла

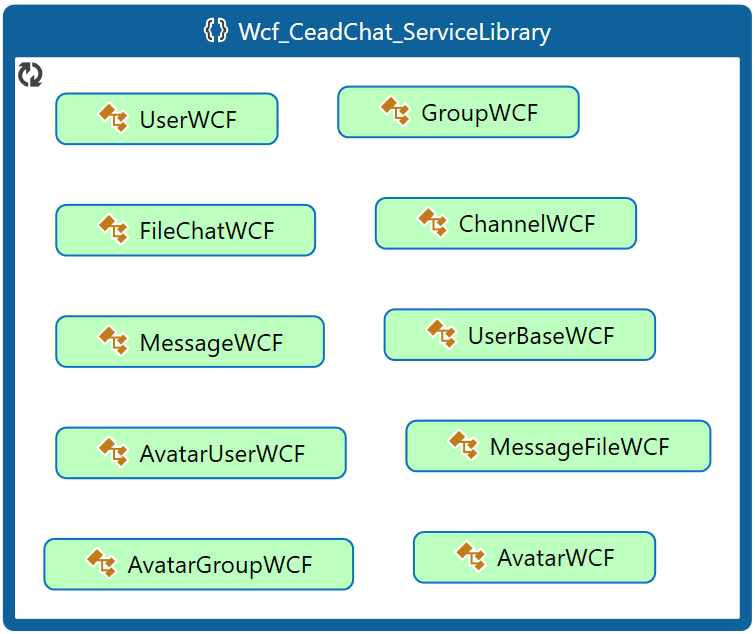
Класс **BlackList** представляет собой модель «черный список» содержит свойства:

* public int Id – уникальный идентификатор записи в чс
* public User Sender - блокирующий
* public User Blocked - заблокированный

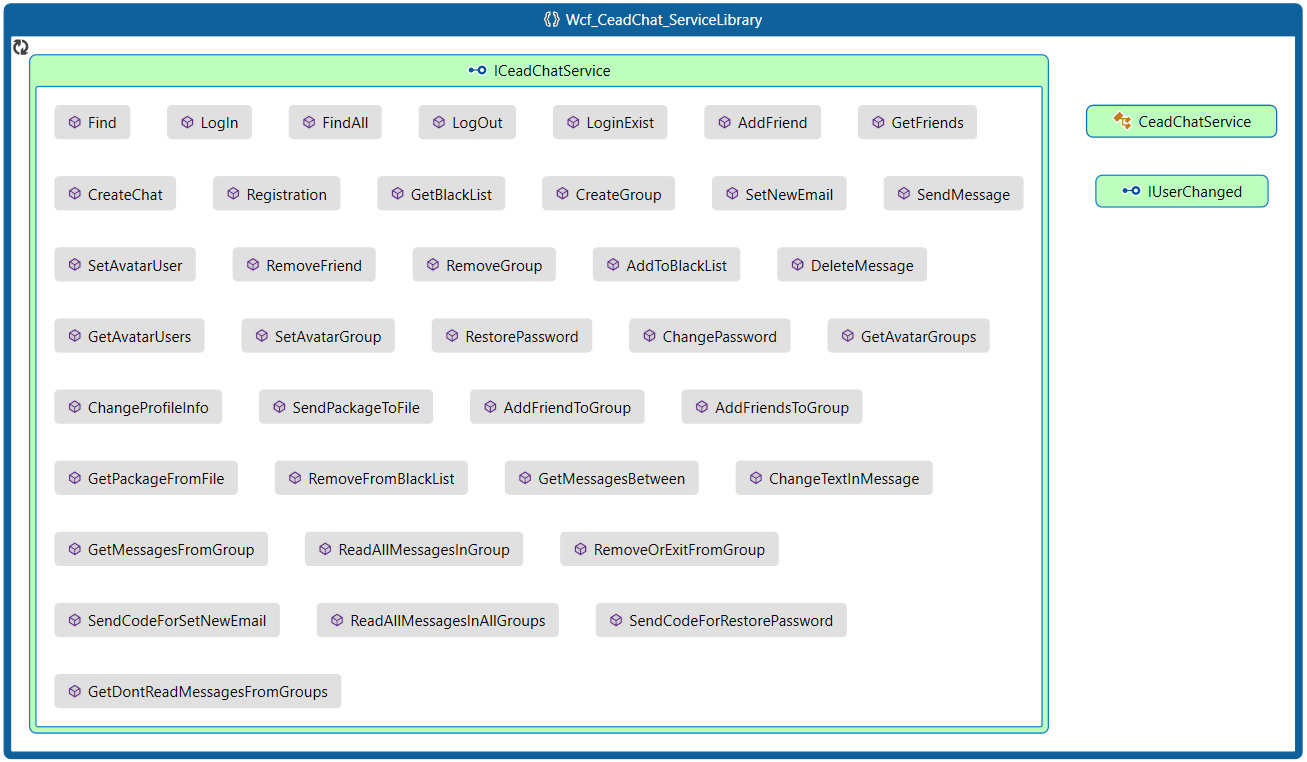
Класс **Recovery** представляет собой модель «код восстановления» содержит свойства:

* public int Id – уникальный идентификатор кода
* public DateTime RequestTime - время отправки кода
* public DateTime RecoveryTime - время приминения кода
* public string ConfirmationCode - код востановления
* public string NewInformation – новый почтовый адрес
* public bool IsWorking – работает
* public int UserId – пользователь который запросил код

## Проектирование сервиса

****

WCF не умеет передавать объекты с рекурсивными ссылками поэтому для транспортировки мы используем классы с постфиксом WCF которые дублируют классы базы данных, но не содержат рекурсивных ссылок.



### Функции сервиса

*public bool* ***AddFriend****(string userLogin)*

добавление пользователя c логином userLogin в список контактов

*public bool* ***AddFriendsToGroup****(GroupWCF group, IEnumerable<UserBaseWCF> friends)*

добавление контактов из списка friends в чат

*public bool* ***AddToBlackList****(UserBaseWCF user)*

блокировка пользователя user

*private void* ***CallUsersInGroup****(ICollection<User> usersCollection, Action<IUserChanged> action)*

отправка Callback всем пользователям из usersCollection

*public bool* ***ChangePassword****(string newPassword, string oldPassword)*

изменение пароля

*public bool* ***ChangeProfileInfo****(string displayName, string login)*

изменение имени и логина, если нужно изменить один параметр, второй можно передать null

*public bool* ***ChangeTextInMessage****(MessageWCF message)*

изменение текста сообщения *message*

*ChatContext* ***Context****(IUserChanged userChanged)*

получение данных из БД, если пользователь userChanged залогинен

*public bool* ***CreateChat****(UserBaseWCF addingUser)*

создание диалога с пользователем addingUser

*public bool* ***CreateGroup****(IEnumerable<UserBaseWCF> newUsers, string nameGroup)*

создание многопользовательского чата с именем nameGroup и пользователями newUsers

*private string* ***CreateToken****(UserBase user)*

генерация токена авторизации для пользователя user

*public bool* ***DeleteMessage****(MessageWCF message)*

удаление сообщения message

*public UserBaseWCF* ***Find****(string login)*

находит и возвращает пользователя по логину, если пользователя с таким логином нет, то вернется null

*public IEnumerable<UserBaseWCF>* ***FindAll****(string login, int maxCount)*

находит maxCount пользователей по Contains(login) и возвращает их

*private string* ***GeneratePasswordHash****(string password)*

генерирует хеш Argon2 для пароля password

*private List<UserBase>* ***GetAllFriends****(User sender)*

получение все связанных с sender пользователей(контакты и пользователи в чатах)

*public IEnumerable<AvatarGroupWCF>* ***GetAvatarGroups*** *(IEnumerable<GroupWCF> groups)*

получение аватаров групп из groups

*public IEnumerable<AvatarUserWCF>* ***GetAvatarUser s****(IEnumerable<UserBaseWCF> users)*

получение аватаров пользователей из users

*public IEnumerable<UserBaseWCF>* ***GetBlackList****()*

получение заблокированных пользователей

*public Dictionary<int, int>* ***GetDontReadMessagesFromGroups*** *(IEnumerable<int> groupsId)*

получение количиства непрочитанных сообщений для групп с id из groupsId

*public IEnumerable<UserBaseWCF>* ***GetFriends****()*

получение контактов для отправляющего

*public IEnumerable<MessageWCF>* ***GetMessagesBetween****(int groupId, int startIdx, int count)*

получает count сообщений из группы с id == groupId начиная со startIdx

*public IEnumerable<MessageWCF>* ***GetMessagesFromGroup****(GroupWCF group)*

получает все сообщения из группы group

*public Package* ***GetPackageFromFile****(int messageId, int packNumber)*

получение куска файла из сообщения с id == messageId и номером packNumber

*public UserWCF* ***LogIn****(string login, string password, string token)*

вход в аккаунт по логину и паролю либо по токену авторизации, если данные авторизации верны, то вернет пользователя иначе null

*public bool* ***LoginExist****(string login)*

проверка логина на существование, если логин существует вернется true

*public bool* ***LogOut****()*

пользователь вышел из аккаунта

*private void* ***NotificationAboutChangeOnlineStatus****(User user)*

изменился статус онлайн у пользователя user

*private string* ***RandomString****(int size)*

генерирует случайную строку размером size

*public bool* ***ReadAllMessagesInAllGroups****()*

прочитать все сообщения во всех группах для отправляющего

*public bool* ***ReadAllMessagesInGroup****(int groupId)*

прочитать все сообщения в группе

*public bool* ***Registration****(UserWCF newUser)*

зарегистрировать пользователя newUser

*public bool* ***RemoveFriend****(UserBaseWCF contact)*

удалить контакт contact

*public bool* ***RemoveFromBlackList****(UserBaseWCF user)*

разблокировать пользователя user

*public bool* ***RemoveGroup****(GroupWCF group)*

удаление чата group

*public bool* ***RemoveOrExitFromGroup****(int groupId, int userIdForRemove)*

удалить пользователя с id == userIdForRemove из группы с id == groupId

*public bool* ***RestorePassword****(string loginOrEmail, string code, string newPassword)*

установка пароля newPassword для пользователя с логином loginOrEmail, при условии, что он ввел правильный code присланный ему на почту

*public bool* ***SendCodeForRestorePassword****(string loginOrEmail)*

шлет на почту пользователя с логином loginOrEmail код для восстановления пароля

*public bool* ***SendCodeForSetNewEmail****(string newEmail, string password)*

если введен правильный пароль password, то на почту newEmail отправится код для подтверждения

*public int* ***SendMessage****(MessageWCF message, long hash)*

отправка сообщения message, hash нужен для верификации сообщения в callback

*void* ***SendMessageToMail****(string email, string title, string text)*

отправка сообщения text с заголовком title на почтовый адрес email

*public bool* ***SendPackageToFile****(int messageId, Package package)*

отправка куска файла Package из сообщения c id == messageId

*public bool* ***SetAvatarGroup****(AvatarGroupWCF avatar)*

установить аватар avatar группе avatar.Group

*public bool* ***SetAvatarUser****(AvatarUserWCF avatar)*

установить аватар пользователю avatar.User

*public bool* ***SetNewEmail****(string recoveryCode)*

если recoveryCode указан верно, пользователю, приславшему запрос устанавливается почтовый адрес закрепленный за этим кодом

*private void* ***Timer\_Elapsed****(object sender, ElapsedEventArgs e)*

проверка на месте ли онлайн пользователи, если callback не дошел, ставим пользователю офлайн статус

*private object* ***TryExecute****(ExceptionDelegate action, bool returnNull = false)*

action выполняется в обертке try, при ошибке вернет null или false

*private static void* ***WriteLog****(string message)*

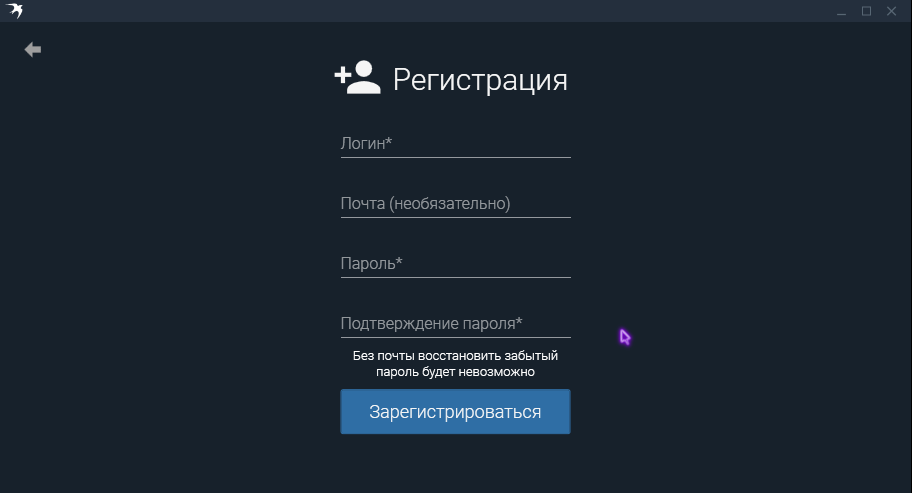
запись message в лог консоли

# Руководство пользователя

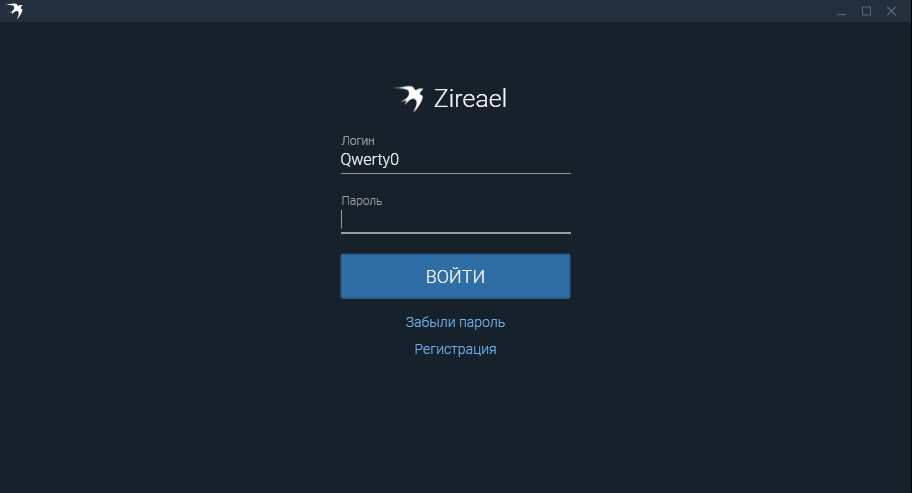
Запуск программы

Для запуска программы необходимо запустить файл с названием client.exe

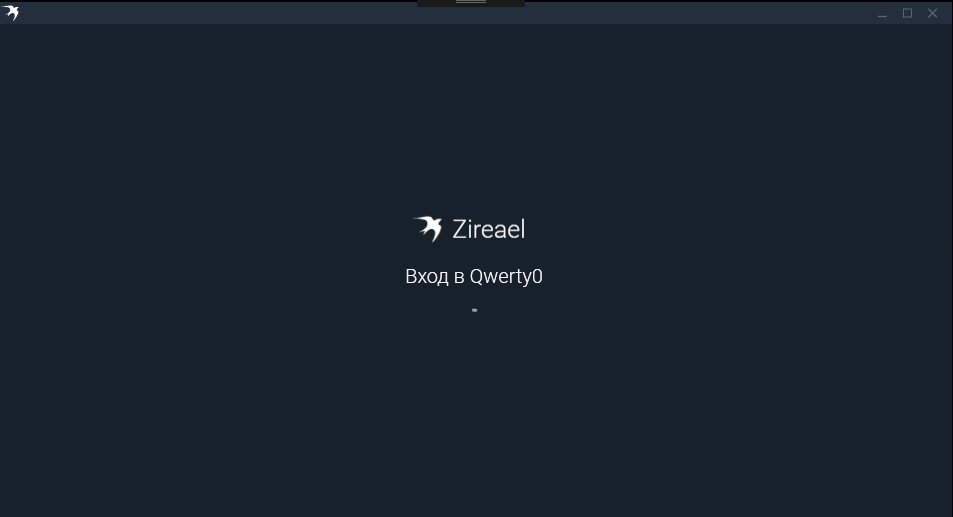
Регистрация



Вход



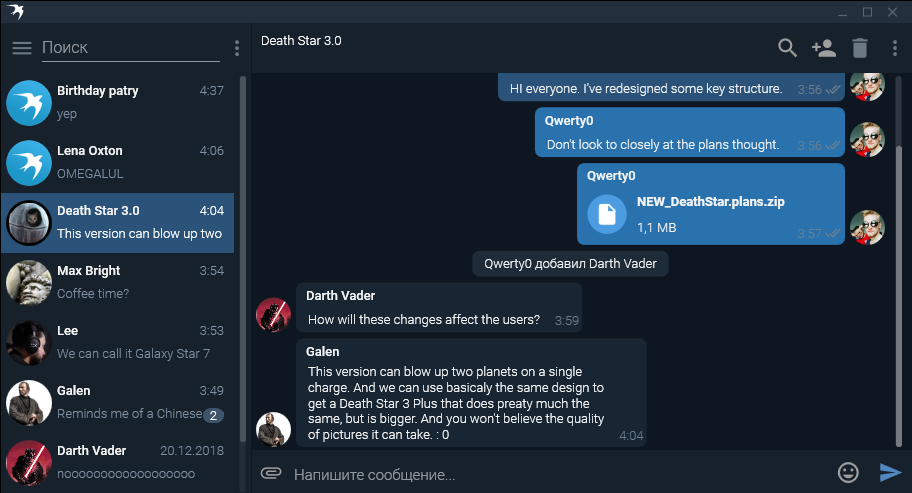
Вход по токену авторизации



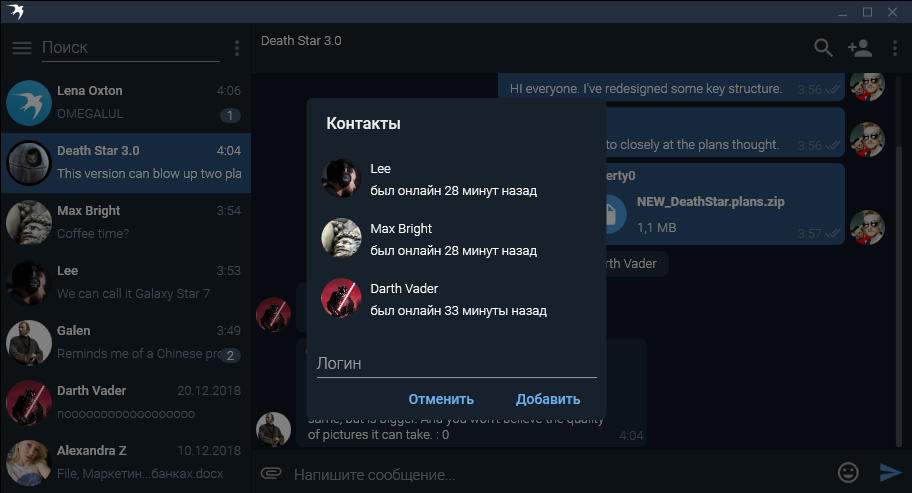
Интерфейс



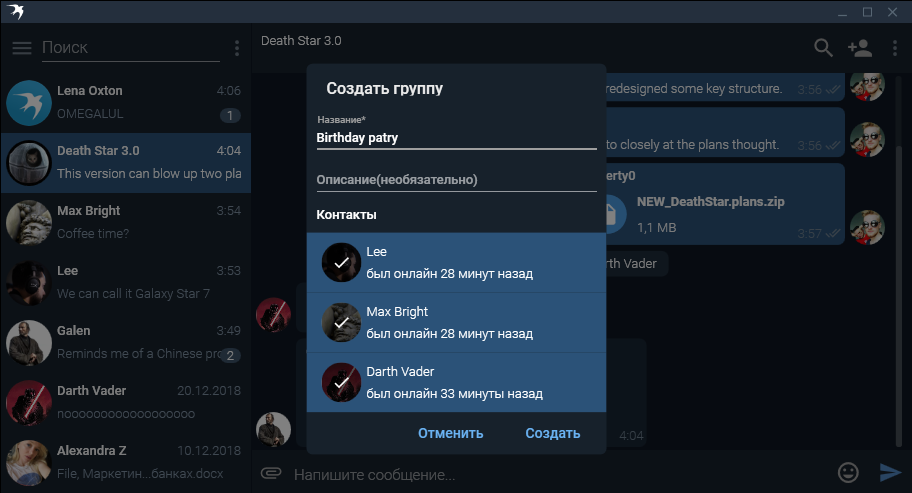
Что бы выделить нужно зажать ЛКМ на сообщении



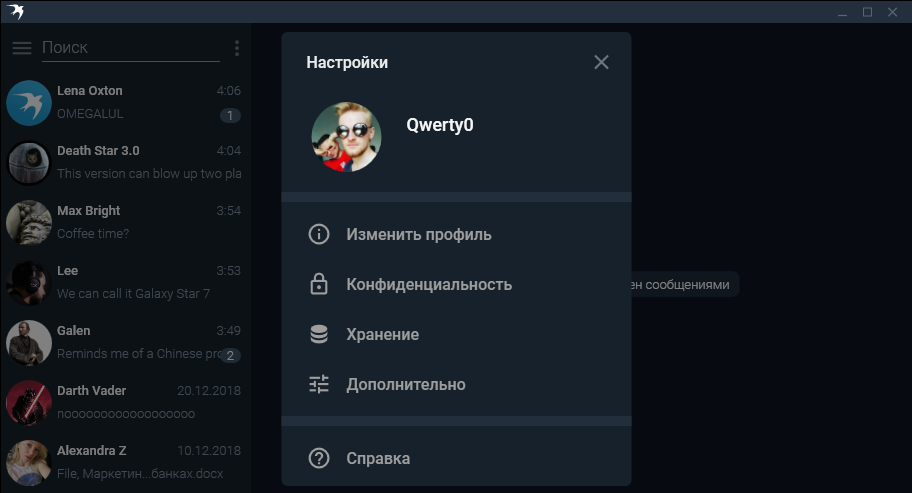
Контакты



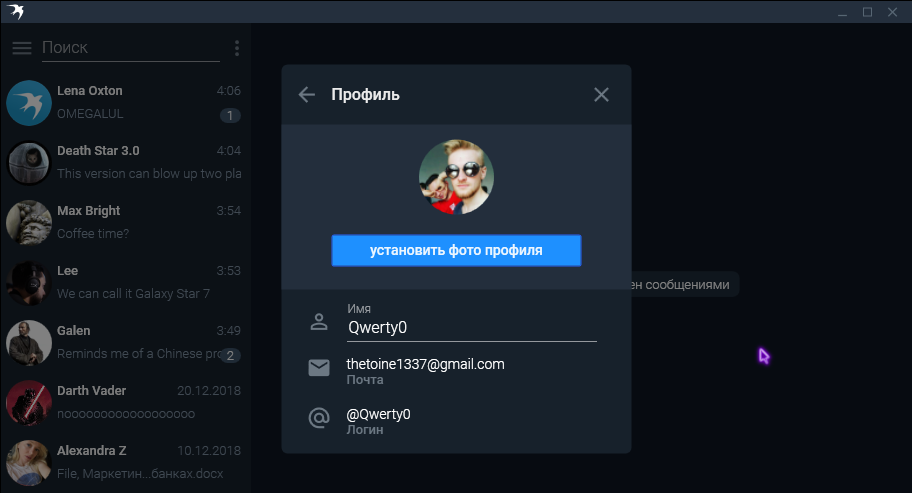
Создание группы

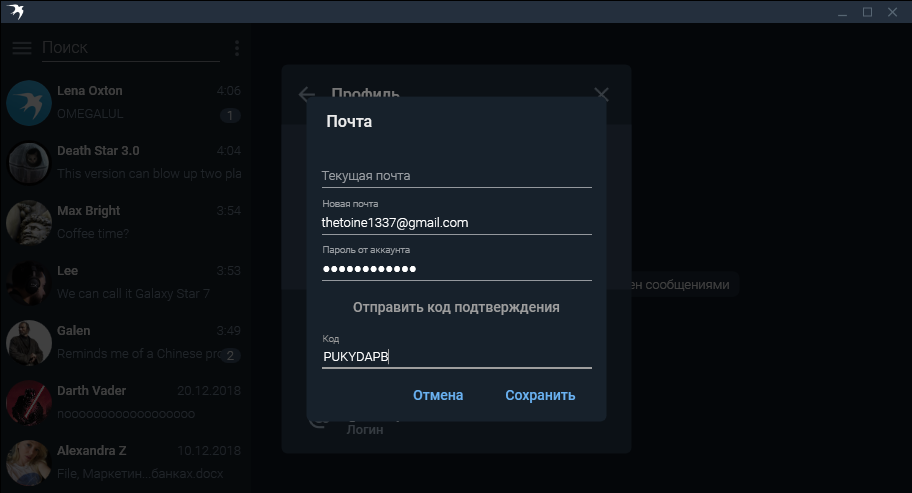


Настройки

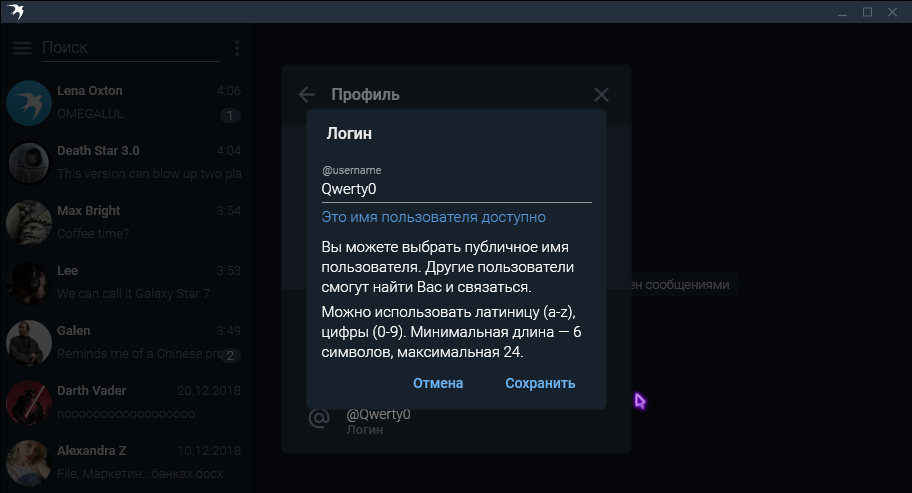


Настройки профиля

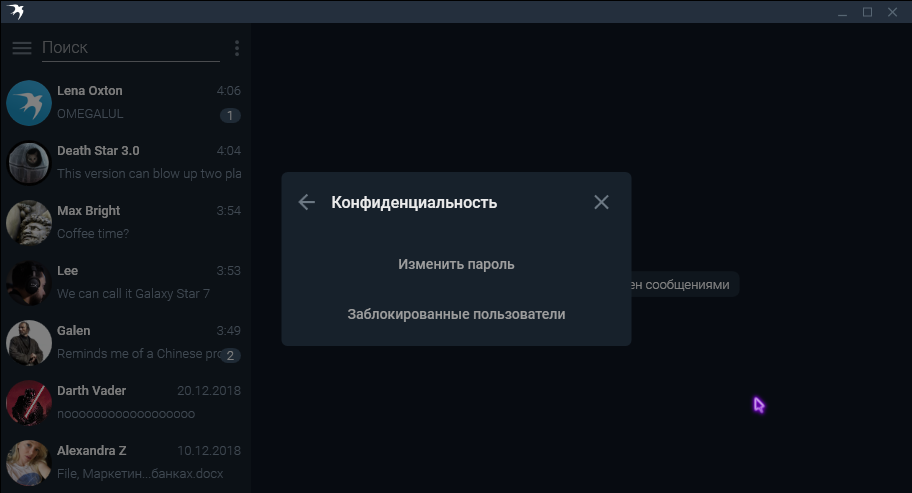


Изменение почты

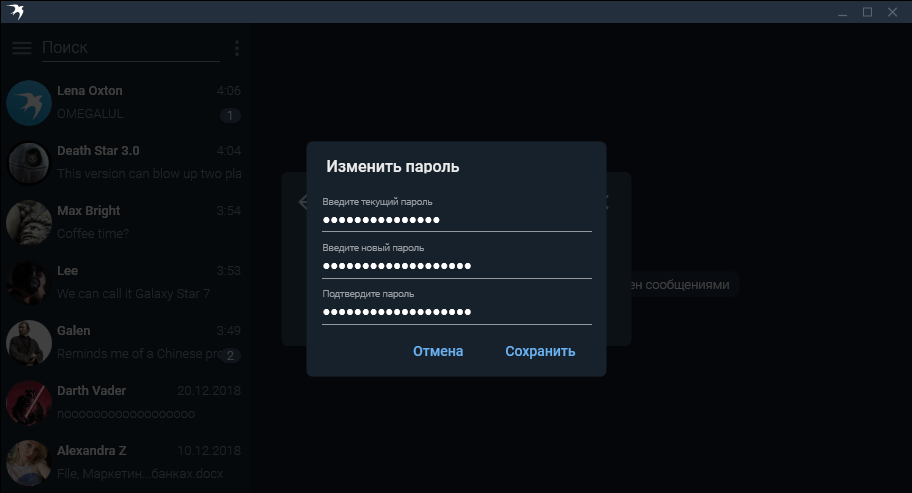
Изменение логина



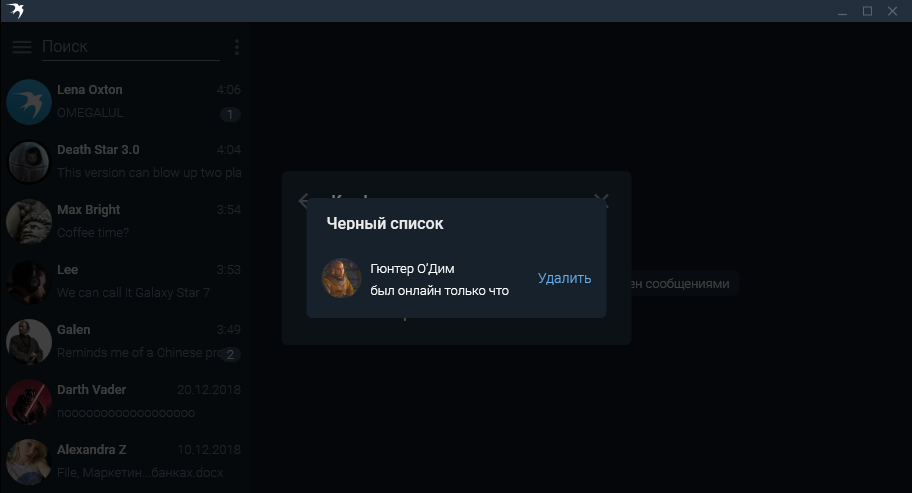
Настройки конфиденциальности



Изменение пароля



Заблокированные пользователи



# Дополнения